

# MANUAL DEL USUARIO DE GENTSTPC

UN PAQUETE DE PROGRAMAS PARA LA  
GENERACION Y LA VERIFICACION DE  
SUCESIONES DE NUMEROS PSEUDOALEATORIOS

AUTOR: ANGEL ADOLFO OLMOS CERVANTES

Tesis de Doctorado en Ingeniería (Orientación  
Investigación de Operaciones) - F.I.- UNAM -2005

"Generación de números pseudoaleatorios eficientes  
en microcomputadores" (ANEXO II)

## INDICE

Capítulo	Página
1. DESCRIPCION DE GENTSTPC	4
2. LOS GENERADORES	5
URN01 . . . . .	8
URN02 . . . . .	9
URN03 . . . . .	10
URN04 . . . . .	10
URN05 . . . . .	11
URN06 . . . . .	12
URN07 . . . . .	13
URN08 . . . . .	13
URN09 . . . . .	14
URN10 . . . . .	15
URN11 . . . . .	16
URN12 . . . . .	17
URN13 . . . . .	18
URN14 . . . . .	18
URN15 . . . . .	19
URN16 . . . . .	20
URN17 . . . . .	20
URN18 . . . . .	21
URN19 . . . . .	21
URN20 . . . . .	22
URN21 . . . . .	23
URN22 . . . . .	23
URN23 . . . . .	24
URN24 . . . . .	24
URN25 . . . . .	24
URN26 . . . . .	25
URN27 . . . . .	25
URN28 . . . . .	25
URN29 . . . . .	26
URN30 . . . . .	26
URN31 . . . . .	27
URN32 . . . . .	27
URN33 . . . . .	28
URN34 . . . . .	28
URN35 . . . . .	29
URN36 . . . . .	29
URN37 . . . . .	30
URN38 . . . . .	30
URN39 . . . . .	31
URN40 . . . . .	31

3. LAS PRUEBAS	32
TST01.....CHI-CUADRADO SOBRE CHI-CUAD. Y K-S SOBRE KOLMOGOROV-SMIRNOV. . . . .	34
TST02.....TEST DE DISTRIBUCION UNIFORME . . . . .	35
TST03.....TEST DEL PRODUCTO REZAGADO . . . . .	35
TST04..... TEST DE LOS HUECOS (GAP) . . . . .	36
TST05.....TEST DEL RECOLECTOR DE CUPONES . . . . .	38
TST06.....TEST K-S DE BONDAD DE AJUSTE . . . . .	40
TST07.....TEST DE PERMUTACION . . . . .	40
TST08.....TEST DEL POKER . . . . .	41
TST09.....TEST DE LAS CORRIDAS (RUNS UP). . . . .	42
TST10.....TEST DE LOS PARES SERIALES . . . . .	43
TST11.....TEST DEL MAXIMO DE T . . . . .	44
TSTM1.... TEST DE CORRELACION SERIAL . . . . .	45
TSTM2.... TESTS DE LA POTENCIA Y EL PERIODO . . . . .	46
TSTM3.... FACTORIZACION FACTORES PRIMOS. . . . .	47
TSTM4.... TEST ESPECTRAL . . . . .	48
TSTENTR...TESTS DE ENTROPIA. . . . .	49
4. UTILIZACION DEL PAQUETE GENTSTPC . . . . .	52
GENERACION DE NUMEROS PSEUDOLEATORIOS USANDO GENTSTPC. . . . .	52
PRUEBAS DE NUMEROS ALEATORIOS CON GENTSTPC	53
5. EJEMPLOS . . . . .	56
OBTENCION DE NUMEROS PSEUDOALEATORIOS CON GENERADORES DE GENTSTPC . . . . .	56
PRUEBAS DE CALIDAD DE LOS GENERADORES DE NUMEROS PSEUDOALEATORIOS. . . . .	57
LLAMADO DE PRUEBAS EMPIRICAS DESDE EL PROGRAMA PRINCIPAL FORTRAN . . . . .	57
LLAMADO DE PRUEBAS TEORICAS DESDE EL PROGRAMA PRINCIPAL FORTRAN. . . . .	58
LLAMADO DE PRUEBAS DE ENTROPIA DESDE EL PROGRAMA PRINCIPAL FORTRAN. . . . .	59
USO DE GENTSTPC PARA VERIFICAR UN GENERADOR DEL USUARIO . . . . .	62
6. RECOMENDACIONES PARA EL USO DE GENERADORES DE GENTSTPC . . . . .	64
ILUSTRACION DE UNA CORRIDA . . . . .	67
7. CODIGO FUENTE FORTRAN 77/90 GENTSTPC1 . . . . .	121
CODIGO FUENTE FORTRAN 77/90 GENTSTPC2 . . . . .	347

# Capítulo 1

## DESCRIPCION DE GENTSTPC

GENTSTPC es un paquete de programas que produce sucesiones de números pseudoaleatorios uniformemente distribuidos en el intervalo (0,1), y que permite verificar la aleatoriedad de tales sucesiones.

En el paquete se incluyen 40 Generadores y 15 Pruebas estadísticas.

Una completa referencia en cuanto a los primeros 20 generadores y al comportamiento de esos generadores frente a las Pruebas puede encontrarse en "The Handbook of Random Number Generation and Testing with TESTRAND Computer Code" de los profesores Edward J. Dudewicz y Thomas G. Ralley, American Sciences Press, Inc., Columbus, Ohio, USA.

Los generadores fueron seleccionados teniendo en cuenta su amplia utilización o sus particulares características satisfactorias frente a las pruebas estadísticas, y su mejor adaptación a las arquitecturas de microcomputadores con palabras-memoria de 32 bits.

Con el objeto de ser aplicados a los generadores se incluyen todas las pruebas teóricas sugeridas por Donald Knuth in "The Art of Computer Programming, Volume 2 / Semi-Numerical Algorithms".

Una prueba especial es la designada TST01 que incluye dos potentes sub-pruebas.

La primera realiza una prueba Chi-cuadrado sobre los valores generados por otras pruebas; mientras que el segundo realiza la prueba de Kolmogorov-Smirnov sobre los valores K-S generados por dos pruebas: la de bondad de ajuste (TST06) y la de los máximos de  $t$  (TST11).

Finalmente, se expone un conjunto de pruebas basadas en Pruebas de Entropía continua, aplicadas a sucesiones de números pseudoaleatorios (cfr. TSTENTR, pág.48).

## Capítulo 2

### LOS GENERADORES

Más abajo se incluye la lista de un conjunto de generadores: URN01 a URN20 que pertenecen al paquete TESTRAND (The Ohio State University, 1981); los generadores URN21 a URN30 (Denison University, 1986); y finalmente, los generadores URN31 a URN40 que constituyen el más reciente aporte al "estado del arte" en la materia.

La lista contiene una breve descripción del origen del generador, el lenguaje fuente en que fue escrito y eventualmente los nombres genéricos usados para describirlos en la literatura publicada.

Posteriormente, para cada generador se suministra la siguiente información: su utilización, la descripción de los parámetros necesarios para invocar a cada rutina; observaciones particulares; rutinas adicionales necesarias y las referencias bibliográficas.

**TABLA DE GENERADORES**

NOMBRE	NOMBRE GENERICICO	REFERENCIAS	LENGUAJE FUENTE
URN01	LLRANDOM	Learmouth y Lewis Tech.Report Naval- Post Grad. School (1973)	Assembler
URN02		Marsaglia y Bray CACM Vol. 11 (1968) pag. 757-759	Fortran
URN03	SUPER-DUPER	Marsaglia, Paul y Ananthanara- yanan McGill School of Computer Science	Assembler
URN04	cfr.URN01; es	LLRANDOM con shuffling	
URN05	GFSR	Lewis Payne y JACM Vol 20 (1973) pag. 456-468	Fortran
URN06		Lurie Mason y Communications on Statistics Vol. 2 (1973)pag. 363-371 Update Manual	Fortran
URN07	KERAND	IRCC Assembler Versión de RANDU	Assembler
URN08	RANDU	IBM Manual SSP	Fortran
URN09	RANDU en SPSS	Nie, Bent y Hull Actualiz. Manual SPSS (1973)	Fortran
URN10	Generador de OMNITAB II	Kruskal CACM Vol 12 (1969) pag. 93-94	Fortran
URN11		Coveyou McPherson y JACM Vol 14 (1967) pag. 100-119	Fortran
URN12		cfr. URN11; versión doble precisión de URN11	
URN13		Ahrens Dieter y Graz Inst. fur Math Assembler Stat., Tech. Report(1974)	Assembler
URN14		Zarling	Fortran
URN15-20		Hoaglin Technical Report Harvard Stat. Dept. NS-340	Assembler
URN21		Swain C.G. y Swain M.	Fortran
URN22	MTH\$RANDOM	VAX-11/780;GPSS-VX	Fortran

URN23		Brody T.A.	Assembler
URN24		Thesen A.	Assembler
URN25		Gason J.	Assembler
URN26		Mathews y Karian	cfr. URN24 Assembler
URN27	GPSS/H	Whittlessey	Fortran
URN28		Lewis Lehmer	y cfr. URN01 Fortran
URN29		Schmidt D.	Fortran
URN30	SIMSCRIPT II.5	Lehmer, Marse y Roberts	Fortran
URN31		Miyazaki	Fortran
URN32		Miyazaki	Fortran
URN33		Miyazaki	Fortran
URN34		Sezgin, Karian y Dudewicz	Fortran
URN35		L'Ecuyer P.	Fortran
URN36		Dudewicz E., Karian Z.	Fortran
URN37		y Marshall	
URN38			
URN39		W.J.J.Rey	Fortran
URN40		Marsaglia y Zaman	Fortran

## COMENTARIOS DEL CODIGO FUENTE DE LOS GENERADORES DE GENTSTPC

SUBROUTINE URN01 (X,IX,NBATCH)

\*\*\*\*\*

INICIALMENTE TESTRAND ADOPTO ESTA RUTINA DEL PAQUETE LLRANDOM EN CODIGO ASSEMBLER ESPECIFICO AL SISTEMA IBM/370 (NAVAL POSTGRADUATE SCHOOL RANDOM NUMBER GENERATOR PACKAGE) EN ESTA VERSION DE GENTSTPC SE DESARROLLA OTRO ALGORITMO.

### PROPOSITO

GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE PRECISION EN (0.0,1.0)

### USO

CALL URN01(X,IX,NBATCH)

### DESCRIPCION DE PARAMETROS

IX...ES EL VALOR INICIAL DE UNA SUCESION Y PUEDE SE CUALQUIER ENTERO ENTRE 1 Y 2147483647.(2\*\*31 -1)

X... ES EL ARREGLO RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE PRECISION EN (0.0,1.0)

NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR CON EL LLAMADO DE ESTA SUBROUTINA

### OBSERVACIONES

IX NO DEBE SER ALTERADO POR EL USUARIO DURANTE LA EJECUCION DEL PROGRAMA, SALVO QUE DECIDA REPETIR UNA SUCESION DE NUMEROS ALEATORIOS.

### SUBROUTINAS Y FUNCIONES REQUERIDAS

NINGUNA

### REFERENCIAS:

1. ALGORITMO DE BRADLEY, FOX Y SCHRAGE", "A GUIDE TO SIMULATION", SPRINGER, 2DA.ED.,1987,P.212.
2. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.
3. LEARMONTH G.P. Y LEWIS P.A.W., "NAVAL POSTGRADUATE SCHOOL RANDOM NUMBER GENERATOR PACKAGE LLRANDOM", TECHNICAL REPORT NPS55LW73061A, DEPARTMENT OF OPERATIONS RESEARCH AND ADMINISTRATIVE SCIENCES, NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIFORNIA, JUNE 1973.
4. DUDEWICZ E.J., "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM NUMBER GENERATOR PACKAGE", TECHNICAL REPORT, DEPARTMENT OF STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.



SUBROUTINE URN02 (X,NN,ML,MM,MK,L,MMM,KKK,NBATCH)  
\*\*\*\*\*

SUBROUTINA DADA POR MARSAGLIA Y BRAY, PARA GENERAR NUMEROS ALEATORIOS EN PUNTO FLOTANTE. EN EL PAQUETE IRCRAND LA RUTINA "RN3" UTILIZA EL MISMO METODO PARA GENERAR NUMEROS ALEATORIOS EN SIMPLE PRECISION (REFERENCIA 2)

#### PROPOSITO

GENERACION DE VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE PRECISION SOBRE (0.0,1.0)

#### USO

"CALL URN02I(ML,MM,MK,L,MMM,KKK,NN,)"  
DESDE EL PROGRAMA PRINCIPAL.  
LUEGO "CALL URN02(X,NN,ML,MM,MK,L,MMM,KKK,NBATCH)"

#### DESCRIPCION DE PARAMETROS

ML ...CONSTANTE ENTERA (MULTIPLICADOR) DEL GENERADOR DE ENTEROS ALEATORIOS  
MM...CONSTANTE ENTERA (MULTIPLICADOR) DEL GENERADOR DE ENTEROS ALEATORIOS  
MK...CONSTANTE ENTERA (MULTIPLICADOR) DEL GENERADOR DE ENTEROS ALEATORIOS  
L...SEMILLA PARA UNA SUCESION ALEATORIA  
MMM...SEMILLA PARA UNA SUCESION ALEATORIA  
KKK...SEMILLA PARA UNA SUCESION ALEATORIA  
NN...VECTOR VARIABLE CON DIMENSION ESPECIFICA 128 DONDE SE ALMACENAN LOS ENTEROS ALEATORIOS USADOS  
X ...ARREGLO RESULTANTE EN PUNTO FLOTANTE DE SIMPLE PRECISION PARA LOS NUMEROS ALEATORIOS EN (0.0,1.0)  
NBATCH...CANTIDAD DE NUMEROS ALEATORIOS A SER GENERADOS POR EL LLAMADO DE ESTA SUBROUTINA

#### OBSERVACIONES

LA SUBROUTINA URN02I INICIALIZA LAS VARIABLES NN,ML,MM,MK,L,MMM,KKK; SI EL USUARIO OPTA POR SUS PROPIOS VALORES NO DEBE INVOCAR EL LLAMADO "CALL URN02I(ML,MM,MK,L,MMM,KKK,NN)", PERO EN CAMBIO DEBE INICIALIZAR DICHAS VARIABLES.

ESTA SUBROUTINA ES ESPECIFICA PARA SISTEMAS IBM/360/370. EL METODO FUE DADO EN REFERENCIA 1.

SUBROUTINAS Y FUNCIONES REQUERIDAS  
NINGUNA

#### REFERENCIAS:

1. MARSAGLIA G. Y BRAY T.A., "ONE-LINE RANDOM NUMBER GENERATORS AND THEIR USE IN COMBINATIONS", COMMUNICATIONS OF THE ACM, VOL.11(1968), PAG. 757-759.
2. DUDEWICZ E.J., "IRCRAND-THE OHIO STATE UNIVERSITY RANDOM NUMBER GENERATOR PACKAGE", TECHNICAL REPORT, DEPARTMENT OF STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
3. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.  
PARA SISTEMAS DISTINTOS A LAS MAQUINAS 360/370:
4. GROSENBAUGH L.R., "MORE ON FORTRAN RANDOM NUMBER GENERATORS" COMMUNICATIONS DE ACM, VOL.12(1969), P.639.

SUBROUTINE URN03 (X,IX,NBATCH)

\*\*\*\*\*

INICIALMENTE TESTRAND ADOPTO ESTA RUTINA DEL PAQUETE SUPER-DUPER EN CODIGO ASSEMBLER ESPECIFICO AL SISTEMA IBM/370 (MCGILL UNIV. SCHOOL OF COMPUTER SCIENCE RANDOM NUMBER GENERATOR PACKAGE).

EN ESTA VERSION DE GENTSTPC SE DESARROLLA OTRO ALGORITMO.

PROPOSITO

GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE PRECISION EN (0.0,1.0)

USO

CALL URN03(X,IX,NBATCH)

DESCRIPCION DE PARAMETROS

IX...ES EL VALOR INICIAL DE UNA SUCESION Y PUEDE SE CUALQUIER ENTERO ENTRE 1 Y 2147483647.(2\*\*31 -1)  
X... ES EL ARREGLO RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE PRECISION EN (0.0,1.0)  
NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR CON EL LLAMADO DE ESTA SUBROUTINA

OBSERVACIONES

IX NO DEBE SER ALTERADO POR EL USUARIO DURANTE LA EJECUCION DEL PROGRAMA, SALVO QUE DECIDA REPETIR UNA SUCESION DE NUMEROS ALEATORIOS.

SUBROUTINAS Y FUNCIONES REQUERIDAS

NINGUNA

REFERENCIAS:

1. ALGORITMO DE BRADLEY, FOX Y SCHRAGE, "A GUIDE TO SIMULATION", SPRINGER, 2DA.ED.,1987,P.211
2. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.

SUBROUTINE URN04

\*\*\*\*\*

ESTA SUBROUTINA CORRESPONDE A UN PAQUETE ESPECIFICAMENTE COMPATIBLE CON LOS COMPUTADORES DEL SISTEMA IBM/370, DONDE EL CODIGO FUENTE ESTA ESCRITO EN LENGUAJE ASSEMBLER, Y POR LO TANTO NO SE INCLUYE EN ESTA VERSION.

REFERENCIAS:

1. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS OHIO,1981.
2. LEARMONTH G.P. Y LEWIS P.A.W., "NAVAL POSTGRADUATE SCHOOL RANDOM NUMBER GENERATOR PACKAGE LLRANDOM", TECHNICAL REPORT NPS55LW73061A, DEPARTMENT OF OPERATIONS RESEARCH AND

ADMINISTRATIVE SCIENCES, NAVAL POSTGRADUATE SCHOOL, MONTEREY,  
CALIFORNIA, JUNE 1973.

3. DUDEWICZ E.J., "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM  
NUMBER GENERATOR PACKAGE", TECHNICAL REPORT, DEPARTMENT OF  
STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.

SUBROUTINE URN05 (MFSR,P,Q,INTSIZ,JJ,IFLAG)

\*\*\*\*\*

METODO GENERALIZADO FEEDBACK SHIFT REGISTER (GFSR) DADO POR T.G.  
LEWIS Y W.H. PAYNE, "GENERALIZED FEEDBACK SHIFT REGISTER PSEUDO-RANDOM  
NUMBER ALGORITHM", JOURNAL OF THE ASSOCIATION FOR COMPUTING MACHINERY,  
VOL.20(1973), PAG.456-468, PARA LA GENERACION DE NUMEROS  
ALEATORIOS EN PUNTO FLOTANTE

PROPOSITO

GENERACION DE NUMEROS ALEATORIOS EN (0.0,1.0)

USO

"CALL URN05I(MFSR,P,JJ,Q,INTSIZ)" SOLO UNA VEZ EN EL PROGRAMA  
PRINCIPAL. DESPUES "CALL URN05 (MFSR,P,Q,INTSIZ,JJ,IFLAG)"

DESCRIPCION DE PARAMETROS

MFSR...ES UNA TABLA DE P NUMEROS ALEATORIOS INICIALES

P...ES UN EXPONENTE DEL POLINOMIO PRIMITIVO  $X^{**P}+X^{**Q}+1$

Q...ES UN EXPONENTE DEL POLINOMIO PRIMITIVO  $X^{**P}+X^{**Q}+1$

DELAY...ES LA DIFERENCIA RELATIVA, EN BITS, ENTRE COLUMNAS DE  
MFSR(.)

INTSIZ...ES EL TAMAÑO INTERO (EN BITS) DEL COMPUTADOR QUE HA SIDO  
UTILIZADO; POR EJEMPLO, 31 PARA LOS SISTEMAS IBM/360/370  
48 PARA EL COMPUTADOR CDC 6000, 35 PARA EL COMPUTADOR SRU  
1100 Y 15 PARA EL COMPUTADOR HP 2100

X...ES EL ARREGLO RESULTANTE EN PUNTO FLOTANTE DE SIMPLE PRECISION  
PARA LOS NUMEROS ALEATORIOS EN (0.0,1.0)

JJ...ES UN PARAMETRO USADO CUANDO RECOMIENZA LA SUCESION

IFLAG...ES UN PARAMETRO QUE ES PASADO DEL PROGRAMA PRINCIPAL  
A LA SUBROUTINA , Y QUE NO DEBE SER CAMBIADO POR EL  
USUARIO.

OBSERVACIONES

URN05I INICIALIZA LAS VARIABLES MFSR,P,DELAY,Q,INTSIZ PARA  
EL SISTEMA IBM /370 Y EL POLINOMIO PRIMITIVO  $X^{**98}+X^{**27}+1$ ;  
SI EL USUARIO DESEA EMPLEAR SUS PROPIOS VALORES NO DEBERA INVOCAR  
"CALL URN05I(MFSR,P,DELAY,Q,INTSIZ)", Y EN TAL CASO INICIALIZAR  
ESAS VARIABLES.

SUBROUTINAS Y FUNCIONES REQUERIDAS

NINGUNA

REFERENCIAS:

1. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER  
GENERATION AND TESTING", AMERICAN SCIENCES PRESS,  
COLUMBUS, OHIO, 1981.

SUBROUTINE URN06 (IX,JX,X,NBATCH)  
\*\*\*\*\*

BASADA EN UNA FUNCION DADA POR D. LURIE AND R.L. MASON:  
"EMPIRICAL INVESTIGATION OF SEVERAL TECHNIQUES FOR COMPUTER  
GENERATION OF ORDER STATISTICS", COMMUNICATIONS IN STATISTICS,  
VOL.2(1973), PAG.363-371, PARA GENERAR NUMEROS ALEATORIOS EN PUNTO  
FLOTANTE EN EL PAQUETE IRCCRAND (REFERENCIA 1), LA FUNCION USADA  
POR ESTE METODO FUE LLAMADA "RN1".

PROPOSITO  
GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE  
PRECISION EN (0.0,1.0)

USO  
CALL URN06(IX,JX,X,NBATCH)

DESCRIPCION DE PARAMETROS  
IX...ES EL VALOR INICIAL DE UNA SUCESION Y PUEDE SE CUALQUIER  
ENTERO ENTRE 1 Y 2147483647.(2\*\*31 -1)  
JX...ES EL VALOR INICIAL DE UNA SUCESION Y PUEDE SER CUALQUIER  
ENTERO ENTRE 1 Y 2147483647.JX NO DEBER SER IGUAL A IX.  
X... ES EL ARREGLO RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO  
FLOTANTE EN SIMPLE PRECISION EN (0.0,1.0)  
NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR  
CON EL LLAMADO DE ESTA SUBRUTINA

OBSERVACIONES  
IX Y JX NO DEBEN SER ALTERADOS POR EL USUARIO DURANTE LA EJECUCION  
DEL PROGRAMA, SALVO QUE EL USUARIO DECIDA REPETIR UNA SUCESION DE  
NUMEROS ALEATORIOS.

SUBRUTINAS Y FUNCIONES REQUERIDAS  
NINGUNA

REFERENCIAS:

1. DUDEWICZ E.J., "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM  
NUMBER GENERATOR PACKAGE", TECHNICAL REPORT, DEPARTMENT OF  
STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
2. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER  
GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS,  
OHIO, 1981.

SUBROUTINE URN07  
\*\*\*\*\*

SUBRUTINA ESCRITA POR C. G. KEANE EN LA OHIO STATE UNIVERSITY,  
11/19/68, EN ASSEMBLER, PARA GENERAR NUMBERS ALEATORIOS ENTEROS Y  
EN PUNTO FLOTANTE. EN EL PAQUETE IRCCRAND (REF.2), UNA SUBRUTINA QUE  
USA ESTE METODO PARA GENERAR NUMEROS ALEATORIOS EN SIMPLE PRECISION  
FUE DENOMINADA "KERAND".

PROPOSITO  
GENERAR VECTORES DE NUMEROS ALEATORIOS DE SIMPLE PRECISION EN PUNTO  
FLOTANTE EN (0.0,1.0), Y NUMEROS ALEATORIOS ENTEROS K SIENDO K

TAL QUE,  $0 < K < 2^{*}31$

ESTA SUBROUTINA ES ESPECIFICA DE LOS SISTEMAS IBM/360/370 ESCRITA EN LENGUAJE ASSEMBLER, Y POR LO TANTO NO SE INCLUYE EN ESTA VERSION.

REFERENCIAS:

1. INTERNATIONAL BUSINESS MACHINES CORPORATION, "SYSTEM/360 SCIENTIFIC SUBROUTINE PACKAGE, VERSION III, PROGRAMMER'S MANUAL, PROGRAM NUMBER 360A-CM-03X", MANUAL GH20-0205-4(FIFTH EDITION), IBM CORPORATION, WHITE PLAINS, NEW YORK, AUGUST 1970, P.77
2. DUDEWICZ E.J., "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM NUMBER GENERATOR PACKAGE", TECHNICAL REPORT, DEPARTMENT OF STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
3. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING" AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.

SUBROUTINE URN08 (IX,X,NBATCH)

\*\*\*\*\*

ESTA SUBROUTINA SE BASA EN LA SUBROUTINA DE SSP DE IBM PARA LA GENERACION DE NUMEROS ALEATORIOS ENTEROS Y EN PUNTO FLOTANTE. EN EL PAQUETE IRCCRAND (REFERENCIA 2), LA SUBROUTINA EMPLEADA PARA GENERAR NUMEROS ALEATORIOS EN SIMPLE PRECISION FUE LLAMADA "RN2"

PROPOSITO

GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE PRECISION EN (0.0,1.0), Y NUMEROS ALEATORIOS ENTEROS K , SIENDO  $0 < K < 2^{*}31$ .

USO

CALL URN08(IX,X,NBATCH)

DESCRIPCION DE PARAMETROS

IX...PARA EL PRIMER VALOR, QUE PUEDE SER CUALQUIER NUMERO ENTERO IMPAR ENTRE 1 Y 999999999.

LOS VALORES SUCESIVOS DE IX DEBERAN SER DEL CONJUNTO DE VALORES PREVIOS DE IY CALCULADOS POR ESTA RUTINA.

X... ES EL VECTOR RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE PRECISION EN (0.0,1.0)

NBATCH...ES LA CANTIDAD DE NUMEROS ALETORIOS QUE SE DESEA GENERAR CON EL LLAMADO DE ESTA SUBROUTINA.

OBSERVACIONES

ESTA SUBROUTINA ES ESPECIFICA DE LOS COMPUTADORES IBM SISTEMAS/360/370.CORRESPONDE A LA SUBROUTINA RANDU DE LA REFERENCIA 1.

SUBROUTINAS Y FUNCIONES REQUERIDAS

NINGUNA

REFERENCIAS:

1. INTERNATIONAL BUSINESS MACHINES CORPORATION, "SYSTEM/360 SCIENTIFIC SUBROUTINE PACKAGE, VERSION III, PROGRAMMER'S MANUAL PROGRAM NUMBER 360A-CM-03X", MANUAL GH20-0205-4 (FIFTH EDITION) IBM CORPORATION, WHITE PLAINS, NEW YORK, AUGUST 1970, P.77
2. DUDEWICZ E.J., "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM NUMBER GENERATOR PACKAGE", TECHNICAL REPORT, DEPARTMENT OF STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
3. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.

SUBROUTINE URN09 (IX,X,NBATCH)

\*\*\*\*\*

ESTA SUBRUTINA ES LA SUBRUTINA SSP DE IBM PARA GENERAR NUMEROS ALEATORIOS ENTEROS Y EN PUNTO FLOTANTE. EL CODIGO USADO DIFIERE DEL CODIGO DE SSP (DONDE SE EMPLEA COMO RN2). EL PRESENTE CODIGO SE ENCUENTRA EN EL LISTADO FUENTE DE N. NIE, D.H. BENT, AND C.H. HULL, "SPSS-STATISTICAL PACKAGE FOR THE SOCIAL SCIENCES", MCGRAW-HILL, INC., NEW YORK, 1970.  
EN EL PAQUETE IRCCRAND (REF.1), LA SUBRUTINA USADA PARA GENERAR NUMEROS ALEATORIOS EN SIMPLE PRECISION SE DENOMINA "RN4".

PROPOSITO

GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE PRECISION EN (0.0,1.0)

USO

CALL URN09(IX,X,NBATCH)

DESCRIPCION DE PARAMETROS

IX...ES EL VALOR INICIAL Y PUEDE SER CUALQUIER ENTERO IMPAR ENTRE 1 Y 999999999.  
X... ES EL VECTOR RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE PRECISION EN (0.0, 1.0)  
NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR CON EL LLAMADO DE ESTA SUBRUTINA.

OBSERVACIONES

IX NO DEBE SER ALTERADO POR EL USUARIO DURANTE LA EJECUCION DEL PROGRAMA, SALVO QUE EL DECIDA REPETIR UNA SUCCESION DE NUMEROS ALEATORIOS.

SUBRUTINAS Y FUNCIONES REQUERIDAS

NINGUNA

REFERENCIAS:

1. DUDEWICZ E.J., "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM NUMBER GENERATOR PACKAGE", TECHNICAL REPORT, DEPARTMENT OF STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.

2. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.

SUBROUTINE URN10 (IX,X,NBATCH)

\*\*\*\*\*

SUBROUTINA DESTACADA POR SU PORTABILIDAD SUGERIDA POR BY J.B. KRUSKAL, Y USADA POR OMNITAB 2, PARA GENERAR NUMEROS ALEATORIOS EN PUNTO FLOTANTE. EN EL PAQUETE IRCCRAND (REFERENCIA 2), LA SUBROUTINA QUE USA ESTE METODO PARA GENERAR NUMEROS ALEATORIOS EN SIMPLE PRECISION SE LLAMA "RN5".

PROPOSITO

GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE PRECISION EN (0.0,1.0)

USO

CALL URN10(IX,X,NBATCH)

DESCRIPCION DE PARAMETROS

- IX...PARA EL PRIMER VALOR, QUE DEBE SER PUESTO IGUAL A 1.LOS VALORES SUCEIVOS DE IX DEBERAN PETENECER AL CONJUNTO DE VALORES PREVIOS DE IY CALCULADOS POR ESTA RUTINA.
- X... ES EL VECTOR RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE PRECISION EN (0.0,1.0)
- NBATCH...ES LA CANTIDAD DE NUMEROS ALETORIOS QUE SE DESEA GENERAR CON EL LLAMADO DE ESTA SUBROUTINA.

OBSERVACIONES

ESTA SUBROUTINA ES GENERAL(REF.1)

SUBROUTINAS Y FUNCIONES REQUERIDAS

NINGUNA

REFERENCIAS:

1. KRUSKAL J.B., "EXTREMELY PORTABLE RANDOM NUMBER GENERATOR", COMMUNICATIONS OF THE ACM, VOL.12(1969), PAG. 93-94.
2. DUDEWICZ E.J., "IRCCRAND-THE OHIO STATE UNIVGRSITY RANDOM NUMBER GENERATOR PACKAGE", TECHNICAL REPORT, DEPARTMENT OF STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
3. DUDEWICZ E.J. Y RALLEYT., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.

SUBROUTINE URN11(IX,X,L,C,ZZ,NBATCH)

\*\*\*\*\*

ESTA SUBROUTINA GENERA NUMEROS ALEATORIOS EN PUNTO FLOTANTE USANDO UN GENERADOR CONGRUENCIAL MIXTO QUE ES ESPECIFICADO POR EL USUARIO. EN EL PAQUETE IRCCRAND (REFERENCIA 2),UNA SUBROUTINA QUE USA ESTE METODO PARA GENERAR NUMEROS ALEATORIOS EN SIMPLE

PRECISION SE DENOMINA "RNCG".

PROPOSITO

GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE  
PRECISION EN (0.0,1.0)

USO

CALL URN11(IX,X,L,C,ZZ,NBATCH)

DESCRIPCION DE PARAMETROS

IX...PARA EL PRIMER VALOR DEBE SER PUESTO A 1.(UN USUARIO QUE  
ENTIENDE LA OPERACION DE ESTE PROGRAMA PUEDE SEGURAMENTE  
USAR OTROS VALORES INICIALES)  
X... ES EL ARREGLO RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO FLO-  
TANTE DE SIMPLE PRECISION EN (0.0,1.0).  
L... ES EL ENTERO MULTIPLICATIVO DEL GENERADOR.  
C... ES LA CONSTANTE ADITIVA DEL GENERADOR. (EL GENERADOR ES  
DENOMINADO "CONGRUENCIAL MIXTO" SI C ES DISTINTO DE CERO  
Y "CONGRUENCIAL MULTIPLICATIVO" SI C=0.)  
ZZ...ES EL MODULO POR EL CUAL LOS NUMEROS SON REDUCIDOS  
NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR  
CON EL LLAMADO DE ESTA SUBROUTINA.

OBSERVACIONES

PARA VER ALGUNAS SELECCIONES HABITUALES DE L,C,ZZ ,REFERENCIA 1

SUBROUTINAS Y FUNCIONES REQUERIDAS

NINGUNA

REFERENCIAS:

1. COVEYOU R.R. Y MACPHERSON R.D., "FOURIER ANALYSIS OF  
UNIFORM RANDOM NUMBER GENERATORS", JOURNAL OF THE ACM, VOL.14  
(1967), PAG.100-119.
2. DUDEWICZ E.J., "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM  
NUMBER GENERATOR PACKAGE", TECHNICAL REPORT, DEPARTMENT OF  
STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
3. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER  
GENERATION AND TESTING", AMERICAN SCIENCES PRESS,  
COLUMBUS, OHIO, 1981.

SUBROUTINE URN12 (IX,X,L,C2,ZZ2,NBATCH)

\*\*\*\*\*

ESTA SUBROUTINA GENERA NUMEROS ALEATORIOS EN PUNTO FLOTANTE  
USANDO UN GENERADOR CONGRUENCIAL MIXTO QUE ES ESPECIFICADO POR  
EL USUARIO. EN EL PAQUETE IRCCRAND (REFERENCIA 2), UNA SUBROUTINA  
QUE USA ESTE METODO PARA GENERAR NUMEROS ALEATORIOS EN SIMPLE  
PRECISION SE DENOMINA "RNCG".

PROPOSITO

GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE  
PRECISION EN (0.0,1.0)



USO

CALL URN12(IX,X,L,C2,ZZ2,NBATCH)

DESCRIPCION DE PARAMETROS

IX...PARA EL PRIMER VALOR DEBE SER PUESTO A 1.(UN USUARIO QUE  
ENTIENDE LA OPERACION DE ESTE PROGRAMA PUEDE SEGURAMENTE  
USAR OTROS VALORES INICIALES). LOS VALORES SUCEIVOS DE IX  
DEBEN TOMARSE DEL CONJUNTO DE LOS VALORES PREVIOS  
IY CALCULADOS POR ESTA RUTINA.  
X... ES EL ARREGLO RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO FLO-  
TANTE DE SIMPLE PRECISION EN (0.0,1.0).  
L... ES EL ENTERO MULTIPLICATIVO DEL GENERADOR.  
C2...ES LA CONSTANTE ADITIVA DEL GENERADOR. (EL GENERADOR ES  
DENOMINADO "CONGRUENCIAL MIXTO" SI C2 ES DISTINTO DE CERO  
Y "CONGRUENCIAL MULTIPLICATIVO" SI C2=0.)  
ZZ2...ES EL MODULO POR EL CUAL LOS NUMEROS SON REDUCIDOS  
NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR  
CON EL LLAMADO DE ESTA SUBROUTINA.

OBSERVACIONES

ESTA SUBROUTINA ES UNA VERSION EN DOBLE PRECISION DE URN11 (LOS  
CALCULOS INTERMEDIOS SON EN DOBLE PRECISION PARA INCREMENTAR LA  
PRECISION).RETORNA NUMEROS ALEATORIOS EN SIMPLE PRECISON.

SUBROUTINAS Y FUNCIONES REQUERIDAS

NINGUNA

REFERENCIAS:

1. COVEYOU R.R. Y MACPHERSON, R.D., "FOURIER ANALYSIS OF  
UNIFORM RANDOM NUMBER GENERATORS", JOURNAL OF THE ACM, VOL.14  
(1967), PAG.100-119.
2. DUDEWICZ E.J., "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM  
NUMBER GENERATOR PACKAGE", TECHNICAL REPORT, DEPARTMENT OF  
STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
3. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER  
GENERATION AND TESTING", AMERICAN SCIENCES PRESS,  
COLUMBUS, OHIO, 1981.

SUBROUTINE URN13 (X,NBATCH)

\*\*\*\*\*

ESTA SUBROUTINA ESTA BASADA EN UNA FUNCION DADA POR J.H. AHRENS Y V.  
DIETER, "NON-UNIFORM RANDOM-NUMBERS", INSTITUT FUR MATH.STATISTIK,  
TECHNISCHE HOCHSCHULE , GRAZ, A 8010 GRAZ, HAMERLINGG. 6, VI, AUSTRIA,  
1974, P.1-10.

PROPOSITO

GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE  
PRECISION EN (0.0,1.0)

USO

CALL URN13(X,NBATCH)

DESCRIPCION DE PARAMETROS

IA.....DIMENSION VECTOR DE NUMEROS ENTEROS

IR.....DIMENSION VECTOR DE NUMEROS REALES  
NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR  
MEDIANTE ESTA SUBROUTINA  
INTIN(1),INTIN(2),INTIN(3),INTIN(4),INTIN(5) VALORES ENTEROS

SUBROUTINAS Y FUNCIONES REQUERIDAS  
NINGUNA

METODO  
UN GENERADOR CONGRUENCIAL MULTIPLICATIVO

REFERENCIAS:

1. AHRENS Y DIETER (1974)
2. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.

SUBROUTINE URN14 (IX,JX,X,NBATCH,TABLE)  
\*\*\*\*\*

ESTA SUBROUTINA ES UNA EXPRESION EN FORTRAN DE UNA PARTE DEL PAQUETE IRAND DE NUMEROS ALEATORIOS DESARROLLADO POR EL DR. RAYMOND Y L. ZARLING EN 1971 EN EL MARIETTA COLLEGE, MARIETTA, OHIO.

PROPOSITO  
GENERA VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE PRECISION EN (0.0,1.0)

USO  
CALL URN14(IX,JX,X,NBATCH,TABLE)

DESCRIPCION DE PARAMETROS

IX...ES EL VALOR INICIAL DE UNA PORCION DEL GENERADOR CONGRUENCIAL  
JX...ES EL INDICE PARA EL PROCESO DE BARAJAR EN LA TABLA  
X...ES EL ARRAY RESULTADO DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPE PRECISION EN (0.0,1.0)  
NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR EN LA LLAMADA A URN14  
TABLE...ES UNA TABLA DE 64 INTEROS ALEATORIOS USADOS INTERNAMENTE POR EL GENERADOR.

OBSERVACIONES  
EL VALOR DE JX ES ASIGNADO A LA MITAD DE LA PALABRA ENTERA EL VALOR ES DIVIDIDO EN 512 PARA CONSTRUIR UN ENTERO ENTRE 0 Y 63 Y LUEGO EMPLEADO COMO INDICE EN UNA TABLA DE 64 ENTEROS ALEATORIOS.  
EL VALOR JX ES CONVERTIDO EN UN NUMERO EN PUNTO FLOTANTE Y GUARDADO EN EL ARREGLO X DE NUMEROS ALEATORIOS A SER RETORNADO.  
EL ENTERO IX ES USADO PARA REEMPLAZAR EL ELEMENTO EN LA TABLA USADO PARA PRODUCIR EL NUMERO ALEATORIO.

SUBROUTINAS Y FUNCIONES REQUERIDAS  
NINGUNA

REFERENCIAS:

1. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.

SUBROUTINE URN15 (IX,X,NBATCH)

\*\*\*\*\*

ESTA SUBRUTINA IMPLEMENTA UN GENERADOR CONGRUENCIAL LINEAL PARA PRODUCIR NUMEROS ALEATORIOS EN PUNTO FLOTANTE. EL GENERADOR FUE DISEÑADO POR DAVID HOAGLIN COMO UNO DE SEIS, CUYO MODULO ES IGUAL A  $2^{31}-1$  QUE SUPERA BIEN EL TEST ESPECTRAL.

PROPOSITO

GENERA VECTORES DE NUMEROS ALEATORIOS DE SIMPLE PRECISION EN PUNTO FLOTANTE EN (0.0,1.0).

ESTA SUBRUTINA ES ESPECIFICA DE LOS SISTEMAS IBM/360/370 ESCRITA EN LENGUAJE ASSEMBLER, Y POR LO TANTO NO SE INCLUYE EN ESTA VERSION.

METODO

REFERENCIAS:

1. DUDEWICZ E.J., "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM NUMBER GENERATOR PACKAGE", TECHNICAL REPORT, DEPARTMENT OF STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
2. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.
3. HOAGLIN D.C., "THEORETICAL PROPERTIES OF CONGRUENTIAL RANDOM-NUMBER GENERATORS: AN EMPIRICAL VIEW", MEMORANDUM NS-340, DEPARTMENT OF STATISTICS, HARVARD UNIVERSITY, CAMBRIDGE, MASSACHUSETTS, NOVEMBER 1976.

SUBROUTINE URN16 (IX,X,NBATCH)

\*\*\*\*\*

ESTA SUBRUTINA IMPLEMENTA UN GENERADOR CONGRUENCIAL LINEAL PARA PRODUCIR NUMEROS ALEATORIOS EN PUNTO FLOTANTE. EL GENERADOR FUE DISEÑADO POR DAVID HOAGLIN COMO UNO DE SEIS, CUYO MODULO ES IGUAL A  $2^{31}-1$  QUE SUPERA BIEN EL TEST ESPECTRAL.

PROPOSITO

GENERA VECTORES DE NUMEROS ALEATORIOS DE SIMPLE PRECISION EN PUNTO FLOTANTE EN (0.0,1.0).

ESTA SUBRUTINA ES ESPECIFICA DE LOS SISTEMAS IBM/360/370 ESCRITA EN LENGUAJE ASSEMBLER, Y POR LO TANTO NO SE INCLUYE EN ESTA VERSION.

REFERENCIAS:

1. DUDEWICZ E.J., "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM

NUMBER GENERATOR PACKAGE", TECHNICAL REPORT, DEPARTMENT OF STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.

2. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.
3. HOAGLIN D.C., "THEORETICAL PROPERTIES OF CONGRUENTIAL RANDOM-NUMBER GENERATORS: AN EMPIRICAL VIEW", MEMORANDUM NS-340, DEPARTMENT OF STATISTICS, HARVARD UNIVERSITY, CAMBRIDGE, MASSACHUSETTS, NOVEMBER 1976.

SUBROUTINE URN17 (IX,X,NBATCH)

\*\*\*\*\*

ESTA SUBRUTINA IMPLEMENTA UN GENERADOR CONGRUENCIAL LINEAL PARA PRODUCIR NUMEROS ALEATORIOS EN PUNTO FLOTANTE. EL GENERADOR FUE DISEÑADO POR DAVID HOAGLIN COMO UNO DE SEIS, CUYO MODULO ES IGUAL A  $2^{*}31-1$  QUE SUPERA BIEN EL TEST ESPECTRAL.

PROPOSITO

GENERA VECTORES DE NUMEROS ALEATORIOS DE SIMPLE PRECISION EN PUNTO FLOTANTE EN (0.0,1.0).  
ESTA SUBRUTINA ES ESPECIFICA DE LOS SISTEMAS IBM/360/370 ESCRITA EN LENGUAJE ASSEMBLER, Y POR LO TANTO NO SE INCLUYE EN ESTA VERSION.

REFERENCIAS:

1. DUDEWICZ E.J., "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM NUMBER GENERATOR PACKAGE", TECHNICAL REPORT, DEPARTMENT OF STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
2. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.
3. HOAGLIN D.C., "THEORETICAL PROPERTIES OF CONGRUENTIAL RANDOM-NUMBER GENERATORS: AN EMPIRICAL VIEW", MEMORANDUM NS-340, DEPARTMENT OF STATISTICS, HARVARD UNIVERSITY, CAMBRIDGE, MASSACHUSETTS, NOVEMBER 1976.

SUBROUTINE URN18 (IX,X,NBATCH)

\*\*\*\*\*

ESTA SUBRUTINA IMPLEMENTA UN GENERADOR CONGRUENCIAL LINEAL PARA PRODUCIR NUMEROS ALEATORIOS EN PUNTO FLOTANTE. EL GENERADOR FUE DISEÑADO POR DAVID HOAGLIN COMO UNO DE SEIS, CUYO MODULO ES IGUAL A  $2^{*}31-1$  QUE SUPERA BIEN EL TEST ESPECTRAL.

PROPOSITO

GENERA VECTORES DE NUMEROS ALEATORIOS DE SIMPLE PRECISION EN PUNTO FLOTANTE EN (0.0,1.0).

ESTA SUBROUTINA ES ESPECIFICA DE LOS SISTEMAS IBM/360/370 ESCRITA EN LENGUAJE ASSEMBLER, Y POR LO TANTO NO SE INCLUYE EN ESTA VERSION.

REFERENCIAS:

1. DUDEWICZ E.J., "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM NUMBER GENERATOR PACKAGE", TECHNICAL REPORT, DEPARTMENT OF STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
2. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.
3. HOAGLIN D.C., "THEORETICAL PROPERTIES OF CONGRUENTIAL RANDOM-NUMBER GENERATORS: AN EMPIRICAL VIEW", MEMORANDUM NS-340, DEPARTMENT OF STATISTICS, HARVARD UNIVERSITY, CAMBRIDGE, MASSACHUSETTS, NOVEMBER 1976.

SUBROUTINE URN19 (IX,X,NBATCH)

\*\*\*\*\*

ESTA SUBROUTINA IMPLEMENTA UN GENERADOR CONGRUENCIAL LINEAL PARA PRODUCIR NUMEROS ALEATORIOS EN PUNTO FLOTANTE. EL GENERADOR FUE DISEÑADO POR DAVID HOAGLIN COMO UNO DE SEIS, CUYO MODULO ES IGUAL A  $2^{*}31-1$  QUE SUPERA BIEN EL TEST ESPECTRAL.

PROPOSITO

GENERA VECTORES DE NUMEROS ALEATORIOS DE SIMPLE PRECISION EN PUNTO FLOTANTE EN (0.0,1.0).

ESTA SUBROUTINA ES ESPECIFICA DE LOS SISTEMAS IBM/360/370 ESCRITA EN LENGUAJE ASSEMBLER, Y POR LO TANTO NO SE INCLUYE EN ESTA VERSION.

REFERENCIAS:

1. DUDEWICZ E.J., "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM NUMBER GENERATOR PACKAGE", TECHNICAL REPORT, DEPARTMENT OF STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
2. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.
3. HOAGLIN D.C., "THEORETICAL PROPERTIES OF CONGRUENTIAL RANDOM-NUMBER GENERATORS: AN EMPIRICAL VIEW", MEMORANDUM NS-340, DEPARTMENT OF STATISTICS, HARVARD UNIVERSITY, CAMBRIDGE, MASSACHUSETTS, NOVEMBER 1976.

SUBROUTINE URN20 (IX,X,NBATCH)

\*\*\*\*\*

ESTA SUBROUTINA IMPLEMENTA UN GENERADOR CONGRUENCIAL LINEAL PARA

PRODUCIR NUMEROS ALEATORIOS EN PUNTO FLOTANTE. EL GENERADOR FUE DISEÑADO POR DAVID HOAGLIN COMO UNO DE SEIS, CUYO MODULO ES IGUAL A  $2^{31}-1$  QUE SUPERA BIEN EL TEST ESPECTRAL.

PROPOSITO

GENERA VECTORES DE NUMEROS ALEATORIOS DE SIMPLE PRECISION EN PUNTO FLOTANTE EN (0.0,1.0).  
ESTA SUBROUTINA ES ESPECIFICA DE LOS SISTEMAS IBM/360/370 ESCRITA EN LENGUAJE ASSEMBLER, Y POR LO TANTO NO SE INCLUYE EN ESTA VERSION.

REFERENCIAS:

1. DUDEWICZ E.J., "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM NUMBER GENERATOR PACKAGE", TECHNICAL REPORT, DEPARTMENT OF STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
2. DUDEWICZ E.J. Y RALLEY T.: "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.
3. HOAGLIN D.C.: "THEORETICAL PROPERTIES OF CONGRUENTIAL RANDOM-NUMBER GENERATORS: AN EMPIRICAL VIEW", MEMORANDUM NS-340, DEPARTMENT OF STATISTICS, HARVARD UNIVERSITY, CAMBRIDGE, MASSACHUSETTS, NOVEMBER 1976.

SUBROUTINE URN21 (M1,M2,M3,NBATCH)

\*\*\*\*\*

PROPOSITO

GENERAR VECTORES DE NUMEROS ALEATORIOS DE SIMPLE PRECISION EN PUNTO FLOTANTE EN (0.0, 1.0)

USO

CALL URN21(M1,M2,M3,NBATCH)

DESCRIPCION DE PARAMETROS

M1,M2,M3...SEMILLAS DE NUMEROS ENTEROS EN EL INTERVALO (0..99999999)  
NBATCH...LA CANTIDAD DE NUMEROS ALEATORIOS A SER GENERADOS.

SUBROUTINAS Y FUNCIONES REQUERIDAS

NINGUNA

METODO

EL METODO DE GENERACION DE NUMEROS ALEATORIOS, FUE DADO POR C. GARDNER SWAIN Y MARGUERITE S. SWAIN, "A UNIFORM RANDOM NUMBER GENERATOR THAT IS REPRODUCIBLE, HARDWARE-INDEPENDENT, AND FAST", JOURNAL OF CHEMICAL INFORMATION AND COMPUTER SCIENCES, VOL 20 (FEB. 1980) PAG.56-58.

SUBROUTINE URN22 (M1,X,NBATCH)

\*\*\*\*\*

ESTA SUBRUTINA GENERA NUMEROS PSEUDO ALEATORIOS CON EL METODO  
USADO POR LA BIBLIOTECA MTH\$RANDOM DE VAX

PROPOSITO:

GENERAR VECTORES DE NUMEROS ALEATORIOS DE SIMPLE PRECISION  
EN PUNTO FLOTANTE EN (0.0, 1.0)

USO

CALL URN22(M1,NBATCH)

DESCRIPCION DE PARAMETROS

M1...SEMILLA DE NUMERO ENTERO EN EL INTERVALO (0..99999999)  
NBATCH...LA CANTIDAD DE NUMEROS ALEATORIOS A SER GENERADOS.

SUBRUTINAS Y FUNCIONES REQUERIDAS

NINGUNA

REFERENCIAS:

BIBLIOTECA DEL SISTEMA VMS, EQUIPO VAX 11/780( MTH\$RANDOM).

SUBROUTINE URN23

\*\*\*\*\*

PROPOSITO

GENERAR VECTORES DE NUMEROS ALEATORIOS DE SIMPLE PRECISION  
EN PUNTO FLOTANTE EN (0.0, 1.0)  
ESTA SUBRUTINA ES ESPECIFICA DEL SISTEMA PASCAL/MACRO VAX 11/780,  
Y POR LO TANTO NO SE INCLUYE EN ESTA VERSION.

REFERENCIAS:

BRODY T.A., "A RANDOM-NUMBER GENERATOR", COMPUTER PHYSICS  
COMMUNICATIONS, VOL 34, 1984, PAG.39-46.

SUBROUTINE URN24

\*\*\*\*\*

PROPOSITO

GENERAR VECTORES DE NUMEROS ALEATORIOS DE SIMPLE PRECISION  
EN PUNTO FLOTANTE EN (0.0, 1.0)  
ESTA SUBRUTINA ES ESPECIFICA DEL SISTEMA PASCAL/MACRO VAX 11/780,  
Y POR LO TANTO NO SE INCLUYE EN ESTA VERSION.

REFERENCIAS:

THESEN ARNE, "AN EFFICIENT GENERATOR OF UNIFORMLY DISTRIBUTED  
RANDOM VARIATES BETWEEN ZERO AND ONE", SIMULATION, VOL 44,  
JANUARY 1985, PAG.17-22.

SUBROUTINE URN25

\*\*\*\*\*

PROPOSITO

GENERAR VECTORES DE NUMEROS ALEATORIOS DE SIMPLE PRECISION  
EN PUNTO FLOTANTE EN (0.0, 1.0)  
ESTA SUBROUTINA ES ESPECIFICA DEL SISTEMA PASCAL/MACRO VAX 11/780,  
Y POR LO TANTO NO SE INCLUYE EN ESTA VERSION.

REFERENCIAS:

1. JASON GAIT, "A NEW NONLINEAR PSEUDORANDOM NUMBER GENERATOR",  
IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, SEPT. 1977, PAG.  
359-363.
2. DATA ENCRYPTION STANDARD, FIPS PUBLICATION, 15 JANUARY 1977.

SUBROUTINE URN26

\*\*\*\*\*

PROPOSITO

GENERAR VECTORES DE NUMEROS ALEATORIOS DE SIMPLE PRECISION  
EN PUNTO FLOTANTE EN (0.0, 1.0)  
ESTA SUBROUTINA ES ESPECIFICA DEL SISTEMA PASCAL/MACRO VAX 11/780,  
Y POR LO TANTO NO SE INCLUYE EN ESTA VERSION.

METODO

ES UNA VARIANTE DEL ALGORITMO:  
THESEN ARNE, "AN EFFICIENT GENERATOR OF UNIFORMLY DISTRIBUTED  
RANDOM VARIATES BETWEEN ZERO AND ONE", SIMULATION, VOL 44,  
JANUARY 1985, PAG.17-22.

SUBROUTINE URN27

\*\*\*\*\*

PROPOSITO

GENERAR VECTORES DE NUMEROS ALEATORIOS DE SIMPLE PRECISION  
EN PUNTO FLOTANTE EN (0.0, 1.0) POR EL METODO DE TAUSWORTHE.

ESTA SUBROUTINA ES ESPECIFICA DEL SISTEMA PASCAL/MACRO VAX 11/780,  
Y POR LO TANTO NO SE INCLUYE EN ESTA VERSION.



REFERENCIAS:

1. HENRIKSEN J.O. Y CRANE R.C., GPSSH USER'S MANUAL, WOLVERINE SOFTWARE CORPORATION, 1983, P.4-10.
2. WHITTLESEY JOHN R.B., "A COMPARISON OF THE CORRELATIONAL BEHAVIOR OF RANDOM NUMBER GENERATORS FOR THE IBM 360", COMMUNICATIONS OF THE ACM, SEPTEMBER 1968, PAG. 642-643.

SUBROUTINE URN28(IX,X,NBATCH)

\*\*\*\*\*

PROPOSITO

GENERAR NUMEROS PSEUDOALEATORIOS EN (0,1) EN PUNTO FLOTANTE CON SIMPLE PRECISION

USO

CALL URN28(IX,X,NBATCH)

DESCRIPCION DE PARAMETROS

IX.....VALOR INICIAL DEL GENERADOR, PUEDE SER: 1,..., 2\*\*32-2  
X.....ES EL VECTOR DE LOS NUMEROS ALEATORIOS EN (0.0, 1.0)  
NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR MEDIANTE ESTA SUBROUTINA

OBSERVACIONES

IX NO DEBE SER ALTERADO POR EL USUARIO DURANTE LA EJECUCION DEL PROGRAMA SALVO QUE SE DESEE REPETIR UNA SUCESION DE NUMEROS

SUBROUTINAS Y FUNCIONES REQUERIDAS

NINGUNA

REFERENCIAS:

SCHRAGE LINUS, "A MORE PORTABLE FORTRAN RANDOM NUMBER GENERATOR." ACM TRANSACTIONS ON MATHEMATICAL SOFTWARE, JUNIO 1979, PAG.132-138.

SUBROUTINE URN29

\*\*\*\*\*

PROPOSITO

GENERAR VECTORES DE NUMEROS ALEATORIOS DE SIMPLE PRECISION EN PUNTO FLOTANTE EN (0.0, 1.0)  
ESTA SUBROUTINA ES ESPECIFICA DEL SISTEMA PASCAL/MACRO VAX 11/780, Y POR LO TANTO NO SE INCLUYE EN ESTA VERSION.

METODO:

ES UN ALGORITMO ANALOGO AL URN01 UTILIZADO EN UNA APLICACION

REFERENCIAS:

SCHMIDT D., "THE 1870S(N,GAMMA) CROSS SECTION AND THE AGE OF THE CHEMICAL ELEMENTS", DENISON UNIVERSITY HONORS PROJECT,1986.

SUBROUTINE URN30(X,NBATCH)

\*\*\*\*\*

PROPOSITO

GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE PRECISION EN (0.0,1.0)

USO

CALL URN30(X,NBATCH)

DESCRIPCION DE PARAMETROS

ISTRM...SEMILLA NUMERO ENTERO ENTRE 1 Y 21474  
IZ...VALOR ACTUAL DEL ARREGLO ZRNG(ISTRM) DE 100 POSICIONES  
NBATCH.....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR MEDIANTE ESTA SUBRUTINA

SUBRUTINAS Y FUNCIONES REQUERIDAS

NINGUNA

REFERENCIAS:

MARSE Y ROBERTS (1983)

SUBROUTINE URN31(X,NBATCH)

\*\*\*\*\*

PROPOSITO

GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE PRECISION EN (0.0,1.0)

USO

CALL URN31(X,NBATCH)

DESCRIPCION DE PARAMETROS

IX...SEMILLA NUMERO ENTERO  
NBATCH.....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR MEDIANTE ESTA SUBRUTINA

SUBRUTINAS Y FUNCIONES REQUERIDAS

NINGUNA

REFERENCIAS:

H. MIYAZAKI (1987)

SUBROUTINE URN32 (X,NBATCH)

\*\*\*\*\*

PROPOSITO

GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE  
PRECISION EN (0.0,1.0)

USO

CALL URN32(X,NBATCH)

DESCRIPCION DE PARAMETROS

IX...SEMILLA NUMERO ENTERO

NBATCH.....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENE-  
RAR MEDIANTE ESTA SUBRUTINA

SUBRUTINAS Y FUNCIONES REQUERIDAS

NINGUNA

REFERENCIAS:

H. MIYAZAKI (1987)

SUBROUTINE URN33(X,NBATCH)

\*\*\*\*\*

PROPOSITO

GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE  
PRECISION EN (0.0,1.0)

USO

CALL URN33(X,NBATCH)

DESCRIPCION DE PARAMETROS

XI...SEMILLA NUMERO ENTERO

NBATCH.....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENE-  
RAR MEDIANTE ESTA SUBRUTINA

SUBRUTINAS Y FUNCIONES REQUERIDAS

NINGUNA

METODO

COMBINACION DE DOS SUCESIONES

REFERENCIAS:

H. MIYAZAKI (1987)

SUBROUTINE URN34 (X,NBATCH)  
\*\*\*\*\*

PROPOSITO  
GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE  
PRECISION EN (0.0,1.0)

USO  
CALL URN34(X,NBATCH)

DESCRIPCION DE PARAMETROS  
KW1...SEMILLA NUMERO ENTERO PARA LA SUCESION {wi}  
KY1... SEMILLA DE LA SUCESION {Yi}  
KZ1... SEMILLA DE LA SUCESION {Zi}  
XR1... VALOR INICIAL REAL PARA EL CALCULO DE INDICES DE LA TABLA  
NBATCH.....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENE-  
RAR MEDIANTE ESTA SUBRUTINA

SUBRUTINAS Y FUNCIONES REQUERIDAS  
NINGUNA

METODO  
COMBINACION DE TRES SUCESIONES CON PERMUTACIONES EN UNA TABLA

REFERENCIAS:  
F. SEZGIN (1987)

SUBROUTINE URN35 (X,NBATCH)  
\*\*\*\*\*

PROPOSITO  
GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE  
PRECISION EN (0.0,1.0)

USO  
CALL URN35(X,NBATCH)

DESCRIPCION DE PARAMETROS  
KW1...SEMILLA NUMERO ENTERO PARA LA SUCESION {wi}  
KY1... SEMILLA DE LA SUCESION {Yi}  
KZ1... SEMILLA DE LA SUCESION {Zi}  
XR1... VALOR INICIAL REAL  
NBATCH.....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENE-  
RAR MEDIANTE ESTA SUBRUTINA

SUBRUTINAS Y FUNCIONES REQUERIDAS  
NINGUNA

METODO  
UTILIZA CUATRO SUCESIONES.

REFERENCIAS:

L' ECUYER P. (1987)

SUBROUTINE URN36

\*\*\*\*\*

PROPOSITO

GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE  
PRECISION EN (0.0, 1.0)

USO

CALL URN36 (X,NBATCH)

DESCRIPCION DE PARAMETROS

IA.....DIMENSION VECTOR DE NUMEROS ENTEROS  
IR.....DIMENSION VECTOR DE NUMEROS REALES  
NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR  
          MEDIANTE ESTA SUBROUTINA  
INTIN(1),INTIN(2),INTIN(3),INTIN(4),INTIN(5) VALORES ENTEROS

SUBROUTINAS Y FUNCIONES REQUERIDAS

NINGUNA

REFERENCIAS:

DUDEWICZ E., KARIAN Z. Y MARSHALL R., "RANDOM NUMBER GENERATION  
ON MICROCOMPUTERS', THE SOCIETY FOR COMPUTER SIMULATION,1985,  
PAG.9-14

SUBROUTINE URN37

\*\*\*\*\*

PROPOSITO

GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE  
PRECISION EN (0.0,1.0)

USO

CALL URN37 (X,NBATCH)

DESCRIPCION DE PARAMETROS

IA.....DIMENSION VECTOR DE NUMEROS ENTEROS  
IR.....DIMENSION VECTOR DE NUMEROS REALES  
NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR  
          MEDIANTE ESTA SUBROUTINA  
INTIN(1) VALOR ENTERO

SUBROUTINAS Y FUNCIONES REQUERIDAS

NINGUNA

REFERENCIAS:

DUDEWICZ E., KARIAN Z. Y MARSHALL R., "RANDOM NUMBER GENERATION ON MICROCOMPUTERS", THE SOCIETY FOR COMPUTER SIMULATION, 1985, PAG.9-14

SUBROUTINE URN38

\*\*\*\*\*

PROPOSITO

GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE PRECISION EN (0.0,1.0)

USO

CALL URN38 (X,NBATCH)

DESCRIPCION DE PARAMETROS

IA.....DIMENSION VECTOR DE NUMEROS ENTEROS  
IR.....DIMENSION VECTOR DE NUMEROS REALES  
NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR MEDIANTE ESTA SUBROUTINA  
INTIN(1) VALOR ENTERO

SUBROUTINAS Y FUNCIONES REQUERIDAS

NINGUNA

REFERENCIAS:

DUDEWICZ E., KARIAN Z. Y MARSHALL R. , "RANDOM NUMBER GENERATION ON MICROCOMPUTERS', THE SOCIETY FOR COMPUTER SIMULATION, 1985, PAG.9-14

SUBROUTINE URN39(X,NBATCH)

\*\*\*\*\*

PROPOSITO

GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE PRECISION EN (0.0,1.0)

USO

CALL URN39(X,NBATCH)

DESCRIPCION DE PARAMETROS

SEED .....SEMILLA NUMERO REAL SIMPLE PRECISION IGUAL A  $2/(1+\text{SQRT}(5))$   
TOTAL39 ...VALOR INICIAL REAL SIMPLE PRECISION  
NBATCH.....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR MEDIANTE ESTA SUBROUTINA

SUBROUTINAS Y FUNCIONES REQUERIDAS

NINGUNA

REFERENCIAS:

REY W.J.J., "CAN I MY RANDOM GENERATOR?", ABSTRACT, INTERN.

CONFERENCE ON BOOTSTRAPPING AN RELATED TECHNIQUES, UNIVERSITY OF  
TRIER, GERMANY (1990)

SUBROUTINE URN40 (X,NBATCH)

\*\*\*\*\*

PROPOSITO

GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE  
PRECISION EN (0.0,1.0)

USO

CALL URN40(X,NBATCH)

DESCRIPCION DE PARAMETROS

SEED .....SEMILLA NUMERO REAL SIMPLE PRECISION

TOTAL40 ...VALOR INICIAL REAL SIMPLE PRECISION

NBATCH.....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENE-  
RAR MEDIANTE ESTA SUBRUTINA

SUBRUTINAS Y FUNCIONES REQUERIDAS

NINGUNA

METODO

UTILIZA TRES SUCESIONES Y DESPLAZAMIENTOS DE BITS

REFERENCIAS:

MARSAGLIA Y ZAMAN

## Capítulo 3

### LAS PRUEBAS

En el cuadro N°2 se listan los pruebas estadísticas que integran GENTSTPC junto con una breve descripción de la función que cada uno realiza y, en particular, si el test almacena valores para luego emplearse en los tests Chi-Cuadrado o Kolmogorov-Smirnov cuando se invoca TST01.

Además se indica: el propósito de la prueba, su uso, una descripción de los parámetros para invocar las respectivas subrutinas, algunos comentarios específicos, los nombres de las subrutinas requeridas, y algunas referencias bibliográficas.

Todos las pruebas fueron escritas en Fortran.

Una completa descripción de cada prueba y los desarrollos teóricos de las fórmulas pueden encontrarse en el cap.3 del libro de Donald Knuth, "The Art of Computer Programming, Volume 2/ SemiNumerical Algorithms".

La prueba TST01 tiene una particular potencia y efectividad. Si los números producidos por un generador son realmente aleatorios entonces una de cada veinte veces la prueba Chi cuadrado debería fracasar con un nivel del 5%.

Almacenando los valores  $C^2$  generados por una prueba, y luego sometiendo esos valores a un test Chi-cuadrado, para medir cuantitativamente su ajuste a una distribución esperada, se obtiene una cantidad que mide la aleatoriedad correspondiente a grandes sucesiones de números producidos por el generador en cuestión.

De manera análoga se puede proceder para el caso de la distribución de Kolmogorov-Smirnov. Valores K-S provenientes de la prueba de bondad de ajuste o del test máximo de t, pueden almacenarse y luego aplicarles el test K-S.

Como el volumen de las salidas producidas por el test TST01 es muy importante, se recomienda leer los detalles de su utilización.

Con referencia a las Pruebas de Entropía, el paquete TSTENTR, elaborado teniendo en cuenta las más recientes aportaciones en pruebas de entropía a distribuciones continuas, requiere como entrada, un archivo de datos numéricos donde se almacenan los números aleatorios producidos por el generador que se desea probar, pudiendo accederse a través de opciones que incluyan además, las pruebas empíricas, las pruebas teóricas, o sólo de entropía.

Comparando con el test TST01(CS2KS2), el tiempo de procesamiento de la prueba de entropía es aproximadamente 4.6 veces la duración del test CS2KS2.



**TABLA DE PRUEBAS**

NOMBRE	DESCRIPCION	GENERA VALORES	
		Chi-cuadr.sobre cuadrado	Chi- K-S sobre K-S
TST01	Realiza Chi-cuadrado sobre Chi-cuad. y K-S sobre K-S		
TST02	Distribución Uniforme (Chi-cuadrado simple)	SI	
TST03	Test del producto rezagado		
TST04	Gap	SI	
TST05	Recolector de cupones	SI	
TST06	Bondad de ajuste de K-S		SI
TST07	Permutación	SI	
TST08	Póker	SI	
TST09	Corridas	SI	
TST10	Pares Seriales	SI	
TST11	Máximo de t		SI
Tests Especiales para Generadores Lineales Congruenciales			
TSTM1	Correlación Serial		
TSTM2	Potencia y Período		
TSTM3	Factorización en Factores Primos		
TSTM4	Test Espectral		
TSTENTR	Tests de entropía continua		

## COMENTARIOS SOBRE EL CODIGO FUENTE DE LAS PRUEBAS DE GENTSTPC

SUBROUTINE TST01

\*\*\*\*\*

TST01 APLICA EL TEST CHI-CUADRADO A VARIABLES SUPUESTAS CHI-CUADRADO Y EL TEST KOLMOGOROV-SMIRNOV A VARIABLES CON DISTRIBUCION SUPUESTA K-S. LAS VARIABLES SUPUESTAS COMO CHI-CUADRADO Y K-S SURGEN DE APLICAR DISTINTOS TESTS AL GENERADOR DE NUMEROS ALEATORIOS PROPUESTO POR EL USUARIO. (TST01 PUEDE SER LLAMADO SOLO UNA VEZ POR GENERADOR)

### PROPOSITO

PRUEBA CHI-CUADRADO A VARIABLES SUPUESTAS CHI-CUADRADO Y PRUEBA K-S A VARIABLES CON DISTRIBUCION SUPUESTA K-S

### USO

PARTE DEL PROCESO DE TST01 SE HACE EN LA RUTINA "REPORT" POR CONVENIENCIA DE PROGRAMACION. ANTES DE SER LLAMADA ESTA RUTINA (TST01) LEE LOS SIGUIENTES 6 PARAMETROS:

VARIABLE	LONG.	NUM./CAR.
TNAME	8	CAR
NCHI	11	NUM
NPRNT	11	NUM
TCODE	8	CAR
CSCODE	8	NUM
KSCODE	8	NUM

### DESCRIPCION DE PARAMETROS

TNAME..DEBE SER "TST01" PARA QUE SE INVOQUE LA RUTINA  
NCHI...CANTIDAD (ENTRE 1 Y 1000) DE LOTES DE NUMEROS ALEAT.(CADA UNO CONTENIENDO NBATCH NUMEROS) A LOS QUE SE APLICARA EL TEST ESPECIFICADO EN CSCODE O KSCODE.  
NPRNT..CANTIDAD DE LOTES PARA LOS QUE SE IMPRIMIRAN LOS RESULTADOS.(SI NCHI=NPRNT=1000 Y SE ESPECIFICAN LOS 7 TESTS, SE IMPRIMIRAN 7000 PAGINAS). NPRNT DEBE SER RELATIVAMENTE PEQUEÑO(NPRNT=10)PARA TENER UNA MUESTRA IMPRESA DE LOS RESULTADOS OBTENIDOS.  
NPRNT PUEDE SER CUALQUIER ENTERO DE 0 A NCHI  
TCODE..ES "CSCS" SI SE DESEA APLICAR EL TEST CHI-CUADRADO A LAS VARIABLES SUPUESTAS CHI-CUAD. "KSKS" SI SE DESEA APLICAR EL TEST K-S A VARIABLES CON DISTRIBUCION SUPUESTA K-S Y "CS2KS2" SI SE DESEA APLICAR AMBOS.  
CSCODE..INDICA CUAL DE LOS TESTS TST02,TST04,TST05,TST07,TST08, TST09,TST10 SE DESEA APLICAR.  
RESPECTIVAMENTE, "1" INDICA QUE SE SELECCIONA Y "0" QUE NO(EJEMPLO:CSCODE=1111011 SELECCIONARA TODOS SALVO TST08,MIENTRAS QUE CSCODE=1111111 SELECCIONARA TODOS)  
KSCODE..INDICA SI SE APLICA TST06 Y/O TST11 PARA LAS VARIABLES CON DISTRIBUCION CONSIDERADA COMO K-S

### OBSERVACIONES:

TST01 ES UN TEST MUY IMPORTANTE, CUYO USO DEBE SER ESTUDIADO CON CUIDADO ANTES DE SER EMPLEADO.

### SUBROUTINAS Y FUNCIONES REQUERIDAS

POINTS, XTRMS, CDFKS, SORT

REFERENCIAS:

1. DUDEWICZ E.J., "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM NUMBER GENERATOR PACKAGE", TECHNICAL REPORT, DEPARTMENT OF STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
2. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.

SUBROUTINE TST02

\*\*\*\*\*

TST02 APLICA EL TEST CHI-CUADRADO PARA UNA DISTRIBUCION UNIFORME

PROPOSITO

DADA UNA LISTA DE N NUMEROS REALES EN PUNTO FLOTANTE ENTRE 0 Y 1 QUE SE SUPONEN ALEATORIOS, SE DETERMINAN CUANTOS CAEN EN CADA UNO DE LOS 100 INTERVALOS (0.00,0.01), (0.01,0.02), ..., (0.98,0.99), (0.99,1.00).

EL TEST CHI-CUADRADO (CON 99 GRADOS DE LIBERTAD) CONSISTE EN DETERMINAR SI LOS NUMEROS EN LOS INTERVALOS CONCUERDAN CON EL MODELO ESPECIFICADO, EN ESTE CASO EL MODELO DE ALEATORIEDAD BAJO EL CUAL UNO "ESPERA" ENCONTRAR N/100 NUMEROS EN CADA UNO DE LOS 100 INTERVALOS.

USO

LA CANTIDAD DE NUMEROS QUE SE ALMACENARAN ES SIEMPRE "NBATCH", EL ARREGLO ENTERO. TST02 UTILIZA LOS SIGUIENTES PARAMETROS:

VARIABLE	LONG.	NUM./CAR.
TNAME	8	CAR

DESCRIPCION DE PARAMETROS

TNAME...TST02

OBSERVACIONES:

EL TEST SE REALIZA A LOS "NBATCH" NUMEROS.

SUBROUTINAS Y FUNCIONES REQUERIDAS:

NINGUNA

REFERENCIAS:

1. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.

TST03 APLICA EL TEST DEL PRODUCTO REZAGADO (LAGGED CORRELATION)

\*\*\*\*\*

PROPOSITO

DADA UNA LISTA DE N NUMEROS REALES EN PUNTO FLOTANTE ENTRE 0 Y 1 QUE SE SUPONEN ALEATORIOS, SE ESPERA QUE EL LAG L DE CORRELACION SERIAL SEA PEQUEÑO. UNA REGLA HEURISTICA (AUN NO JUSTIFICADA)

TEORICAMENTE EN LA LITERATURA) ES QUE EL 95% DE LAS VECES SE PUEDE ENCONTRAR, BAJO ALEATORIEDAD, UN LAG L DE CORRELACION SERIAL ENTRE  $-(2*\sqrt{N}+1)/N$  Y  $(2*\sqrt{N}-1)/N$ . SI ESTO NO SE VERIFICA, EL GENERADOR ES SOSPECHOSO. COMO SE HAN PROPUESTO DISTINTAS FORMULAS DE CORRELACION SERIAL PARA UN LAG L, CALCULAMOS 4 DE ELLAS PARA QUE EL USUARIO PUEDA VER COMO AFECTA AL RESULTADO LA FORMULA ELEGIDA; SE ESPERA QUE LA DIFERENCIA POR CAMBIO DE FORMULA SEA MINIMA.

USO

PARAMETROS UTILIZADOS POR TST03:

VARIABLE	LONG.	NUM./CAR.
TNAME	8	CAR
IPARM(1)	11	NUM
IPARM(2)	11	NUM

DESCRIPCION DE PARAMETROS

IPARM(1)...CANTIDAD (ENTRE IPARM(2)+2 Y NBATCH, INCLUSIVE) DE NUMEROS ALEATORIOS A SER USADOS EN EL TEST DE CORRELACION SERIAL

IPARM(2)...CANTIDAD DE LAGS PARA LOS QUE EL LAG DE CORRELACION SERIAL L SERA CALCULADO. SE DEBE TENER  $1 \leq IPARM(2) \leq 20$ . EL VALOR POR DEFECTO ES  $IPARM(2)=20$ .

OBSERVACIONES:

SUBROUTINAS Y FUNCIONES REQUERIDAS  
NINGUNA

REFERENCIAS:

1. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.
2. KNUTH D.E., "THE ART OF COMPUTER PROGRAMMING, VOLUME 2/ SEMINUMERICAL ALGORITHMS", ADDISON-WESLEY PUBLISHING COMPANY, INC., READING, MASSACHUSETTS, 1969, PAG. 64-65.

SUBROUTINE TST04

\*\*\*\*\*

TST04 APLICA EL TEST DEL COLECCIONISTA DE CUPONES PARA PROBAR ALEATORIEDAD.

PROPOSITO:

DADA UNA LISTA DE N NUMEROS REALES EN PUNTO FLOTANTE ENTRE 0 Y 1 QUE SE SUPONEN ALEATORIOS, SE CONVIERTEN A NUMEROS ENTEROS 1, 2, ..., D DE

LA SIGUIENTE FORMA:

REEMPLAZAR X POR 1 SI  $0.0 \leq X < 1/D$   
 REEMPLAZAR X POR 2 SI  $1/D \leq X < 2/D$   
 REEMPLAZAR X POR 3 SI  $2/D \leq X < 3/D$

. . .  
. . .

REEMPLAZAR X POR D SI  $(D-1)/D \leq X \leq 1.0$ .

LUEGO LA NUEVA SUCESION  $\{Y_i\}$  DE NUMEROS ENTEROS DEBE SER ALEATORIA.

PARA COMPROBARLO APLICAMOS EL TEST DEL COLECCIONISTA DE CUPONES QUE CONSISTE EN BUSCAR SUB-SUCESIONES O SEGMENTOS DE NUMEROS ENTEROS DONDE SE PUEDA IDENTIFICAR A LOS ELEMENTOS  $1, 2, \dots, D$  QUE APARECEN EN CUALQUIER ORDEN EN  $\{Y_i\}$ , MEDIANTE UNA BUSQUEDA PROGRESIVA.

UNA VEZ COMPLETADO UN CONJUNTO  $\{1, 2, \dots, D\}$  SE CUENTAN LOS ELEMENTOS QUE FUE NECESARIO OBSERVAR PARA OBTENER ESE CONJUNTO. TAL CANTIDAD SE LLAMA LONGITUD DE LA SUBSUCESION O SEGMENTO. SE RECOMIENZA CON EL ELEMENTO QUE SIGUE INMEDIATAMENTE EN  $\{Y_i\}$  PARA FORMAR UN NUEVO CONJUNTO COMPLETO. SEA Q LA CANTIDAD DE TERMINOS DE LA SUCESION QUE DEBEN EXAMINARSE PARA ENCONTRAR UN CONJUNTO COMPLETO  $1, 2, \dots, D$ .

ENTONCES LOS POSIBLES VALORES DE Q SON  $D, D+1, D+2, \dots$ . CUANDO LA VERDADERA ALEATORIEDAD PREVALECE, LAS PROBABILIDADES SIGUIENTES SON CONOCIDAS.

$P(Q=D)$

$P(Q=D+1)$

$P(Q=D+2)$

.

.

.

SI SE CALCULAN TALES VALORES Q REPETIDAMENTE, UN TEST CHI-CUADRADO PUEDE REALIZARSE PARA VER SI LOS NUMEROS OBSERVADOS EN CADA CATEGORIA CONCUERDAN CON LAS PROBABILIDADES TEORICAS. ASI SE PRODUCE UN TEST DE ALEATORIEDAD DE LA SUCESION BASICA DE NUMEROS.

ESTE TEST SE REALIZA PARA TRES CASOS:

D=5 CON CATEGORIAS

Q=5

Q=6

.

.

.

Q=19

Q>=20;

D=5 APLICADO A  $FR(100X)$ , DONDE  $FR(Z)$  DENOTA LA PARTE FRACCIONARIA DE Z, CON CATEGORIAS

Q=5

Q=6

.

.

.

Q=19

Q>=20;

D=10 APLICADO A LOS X CON CATEGORIAS

$10 \leq Q \leq 19$

$20 \leq Q \leq 23$

$24 \leq Q \leq 27$

$28 \leq Q \leq 32$

$33 \leq Q \leq 39$

Q>=40

USO:

TST04 EMPLEA LOS SIGUIENTES PARAMETROS:

VARIABLE	LONG.	NUM./CAR.
TNAME	8	CAR

IPARM(1)	11	NUM
IPARM(2)	11	NUM

DESCRIPCION DE PARAMETROS

IPARM(1)...INDICE PARA COMIENZO DE LA LISTA DE N BATCH NUMEROS ALEATORIOS  
 IPARM(2)...INDICE PARA EL FINAL DE LA LISTA DE N BATCH NUMEROS ALEATORIOS

SUBROUTINAS Y FUNCIONES REQUERIDAS

PNTCHI (NO IMPLEMENTADA EN ESTA VERSION)

REFERENCIAS:

1. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.
2. GREENWOOD R.E., "COUPON COLLECTOR'S TEST FOR RANDOM DIGITS", MATHEMATICAL TABLES AND OTHER AIDS TO COMPUTATION, VOL.9(1955), PAG.1-5.

SUBROUTINE TST05

\*\*\*\*\*

TST05 APLICA EL TEST DE INTERVALOS (GAPS) O HUECOS PARA ALEATORIEDAD

PROPOSITO:

DADA UNA LISTA DE N NUMEROS REALES EN PUNTO FLOTANTE ENTRE 0 Y 1 QUE SE SUPONEN ALEATORIOS, SE EXAMINA CADA UNO PARA VER SI SE ENCUENTRA O NO ENTRE LOS NUMEROS ALFA Y BETA (ALFA < BETA). CADA NUMERO QUE ESTA ENTRE ALFA Y BETA ES REEMPLAZADO POR UN 1 Y LOS QUE NO, POR UN 0.  
 ESTO CONVIERTE LOS N NUMEROS EN UNA SUCESION DE 0 Y 1.  
 SEA  $P = BETA - ALFA$ . SI LOS N NUMEROS SON ALEATORIOS, LA PROBABILIDAD DE QUE OCURRAN J CEROS DESPUES DE UN 1 Y ANTES DEL SIGUIENTE 1 ES:  
 $P(J) = P * ((1 - P) ** J)$ ,  $J = 0, 1, 2, \dots$

LUEGO DE OBSERVAR LA FRECUENCIA EN QUE OCURREN LAS CATEGORIAS, SE REALIZA UN TEST CHI-CUADRADO PARA VER SI LOS NUMEROS OBSERVADOS EN CADA CATEGORIA CONCUERDAN CON LAS PROBABILIDADES TEORICAS. ASI PRODUCE UN TEST DE ALEATORIEDAD DE LA SUCESION BASICA DE NUMEROS.  
 KENDALL Y BABINGTON SMITH PROPUSIERON ESTE TEST EN UN CONTEXTO SUCESIONES DE LOS DIGITOS 0,1,2,3,4,5,6,7,8,9) QUE ES EQUIVALENTE A ELEGIR ALFA=0.0 Y BETA=0.1. EL PROCEDIMIENTO SE LLAMA TEST "GAP". EL TEST HA SIDO CODIFICADO PARA SER REALIZADO CON TRES VALORES DE (ALFA, BETA):  
 (ALFA, BETA) = (0.0, 0.5)... LLAMADO TEST "CORRIDAS POR DEBAJO DE LA MEDIA"  
 (ALFA, BETA) = (0.5, 1.0)... LLAMADO TEST "CORRIDAS POR ARRIBA DE LA MEDIA"  
 (ALFA, BETA) = (0.333, 0.667) ... ESPECIFICADO POR EL USUARIO

USO

LOS PARAMETROS DE LA RUTINA TST05 SON:

VARIABLE	LONG.	NUM./CAR.
TNAME	8	CAR
IPARM(1)	11	NUM
IPARM(2)	11	NUM
IPARM(3)	11	NUM
IPARM(4)	11	NUM

DESCRIPCION DE PARAMETROS

IPARM(1)...ENTERO 1000\*ALFA, DONDE ALFA TIENE A LO SUMO 3 DECIMALES

IPARM(2)...ENTERO 1000\*BETA, DONDE BETA TIENE A LO SUMO 3 DECIMALES

IPARM(3)...INDICE PARA COMENZAR EN LA LISTA DE NUMEROS ALEATORIOS

IPARM(4)...INDICE PARA TEMINAR EN LA LISTA DE NUMEROS ALEATORIOS

OBSERVACIONES:

EL NUMERO GN DE GAPS USADOS ES EL NUMERO (ALEATORIO) ENCONTRADO ENTRE LOS (IPARM(4)-IPARM(3)+1) NUMEROS ANALIZADOS. TODAS LAS CATEGORIAS J DONDE GN\*P(J)>=4 SON USADAS EN EL TEST CHI-CUADRADO COMO CATEGORIAS INDIVIDUALES MIENTRAS QUE EL RESTO DE LAS CATEGORIAS SON AGRUPADAS EN UNA SOLA PARA EL TEST CHI-CUADRADO.

SUBROUTINAS Y FUNCIONES REQUERIDAS

PNTCHI(NO IMPLEMENTADA EN ESTA VERSION), POINTS

REFERENCIAS :

1. DUDEWICZ E.J. Y RALLEY T.. "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.
2. KENDALL M.G. Y BABINGTON SMITH B.. "RANDOMNESS AND RANDOM SAMPLING NUMBERS", JOURNAL OF THE ROYAL STATISTICAL SOCIETY, VOL.101(1938), PAG. 147-166.

SUBROUTINE TST06

\*\*\*\*\*

PROPOSITO:

DADA UNA LISTA DE N NUMEROS REALES EN PUNTO FLOTANTE ENTRE 0 Y 1 QUE SE SUPONEN ALEATORIOS, EL TEST KOLMOGOROV-SMIRNOV EVALUA SU ALEATORIEDAD COMPARANDO LAS DESVIACIONES DE SU FUNCION DE DISTRIBUCION, EN ESTE CASO, SUPUESTA UNIFORME EN 0.0 A 1.0, CON LA FUNCION DE DISTRIBUCION EMPIRICA. ESTAS DESVIACIONES PUEDEN EVALUARSE EMPLEANDO LAS MEDIDAS:

$K+ = \text{SQRT}(N) * \text{MAX}(\text{EMPIRICA}(X) - \text{ESTIMADA}(X))$   
 $K- = \text{SQRT}(N) * \text{MAX}(\text{ESTIMADA}(X) - \text{EMPIRICA}(X))$   
 $K* = \text{MAX}(K+, K-)$

TST06 CALCULA E IMPRIME K+, K-, K\* CON LOS PUNTOS PORCENTUALES DE SU RESPECTIVO A DISTRIBUCION

USO

EL ARREGLO DE NBATCH NUMEROS ALEATORIOS ALMACENADOS ES USADO PARA REASIGNAR VALORES AL ARREGLO ORIGINAL LUEGO DE HABER EJECUTADO TST06.

LOS PARAMETROS DE LA RUTINA TST06 SON:

VARIABLE	LONG.	NUM./CAR.
TNAME	8	CAR
IPARM(1)	11	NUM
IPARM(2)	11	NUM
IPARM(3)	11	NUM

DESCRIPCION DE PARAMETROS

IPARM(1)...CANTIDAD (ENTRE 1 Y NBATCH) DE NUMEROS ALEAT. A SER USADOS EN CADA TEST K-S

IPARM(2)...CANTIDAD DE CONJUNTOS DE IPARM(1) NUMEROS CADA UNO QUE SERAN TESTEADOS. DEBE SER IPARM(1)\*IPARM(2)<=NBATCH.

IPARM(3)...INDICE PARA COMENZAR EN LA LISTA DE NUMEROS ALEATORIOS DEBE VERIFICARSE: IPARM(3)-1+IPARM(1)\*IPARM(2)<= NBATCH.

SUBROUTINAS Y FUNCIONES REQUERIDAS

KSTPL, CDFKS, XTRMS, SORT

REFERENCIAS:

1. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.

SUBROUTINE TST07

\*\*\*\*\*

TST07 APLICA EL TEST DE PERMUTACION PARA PROBAR ALEATORIEDAD

PROPOSITO:

DADA UNA LISTA DE N NUMEROS REALES EN PUNTO FLOTANTE ENTRE 0 Y 1 QUE SE SUPONEN ALEATORIOS, SE TOMAN SUCEIVOS CONJUNTOS DE T NUMEROS. CADA UNO DE ESOS CONJUNTOS TIENE T-FACTORIAL ORDENES POSIBLES (CUANDO CLASIFICAMOS LOS T NUMEROS COMO MAYOR, SEGUNDO MAYOR, ..., MENOR), Y CUANDO PREVALECE ALEATORIEDAD VERDADERA, LAS PROBABILIDADES DE LOS T-FACTORIAL ORDENES SON IGUALES A 1/(T-FACTORIAL). LUEGO DE OBSERVAR LA FRECUENCIA EN QUE OCURRE CADA ORDENAMIENTO CUANDO SE EXAMINAN MUCHOS GRUPOS DE T ENTEROS, SE REALIZA UN TEST CHI-CUAD. PARA VER SI LOS NUMEROS OBSERVADOS EN CADA CATEGORIA ESTAN DE ACUERDO CON IGUALES PROBABILIDADES PARA LAS CATEGORIAS, ASI SE PRODUCE UN TEST DE ALEATORIEDAD DE LA SUCESION DE NUMEROS.

ESTE TEST ES REALIZADO PARA T=3,4,5.

LOS PARAMETROS DE TST07 SON:

VARIABLE	LONG.	NUM./CAR.
TNAME	8	CAR
IPARM(1)	11	NUM



## DESCRIPCION DE PARAMETROS

IPARM(1)...INDICE DEL PRIMER NUMERO EN LA LISTA DE NUMALEA NUMEROS ALEATORIOS.

IPARM(2)...INDICE DEL ULTIMO NUMERO EN LA LISTA DE NABTCH NUMEROS ALEATORIOS.

## SUBROUTINAS Y FUNCIONES REQUERIDAS:

PNTCHI(NO IMPLEMENTADA EN ESTA VERSION)

## REFERENCIAS:

1. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.

## SUBROUTINE TST08

\*\*\*\*\*

TST08 APLICA EL TEST DEL POKER

## PROPOSITO:

DADA UNA LISTA DE N NUMEROS REALES EN PUNTO FLOTANTE ENTRE 0 Y 1 QUE SE SUPONEN ALEATORIOS, SE LOS CONVIERTE A NUMEROS ENTEROS 1,2,3,4,5,6,7,8,9,10 DE LA SIGUIENTE FORMA:

REEMPLAZAR X POR 1 SI  $0.0 \leq X < 0.1$

REEMPLAZAR X POR 2 SI  $0.1 \leq X < 0.2$

REEMPLAZAR X POR 3 SI  $0.2 \leq X < 0.3$

REEMPLAZAR X POR 4 SI  $0.3 \leq X < 0.4$

REEMPLAZAR X POR 5 SI  $0.4 \leq X < 0.5$

REEMPLAZAR X POR 6 SI  $0.5 \leq X < 0.6$

REEMPLAZAR X POR 7 SI  $0.6 \leq X < 0.7$

REEMPLAZAR X POR 8 SI  $0.7 \leq X < 0.8$

REEMPLAZAR X POR 9 SI  $0.8 \leq X < 0.9$

REEMPLAZAR X POR 10 SI  $0.9 \leq X \leq 1.0$ .

LUEGO LOS NUMEROS ENTEROS SERAN ALEATORIOS ENTRE 1 Y 10. EL TEST DEL POKER SEGUN KENDALL Y BABINGTON SMITH PROPONE: TOMAR SUCESIVOS CONJUNTOS DE 5 ENTEROS, Y PARA CADA UNO, DETERMINAR SI CONTIENEN EL MISMO NUMERO ("AAAAA" DONDE A ES CUALQUER ENTERO ENTRE 1 Y 10), O BIEN UN ENTERO REPETIDO 4 VECES Y UNO DISTINTO ("AAAAB"), 3 IGUALES Y DOS DE OTRO ("AAABB"), TRES IGUALES Y DOS DISTINTOS ("AAABC"), 2 DE CADA UNO Y EL TERCERO DIFERENTE ("AABBC"), DE LA FORMA ("AABCD") O ("ABCDE").

UNA DE ESTAS POSIBILIDADES OCURRIRA Y CUANDO HAY ALEATORIEDAD, LA PROPORCION DE OCURRENCIA DE CADA UNA SERA IGUAL A LAS PROBABILIDADES:

P(AAAAA) = .0001                      P(AAAAB) = .0045

P(AAABB) = .0090                      P(AAABC) = .0720

P(AABBC) = .1080                      P(AABCD) = .5040

P(ABCDE) = .3024.

LUEGO, SE REALIZA UN TEST CHI-CUADRADO PARA VER SI LOS NUMEROS OBSERVADOS EN CADA CATEGORIA CONCUERDAN CON LAS PROBABILIDADES TEORICAS MENCIONADAS MIENTRAS QUE EL TEST UTILIZA 7 "REPARTICIONES" DEL CONJUNTO DE 5 VALORES ENTEROS, PUEDEN TOMARSE OTRAS. EL PROGRAMA REALIZA TAMBIEN EL TEST DEL POKER PARA LAS REPARTICIONES BASADAS EN LOS ENTEROS DIFERENTES QUE HAY EN EL 5, PARA LAS CUALES, LAS PROBABILIDADES BAJO ALEATORIEDAD SON:

P(1 DIFERENTE) = .0001                    P(2 DIFERENTE) = .0135  
P(3 DIFERENTE) = .1800                    P(4 DIFERENTE) = .5040  
P(5 DIFERENTE) = .3024.

NO ES NECESARIO RESTRINGIRSE A MANOS DE 5 CARTAS. TAMBIEN SE REALIZA EL TEST DEL POKER PARA MANOS DE 4 CARTAS CON 5 REPARTICIONES, CON PROBABILIDADES:

P(AAAA) = .001                    P(AAAB) = .036  
P(AABB) = .027                    P(AABC) = .432  
P(ABCD) = .504

USO:

LOS PARAMETROS DE LA RUTINA TST08 SON:

VARIABLE	LONG.	NUM./CAR.
TNAME	8	CAR
IPARM(1)	11	NUM
IPARM(2)	11	NUM

DESCRIPCION DE PARAMETROS

IPARM(1)...INDICE PARA COMENZAR EN LA LISTA DE NUMEROS ALEATORIOS  
IPARM(2)...INDICE PARA FINALIZAR EN LA LISTA DE NUMEROS ALEATORIOS

SUBROUTINAS Y FUNCIONES REQUERIDAS:

PNTCHI (NO IMPLEMENTADA EN ESTA VERSION)

REFERENCIAS:

1. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981
2. KENDALL M.G. Y BABINGTON SMITH B., "RANDOMNESS AND RANDOM SAMPLING NUMBERS", JOURNAL OF THE ROYAL STATISTICAL SOCIETY, VOL.101(1938)

SUBROUTINE TST09

\*\*\*\*\*

TST09 APLICA EL TEST DE LAS CORRIDAS (RUNS UP) PARA PROBAR ALEATORIEDAD.

PROPOSITO

DADA UNA LISTA DE N NUMEROS REALES EN PUNTO FLOTANTE ENTRE 0 Y 1 QUE SE SUPONEN ALEATORIOS, SE ESPERA QUE EL TAMAÑO DE "RUNS UP" OBSERVADOS SIGA UNA CIERTA DISTRIBUCION DE PROBABILIDAD. SI DETERMINAMOS EL NUMERO COUNT(R) DE RUNS DE TAMAÑO R (1<=R<=5) Y EL NUMERO COUNT(6) DE RUNS DE TAMAÑO >=6, PUEDE USARSE UN TEST CHI-CUADRADO PARA VER SI

LOS NUMEROS SON ALEATORIOS. (NOTAR QUE SE TIENE UN TEST CHI-CUADRADO COMPLEJO YA QUE LOS TAMAÑOS DE LOS RUNS ESTAN CORRELACIONADOS. ES DECIR, SE ESPERA QUE UN RUN LARGO SEA SEGUIDO POR UNO CORTO). EL TEST SE APLICA A 3 CONJUNTOS DE NUMEROS ALEATORIOS:

LISTA BASICA DE LOS X - LISTA DE LOS FR(10X) - LISTA DE FR(100X)  
DONDE FR (Z) DENOTA LA PARTE FRACCIONARIA DE Z.

USO:

TST09 TIENE LOS SIGUIENTES PARAMETROS:

VARIABLE	LONG.	NUM./CAR.
TNAME	8	CAR
IPARM(1)	11	NUM
IPARM(2)	11	NUM

DESCRIPCION DE PARAMETROS

IPARM(1)...INDICE DEL ARREGLO DONDE SE LOCALIZA EL PRIMERO A USAR  
 IPARM(2)...INDICE DEL ARREGLO DONDE SE LOCALIZA EL ULTIMO A USAR

OBSERVACIONES:

ESTA RUTINA ASUME QUE LA PROBABILIDAD DE TENER X IGUALES EN LA LISTA DE N NUMEROS ES CERO. CUANDO ESTO NO ES VERDADERO, LOS RESULTADOS SON SOLO APROXIMADOS.

SUBROUTINAS Y FUNCIONES REQUERIDAS:

PNTCHI (NO IMPLEMENTADA EN ESTA VERSION)

METODO

REFERENCIAS:

1. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981

SUBROUTINE TST10

\*\*\*\*\*

PROPOSITO:

DADA UNA LISTA DE N NUMEROS REALES EN PUNTO FLOTANTE ENTRE 0 Y 1, QUE SE SUPONENEN ALEATORIOS, SEA UN ENTERO  $D \geq 2$ , SE CONSIDERA EL REEMPLAZO DE DICHS NUMEROS POR ENTEROS 1, 2, ..., D MEDIANTE  $1 + \text{INT}(D * X)$  DONDE  $\text{INT}(D * X)$  ES LA PARTE ENTERA DE  $D * X$  (PERO SI  $D * X = D$ , TOMAMOS  $\text{INT}(D * X) = D - 1$ ).

LOS NUEVOS NUMEROS SERAN ALEATORIOS ENTRE 1 Y D. PARA ESTOS ENTEROS KENDALL Y BABINGTON SMITH PROPONEN EL TEST "SERIAL PAIRS": TOMAR SUCESIVOS PARES DE ENTEROS Y DETERMINAR EN CUAL DE LAS  $D^2$  CATEGORIAS POSIBLES

(1,1), (1,2), (1,3), ..., (1,D-1), (1,D),  
 (2,1), (2,2), (2,3), ..., (2,D-1), (2,D),  
 .  
 .  
 (D,1), (D,2), (D,3), ..., (D,D-1), (D,D)

CAE CADA UNO. CUANDO HAY ALEATORIEDAD LAS CATEGORIAS SON EQUIPROBABLES CON  $\text{PROB.} 1/(D^2)$ . LUEGO DE OBSERVAR LAS OCURRENCIAS EN CADA CATEGORIA SE REALIZA UN TEST CHI-CUAD. PARA VER SI LOS NUMEROS OBSERVADOS EN CADA CATEGORIA COINCIDEN CON LAS PROBAB. TEORICAS.

AQUI SE REALIZA EL TEST PARA  $D=3$ ,  $D=10$  Y  $D=20$

PARAMETROS:

VARIABLE	LONG.	NUM./CAR.
TNAME	8	CAR
IPARM(1)	11	NUM
IPARM(2)	11	NUM



TNAME	8	CAR
IPARM(1)	11	NUM
IPARM(2)	11	NUM
IPARM(3)	11	NUM
IPARM(4)	11	NUM

DESCRIPCION DE PARAMETROS

IPARM(1)...VALOR DE T A SER USADO (SE PERMITEN 2,3,4,...).  
 IPARM(2)...CANTIDAD DE MAXIMOS (DE T NUMEROS ALEAT. CADA UNO) A SER USADOS EN CADA TEST DE K-S.  
 IPARM(3)...CANTIDAD DE CONJUNTOS DE IPARM(2) MAXIMOS CADA UNO QUE SERAN VERIFICADOS. DEBE COMPROBARSE:  
 $T * IPARM(2) * IPARM(3) \leq NBATCH$ .  
 IPARM(4)...INDICE PARA COMENZAR EL ARREGLO DE NUMEROS ALEATORIOS DEBE SER  $IPARM(4) - 1 + T * IPARM(2) * IPARM(3) \leq NBATCH$ .

SUBROUTINAS Y FUNCIONES REQUERIDAS:

KSTPL, CDFKS, XTRMS, SORT

REFERENCIAS:

1. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.

SUBROUTINE TSTM1

\*\*\*\*\*

TSTM1 CALCULA LA CORRELACION SERIAL DE UN GENERADOR DE NUMEROS ALEATORIOS CONGRUENCIAL (MIXTO) ESPECIFICADO POR EL USUARIO.

PROPOSITO:

DADO UN METODO PARA GENERAR NUMEROS REALES EN PUNTO FLOTANTE ENTRE 0 Y 1 (INCLUSIVE), QUE SE SUPONEN ALEATORIOS, ES DESEABLE QUE SU CORRELACION SERIAL LAG1 ESTE CERCA DE CERO. TST03 VERIFICA ESTO EMPIRICAMENTE. SIN EMBARGO PARA CIERTOS TIPOS DE GENERADORES, LA CORRELACION SERIAL LAG1 PUEDE SER CALCULADA A TRAVES DE FORMULAS TEORICAS. ESTA SUBROUTINA REALIZA DICHO CALCULO PARA GENERADORES DE NUMEROS ALEATORIOS DE LA FORMA:

$$X(I+1) = L * X(I) + C \text{ MOD } ZZ, I=0,1,2,\dots$$

USO:

CUANDO SE EJECUTA EL COMANDO "CALL TESTER" EN EL PROG.PRINCIPAL, UN ARCHIVO DE ENTRADA CON LOS DATOS NECESARIOS PRODUCE UNA LLAMADA DE LA RUTINA TSTM1. LOS PARAMETROS DE ESTA SUBROUTINA SON:

VARIABLE	LONG.	NUM./CAR.
TNAME	8	CAR
IPARM(1)	11	NUM
IPARM(2)	11	NUM
IPARM(3)	11	NUM

DESCRIPCION DE PARAMETROS:

IPARM(1)...ES LA CONSTANTE MULTIPLICATIVA L=A DEL GENERADOR.  
 IPARM(2)...CONSTANTE ADITIVA DEL GENERADOR.(EL GENERADOR ES LLAMADO "CONGRUENCIAL MIXTO" SI ESTA CONSTANTE C ES DISTINTA DE 0, Y "CONGRUENCIAL MULTIPLICATIVO" SI C=0)  
 IPARM(3)...MODULO ZZ DEL GENERADOR.

SUBROUTINAS Y FUNCIONES REQUERIDAS:  
 GCD

METODO

VER REFERENCIA 3.PARA EL ALGORITMO EXACTO USADO CUANDO L Y ZZ SON RELATIVAMENTE PRIMOS. REFERENCIA 1 PARA LA APROXIMACION USADA DE OTRA FORMA Y REFERENCIA 2 PARA DISCUSION Y EJEMPLOS

REFERENCIAS:

1. COVEYOU R.R., "SERIAL CORRELATION IN THE GENERATION OF PSEUDO-RANDOM NUMBERS", JOURNAL OF THE ACM, VOL.7(1960), PAG.72-74.
2. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.
3. KNUTH D.E., "THE ART OF COMPUTER PROGRAMMING, VOLUME 2/ SEMINUMERICAL ALGORITHMS", ADDISON-WESLEY PUBLISHING COMPANY, INC., READING, MASSACHUSETTS, 1969, PAG.72,75-76.

SUBROUTINE TSTM2

\*\*\*\*\*

TSTM2 CALCULA EL PERIODO/POTENCIA  $P(X(I+1)<X(I))$  DE UN GENERADOR DE NUMEROS ALEATORIOS CONGRUENCIAL (MIXTO) ESPECIFICADO POR EL USUARIO

PROPOSITO

DADO UN METODO DE GENERACION DE NUMEROS REALES EN PTO. FLOTANTE ENTRE 0 Y 1 (INCLUSIVE) EL CUAL SE SUPONE ALEATORIO, EL PROPOSITO ES QUE NO "CICLE" PRONTO, Y QUE  $P(X(I+1)<X(I))$  ESTE "CERCA" A 0.5.

ESTA SUBROUTINA REALIZA CALCULOS RELACIONADOS CON ESTE OBJETIVO PARA GENERADORES DE NUMEROS ALEATORIOS QUE SON DE ESTA FORMA

$$X(I+1) = A * X(I) + \text{MOD } ZZ, I=0,1,2,\dots$$

USO

LOS PARAMETROS DE TSTM2 SON:

VARIABLE	LONG.	NUM./CAR.
TNAME	8	CAR
IPARM(1)	11	NUM
IPARM(2)	11	NUM
IPARM(3)	11	NUM

DESCRIPCION DE PARAMETROS

IPARM(1)...ES EL MULTIPLICADOR ENTERO A DEL GENERADOR.

IPARM(2)...ES LA CONSTANTE ADITIVA DEL GENERADOR. (EL GENERADOR

ES DENOMINADO "CONGRUENCIAL MIXTO" SI ESTA CONSTANTE  
NO ES IGUAL A CERO, Y "CONGRUENCIAL MULTIPLICATIVO" SI  
C=0)  
IPARM(3)...ES EL MODULO ZZ POR EL CUAL LOS NUMEROS SON REDUCIDOS.

OBSERVACIONES:

SI UNO O MAS DE LOS A,C,ZZ CONSIDERADOS EXCEDEN EL VALOR ENTERO  
MAXIMO DISPONIBLE DE UN MICROCOMPUTADOR ( $2^{31}-1$ ) DE  
32 BITS, ENTONCES ESTA SUBROUTINA NO PUEDE USARSE.

SUBROUTINAS Y FUNCIONES REQUERIDAS

GCD, TSTM3, ORDER

METODO

VER REFERENCIA 1. PARA DISCUSION Y EJEMPLOS, Y REFERENCIA 2. PARA  
UN RESUMEN DE LOS RESULTADOS BASICOS CODIFICADOS EN ESTE PROGRAMA:

REFERENCIAS:

1. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER  
GENERATION AND TESTING", AMERICAN SCIENCES PRESS,  
COLUMBUS, OHIO, 1981.
2. KNUTH D.E., "THE ART OF COMPUTER PROGRAMMING, VOLUME 2/  
SEMINUMERICAL ALGORITHMS", ADDISON-WESLEY PUBLISHING COMPANY,  
INC., READING, MASSACHUSETTS, 1969, PAG.9-10,15-19,21-23,69,  
294-313

SUBROUTINE TSTM3 (N,INTDIV,INDX,ISW)

\*\*\*\*\*

TSTM3 CALCULA LOS FACTORES PRIMOS DE UN ENTERO ESPECIFICADO.  
EL ENTERO MENCIONADO ES FRECUENTEMENTE USADO COMO LA CONSTANTE  
MULTIPLICATIVA DE UN GENERADOR CONGRUENCIAL DE NUMEROS ALEATORIOS  
(MULTIPLICATIVO O MIXTO)

PROPOSITO

CALCULAR E IMPRIMIR LA FACTORIZACION PRIMA DE UN ENTERO DADO

USO:

LA SUBROUTINA TSTM3 UTILIZA LOS SIGUIENTES PARAMETROS:

VARIABLE	LONG.	NUM./CAR.
TNAME	8	CAR
N	11	NUM

DESCRIPCION DE PARAMETROS:

N.....ES EL ENTERO A SER FACTORIZADO EN FACTORES PRIMOS.  
INTDIV...ARREGLO DE 31X2 DONDE SE GUARDA LA FACTORIZACION DE N.  
INTDIV(K,1)ES EL K-ESIMO PRIMO DISTINTO QUE OCURRE EN LA  
FACTORIZACION DE N.  
INTDIV(K,2)ES LA POTENCIA CON LA CUAL APARECE ESTE  
PRIMO EN N.  
INDX.....CANTIDAD DE FACTORES PRIMOS DISTINTOS DE N.

ISW.....SE USA PARA IMPRIMIR LOS RESULTADOS CUANDO ISW=1, O NO IMPRIMIRLOS CUANDO ISW=0.

NOTA:

EL ENTERO A SER FACTORIZADO PUEDE NO EXCEDER EL VALOR ENTERO MAXIMO DISPONIBLE DE UN MICROMPUTADOR (2147483647=2\*\*31-1) DE 32 BITS.

SUBROUTINAS Y FUNCIONES REQUERIDAS:

NINGUNA

REFERENCIAS:

1. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981.
2. KNUTH D.E., "THE ART OF COMPUTER PROGRAMMING, VOLUME 2/ SEMINUMERICAL ALGORITHMS", ADDISON-WESLEY PUBLISHING COMPANY, INC., READING, MASSACHUSETTS, 1969, PAG.338-340.

SUBROUTINE TSTM4 (A,M)

\*\*\*\*\*

TSTM4 CALCULA LAS CANTIDADES D Y R PARA ESTABLECER UNA MEDIDA DE LA UNIFORMIDAD EN LAS DIMENSIONES T=2,3,4,5,6 PARA LOS NUMEROS ALEATORIOS GENERADOS POR UN ALGORITMO CONGRUENCIAL MULTIPLICATIVO EN UN PERIODO.

LA CANTIDAD D, EN LA DIMENSION  $n=T$ , SE DEFINE COMO LA MINIMA DISTANCIA ENTRE LOS HIPERPLANOS PARALELOS GENERADOS A PARTIR DE  $\{U_i\}$  POR LAS N-UPLAS  $(U_i, \dots, U_{n+i-1})$

LA CANTIDAD R SE DEFINE COMO EL COCIENTE ENTRE D Y B, DONDE B ES UNA COTA SUPERIOR DE D EN LA DIMENSION T PROPUESTA POR MARSAGLIA, QUE ALCANZA UN VALOR 1 COMO MINIMO.

PROPOSITO:

CALCULAR E IMPRIMIR LAS MEDIDAS D Y R

USO:

LA SUBROUTINA TSTM4 UTILIZA LOS SIGUIENTES PARAMETROS:

A ... FACTOR MUTIPLICATIVO DEL GENERADOR CONGRUENCIAL

M ... MODULO DEL GENERADOR

SUBROUTINAS Y FUNCIONES REQUERIDAS:

NINGUNA

REFERENCIAS:

1. COVEYOU R.R. Y MACPHERSON R.D., "FOURIER ANALYSIS OF UNIFORM RANDOM NUMBER GENERATORS", JOURNAL OF THE ACM, VOL.14 1967, PAG.100-119
2. DUDEWICZ E.J. Y RALLEY T., "THE HANDBOOK OF RANDOM NUMBER GENERATION AND TESTING", AMERICAN SCIENCES PRESS,



COLUMBUS, OHIO, 1981.

3. HOAGLIN D.C., "THEORETICAL PROPERTIES OF CONGRUENTIAL RANDOM NUMBER GENERATORS:UN EMPIRICAL VIEW", MEMORANDUM NS-340, DEPARTMENT OF STATISTICS, HARVARD UNIVERSITY, CAMBRIDGE, MASS.
4. KNUTH D.E., "THE ART OF COMPUTER PROGRAMMING, VOLUME 2/ SEMINUMERICAL ALGORITHMS", ADDISON-WESLEY PUBLISHING COMPANY, MASSACHUSETTS, 1969 Y 2DA.EDICION,1981.

SUBROUTINE TSTENTR

\*\*\*\*\*

PRUEBAS DE ENTROPIA CONTINUA EMPLEANDO  
LOS SIGUIENTES TESTS: VASICEK - Hmn  
Y DUDEWICZ-VAN DER MEULEN - H\*ALFA(m,n)

\*\*\*\*\*C

X: ARREGLO UNIDIMENSIONAL DE 'CANTELEM' NUMEROS ALEATORIOS  
EN DOBLE PRECISION, PARA CADA SUCESION GENERADA A TRAVES  
DE 'NLOTES' LLAMADOS AL GENERADOR URNWN

M: ARREGLO DE VALORES ENTEROS POSITIVOS  
(M=1,2,...,10,15,20,30,40) PARA UTILIZAR EN EL CALCULO DE  
Hmn Y H\*ALFA(m,n)EN CADA SUCESIÓN.

DESCRIPCION DE ARCHIVOS

ARCHDAT: ARCHIVO DIRECTO CON FORMATO F11.10 CONTENIENDO  
NUMLOTES\*CANTELEM NUMEROS ALEATORIOS GENERADOS POR URNWN

\*\*\*\*\*

SUBROUTINE SORTENTROPIA(R,NN)  
DOUBLE PRECISION R(10000)

\*\*\*\*\*

SUBROUTINE LOSEMES(LOSM,IM)

\*\*\*\*\*

SUBROUTINE EUXXFD10(NUMLOTES,CANTELEM)  
PARA CASOS DONDE SE OBTIENEN VALORES INFINITOS

PROPOSITO:

ANALOGO AL PROGRAMA EUXXFIND, PARA SER UTILIZADO CUANDO EXISTAN  
ALGUNOS M PARA LOS CUALES SE OBTIENEN VALORES INFINITOS  
(V.GR. URN10, URN13, URN39)

- CALCULAR EL ESTIMADOR DE ENTROPIA DE VASICEK: Hmn
- DETERMINAR EL ESTIMADOR ASINTOTICO AL PERCENTIL 100xALFA
- PROBAR EL TEST DE DUDEWICZ-VAN DER MEULEN:  
Hmn <= H\*ALFA(m,n)  
CALCULANDO PORCENTAJES DE VERIFICACIONES

DESCRIPCION DE PARAMETROS

X: ARREGLO UNIDIMENSIONAL PARA CADA SUCESION DE 'CANTELEM'  
NUMEROS ALEATORIOS GENERADOS A TRAVES DE 'NUMLOTES'  
LLAMADOS AL GENERADOR URNWN

M: ARREGLO DE VALORES ENTEROS POSITIVOS (M=1,2,...,10,15,20,  
30,40) PARA UTILIZAR EN EL CÁLCULO DE Hmn Y H\*ALFA(m,n)

EN CADA SUCESION.  
RESU,H14,PROMH: ARREGLOS PARA ALMACENAR VALORES DE LOS  
ESTADISTICOS.

DESCRIPCION DE ARCHIVOS

ARCHDAT: ARCHIVO DE ACCESO DIRECTO CON FORMATO F11.10 CONTENIENDO  
NUMLOTES\*CANTELEM NÚMEROS ALEATORIOS GENERADOS POR URNWN  
HMNUXX.DAT:ARCHIVO DE RESULTADOS DE VERIFICACIONES DE UN TEST.  
PRHUXX.DAT:ARCHIVO DE PROMEDIOS, DE LONGITUD QUINCE.  
HF14UXX.DAT:ARCHIVO DE REGISTROS DE LONGITUD QUINCE, PARA LOS  
VALORES DE H\*ALFA(m,n)

\*\*\*\*\*

SUBROUTINE EUXXFIND (NUMLOTES,CANTELEM)  
PARA CASOS DONDE SE OBTIENEN VALORES FINITOS

PROPOSITO:

- CALCULAR EL ESTIMADOR DE ENTROPÍA DE VASICEK :Hmn
- DETERMINAR EL ESTIMADOR ASINTOTICO AL PERCENTIL 100\*ALFA MEDIANTE
- APROXIMACIÓN  $H^*ALFA$  (FORM. 11, cfr. 2.4.3)
- PROBAR EL TEST DE DUDEWICZ-VAN DER MEULEN:  $Hmn \leq H^*alfa(m,n)$   
CALCULANDO PORCENTAJES DE VERIFICACIONES

DESCRIPCION DE PARAMETROS

X: ARREGLO UNIDIMENSIONAL PARA CADA SUCESION DE 'NUMLOTES' NUMEROS  
ALEATORIOS GENERADOS A TRAVÉS DE 1000 LLAMADOS AL GENERADOR  
URNWN

M: ARREGLO DE VALORES ENTEROS POSITIVOS  
(M=1,2,...,10,15,20,30,40)

PARA UTILIZAR EN EL CALCULO DE Hmn Y  $H^*ALFA(m,n)$  EN CADA  
SUCESIÓN

RESU,H14,PROMH: ARREGLOS PARA ALMACENAR VALORES DE LOS  
ESTADISTICOS

DESCRIPCION DE ARCHIVOS

ARCHDAT:ARCHIVO DE ACCESO DIRECTO CON FORMATO F11.10 CONTENIENDO  
NUMLOTES\*CANTELEM NUMEROS ALEATORIOS GENERADOS POR URNWN  
HMNUXX.DAT: ARCHIVO DE RESULTADOS DE VERIFICACIONES DE UN TEST.  
PRHUXX.DAT: ARCHIVO DE PROMEDIOS, DE LONGITUD QUINCE.  
HF14UXX.DAT: ARCHIVO DE REGISTROS DE LONGITUD QUINCE, PARA LOS  
VALORES DEL ESTIMADOR  $H^*ALFA(m,n)$

\*\*\*\*\*

SUBROUTINE EUXXFINE(NUMLOTES)  
RESULTADOS DE LOS TESTS DE ENTROPIA

PROPOSITO:

IMPRIME LOS RESULTADOS DE CALCULAR EL ESTIMADOR DE ENTROPIA DE  
VASICEK; DEL ESTIMADOR ASINTOTICO AL PERCENTIL 100\*ALFA ; Y DE LAS  
PRUEBAS DEL TEST DE DUDEWICZ-VAN DER MEULEN.

DESCRIPCION DE PARAMETROS

M: ARREGLO DE VALORES ENTEROS POSITIVOS  
(M=1,2,...,10,15,20,30,40)

SM, PROMSM, PM, PMH, F14 Y PF14: ARREGLOS DE LONGITUD 14 PARA  
ALMACENAR VALORES PROMEDIOS.

DESCRIPCION DE ARCHIVOS

HMNUXX.DAT: ARCHIVO DE RESULTADOS DE VERIFICACIONES DE UN TEST.  
PRHUXX.DAT : ARCHIVO DE PROMEDIOS DE LONGITUD QUINCE.

HF14UXX.DAT: ARCHIVO DE REGISTROS DE LONGITUD QUINCE PARA LOS  
VALORES DEL ESTIMADOR  $H^*ALFA(m,n)$   
OBSERVACION: SU USO ES COMUN PARA PROCESAR EN SU ULTIMA ETAPA TODOS  
LOS C RESULTADOS OBTENIDOS EN LOS PROGRAMAS EJECUTADOS  
PRECEDENTEMENTE:  
EUXXMM10, EUXMDP10, EUXXFIND, EUXXFD10 Y EUXFINDP.

REFERENCIAS:

1. DUDEWICZ E.J. y VAN DER MEULEN E.C., "ENTROPY-BASED TESTS OF UNIFORMITY", JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION, 1981.
2. VASICEK O., "A TEST OF NORMALITY BASED ON SQMPLE ENTROPY", JOURNAL OF DE ROYAL STATISTICAL SOCIETY, 1976.
3. DUDEWICZ E.J., VAN DER MEULEN E.C, SRIRAM M.G. Y TEOH NICK K.W., "ENTROPY-BASED RANDOM NUMBER EVALUATION", AMERICAN JOURNAL OF MATHEMATICAL AND MANAGEMENT SCIENCE, AMERICAN SCIENCES PRESS, 1995.

\*\*\*\*\*

## Capítulo 4

### UTILIZACION DEL PAQUETE GENTSTPC

Los procedimientos que conforman el paquete GENTSTPC son invocados a través de un programa de control, TESTER, que lee los archivos de datos generados por el programa ENTRADA, y luego transfiere el control a las distintas subrutinas de GENTSTPC.

#### GENERACION DE NUMEROS PSEUDOALEATORIOS USANDO GENTSTPC

Para crear un arreglo de números pseudoaleatorios, el usuario provee a través del programa ENTRADA los siguientes datos:

- El nombre de uno de los generadores : URN01, URN02, URN03, ... , URN40.

En el caso que se trate de un generador provisto por el propio usuario deberá ingresar URNWN. Aquí se distinguen dos subcasos: generador incluido en el paquete GENTSTPC o un archivo de números aleatorios generado externamente.

- A continuación seleccionará la opción "N" correspondiente a "no aplicar ningún test" a los números generados.

- El dato siguiente que requiere ENTRADA es el ingreso de la cantidad NUMALE de números aleatorios, entre 0 y 10000 , que serán generados en cada llamada del generador, o a ser usados desde el archivo ARCHDAT, en el caso de utilizar la opción URNWN con un archivo de números aleatorios generados externamente , con otros métodos y/o paquetes.

- El ingreso de NUMPR (no superior a NUMALE) es requerido con el fin de especificar la cantidad de números aleatorios que podrán ser impresos en ocasión del primer llamado al generador.

- Las opciones disponibles con respecto a la inicialización de las variables del generador son:

- 0: Los valores iniciales serán dados por defecto.
- 1: Los valores iniciales serán provistos por el usuario.
- 2: Este valor sólo se usa cuando los números pseudoaleatorios provienen del archivo ARCHDAT.
- 3: Opción no habilitada.
- 4: Sólo utilizable con los generadores URN02, URN03, URN04 donde se requieren largas tablas de inicialización. En esta versión de GENTSTPC la opción 4 no está habilitada porque no emplea compiladores ASSEMBLER en estas subrutinas.

## PRUEBAS DE NUMEROS PSEUDOALEATORIOS CON GENTSTPC

Para probar la aleatoriedad de arreglos de números aleatorios el usuario provee una serie de parámetros a través del programa ENTRADA, que se ocupa de crear un conjunto de archivos de datos vinculados con las pruebas: TST01,TST02,...,TST11,TSTM1, TSTM2,...,TSTM4.

Recientemente se agregaron a los cuatro últimos tests teóricos, varias pruebas basadas en Entropía de funciones de densidad continua, TSTENTR, que incluyen los estimadores de Vasicek y Dudewicz-Vander Meulen.

En una primera ventana del programa ENTRADA se pide seleccionar una de las siguientes opciones a ingresar:

E : Si aplicará Tests Empíricos (y teóricos, eventualmente)

T : Si aplicará Tests Teóricos

En la pantalla siguiente, "TESTS", se lee una "información previa a la elección de los tests": los tests "teóricos" TSTM2 y TSTM3 no pueden ser usados conjuntamente, ya que el primero usa al segundo para su programación.

Considerados independientemente los parámetros requeridos por cada test son:

- TST02, no requiere parámetros.
- TST03, pide ingresar uno de los coeficientes del generador congruencial, u otro número que se desee descomponer en sus factores primos.
- TST04, los índices del inicio y del fin de la lista de NUMALE números.
- TST05, los valores enteros con 3 dígitos de **a,b** extremos del intervalo (ALFA,BETA) de modo que se verifique  $a + b = 1$ ; y los índices inicio y fin de la lista NUMALE.
- TST06, tres parámetros: NLOT, cantidad de números en cada muestra, entre 1 y NUMALE; cantidad de conjuntos (con NLOT elementos) a ser probados; y el índice de iniciación de la lista de NUMALE números.
- TST07, dos parámetros: los índices del inicio y del fin de la lista de NUMALE números.
- TST08, dos parámetros: los índices del inicio y del fin de la lista de NUMALE números.
- TST09, dos parámetros: los índices del inicio y del fin de la lista de NUMALE números.
- TST10, dos parámetros: los índices del inicio y del fin de la lista de NUMALE números.

- TST11, cuatro parámetros: valor de T ( T-uplas; T=2,3,4,...) ; cantidad de máximos en cada T-upla a usarse en cada test K-S ; cantidad de conjuntos a ser verificados; el índice inicial de la lista de NUMALE números pseudoaleatorios.

El test TST01 requiere un tratamiento particular. Una vez seleccionada esta opción en la pantalla "TESTS", se ingresan los siguientes parámetros:

- NLOT , cantidad de lotes, menor o igual que 1000, cada uno formado de NUMALE números pseudoaleatorios, a los cuales se les aplicarán los tests especificados en las opciones CSCODE y/o KSCODE.
- NIMPR , la cantidad de lotes cuyos resultados serán impresos por pantalla, con la restricción que sea menor o igual que NLOT.
- TCODE : tiene tres valores posibles CSCS, KSKS y CS2KS2
  - . CSCS : se usa si se aplica Chi-cuadrado a variables que se suponen Chi-cuadrado.
  - . KSKS : si se aplica K-S a variables con distribución supuesta K-S.
  - . CS2KS2: cuando se aplica Chi-cuadrado a variables que se las supone Chi-cuadrado; y K-S sobre variables con distribución que se supone K-S.
- CSCODE : se debe utilizar con el parámetro TCODE = CSCS. Indica cuáles de los siguientes siete tests serán empleados, previo a la aplicación del test Chi-cuadrado a los resultados. Se requiere ingresar "1" en caso afirmativo o "0" para el caso contrario.
  - TST02 : test de distribución uniforme
  - TST04 : test del recolector de cupones
  - TST05 : test gap
  - TST07 : test de permutación
  - TST08 : test del póker
  - TST09 : test de las corridas
  - TST10: test de pares seriales.

Casos de tests "teóricos" (opción "T" )

- TSTM1(correlación serial): Se ingresan los tres parámetros de un generador congruencial lineal bajo dos formas, según asuman valores enteros menores que  $2^{*}31$  o iguales o mayores que  $2^{*}31$ .
- TSTM2 (potencia-período del generador): Se ingresan los parámetros A, C y M con la restricción de no superar once dígitos como máximo en cada caso.
- TSTM3 (factorización): Se requiere el ingreso de uno de los parámetros del generador congruencial lineal, o bien cualquier otro número para ser descompuesto en sus factores primos.
- TSTM4 (espectral): Los parámetros A (constante multiplicativa) y M (módulo) no deben exceder  $2^{*}31 - 1$  y se ingresan bajo la forma de notación científica.

### Caso de tests de "entropía"

Se ingresa a esta batería de pruebas mediante la opción 'URNWN' que es habilitada por un ventana del programa ENTRADA.EXE.

Eligiendo la opción '2' (que permite el empleo de un archivo de datos formado por las sucesiones de números a verificar, y el cual es designado ARCHDAT), se obtiene el mensaje 'Seleccionar una de las siguientes opciones e ingresar':

E: Si aplicará Tests Empíricos (y Teóricos eventualmente)

T: Si aplicará Tests de Entropía

Luego, deben introducirse los datos NUMLOTES (igual a la cantidad de lotes o sucesiones de números pseudoaleatorios) y CANTELEM (cantidad de números que contiene cada lote).

#### NOTAS:

1. El programa principal se denomina TSTENTR (entropía continua) requiere de un archivo de datos numéricos, ARCHDAT, (formato F11.10) que posea el total de números pseudoaleatorios producidos por el generador que se desea probar, inclusive los URN\*\*.
2. El archivo ARCHDAT debe poseer una cantidad de datos igual al producto de NUMLOTES\*CANTELEM.
3. Además del archivo de resultados, 'Salida.doc', los datos correspondientes a los percentiles  $P_m = 100 * \Phi(F)$ , se almacenan en el archivo localizado en: C:\Gentstpc\Val\_fi.txt

Empleando un soft estadístico se procesan estos últimos valores a fin de determinar  $P_m$  que de acuerdo al algoritmo propuesto (cfr. 3.5.3) debe estar entre las cotas:  $5 \leq P_m \leq 50$ , para  $m = 2, 3, 4, 5, 6, 7, 8, 9, 10, 15$  y 20.

Ejemplo(URN02, empleando MAPLE V):

```
> fopen(`val_fi.txt`, READ, TEXT):  
> with(stats):  
> valor:=readdata(`val_fi.txt`):  
> statevalf[cdf, normald[0,1]](valor);
```

En este caso se verifica desde el tercer valor de la sucesión de más abajo: 7.18944868 que corresponde a  $m = 20$ ; 16.80928001 para  $m = 15$ ; hasta el valor 47.32714548 correspondiente a  $m = 2$

```
[.3391540510e-4, .4369168722e-2, .7189444868e-1, .1680928001,  
.2966854693, .3231106919, .3485602184, .3725290821, .3965336331,  
.4183776068, .4360495357, .4520885717, .4632497026, .4732714548]
```

## Capítulo 5

### EJEMPLOS

Los ejemplos ilustran algunas formas de construir sucesiones de números pseudoaleatorios y su posterior verificación mediante las pruebas estadísticas del paquete GENTSTPC.

#### OBTENCION DE NUMEROS PSEUDOALEATORIOS CON GENERADORES DE GENTSTPC

En el Ej. N°1 se muestra la configuración de uno de los archivos creados para generar diez mil números pseudoaleatorios con el generador URN08 de los cuales se mostrarán por pantalla los cien primeros, con inicialización por defecto, sin cálculos de tiempos.

Ejemplo N°1

```
URN08      10000NO      100                0
```

En el Ej. N°2 el generador URN08 es inicializado desde el teclado con la opción "1".

En el Ej. N°3 el generador URNWN es proporcionado por el usuario, y el archivo creado es similar al ejemplo anterior.

Ejemplo N°2

```
URN08      10000NO      100                1
```

Ejemplo N°3

```
URNWN      10000NO      100                2
```

En este paquete está prevista la inicialización en tiempo de ejecución de los siguientes generadores (opción "1"):

- URN02, requiere el ingreso de una tabla de 128 números hexadecimales.
- URN07, URN08, URN09, URN10, URN15, URN16, URN17, URN18, URN19, URN20  
Se ingresará una variable.
- URN05, una tabla de 98 números enteros.
- URN11, 4 valores iniciales.
- URN12, 4 valores iniciales.
- URN14, 2 valores iniciales.

Observación:

Los tipos de datos y su significado se encuentran descriptos en Cap.2.



**PRUEBAS DE CALIDAD DE LOS GENERADORES DE NUMEROS PSEUDOALEATORIOS**

**LLAMADO DE PRUEBAS EMPIRICAS DESDE EL PROGRAMA PRINCIPAL FORTRAN**

Los Ejemplos N° 4 y 5 utilizan un generador, URN38, y para la verificación de las sucesiones de números se consideran, respectivamente, las siguientes opciones, que desde el programa de ingreso de datos ENTRADA, originan los respectivos archivos de datos:

E : Tests empíricos TST02,TST03,...,TST11, en forma independiente

E : Test empírico TST01, opción CS2KS2.

Ejemplo N°4

URN38	10000NO	100		0
TST02				
TST03	1000		20	
TST04		1	10000	
01T05		333	667	1 10000
01T06		1000	10	1
01T07		1	10000	
01T08		1	10000	
01T09		1	10000	
01T10		1	10000	
01T11		2	500	10 1

Ejemplo N°5

URN38	10000NO	100		0
TST03		10000	20	
TST01		1000	1	CS2KS211111110 11000000
01T02				
01T04		1	10000	
01T05		333	667	1 10000
01T06		1000	10	1
01T07		1	10000	
01T08		1	10000	
01T09		1	10000	
01T10		1	10000	
01T11		2	500	10 1

## LLAMADO DE PRUEBAS TEORICAS DESDE EL PROGRAMA PRINCIPAL FORTRAN

El Ejemplo N° 6 corresponde a uno de los archivos de datos que usan los cuatro tests teóricos aplicados a un generador congruencial lineal, URN15.

El programa ENTRAD permite elegir independientemente a TSTM1, TSTM2, TSTM3 o TSTM4, para los cuales pide la entrada de uno o más de los siguientes parámetros: multiplicador A, constante aditiva C y módulo M, a fin de organizar sendos archivos con los datos necesarios.

Otra forma, es ingresar TSTM1, TSTM2 y TSTM4 en una sola corrida, para obtener los resultados de los cuatro tests simultáneamente.

Ejemplo N°6

A = 764261123 ; C = 0 ; M = 2\*\*31 - 1

URN15	ONO	0	0
TSTM1	764261123	0	2147483647
TSTM2	764261123	0	2147483647
TSTM3	2147483647		
TSTM4	764261123		2147483647

## LLAMADO DE PRUEBAS DE ENTROPIA DESDE EL PROGRAMA PRINCIPAL FORTRAN

En la pantalla de entrada se pide el ingreso de los siguientes parámetros:

NUMLOTES = Cantidad de lotes

CANTELEM = Cantidad de números que contiene cada lote

Debe verificarse que el archivo ARCHDAT contenga una cantidad de números aleatorios igual al producto NUMLOTES \* CANTELEM.

El paquete de programas se incorpora a GENTSTPC mediante un conjunto de subrutinas encabezadas por la subrutina denominada SUBROUTINE TSTENTR. Todos los archivos que se construyen a partir de ARCHDAT y los algoritmos de cálculo se vinculan entre sí, sin intervención del usuario.

Las salidas, de modo análogo al paquete central GENTSTPC, pueden obtenerse como parte del archivo de resultados, como se explica en las primeras pantallas de ayuda de este soft.

A continuación, se ilustra como ejemplo, el caso de un archivo ARCHDAT formado por los números aleatorios generados con URN03.

Ejemplo N° 7

NUMLOTES = 1000

CANTELEM = 10000

```
*****
*LAS SIGUIENTES SALIDAS HAN SIDO PRODUCIDAS POR EL SISTEMA: "GENTSTPC"
*
*EL PAQUETE ESTA CONSTITUIDO POR RUTINAS PARA LA GENERACION DE NUMEROS
*
*PSEUDOALEATORIOS Y TESTS ESTADISTICOS PROGRAMADOS PARA
*
*MICROCOMPUTADORES.*
*****
ESTOS SON LOS      100 NUMEROS A SER IMPRESOS POR EL GENERADOR URNWN      :
```

```
0.00003052 0.00018311 0.00082399 0.00329594 0.01235973
0.04449497 0.15573221 0.53393859 0.80204165 0.00680240
0.82243967 0.87341642 0.83854151 0.17050117 0.47613364
0.32229128 0.64854503 0.99064845 0.10698555 0.72607750
0.39359507 0.82687283 0.41888145 0.07143317 0.65866596
0.30909729 0.92658991 0.77766407 0.32667512 0.96107394
0.82636762 0.30854017 0.41393226 0.70673209 0.51500219
0.72942436 0.74152619 0.88433808 0.63229275 0.83471394
0.31764895 0.39346805 0.50196785 0.47059470 0.30585757
0.59979320 0.84604096 0.67810702 0.45427337 0.62267733
0.64760345 0.28152490 0.86071849 0.63058692 0.03705499
0.54704756 0.94879031 0.76931411 0.07677165 0.53680295
0.52987301 0.34801140 0.31921121 0.78316462 0.82608688
0.90803969 0.01345624 0.90838021 0.32917500 0.79962826
0.83519453 0.81451303 0.37032747 0.89134735 0.01513707
0.06869613 0.27594313 0.03739363 0.74087352 0.10869840
0.98432887 0.92768747 0.70716512 0.89380366 0.99833602
```

```

0.94578326 0.68967527 0.62600207 0.54893512 0.65959227
0.01713738 0.16649392 0.84472710 0.56991738 0.81696045
0.77250618 0.28239280 0.74180120 0.90927202 0.77942133

```

\*\*\*\*\*

TESTS DE ENTROPIA CONTINUA

CALCULO DE H<sub>m,n</sub> PARA M=1,...,10,15,20,30,40 (ARCHIVO ARCHDAT)

USARA EUXFIND, PUES SE OBTIENEN VALORES FINITOS

RESULTADOS TEST ENTROPIA HMN - GENERADOR URNWN

M	H <sub>m,n</sub> (PROMEDIO) <= H*(m,n)	VALOR F EN 100*F <sub>i</sub> (F)
40	-0.07322548	-3.95728707
30	-1.08185768	-2.60315871
20	-4.07841444	-1.47488415
15	-6.76388788	-0.99661326
10	-11.03334618	-0.59688860
9	-12.45340347	-0.52591372
8	-14.18816853	-0.45646897
7	-16.42619896	-0.39183936
6	-19.37399483	-0.32570907
5	-23.45664787	-0.25925627
4	-29.38262367	-0.19665924
3	-39.72627258	-0.14363284
2	-60.29411697	-0.10342918
1	-124.21064758	-0.06012384

EL PTO. PERCENTIL 100\*F<sub>i</sub>(F), SE OBTIENE POR METODOS ESTADISTICOS  
CFR. MANUAL DEL USUARIO

SE HAN LEIDO Y PROCESADO TODOS LOS DATOS SUMINISTRADOS POR LOS  
ARCHIVOS DE ENTRADA Y, POR LO TANTO, LA EJECUCION HA FINALIZADO.

El método estadístico utilizado para procesar los valores finales de los tests de Entropía, columna 3, Valor F en 100\*F<sub>i</sub>(F), consiste en determinar el valor de la función de densidad acumulativa correspondiente a la distribución normal estándar N(0,1):

Empleando, por ejemplo, un archivo de 14 datos, en MAPLE V R4, se obtiene, luego de ingresar los respectivos comandos, los valores de P<sub>m</sub> que aparecen en la segunda columna del cuadro que sigue, donde P<sub>m</sub> = 100\*Φ(.)

```

> fopen(`valf.txt`,READ,TEXT):
> with(stats):
> valor:=readdata(`valf.txt`) :
> statevalf[cdf,normald[0,1]](valor);
> close(`valf.txt`);
>

```

M	$P_m = 100 * \Phi(.)$
40	.3790290455e-2
30	.4618459140
20	7.012184615
15	15.94761336
10	27.52908795
9	29.94740780
8	32.40263898
7	34.75884555
6	37.23222407
5	39.77187589
4	42.20471045
3	44.28952035
2	45.88111771
1	47.60285014

Teniendo en cuenta el algoritmo (cfr. 2.4.3), en este caso se verifica el criterio de entropía:  $5 \leq P_m \leq 50$ , para todo M entre 2 y 20.

## USO DE GENTSTPC PARA VERIFICAR UN GENERADOR DEL USUARIO

Los Ejemplos N° 8 y N° 9 corresponden al caso de verificación de la calidad de números pseudoaleatorios generados por el usuario bajo dos modos:

Ejemplo N° 8: Utilizando el código fuente de la subrutina URNWN incluida en GENTSPC

ESTA SUBROUTINA SE PROVEE COMO UN PROTOTIPO DE LA FORMA EN LA QUE DEBE PROCESARSE EN GENTSTPC UN GENERADOR PROVISTO POR EL USUARIO. SI EL USUARIO PROVEE OTRA SUBROUTINA, DEBE SOBREScribir ESTA. SI EL USUARIO INTENTA USAR URNWN SIN DAR SU PROPIO CODIGO, APARECERA UN MENSAJE DE ERROR Y TERMINARA LA EJECUCION.

UNA SECUENCIA TIPICA DE LLAMADO EN EL PROGRAMA PRINCIPAL DEL USUARIO PUEDE SER (CON INICIALIZACION PROVISTA POR EL USUARIO):

```
COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
IA=1
IR=0
INTIN(1)=524287
CALL TESTER
```

DE OTRO MODO, SI EN VEZ DE INICIALIZACION POR EL USUARIO FUERAN DATOS EN ARCHIVOS DE ENTRADA, SE TENDRIA:

```
COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
IA=1
IR=0
CALL TESTER
```

```

SUBROUTINE URNWN(X,NBATCH)
COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
DIMENSION X(NBATCH)
CHARACTER RESP,C1
OPEN(UNIT=60,ACCESS='DIRECT',RECL=1,FORM='FORMATTED',FILE=
* 'OPCION' )
12 READ(60,12)C1
FORMAT(1A)
IF(C1.EQ.'S')THEN
READ(60,12)RESP
GOTO 88
ENDIF
OPEN(UNIT=63,ACCESS='DIRECT',RECL=1,FORM='FORMATTED',FILE=
* 'GENOARCH' )
READ(63,12)RESP
CLOSE(63)
CLOSE(60)
OPEN(UNIT=60,ACCESS='DIRECT',RECL=1,FORM='FORMATTED',FILE=
* 'OPCION' )
WRITE(60,12)'S'

WRITE(60,12)RESP
88 IF(RESP.EQ.' 1' ) THEN
J=INTIN(1)
DO 10 I = 1,NBATCH
J=J*1220703125
IF(J.LT.0)J=J+2147483647+1
X(I)=FLOAT(J)*0.4656613E-9
10 CONTINUE
INTIN(1)=J
ELSE IF(RESP.EQ.' 2' ) THEN
DO 20 IK=1,NBATCH
READ(90,900)XXX
```

```

          X(IK)=XXX
900      FORMAT(F11.10)
        20  CONTINUE
          ELSE
          WRITE(*,*)'HA INGRESADO UN VALOR INCORRECTO'
          STOP
          ENDIF
          CLOSE(60)
          RETURN
        END

```

URNWN es invocada con dos parámetros explícitos: X, arreglo donde se almacenan los números aleatorios generados, y NBATCH (NUMALEA) que indica la cantidad de números a ser generados en cada llamada. Otra información adicional necesaria al generador son las semillas y sus respectivos valores que son pasadas en el bloque común OWNPRM. Además, la cantidad de valores enteros de inicialización IA y el total de valores en punto flotante, IR, que son dados en los arreglos INTIN y REALIN, respectivamente. Aquellos pueden ser leídos en tiempo de ejecución o bien se escriben formando parte del programa principal (IOPT=2).

#### Ejemplo N° 9

Proveyendo un archivo "ARCHDAT" de caracteres ASCII con registros de ancho igual a once, bajo el formato F11.10, el que fue producido por otro tipo de generador y procesado bajo un ambiente y lenguajes diferentes al de GENTSTPC. Este caso está contemplado dentro de la opción de generadores del programa ENTRADA y naturalmente el tamaño del archivo ARCHDAT debe ser tal que las cantidades NUMALEA (NBATCH) sean suficientes para el test que se usará; si se tratara de TST01 con la opción CS2KS2 sería necesario tener 1000 (mil) lotes de NUMALEA = 10000 (diez mil) números cada uno.

## Capítulo 6

### RECOMENDACIONES PARA EL USO DE LOS GENERADORES GENTSTPC

Los generadores de GENTSTPC han sido verificados con los tests estadísticos de este paquete.

En particular, a cada generador se le ha aplicado la prueba Chi-cuadrado sobre Chi-cuadrado (TST01) bajo el mismo formato con que reportaron los resultados de aplicar TESTRAND los profesores E.J. Dudewicz y T. Ralley en "The Handbook of Random Number Generation and Testing", American Sciences Press, Columbus, Ohio, 1981.

Los cálculos se efectuaron con la descripción detallada en el procedimiento que se encuentra en la SUBROUTINE GAMA de GENTSTPC, para una cantidad de lotes NLOT (NCHI) igual a 1000, con NUMALEA (NBATCH) igual a 10000 números aleatorios cada lote.

Los tests usados son en total 19:

1. DISTR.UNIF.(TST02)
2. CUPONES D=5(TST04)
3. CUPONES FR(100R)
4. CUPONES D=10
5. GAP ARRIBA MEDIA(TST05)
6. GAP BAJO MEDIA
7. GAP (0.33,0.667)
8. PERMUTACIONES 3S(TST07)
9. PERMUTACIONES 4S
10. PERMUTACIONES 5S
11. POKER (4,4) (TST08)
12. POKER (5,6)
13. POKER (5,4)
14. CORRIDAS R(TST09)
15. CORRIDAS FR(10R)
16. CORRIDAS FR(100R)
17. PARES SERIALES 3X3(TST10)
18. PARES SERIALES 10X10
19. PARES SERIALES 20X20

En la subrutina GAMA se calcula el valor aproximado  $G'$  de la probabilidad  $G$ , gama, a partir de la tabla  $(G,y)$ ,  $G'$  tal que  $y \leq T(i)$ , mediante interpolación.

Es decir :

$G' = P [CHI-CUAD \leq T(I)]$ ; donde el estadístico  $T(i) = \text{Chi-Cuadrado Total}$  es aplicado sobre 100 intervalos.

Los valores interpolados son los  $y = T(i) = \text{SUMCHI}$ , sumas de 100 contribuciones al test Chi-cuadrado en cada uno de los 19 tests Chi-cuadrado sobre Chi-cuadrado(CSCS).

Cada test recibe 1000 valores  $T(i)$ , cada uno proveniente de 10000 números pseudoaleatorios obtenidos por un generador URN\*\*.

Se considera que un buen generador debe tener alrededor de la mitad de los 19 valores de  $G'$  por encima de 0.50.



Si  $G' \geq 0.99$  el test falla y si  $0.95 \leq G' \leq 0.99$ , una vez, entonces el generador es sospechoso.

Los generadores URN01,URN02,URN11,URN12,URN13,URN14, URN15, URN22,URN28, URN30, URN31, URN35, URN36, URN37 y URN39 pasan la prueba TST01(CS2KS2).

Como ejemplo se muestra en el Cuadro N° 6.1 uno de los mejores resultados arrojados por GENTSTPC para un generador, el URN13.

Por otra parte, en el caso de URN01, a pesar de sus muy buenos resultados en TST01 con los 19 tests, falla en el test espectral, como se muestra en el Cuadro N° 6.2, lo mismo que URN11,12,14,28,31 y 37.

Si tenemos en cuenta además las Pruebas de Entropía, los Generadores que superan todas las pruebas son: URN02,URN13,URN22, URN35 y URN39. (No se programaron en GENTSTPC: URN04,07,15-19,23-27).

Esta salida, que constituye sólo la parte final del reporte completo que se agrega en el Apéndice, ha sido procesada por GENTSTPC de acuerdo a los datos introducidos por el programa ENTRADA, como se detalla en la introducción (pág.1) de dicho Apéndice.

Cuadro N° 6.1

GENERADOR: URN13

T E S T	T(I)	P[CHI-CUAD<=T(I)]
DISTR.UNIF.(TST02)	92.40	0.33
CUPONES D=5(TST04)	97.20	0.20
CUPONES FR(100R)	95.40	0.41
CUPONES D=10	98.40	0.50
GAP ARRIBA MEDIA(TST05)	82.60	0.11
GAP BAJO MEDIA	112.60	0.83
GAP (0.33,0.667)	102.40	0.61
PERMUTACIONES 3S(TST07)	91.40	0.30
PERMUTACIONES 4S	99.40	0.53
PERMUTACIONES 5S	113.40	0.84
POKER (4,4) (TST08)	107.80	0.74
POKER (5,6)	96.00	0.43
POKER (5,4)	90.40	0.28
CORRIDAS R(TST09)	102.60	0.61
CORRIDAS FR(10R)	93.80	0.37
CORRIDAS FR(100R)	119.00	0.92
PARES SERIALES 3X3(TST10)	77.00	0.04
PARES SERIALES 10X10	103.00	0.62
PARES SERIALES 20X20	117.00	0.89
CANTIDAD DE $G' = P[CHI-CUAD \leq T(I)] \geq 0.99$		0
CANTIDAD DE $G' = P[CHI-CUAD \leq T(I)] \geq 0.95$		0
CANTIDAD DE $G' = P[CHI-CUAD \leq T(I)] \geq 0.50$		10

Cuadro N° 6.2

GENERADOR: URN01  
TEST ESPECTRAL

---

LOS VALORES DE R QUE SEAN PROXIMOS A 1 SON SATISFACTORIOS LA DIMENSION:  $2 \leq T \leq 6$  ES ADECUADA EN MUCHAS APLICACIONES (REF. KNUTH D., P.101) . LA DISTANCIA INTERPLANAR D DEBE VERIFICAR  $D \leq COTA$  EL TEST ESPECTRAL HA SIDO APLICADO (RUTINA TSTM4) Y HA PRODUCIDO LOS SIGUIENTES RESULTADOS:

DIMENSION T = 2  
NU = 0.1680700003D+05  
1/NU = 0.5949901816D-04  
D = 0.5949901970D-04

COTA = 0.3051757813D-04  
R = 0.3899268255D+01  
PARA T= 2 EL TEST NO SATISFACE:  $D \leq COTA$

---

DIMENSION T = 3  
NU = 0.6389029660D+03  
1/NU = 0.1565182904D-02  
D = 0.1565182931D-02  
COTA = 0.9765625000D-03  
R = 0.3667810188D+01

PARA T= 3 EL TEST NO SATISFACE:  $D \leq COTA$

---

DIMENSION T = 4  
NU = 0.1472480900D+03  
1/NU = 0.6791259569D-02  
D = 0.6791259628D-02  
COTA = 0.5524271728D-02  
R = 0.3229038095D+01

---

DIMENSION T = 5  
NU = 0.6662582082D+02  
1/NU = 0.1500919595D-01  
D = 0.1500919554D-01  
COTA = 0.1562500000D-01  
R = 0.2859605477D+01

---

DIMENSION T = 6  
NU = 0.2991655060D+02  
1/NU = 0.3342631352D-01  
D = 0.3342631459D-01  
COTA = 0.3125000000D-01  
R = 0.3561021835D+01

---

**ILUSTRACION DE UNA CORRIDA DE GENTSTPC  
INGRESO DE DATOS Y MUESTRA DE UNA SALIDA IMPRESA**

En el presente Apéndice del Manual del Usuario del sistema GENTSTPC se incluye información acerca de los DATOS y las SALIDAS para un ejemplo consistente en la **generación** de números pseudoaleatorios empleando un generador denominado URN13 del paquete, y la **verificación** de los tests estadísticos empíricos a través del denominado TST01.

**A) DATOS**

Utilizando el programa ejecutable **ENTRADA.EXE** se ingresan los siguientes datos:

- Nombre del generador: URN13
- Selección del tipo de Tests: E (Empíricos)
- Cantidad de números aleatorios a ser generados en cada llamada del generador : 10000 (diez mil)
- Cantidad de números aleatorios a imprimir por pantalla sólo en la primera llamada al generador : 100
- Opción respecto a inicialización: 0 (defecto)
- Selección de Test : TST01
- Cantidad de lotes NLOT con NUMALE cada uno a los que se les aplicará los tests especificados en CSCODE y/o KSCODE : 1000 (mil)
- Cantidad NIMPR de lotes cuyos resultados serán impresos en la pantalla : 1
- Ingreso de TCODE: CS2KS2 (aplica Chi-cuadrado sobre Chi-cuadrado y Kolmogorov-Smirnov sobre K-S)
- Entra para TCODE = CSCS: "1" en cada test TST02,TST04,TST05,TST07, TST08,TST09 y TST10.
- Entra para TCODE=KSKS: "1" en cada test TS06 y TST11.
- Los parámetros que en cada uno de los tests antes seleccionados, salvo TST02 (que no posee parámetros), se piden por pantalla, y que se resumen:

```
TST04: P1= 1 ; P2=10000
TST05: P1= 333 ; P2= 667 ;P3=1 ; P4=10000
TST06: P1=1000 ; P2= 10 ;P3=1
```

Los tests TST07 , TST08 , TST09 y TST10 : P1=1 ; P2=10000

TST11: P1=2 ; P2=500 ; P3=10 ; P4=1

Observación:

Automáticamente quedan creados 12 archivos de datos con los cuales se ejecutará GENTSTPC(VERSIÓN GENTSTPC2.EXE):

INDICE,ARCH,ARCH1,ARCH2,ARCHT04,ARCHT05,ARCHT06,ARCHT07,ARCHT08,ARCHT09,ARCHT10 y ARCHT11.

El archivo ARCH tiene el siguiente formato:

URN13	10000NO	100		0	
TST01	1000		1	CS2KS211111110	11000000
01T02					
01T04		1	10000		
01T05	333		667	1	10000
01T06	1000		10	1	
01T07		1	10000		
01T08		1	10000		
01T09		1	10000		
01T10		1	10000		
01T11		2	500	10	1

## B) SALIDAS

Las salidas corresponden a los títulos como aparecen impresos los reportes en forma consecutiva:

- Impresión de los 100 primeros números generados por URN13
- Salidas de TST02 (test de distribución uniforme)
- Salidas de TST04 (cupones)
- Salidas de TST05 (gap)
- Salidas de TST07 (permutación)
- Salidas de TST08 (póker)
- Salidas de TST09 (corridas)
- Salidas de TST10 (pares seriales)
- Salidas de TST06 Kolmogorov-Smirnov :1000 muestras de 10 submuestras cada una.
- Salidas de TST11 Test del máximo t : aplicación a 1000 muestras con 10 submuestras de 10000 números cada una, a los tests TST02,TST05,TST07,TST08,TST09,TST10, TST06 y TST11)
- Salidas de TST01 Aplicación a 1000 grupos de valores Chi-Cuadrado generados por los tests TST02,TST05,TST07,TST08, TST09,TST10,TST06 y TST11.
- Salidas de los 19 Tests y resultado del Test Chi- cuadrado Total con relación a la tabla de gama con 99 grados de libertad.

\*\*\*\*\*  
\* LAS SIGUIENTES SALIDAS HAN SIDO PRODUCIDAS POR EL SISTEMA: " GENTSTPC " \*  
\* EL PAQUETE ESTA CONSTITUIDO POR RUTINAS PARA LA GENERACION DE NUMEROS \*  
\* PSEUDOALEATORIOS Y TESTS ESTADISTICOS PROGRAMADOS PARA MICROCOMPUTADORES.\*  
\*\*\*\*\*

\*\*\*\*\*  
\* SE HA ENCONTRADO UN GENERADOR EN LOS DATOS DE ENTRADA.TODOS LOS TESTS \*  
\* DISPONIBLES SERAN APLICADOS A ESTE GENERADOR HASTA EL FIN DE LA EJECUCION.\*  
\* \*  
\* SE ASIGNARA UN NUMERO DE CODIGO AL PRESENTE GENERADOR \*  
\* EL CODIGO ASIGNADO ES GENCODE: 1 \*  
\*\*\*\*\*  
LA INICIALIZACION ESPECIFICADA ANTES DE QUE ALGUN NUMERO SEA GENERADO ES:

INTIN(1)=	524287	INTIN(2)=	15
INTIN(3)=	32767	INTIN(4)=	20251
INTIN(5)=	24173		

LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO SIGUIENTE A LOS 10000  
YA GENERADOS) ES

INTIN(1)=	524287	INTIN(2)=	65707
INTIN(3)=	447	INTIN(4)=	20251
INTIN(5)=	24173		

ESTOS SON LOS 100 NUMEROS A SER IMPRESOS POR EL GENERADOR URN13:

0.79629517	0.84356689	0.04583740	0.14498901	0.21414185
0.46328735	0.83062744	0.65811157	0.52328491	0.91583252
0.29458618	0.20974731	0.38656616	0.67630005	0.51638794
0.26010132	0.21218872	0.44583130	0.64706421	0.70370483
0.50610352	0.17663574	0.77621460	0.88482666	0.67355347
0.06192017	0.96560669	0.92132568	0.45709229	0.92837524
0.13885498	0.27487183	0.19924927	0.46823120	0.51364136
0.50451660	0.11660767	0.74890137	0.26461792	0.14141846
0.77322388	0.43716431	0.10418701	0.62228394	0.94070435
0.17605591	0.46881104	0.21875000	0.03256226	0.55130005
0.99826050	0.90039063	0.30020142	0.08529663	0.99526978
0.06900024	0.97830200	0.69580078	0.55328369	0.10113525
0.56661987	0.05700684	0.03964233	0.63699341	0.43643188
0.30044556	0.62158203	0.55291748	0.94821167	0.49343872
0.52301025	0.55471802	0.26889038	0.33193970	0.69882202
0.32998657	0.07574463	0.48617554	0.61517334	0.19610596
0.33441162	0.03594971	0.82232666	0.18218994	0.33862305
0.43316650	0.00817871	0.48178101	0.19943237	0.42944336
0.21966553	0.21453857	0.97070313	0.40591431	0.93972778
0.43362427	0.59918213	0.54284668	0.70706177	0.31698608

LA CANTIDAD DE MUESTRAS DE NROS. ALEAT. A SER GENERADOS  
PARA SU VERIFICACION ES: 1000

LA CANTIDAD DE ESTOS, DESDE LOS CUALES SE REPORTARAN LOS RESULTADOS  
DE LOS TESTS INDIVIDUALES ES 1

TESTS QUE SUMINISTRAN VALORES CHI-CUADRADO A TST01:

UDISTB  
CUPON  
GAP  
PERMUT  
POKER  
CORRID  
PARSER

TESTS QUE SUMINISTRAN VALORES KOLMOGOROV-SMIRNOV A TST01:

K-S  
MAX T

\*\*\*\*\*

TST02: TEST DE DISTRIBUCION UNIFORME

EL TEST CHI-CUADRADO FUE APLICADO (RUTINA TST02) YHA PRODUCIDO  
LOS SIGUIENTES RESULTADOS:

NUMERO DE MUESTRA: 1

-----

INDICE DE INTERVALO	COTA SUPERIOR DEL INTERVALO	FRECUENCIA ESPERADA	FRECUENCIA OBSERVADA	CONTRIBUCION DE CHICUADRADO
1	0.01	100.00	94	0.36
2	0.02	100.00	106	0.36
3	0.03	100.00	94	0.36
4	0.04	100.00	87	1.69
5	0.05	100.00	93	0.49
6	0.06	100.00	96	0.16
7	0.07	100.00	96	0.16
8	0.08	100.00	96	0.16
9	0.09	100.00	98	0.04
10	0.10	100.00	79	4.41
11	0.11	100.00	99	0.01
12	0.12	100.00	99	0.01
13	0.13	100.00	91	0.81
14	0.14	100.00	107	0.49
15	0.15	100.00	95	0.25
16	0.16	100.00	92	0.64
17	0.17	100.00	112	1.44
18	0.18	100.00	130	9.00
19	0.19	100.00	107	0.49
20	0.20	100.00	104	0.16
21	0.21	100.00	103	0.09
22	0.22	100.00	112	1.44
23	0.23	100.00	107	0.49
24	0.24	100.00	104	0.16
25	0.25	100.00	111	1.21
26	0.26	100.00	88	1.44
27	0.27	100.00	119	3.61
28	0.28	100.00	108	0.64
29	0.29	100.00	87	1.69
30	0.30	100.00	108	0.64
31	0.31	100.00	106	0.36
32	0.32	100.00	79	4.41
33	0.33	100.00	80	4.00
34	0.34	100.00	97	0.09
35	0.35	100.00	106	0.36
36	0.36	100.00	103	0.09
37	0.37	100.00	89	1.21
38	0.38	100.00	101	0.01
39	0.39	100.00	91	0.81
40	0.40	100.00	107	0.49
41	0.41	100.00	100	0.00
42	0.42	100.00	117	2.89
43	0.43	100.00	94	0.36
44	0.44	100.00	106	0.36
45	0.45	100.00	103	0.09
46	0.46	100.00	95	0.25
47	0.47	100.00	105	0.25
48	0.48	100.00	85	2.25
49	0.49	100.00	88	1.44
50	0.50	100.00	98	0.04
51	0.51	100.00	112	1.44
52	0.52	100.00	94	0.36



USANDO LOS NUMEROS ALEATORIOS COMO HAN SIDO GENERADOS, CON D=5:

EL NUMERO DE SEGMENTOS ENCONTRADOS ES: 862  
EL ULTIMO SEGMENTO COMPLETO TERMINA EN: 9995  
LA MAXIMA LONGITUD DE UN SEGMENTO ES: 42

LONG. SEGMENTO	No. ESPERADO	No. OBSERVADO
5	33.101	33
6	66.202	75
7	86.062	88
8	92.682	80
9	90.087	83
10	82.302	92
11	72.250	63
12	61.753	54
13	51.817	53
14	42.920	47
15	35.223	39
16	28.713	26
17	23.293	24
18	18.828	21
19	15.179	14
20 O MAYOR	61.589	70

CHI-CUADRADO PARA 15 GRADOS DE LIBERTAD ES: 9.395  
PROBABILIDAD P DE QUE UNA VAR. ALEATORIA CHI-CUADRADO CON 15 GRADOS DE LIBERTAD SEA  $\leq$  X:

P	.0001	.001	.01	.05	.30	.50	.70	.95	.99	.999	.9999
-----											
X	2.14	3.48	5.23	7.26	11.7	14.3	17.3	25.0	30.6	37.7	44.3

RESULTADOS DEL TEST DEL RECOLECTOR DE CUPONES (CONTINUACION)  
UTILIZANDO LA PARTE FRACCIONARIA DE 100\*(NROS. ALEATORIOS) CON D = 5

EL NUMERO DE SEGMENTOS ENCONTRADOS ES: 877  
EL ULTIMO SEGMENTO COMPLETO TERMINA EN: 9988  
LA MAXIMA LONGITUD DE UN SEGMENTO ES: 40

LONG. SEGMENTO	NRO. ESPERADO	NRO. OBSERVADO
5	33.677	34
6	67.354	57
7	87.560	96
8	94.295	107
9	91.655	86
10	83.734	73
11	73.507	75
12	62.827	49
13	52.719	60
14	43.667	59
15	35.836	43
16	29.213	27
17	23.698	12
18	19.155	20
19	15.443	18
20 O MAYOR	62.660	61

CHI-CUADRADO PARA 15 GRADOS DE LIBERTAD ES: 23.187  
PROBABILIDAD P DE QUE UNA VAR. ALEATORIA CHI-CUADRADO CON 15 GRADOS DE LIBERTAD SEA  $\leq$  X:



```

P| .0001 .001 .01 .05 .30 .50 .70 .95 .99 .999 .9999
-----
X| 2.14 3.48 5.23 7.26 11.7 14.3 17.3 25.0 30.6 37.7 44.3

```

RESULTADOS DEL TEST DEL RECOLECTOR DE CUPONES (CONTINUACION)  
USANDO UNA REPARTICION DE LONGITUD 40 O MENOR, COND = 10

EL NUMERO DE SEGMENTOS ENCONTRADOS ES: 355  
EL ULTIMO SEGMENTO COMPLETO FINALIZA EN: 9989  
LA LONGITUD MAXIMA DE UN SEGMENTO ES: 60

LONG. SEGMENTO	No. ESPERADO	No. OBSERVADO
10 A 19	61.490	64
20 A 23	62.098	62
24 A 27	60.885	71
28 A 32	60.826	64
33 A 39	54.017	42
40 O MAYOR	55.683	52

CHI-CUADRADO PARA 5 GRADOS DE LIBERTAD ES: 4.865  
PROBABILIDAD P DE QUE UNA VAR. ALEATORIA CHI-CUADRADO CON 5 GRADOS  
DE LIBERTAD SEA <= X:

```

P| .0001 .001 .01 .05 .30 .50 .70 .95 .99 .999 .9999
-----
X| .082 .210 .554 1.15 3.00 4.35 6.06 11.1 15.1 20.5 25.7

```

\*\*\*\*\*  
\*\*\*\*\*

TST05: TEST GAP (TEST DE LOS INTERVALOS O HUECOS)

EL TEST GAP FUE APLICADO (RUTINA TST05) Y SE HAN OBTENIDO  
LOS SIGUIENTES RESULTADOS:

NUMERO DE MUESTRA: 1  
-----

INTERVALO DEL TEST 0.0 A 0.5 - CORRIDAS POR DEBAJO DE LA MEDIA

LA CANTIDAD DE "GAPS" ENCONTRADOS ES: 4973  
EL ULTIMO GAP FINALIZA EN: 9999  
LA MAXIMA LONGITUD DE GAP ENCONTRADA ES: 14

LONG. GAP	GAPS ESPERADOS	GAPS OBSERVADOS
0	2486.500	2472
1	1243.250	1254
2	621.625	622
3	310.813	317
4	155.406	133
5	77.703	92
6	38.852	33
7	19.426	29
8	9.713	11
9	4.856	7
10	4.856	3

CHI-CUADRADO PARA 10 GRADOS DE LIBERTAD ES: 13.588  
PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-CUADRADO  
CON 10 GRADOS DE LIBERTAD SEA <=X:

```

P| .0001 .001 .01 .05 .30 .50 .70 .95 .99 .999 .9999
-----
X| 0.889 1.48 2.56 3.94 7.27 9.34 11.78 18.3 23.2 29.6 35.6

```

INTERVALO DEL TEST 0.5 A 1.0 - CORRIDAS POR DEBAJO DE LA MEDIA

EL NUMERO DE GAPS ENCONTRADOS ES: 5028  
EL ULTIMO GAP FINALIZA EN: 10000  
LA MAXIMA LONGITUD DE UN GAP ENCONTRADA ES: 11

LONG. GAP	GAPS ESPERADOS	GAPS OBSERVADOS
0	2514.000	2527
1	1257.000	1254
2	628.500	627
3	314.250	310
4	157.125	146
5	78.563	87
6	39.281	43
7	19.641	23
8	9.820	5
9	4.910	4
10	4.910	2

CHI-CUADRADO PARA 10 GRADOS DE LIBERTAD ES: 7.016  
PROBABILIDAD P DE QUE UN VARIABLE ALEATORIA CHI-CUADRADO  
CON 10 GRADOS DE LIBERTAD SEA  $\leq X$ :

P	.0001	.001	.01	.05	.30	.50	.70	.95	.99	.999	.9999
X	0.889	1.48	2.56	3.94	7.27	9.34	11.78	18.3	23.2	29.6	35.6

INTERVALO DEL TEST 0.333 A 0.667

LA CANTIDAD DE GAPS ENCONTRADOS ES: 3367  
EL ULTIMO GAP FINALIZA EN: 10000  
LA MAXIMA LONGITUD ENCONTRADA ES: 25

LONG. GAP	GAPS ESPERADOS	GAPS OBSERVADOS
0	1124.578	1122
1	748.969	769
2	498.813	491
3	332.210	344
4	221.252	203
5	147.354	156
6	98.137	91
7	65.360	64
8	43.529	50
9	28.991	25
10	19.308	12
11	12.859	13
12	8.564	8
13	17.077	19

CHI-CUADRADO PARA 13 GRADOS DE LIBERTAD ES: 8.175  
PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-CUADRADO  
CON 13 GRADOS DE LIBERTAD SEA  $\leq X$ :

P	.0001	.001	.01	.05	.30	.50	.70	.95	.99	.999	.9999
X	1.733	2.62	4.11	5.89	9.93	12.34	15.12	22.4	27.7	34.5	40.9

\*\*\*\*\*  
\*\*\*\*\*

TST07: TEST DE PERMUTACION

EL TEST DE PERMUTACION FUE APLICADO (RUTINA TST07) Y HA PRODUCIDO

LOS SIGUIENTES RESULTADOS:

NUMERO DE MUESTRA: 1

-----

LA CANTIDAD DE NUMEROS ALEATORIOS UTILIZADOS EN EL TESTING ES: 10000

PERMUTACIONES DE 3 OBJETOS

LA CANTIDAD DE GRUPOS DE 3 NUMEROS EXAMINADOS ES: 3333  
LA FRECUENCIA ESPERADA DE CADA TIPO DE PERMUTACIONES: 555.333  
LA CANTIDAD DE TUPLAS DE 3 QUE FALLAN PARA SER PERMUTACIONES (Y ENTONCES SON EXCLUIDAS DE CONSIDERACION) ES: 1

PERMUTACION:	(1 2 3)	(1 3 2)	(3 2 1)	(2 1 3)	(2 3 1)	(3 1 2)
OCURRENCIAS:	567	542	561	554	537	571

CHI-CUADRADO CON 5 GRADOS DE LIBERTAD ES: 1.673  
PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-CUADRADO  
CON 5 GRADOS DE LIBERTAD SEA  $\leq$  X:

P	.0001	.001	.01	.05	.30	.50	.70	.95	.99	.999	.9999
-----											
X	.082	.210	.554	1.15	3.00	4.35	6.06	11.1	15.1	20.5	25.7

PERMUTACIONES DE 4 OBJETOS

LA CANTIDAD DE GRUPOS DE 4 NUMEROS EXAMINADOS ES: 2500  
LA FRECUENCIA ESPERADA DE CADA TIPO DE PERMUTACIONES: 104.083  
LA CANTIDAD DE TUPLAS DE 4 QUE FALLAN PARA SER PERMUTACIONES (Y ENTONCES SON EXCLUIDAS DE CONSIDERACION) ES: 2

PERMUTACION:	(1 2 3 4)	(1 2 4 3)	(1 4 3 2)	(4 2 3 1)	(1 3 2 4)	(1 3 4 2)
OCURRENCIAS:	99	125	106	101	95	92

PERMUTACION:	(1 4 2 3)	(4 3 2 1)	(3 2 1 4)	(3 2 4 1)	(3 4 1 2)	(4 2 1 3)
OCURRENCIAS:	103	104	111	94	90	98

PERMUTACION:	(2 1 3 4)	(2 1 4 3)	(2 4 3 1)	(4 1 3 2)	(2 3 1 4)	(2 3 4 1)
OCURRENCIAS:	112	112	124	105	106	116

PERMUTACION:	(2 4 1 3)	(4 3 1 2)	(3 1 2 4)	(3 1 4 2)	(3 4 2 1)	(4 1 2 3)
OCURRENCIAS:	94	110	125	95	84	97

CHI-CUADRADO CON 23 GRADOS DE LIBERTAD ES: 27.572  
PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-CUADRADO  
CON 23 GRADOS DE LIBERTAD SEA  $\leq$  X:

P	.0001	.001	.01	.05	.30	.50	.70	.95	.99	.999	.9999
-----											
X	5.75	7.53	10.2	13.1	19.1	22.3	26.0	35.2	41.6	49.7	57.1

PERMUTACIONES DE 5 OBJETOS

LA CANTIDAD DE GRUPOS DE 5 NUMEROS EXAMINADOS ES: 2000  
LA FRECUENCIA ESPERADA DE CADA TIPO DE PERMUTACIONES: 16.658  
LA CANTIDAD DE TUPLAS DE 5 QUE FALLAN PARA SER PERMUTACIONES (Y ENTONCES SON EXCLUIDAS DE CONSIDERACION) ES: 1

PERMUT.:	(1 2 3 4 5)	(1 2 3 5 4)	(1 2 5 4 3)	(1 5 3 4 2)	(5 2 3 4 1)	(1 2 4 3 5)
OCUR.:	21	15	20	16	10	9

PERMUT.:	(1 2 4 5 3)	(1 2 5 3 4)	(1 5 4 3 2)	(5 2 4 3 1)	(1 4 3 2 5)	(1 4 3 5 2)
----------	-------------	-------------	-------------	-------------	-------------	-------------

OCUR.:	17	15	18	15	29	13
PERMUT.:(1 4 5 2 3)	(1 5 3 2 4)	(5 4 3 2 1)	(4 2 3 1 5)	(4 2 3 5 1)	(4 2 5 1 3)	
OCUR.:	17	17	15	25	12	19
PERMUT.:(4 5 3 1 2)	(5 2 3 1 4)	(1 3 2 4 5)	(1 3 2 5 4)	(1 3 5 4 2)	(1 5 2 4 3)	
OCUR.:	13	18	27	26	15	23
PERMUT.:(5 3 2 4 1)	(1 3 4 2 5)	(1 3 4 5 2)	(1 3 5 2 4)	(1 5 4 2 4)	(5 3 4 2 1)	
OCUR.:	12	16	15	15	15	13
PERMUT.:(1 4 3 2 5)	(1 4 3 2 5)	(1 4 5 3 2)	(1 5 2 3 4)	(5 4 2 3 1)	(4 3 2 1 5)	
OCUR.:	10	14	15	13	18	20
PERMUT.:(4 3 2 5 1)	(4 3 5 1 2)	(4 5 2 1 3)	(5 3 2 1 4)	(3 2 1 4 5)	(3 2 1 5 4)	
OCUR.:	11	15	11	16	24	19
PERMUT.:(3 2 5 4 1)	(3 5 1 4 2)	(5 2 1 4 3)	(3 2 4 1 5)	(3 2 4 5 1)	(3 2 5 1 4)	
OCUR.:	12	15	16	12	12	21
PERMUT.:(3 5 4 1 2)	(5 2 4 1 3)	(3 4 1 2 5)	(3 4 1 5 2)	(3 4 5 2 1)	(3 5 1 2 4)	
OCUR.:	19	12	17	16	25	12
PERMUT.:(5 4 1 2 3)	(4 2 1 5 3)	(4 2 1 5 3)	(4 2 5 3 1)	(4 5 1 3 2)	(5 2 1 3 4)	
OCUR.:	16	17	15	22	16	16
PERMUT.:(2 1 3 4 5)	(2 1 3 5 4)	(2 1 5 4 3)	(2 5 3 4 1)	(5 1 3 4 2)	(2 1 4 3 5)	
OCUR.:	19	15	16	19	16	13
PERMUT.:(2 1 4 5 3)	(2 1 5 3 4)	(2 1 5 4 3)	(5 1 4 3 2)	(2 4 3 1 5)	(2 4 3 5 1)	
OCUR.:	18	24	10	17	22	21
PERMUT.:(2 4 5 1 3)	(2 5 3 1 4)	(5 4 3 1 2)	(4 1 3 2 5)	(4 1 3 5 2)	(4 1 5 2 3)	
OCUR.:	15	18	19	16	20	17
PERMUT.:(4 5 3 2 1)	(5 1 3 2 4),	(2 3 1 4 5)	(2 3 1 5 4)	(2 3 5 4 1)	(2 5 1 4 3)	
OCUR.:	12	11	17	17	12	14
PERMUT.:(5 3 1 4 2)	(2 3 4 1 5)	(2 3 4 5 1)	(2 3 5 1 4)	(2 5 4 1 3)	(5 3 4 1 2)	
OCUR.:	16	21	18	20	11	19
PERMUT.:(2 4 1 3 5)	(2 4 5 3 1)	(2 4 5 3 1)	(2 5 1 3 4)	(5 4 1 3 2)	(4 3 1 2 5)	
OCUR.:	18	23	13	20	21	14
PERMUT.:(4 3 1 5 2)	(4 3 1 2 3)	(4 5 1 2 3)	(5 3 1 2 4)	(3 1 2 4 5)	(3 1 2 5 4)	
OCUR.:	16	21	14	22	18	14
PERMUT.:(3 1 5 4 2)	(3 5 2 4 1)	(5 1 2 4 3)	(3 1 4 2 5)	(3 1 4 5 2)	(3 1 5 2 4)	
OCUR.:	17	10	16	14	14	20
PERMUT.:(3 5 4 2 1)	(5 1 4 2 3)	(3 4 2 1 5)	(3 4 2 5 1)	(3 4 5 1 2)	(3 5 2 1 4)	

OCUR.: 25 18 12 14 16 15  
 PERMUT.:(5 4 2 1 3) (4 1 2 3 5) (4 1 2 5 3) (4 1 5 3 2) (4 5 2 3 1) (5 1 2 3  
 4)  
 OCUR.: 25 17 17 17 12 15

CHI-CUADRADO CON 119 GRADOS DE LIBERTAD ES: 115.557  
 PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-CUADRADO  
 CON 119 GRADOS DE LIBERTAD SEA <= X:

P	.0001	.001	.01	.05	.30	.50	.70	.95	.99	.999	.9999
-----											
X	70.0	77.0	86.1	94.8	110.0	118.3	126.6	145.5	157.8	172.4	185.1

\*\*\*\*\*  
 \*\*\*\*\*

TST08: TEST DEL POKER

EL TEST DEL POKER HA SIDO APLICADO (RUTINA TST08) Y HA PRODUCIDO  
 LOS SIGUIENTES RESULTADOS:

NUMERO DE MUESTRA: 1  
 -----

EL TEST DEL POKER FUE APLICADO A 2500 GRUPOS DE 4NUMEROS ALEATORIOS Y 2000  
 GRUPOS DE 5 NUMEROS ALEATORIOS.  
 PARA UN GRUPO DE NUMEROS ALEATORIOS, EL TEST PRUEBA QUE TIPO DE REPARTICION  
 OCURRE.

MANOS DE 4 CON 4 REPARTICIONES

TIPO	ESPERADAS	OBSERVADAS
AAAA	2.5000	3
AAAA OR AAAB	92.5000	95
AABB	67.5000	68
AABC	1080.0000	1075
ABCD	1260.0000	1262

CHI-CUADRADO PARA 3 GRADOS DE LIBERTAD ES: 0.10  
 PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-CUADRADO  
 CON 3 GRADOS DE LIBERTAD SEA <= X:

P	.0001	.001	.01	.05	.30	.50	.70	.95	.99	.999	.9999
-----											
X	.005	.024	.114	.352	1.42	2.37	3.66	7.81	11.3	16.3	21.1

MANOS DE 5 CON 6 REPARTICIONES

TIPO	ESPERADAS	OBSERVADAS
AAAAA	0.2000	0
AAAAA OR AAAAB	9.2000	9
AAABB	18.0000	18
AAABC	144.0000	142
AABBC	216.0000	222
AABCD	1008.0000	993
ABCDE	604.8000	616

CHI-CUADRADO PARA 5 GRADOS DE LIBERTAD ES: 0.63  
 PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-CUADRADO  
 CON 5 GRADOS DE LIBERTAD SEA <= X:

P	.0001	.001	.01	.05	.30	.50	.70	.95	.99	.999	.9999
-----											
X	.082	.210	.554	1.15	3.00	4.35	6.06	11.1	15.1	20.5	25.7

-----  
MANOS DE 5 CON 4 REPARTICIONES

TIPO	ESPERADAS	OBSERVADAS
1 DIFERENTE	0.2000	0
1 O 2 DIFERENTES	27.2000	27
3 DIFERENTES	360.0000	364
4 DIFERENTES	1008.0000	993
5 DIFERENTES	604.8000	616

CHI-CUADRADO PARA 3 GRADOS DE LIBERTAD ES: 0.48  
PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-CUADRADO  
CON 3 GRADOS DE LIBERTAD SEA  $\leq$  X:

P	.0001	.001	.01	.05	.30	.50	.70	.95	.99	.999	.9999
X	.005	.024	.114	.352	1.42	2.37	3.66	7.81	11.3	16.3	21.1

-----

\*\*\*\*\*  
\*\*\*\*\*

TST09: TEST DE LAS CORRIDAS (RUNS UP)

EL TEST "RUNS UP" FUE APLICADO (RUTINA TST09) Y HA PRODUCIDO  
LOS SIGUIENTES RESULTADOS:

NUMERO DE MUESTRA: 1  
-----

CADA MUESTRA CONSISTE DE 10000 NUMEROS ALEATORIOS

UTILIZANDO EL NUMERO ALEATORIO X COMO GENERADO

TABLA DE "RUNS UP"

1717 CORRIDAS DE LONGITUD 1  
2049 CORRIDAS DE LONGITUD 2  
947 CORRIDAS DE LONGITUD 3  
248 CORRIDAS DE LONGITUD 4  
55 CORRIDAS DE LONGITUD 5  
12 CORRIDAS DE LONGITUD 6 O MAS

LA CORRIDA MAXIMA ES DE LONGITUD: 9  
VAR. ESTADISTICA, APROX.CHI-CUADRADO CON 6 GRADOS DE LIBERTAD: 8.49

UTILIZANDO LA PARTE FRACCIONARIA DE  $10 \cdot X$ :

TABLA DE "RUNS UP"

1680 CORRIDAS DE LONGITUD 1  
2100 CORRIDAS DE LONGITUD 2  
917 CORRIDAS DE LONGITUD 3  
257 CORRIDAS DE LONGITUD 4  
55 CORRIDAS DE LONGITUD 5  
11 CORRIDAS DE LONGITUD 6 O MAS

LA CORRIDA MAXIMA ES DE LONGITUD: 6  
VAR. ESTADISTICA, APROX.CHI-CUADRADO CON 6 GRADOS DE LIBERTAD: 2.09

UTILIZANDO LA PARTE FRACCIONARIA DE  $10 \cdot X$ :

TABLA DE "RUNS UP"

1686 CORRIDAS DE LONGITUD 1  
 2081 CORRIDAS DE LONGITUD 2  
 909 CORRIDAS DE LONGITUD 3  
 261 CORRIDAS DE LONGITUD 4  
 58 CORRIDAS DE LONGITUD 5  
 15 CORRIDAS DE LONGITUD 6 O MAS

LA CORRIDA MAXIMA ES DE LONGITUD: 7  
 VAR. ESTADISTICA, APROX.CHI-CUADRADO CON 6 GRADOS DE LIBERTAD: 2.21

PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-CUADRADO  
 CON 6 GRADOS DE LIBERTAD SEA <= X:

P	.0001	.001	.01	.05	.30	.50	.70	.95	.99	.999	.9999
X	.172	.381	.872	1.63	3.83	5.35	7.23	12.6	16.8	22.5	27.9

\*\*\*\*\*  
 \*\*\*\*\*

TST10: PARES SERIALES

EL TEST DE PARES SERIALES FUE APLICADO (RUTINA TST10) Y SE HAN OBTENIDO  
 LOS SIGUIENTES RESULTADOS:

NUMERO DE MUESTRA: 1  
 -----

LA CANTIDAD DE PARES DE NUMEROS ALEATORIOS PROBADOS ES: 5000

PARES DE NUMEROS(Q,R) CON Q Y R ENTRE 1 Y 3

NUMEROS EN CATEGORIA(Q,R)

	R=1	R=2	R=3
Q=1	549	554	535
Q=2	568	551	564
Q=3	564	570	545

EL NUMERO ESPERADO EN CADA CATEGORIA ES: 555.56  
 CHI-CUADRADO PARA 8 GRADOS DE LIBERTAD ES: 1.99  
 PROBABILIDAD P QUE UNA VAR. ALEAT. CHI-CUADRADO CON 8 G. DE LIBERTAD  
 SEA <= X:

P	.0001	.001	.01	.05	.30	.50	.70	.95	.99	.999	.9999
X	.464	.857	1.65	2.73	5.53	7.34	9.52	15.5	20.1	26.1	31.8

PARES DE NUMEROS(Q,R) CON Q Y R ENTRE 1 Y 10

NUMEROS EN CATEGORIA(Q,R)

	R=1	R=2	R=3	R=4	R=5	R=6	R=7	R=8	R=9	R=10
Q= 1	41	51	56	31	47	53	45	42	48	55
Q= 2	39	48	51	49	57	45	49	58	43	51

Q= 3	46	57	73	51	47	63	61	41	54	40
Q= 4	49	57	36	50	45	55	56	30	47	58
Q= 5	40	63	56	44	48	44	54	49	59	58
Q= 6	51	62	59	44	49	49	44	54	47	57
Q= 7	51	36	57	52	50	42	47	40	68	43
Q= 8	52	57	48	52	46	59	62	42	49	48
Q= 9	58	54	37	62	34	67	50	40	54	49
Q=10	43	61	41	41	53	44	45	60	52	48

EL NUMERO ESPERADO EN CADA CATEGORIA ES: 50.00  
 CHI-CUADRADO PARA 99 GRADOS DE LIBERTAD ES: 129.52  
 PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-CUADRADO  
 CON 99 GRADOS DE LIBERTAD SEA <= X:

P	.0001	.001	.01	.05	.30	.50	.70	.95	.99	.999	.999;
X	55.0	61.1	69.2	77.0	91.5	98.3	105.9	123.2	134.6	148.2	160.1

PARES DE NUMEROS(Q,R) CON Q Y R ENTRE 1 Y 20

NUMEROS EN CATEGORIA(Q,R)

	R=1	R=2	R=3	R=4	R=5	R=6	R=7	R=8	R=9	R=10
Q= 1	11	11	6	16	16	16	6	9	5	16
Q= 2	9	10	11	18	8	16	9	7	14	12
Q= 3	12	11	9	10	15	9	12	12	13	13
Q= 4	11	5	13	16	11	16	10	15	16	15
Q= 5	11	14	13	12	19	14	22	10	11	12
Q= 6	10	11	13	19	21	19	7	12	8	16
Q= 7	13	8	14	21	9	5	13	15	8	8
Q= 8	15	13	14	8	16	6	11	11	16	13
Q= 9	15	9	14	19	23	15	13	11	13	12
Q=10	7	9	13	17	12	6	14	6	11	12
Q=11	10	15	13	9	11	18	12	12	9	17
Q=12	9	17	17	23	17	13	9	11	11	12
Q=13	15	8	6	12	11	10	13	21	11	11
Q=14	11	17	4	14	19	17	7	11	11	17
Q=15	14	16	18	13	15	14	16	7	8	14
Q=16	10	12	14	12	6	13	13	16	11	13
Q=17	13	11	14	13	5	10	13	18	14	8
Q=18	17	17	5	22	11	11	15	16	4	8



Q=19	10	14	17	14	8	6	11	14	19	7
Q=20	12	7	14	16	9	18	7	9	12	15

	R=11	R=12	R=13	R=14	R=15	R=16	R=17	R=18	R=19	R=20
Q= 1	11	15	12	15	13	13	8	10	19	11
Q= 2	15	12	7	11	5	11	13	17	13	12
Q= 3	8	16	11	14	23	15	11	7	14	14
Q= 4	10	11	11	13	5	15	12	13	9	14
Q= 5	17	15	19	16	9	12	17	14	9	9
Q= 6	15	16	13	13	12	8	14	9	12	10
Q= 7	17	14	15	13	7	7	4	15	13	16
Q= 8	12	12	15	13	11	5	12	16	11	18
Q= 9	14	12	22	13	12	17	11	22	17	11
Q=10	8	10	9	10	11	9	11	15	18	12
Q=11	15	15	7	12	11	17	13	14	18	8
Q=12	12	7	13	12	16	10	10	10	18	13
Q=13	5	10	12	12	13	7	17	20	12	8
Q=14	14	13	15	8	12	8	15	16	8	15
Q=15	13	16	16	15	8	12	10	12	13	14
Q=16	13	17	19	12	10	12	14	13	15	6
Q=17	18	15	10	15	12	9	18	7	12	17
Q=18	19	15	14	11	11	8	15	14	9	11
Q=19	10	9	11	11	15	19	14	12	13	13
Q=20	11	14	12	11	13	13	12	14	12	10

EL NUMERO ESPERADO EN CADA CATEGORIA ES: 12.50  
 CHI-CUADRADO PARA 399 GRADOS DE LIBERTAD ES: 432.16  
 PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-CUADRADO  
 CON 399 GRADOS DE LIBERTAD SEA <= X:

P	.0001	.001	.01	.05	.30	.50	.70	.95	.99	.999	.9999
X	302.4	317.4	336.2	353.7	383.7	398.3	413.3	446.6	467.6	492.0	512.7

\*\*\*\*\*  
 \*\*\*\*\*

TEST06: KOLMOGOROV-SMIRNOV.

EL TEST KOLMOGOROV-SMIRNOV FUE APLICADO (RUTINA TST06)

Y SE HAN OBTENIDO LOS SIGUIENTES RESULTADOS:

ESPECIFICACIONES DEL USUARIO:

LA CANTIDAD DE NUMEROS EN CADA TEST KS ES: 1000

LA CANTIDAD DE TESTS POR MUESTRA A SER EJECUTADOS ES: 10

-----  
SUBMUESTRA 1 (DE 10) EN LA MUESTRA 1 DE NUMEROS ALEATORIOS

EL TEST K-S CON N= 1000 PRODUCE: K+= 31.0159 K- = 29.9866 K = 31.0159

AQUI, LA SIGNIFICANCIA PUEDE EVALUARSE UTILIZANDO LOS HECHOS (OBTENIDOS DE

DISTRIBUCIONES LIMITES DE K+, K-, K) DE QUE:

P(K+ <= 31.0159 )= 1.000

P(K- <= 29.9866 )= 1.000

P(K >= 31.0159 )= 1.000

-----  
SUBMUESTRA 2 (DE 10) EN LA MUESTRA 1 DE NUMEROS ALEATORIOS

EL TEST K-S CON N= 1000 PRODUCE: K+= 29.8723 K- = 29.5908 K = 29.8723

AQUI, LA SIGNIFICANCIA PUEDE EVALUARSE UTILIZANDO LOS HECHOS (OBTENIDOS DE

DISTRIBUCIONES LIMITES DE K+, K-, K) DE QUE:

P(K+ <= 29.8723 )= 1.000

P(K- <= 29.5908 )= 1.000

P(K >= 29.8723 )= 1.000

-----  
SUBMUESTRA 3 (DE 10) EN LA MUESTRA 1 DE NUMEROS ALEATORIOS

EL TEST K-S CON N= 1000 PRODUCE: K+= 31.0859 K- = 30.6942 K = 31.0859

AQUI, LA SIGNIFICANCIA PUEDE EVALUARSE UTILIZANDO LOS HECHOS (OBTENIDOS DE

DISTRIBUCIONES LIMITES DE K+, K-, K) DE QUE:

P(K+ <= 31.0859 )= 1.000

P(K- <= 30.6942 )= 1.000

P(K >= 31.0859 )= 1.000

-----  
SUBMUESTRA 4 (DE 10) EN LA MUESTRA 1 DE NUMEROS ALEATORIOS

EL TEST K-S CON N= 1000 PRODUCE: K+= 29.9194 K- = 30.0271 K = 30.0271

AQUI, LA SIGNIFICANCIA PUEDE EVALUARSE UTILIZANDO LOS HECHOS (OBTENIDOS DE

DISTRIBUCIONES LIMITES DE K+, K-, K) DE QUE:

P(K+ <= 29.9194 )= 1.000

P(K- <= 30.0271 )= 1.000

P(K >= 30.0271 )= 1.000

-----  
SUBMUESTRA 5 (DE 10) EN LA MUESTRA 1 DE NUMEROS ALEATORIOS

EL TEST K-S CON N= 1000 PRODUCE: K+= 29.6022 K- = 30.8234 K = 30.8234

AQUI, LA SIGNIFICANCIA PUEDE EVALUARSE UTILIZANDO LOS HECHOS (OBTENIDOS DE

DISTRIBUCIONES LIMITES DE K+, K-, K) DE QUE:

P(K+ <= 29.6022 )= 1.000  
P(K- <= 30.8234 )= 1.000  
P(K >= 30.8234 )= 1.000

-----  
SUBMUESTRA 6 (DE 10) EN LA MUESTRA 1 DE NUMEROS ALEATORIOS

EL TEST K-S CON N= 1000 PRODUCE: K+= 30.2578 K- = 30.9164 K = 30.9164

AQUI, LA SIGNIFICANCIA PUEDE EVALUARSE UTILIZANDO LOS HECHOS (OBTENIDOS DE

DISTRIBUCIONES LIMITES DE K+, K-, K) DE QUE:

P(K+ <= 30.2578 )= 1.000  
P(K- <= 30.9164 )= 1.000  
P(K >= 30.9164 )= 1.000

-----  
SUBMUESTRA 7 (DE 10) EN LA MUESTRA 1 DE NUMEROS ALEATORIOS

EL TEST K-S CON N= 1000 PRODUCE: K+= 29.6293 K- = 29.1270 K = 29.6293

AQUI, LA SIGNIFICANCIA PUEDE EVALUARSE UTILIZANDO LOS HECHOS (OBTENIDOS DE

DISTRIBUCIONES LIMITES DE K+, K-, K) DE QUE:

P(K+ <= 29.6293 )= 1.000  
P(K- <= 29.1270 )= 1.000  
P(K >= 29.6293 )= 1.000

-----  
SUBMUESTRA 8 (DE 10) EN LA MUESTRA 1 DE NUMEROS ALEATORIOS

EL TEST K-S CON N= 1000 PRODUCE: K+= 31.2530 K- = 30.7185 K = 31.2530

AQUI, LA SIGNIFICANCIA PUEDE EVALUARSE UTILIZANDO LOS HECHOS (OBTENIDOS DE

DISTRIBUCIONES LIMITES DE K+, K-, K) DE QUE:

P(K+ <= 31.2530 )= 1.000  
P(K- <= 30.7185 )= 1.000  
P(K >= 31.2530 )= 1.000

-----  
SUBMUESTRA 9 (DE 10) EN LA MUESTRA 1 DE NUMEROS ALEATORIOS

EL TEST K-S CON N= 1000 PRODUCE: K+= 30.3326 K- = 29.9150 K = 30.3326

AQUI, LA SIGNIFICANCIA PUEDE EVALUARSE UTILIZANDO LOS HECHOS (OBTENIDOS DE

DISTRIBUCIONES LIMITES DE K+, K-, K) DE QUE:

P(K+ <= 30.3326 )= 1.000  
P(K- <= 29.9150 )= 1.000  
P(K >= 30.3326 )= 1.000

-----  
SUBMUESTRA 10 (DE 10) EN LA MUESTRA 1 DE NUMEROS ALEATORIOS

EL TEST K-S CON N= 1000 PRODUCE: K+= 30.2164 K- = 30.4393 K = 30.4393

AQUI, LA SIGNIFICANCIA PUEDE EVALUARSE UTILIZANDO LOS HECHOS (OBTENIDOS DE

DISTRIBUCIONES LIMITES DE K+, K-, K) DE QUE:

P(K+ <= 30.2164 )= 1.000  
P(K- <= 30.4393 )= 1.000  
P(K >= 30.4393 )= 1.000

\*\*\*\*\*  
\*\*\*\*\*

TST11: MAX DE T

EL TEST K-S EN MAX DE T FUE APLICADO (RUTINA TST11) Y SE HAN OBTENIDO  
LOS SIGUIENTES RESULTADOS:

ESPECIFICACIONES DEL USUARIO:

EL TEST USA EL MAXIMO DE LOS CONJUNTOS DE 2 VALORES  
HAY 10 REPETICIONES DEL TEST POR MUESTRA.

-----  
SUBMUESTRA 1 (DE 10) EN LA MUESTRA 1 DE NUMEROS ALEATORIOS  
-----  
SUBMUESTRA 2 (DE 10) EN LA MUESTRA 1 DE NUMEROS ALEATORIOS  
-----  
SUBMUESTRA 3 (DE 10) EN LA MUESTRA 1 DE NUMEROS ALEATORIOS  
-----  
SUBMUESTRA 4 (DE 10) EN LA MUESTRA 1 DE NUMEROS ALEATORIOS  
-----  
SUBMUESTRA 5 (DE 10) EN LA MUESTRA 1 DE NUMEROS ALEATORIOS  
-----  
SUBMUESTRA 6 (DE 10) EN LA MUESTRA 1 DE NUMEROS ALEATORIOS  
-----  
SUBMUESTRA 7 (DE 10) EN LA MUESTRA 1 DE NUMEROS ALEATORIOS  
-----  
SUBMUESTRA 8 (DE 10) EN LA MUESTRA 1 DE NUMEROS ALEATORIOS  
-----  
SUBMUESTRA 9 (DE 10) EN LA MUESTRA 1 DE NUMEROS ALEATORIOS  
-----  
SUBMUESTRA 10 (DE 10) EN LA MUESTRA 1 DE NUMEROS ALEATORIOS  
\*\*\*\*\*

VALORES CHI-2 PARA CADA MUESTRA DE NUMEROS ALEATORIOS  
DEL TEST DISTRIB.UNIF.(TST02) USANDO 1000 MUESTRAS DE 10000 NUMEROS  
ALEATORIOS CADA UNA

MUESTRA	CHI-2	MUESTRA	CHI-2	MUESTRA	CHI-2	MUESTRA	CHI-2	MUESTRA	CHI-2
1	97.88	51	132.56	101	108.54	151	123.28	201	94.96
2	109.60	52	119.36	102	108.34	152	118.70	202	100.46
3	113.34	53	102.28	103	128.18	153	90.10	203	88.54
4	84.48	54	94.90	104	92.06	154	104.40	204	118.10
5	103.64	55	113.54	105	73.94	155	106.54	205	98.96
6	115.14	56	76.18	106	78.42	156	108.94	206	86.28
7	125.18	57	105.56	107	83.00	157	106.72	207	87.64
8	115.64	58	93.96	108	98.56	158	96.84	208	109.12
9	103.42	59	97.32	109	97.88	159	114.84	209	100.02
10	114.70	60	89.62	110	83.36	160	116.64	210	104.08
11	104.04	61	135.16	111	111.06	161	93.52	211	106.26
12	93.68	62	77.12	112	81.98	162	85.68	212	131.60
13	84.72	63	109.46	113	109.16	163	89.80	213	89.26
14	92.22	64	120.76	114	107.12	164	72.04	214	64.52
15	79.36	65	89.50	115	107.84	165	106.08	215	107.66
16	95.52	66	82.36	116	86.16	166	85.70	216	89.58
17	82.52	67	95.60	117	89.56	167	97.46	217	124.28
18	114.60	68	112.02	118	92.42	168	91.40	218	124.80
19	99.74	69	93.76	119	97.76	169	92.56	219	118.94
20	104.98	70	105.40	120	121.20	170	117.02	220	94.06
21	89.38	71	108.96	121	77.46	171	99.04	221	92.36
22	103.28	72	97.88	122	108.96	172	93.54	222	117.24
23	105.12	73	103.22	123	97.96	173	91.16	223	123.82
24	114.30	74	102.82	124	116.34	174	96.54	224	104.58
25	96.02	75	84.58	125	82.46	175	93.72	225	141.80
26	108.68	76	103.96	126	89.04	176	125.62	226	87.92
27	93.48	77	112.14	127	113.24	177	111.60	227	88.82
28	88.66	78	69.30	128	83.88	178	81.42	228	111.16

29	82.54	79	114.28	129	91.88	179	101.92	229	82.88
30	88.34	80	93.62	130	98.18	180	82.12	230	78.36
31	89.58	81	69.02	131	94.80	181	103.28	231	119.06
32	98.76	82	85.80	132	76.74	182	103.64	232	93.70
33	92.20	83	99.68	133	122.04	183	70.68	233	81.62
34	103.74	84	93.66	134	96.94	184	96.18	234	87.60
35	120.20	85	97.20	135	110.18	185	101.30	235	89.70
36	83.84	86	113.86	136	82.58	186	117.42	236	104.52
37	126.26	87	110.98	137	107.50	187	96.34	237	108.10
38	79.12	88	109.34	138	88.18	188	105.16	238	85.34
39	75.86	89	108.72	139	95.40	189	85.74	239	111.70
40	122.98	90	76.98	140	100.44	190	93.02	240	117.70
41	111.34	91	103.80	141	61.68	191	94.42	241	84.98
42	99.78	92	101.02	142	106.94	192	76.62	242	104.96
43	86.92	93	94.26	143	96.14	193	88.58	243	86.82
44	92.96	94	83.54	144	76.18	194	74.32	244	101.90
45	116.88	95	93.22	145	101.50	195	84.38	245	112.42
46	115.16	96	95.36	146	99.96	196	106.78	246	101.08
47	73.78	97	113.22	147	107.16	197	114.28	247	86.32
48	86.64	98	89.44	148	107.04	198	87.80	248	125.12
49	92.68	99	119.56	149	98.26	199	76.64	249	108.98
50	103.64	100	96.68	150	75.10	200	87.52	250	102.04

- - - - - siguen muestras N° 251...751...1000 - - - - -

MUESTRA	CHI-2	MUESTRA	CHI-2	MUESTRA	CHI-2	MUESTRA	CHI-2	MUESTRA	CHI-2
751	118.46	801	110.52	851	103.30	901	102.10	951	78.84
752	82.28	802	103.64	852	80.82	902	83.26	952	104.62
753	79.06	803	96.86	853	93.16	903	111.00	953	114.28
754	87.30	804	109.90	854	113.60	904	80.42	954	97.70
755	116.10	805	113.94	855	104.16	905	96.22	955	84.98
756	100.46	806	110.98	856	110.04	906	108.28	956	84.18
757	104.30	807	104.12	857	83.10	907	107.58	957	106.48
758	109.70	808	118.36	858	102.94	908	83.70	958	107.58
759	97.52	809	102.74	859	88.28	909	91.76	959	85.96
760	101.42	810	111.64	860	107.62	910	99.64	960	78.40
761	71.90	811	85.38	861	79.20	911	123.84	961	109.94
762	89.12	812	105.64	862	84.62	912	92.26	962	109.24
763	101.94	813	87.84	863	101.58	913	92.10	963	84.18
764	91.32	814	87.06	864	114.98	914	109.18	964	80.72
765	90.42	815	76.68	865	103.94	915	99.26	965	91.06
766	106.68	816	99.10	866	83.08	916	131.40	966	91.42
767	111.20	817	108.86	867	109.20	917	103.42	967	115.98
768	104.64	818	114.90	868	115.66	918	100.48	968	98.34
769	128.22	819	117.56	869	117.24	919	82.82	969	105.70
770	76.66	820	77.78	870	111.50	920	114.76	970	106.22
771	88.08	821	93.46	871	99.08	921	81.60	971	106.30
772	114.02	822	80.38	872	96.18	922	94.84	972	105.48
773	96.64	823	102.80	873	84.58	923	97.72	973	110.06
774	82.90	824	84.88	874	86.22	924	117.52	974	82.66
775	100.86	825	101.30	875	97.08	925	104.88	975	97.12
776	82.98	826	90.06	876	105.46	926	104.34	976	90.00
777	99.40	827	91.04	877	98.30	927	115.10	977	93.38
778	100.18	828	104.94	878	90.04	928	140.26	978	110.18
779	73.28	829	78.94	879	97.72	929	109.18	979	104.68
780	98.40	830	90.96	880	91.60	930	97.22	980	112.96
781	87.62	831	86.04	881	112.86	931	115.58	981	102.96
782	104.52	832	107.68	882	117.82	932	86.34	982	119.52
783	72.02	833	99.04	883	91.34	933	79.48	983	107.34
784	104.44	834	76.26	884	73.92	934	105.96	984	96.44
785	95.58	835	111.82	885	86.60	935	80.74	985	102.16
786	75.66	836	87.68	886	118.44	936	84.68	986	106.82
787	100.48	837	71.86	887	90.06	937	102.32	987	85.42
788	95.80	838	87.24	888	91.20	938	103.74	988	74.92
789	91.88	839	80.72	889	122.42	939	88.50	989	129.50
790	93.88	840	88.14	890	92.72	940	105.40	990	88.84
791	85.88	841	128.34	891	97.82	941	79.66	991	92.36
792	82.06	842	78.22	892	88.08	942	94.76	992	100.74
793	106.40	843	112.54	893	100.44	943	94.50	993	113.62
794	90.14	844	104.38	894	101.00	944	110.60	994	96.40
795	143.68	845	81.72	895	105.16	945	82.16	995	89.92
796	103.50	846	100.68	896	87.86	946	97.18	996	94.62
797	100.04	847	84.98	897	88.32	947	115.88	997	100.28
798	96.18	848	102.54	898	99.44	948	111.86	998	123.64
799	86.74	849	90.96	899	96.42	949	121.30	999	99.36
800	105.10	850	119.16	900	84.64	950	93.18	1000	61.40

\*\*\*\*\*

VALORES CHI-2 PARA CADA MUESTRA DE NUMEROS ALEATORIOS  
 DEL TEST DE CUPONES (TST04) USANDO 1000 MUESTRAS DE 10000 NUMEROS  
 ALEATORIOS CADA UNA

MUESTRA	CHI-2 ENTEROS 1-5	CHI-2 ENTEROS 1-5 FRAC(100*X)	CHI-2 ENTEROS 1-10	MUESTRA	CHI-2 ENTEROS 1-5	CHI-2 ENTEROS 1-5 FRAC(100*X)	CHI-2 ENTEROS 1-10
1	9.40	23.19	4.87	51	7.50	7.69	3.76
2	14.99	13.48	6.25	52	8.05	12.00	2.18
3	14.32	15.04	4.95	53	7.78	16.42	1.74
4	10.44	14.98	2.74	54	15.02	25.26	2.74
5	17.08	17.23	6.76	55	21.51	10.98	2.11
6	20.83	12.57	13.56	56	26.29	18.05	10.32
7	15.05	5.35	2.01	57	28.47	8.73	2.89
8	19.54	17.46	6.76	58	12.06	15.95	1.13
9	20.18	18.60	3.99	59	15.76	10.55	3.09
10	6.94	9.78	7.74	60	16.56	13.98	8.06
11	9.95	18.52	2.65	61	22.93	22.42	3.15
12	16.37	13.53	5.53	62	10.82	14.07	1.72
13	17.88	9.21	3.97	63	13.90	20.37	3.43
14	20.54	11.19	7.43	64	14.01	19.97	10.68
15	13.64	14.45	3.30	65	13.26	21.90	3.06
16	11.80	9.47	2.15	66	16.76	10.93	6.61

17	27.33	14.00	2.62	67	17.29	16.76	5.15
18	8.53	11.77	2.04	68	11.73	18.47	1.95
19	15.07	9.39	6.11	69	12.51	17.16	4.74
20	15.93	12.06	5.99	70	11.77	30.39	4.47
21	20.93	17.05	2.87	71	22.22	14.57	4.67
22	8.17	13.60	5.89	72	12.60	14.58	8.17
23	15.57	18.66	14.75	73	17.23	6.70	2.55
24	17.06	25.71	2.53	74	14.75	17.43	5.93
25	7.88	13.78	1.55	75	7.71	17.93	3.34
26	14.02	13.90	5.16	76	18.46	13.68	6.44
27	14.71	13.99	9.07	77	13.75	13.95	8.47
28	10.45	6.72	3.92	78	12.50	24.46	6.20
29	17.32	14.14	2.99	79	11.54	14.06	2.53
30	4.10	14.23	17.15	80	18.77	16.04	12.90
31	15.59	24.99	1.36	81	17.92	11.12	6.01
32	17.43	14.96	3.41	82	29.27	14.14	14.01
33	14.47	15.80	6.36	83	14.46	7.22	11.26
34	14.15	9.06	8.68	84	12.06	10.21	3.06
35	11.53	9.67	0.51	85	10.62	16.82	1.16
36	15.67	16.64	3.57	86	20.80	14.95	3.91
37	14.86	6.78	2.70	87	15.77	17.06	2.13
38	19.77	12.66	1.91	88	7.77	13.66	5.21
39	7.19	13.98	4.02	89	12.50	20.79	7.15
40	15.13	15.76	1.64	90	10.16	11.24	4.39
41	20.78	22.52	2.45	91	27.86	17.56	0.87
42	18.80	14.53	4.21	92	11.23	12.09	4.00
43	15.04	8.30	3.70	93	15.28	34.16	5.37
44	19.97	11.66	4.71	94	12.21	13.94	3.24
45	8.17	10.13	8.88	95	26.57	17.99	8.38
46	15.67	8.55	0.63	96	18.39	16.42	4.26
47	14.23	18.32	4.53	97	19.64	18.53	6.53
48	11.73	16.79	3.89	98	15.08	9.23	4.33
49	18.31	11.94	5.81	99	16.53	19.78	9.81
50	14.82	19.03	2.77	100	12.67	20.77	0.79

- - - - siguen muestras N° 101...801 a 1000 - - - -

MUESTRA	CHI-2 ENTEROS 1-5	CHI-2 ENTEROS 1-5 FRAC(100*X)	CHI-2 ENTEROS 1-10	MUESTRA	CHI-2 ENTEROS 1-5	CHI-2 ENTEROS 1-5 FRAC(100*X)	CHI-2 ENTEROS 1-10
801	14.38	17.80	1.91	851	24.23	9.31	7.81
802	14.85	9.33	12.27	852	10.12	18.16	7.43
803	14.43	12.55	2.45	853	4.85	4.74	3.73
804	21.31	16.57	1.95	854	11.77	10.54	2.63
805	15.03	14.74	5.23	855	14.24	11.49	4.14
806	17.59	11.57	1.37	856	17.61	7.47	10.70
807	9.83	12.60	4.69	857	15.82	8.84	3.90
808	27.40	6.64	8.73	858	13.62	10.37	3.20
809	9.59	25.24	8.30	859	11.34	17.19	1.41
810	12.91	10.02	3.62	860	8.72	12.49	18.39
811	11.07	22.70	6.59	861	20.29	17.37	1.33
812	11.14	13.44	4.87	862	12.86	7.83	13.46
813	6.02	6.21	10.86	863	9.43	12.24	1.70
814	21.50	11.04	3.30	864	9.80	17.69	3.84
815	8.28	16.78	4.79	865	16.56	9.35	2.96
816	13.65	13.82	1.25	866	9.24	8.01	4.70
817	15.31	10.34	5.89	867	17.47	13.14	9.79
818	9.70	6.63	7.96	868	14.80	10.62	3.39
819	14.68	13.54	0.92	869	15.07	15.34	1.06
820	9.74	6.08	4.98	870	21.36	19.65	2.33
821	10.99	16.59	1.62	871	9.97	14.61	1.84
822	9.48	16.93	10.41	872	11.89	12.08	2.43
823	13.18	10.18	14.33	873	18.84	6.92	8.21
824	17.89	9.01	11.90	874	21.91	15.67	2.56
825	23.65	14.19	3.52	875	29.89	7.27	4.10
826	12.48	15.92	9.16	876	9.28	15.71	0.92
827	30.21	11.64	4.09	877	15.91	16.60	1.76
828	27.56	13.11	4.72	878	18.79	6.16	5.34
829	19.83	29.90	1.93	879	9.97	9.07	1.77
830	19.74	18.67	10.38	880	21.68	13.93	0.65
831	12.61	11.18	9.62	881	18.75	18.36	6.63
832	8.22	10.16	12.42	882	12.10	12.73	2.82
833	10.05	11.57	0.91	883	8.43	20.00	7.44
834	16.11	10.62	2.65	884	9.63	10.24	1.61
835	19.08	13.60	4.58	885	12.53	14.99	1.32
836	9.61	15.04	2.92	886	15.57	10.96	7.93
837	7.49	10.80	2.77	887	15.60	15.03	2.51
838	19.27	15.76	10.22	888	5.18	11.48	10.27
839	11.63	13.26	8.16	889	12.60	16.29	1.53
840	13.32	9.17	2.46	890	5.63	19.44	2.05
841	19.50	11.42	9.38	891	15.08	19.92	3.75
842	20.21	13.49	6.12	892	23.03	20.53	6.50
843	18.99	12.84	1.63	893	18.17	23.02	5.57
844	14.85	17.95	3.15	894	17.24	12.53	5.02
845	15.65	14.51	5.81	895	11.98	13.52	12.73
846	27.69	29.63	2.59	896	17.48	15.93	2.30
847	13.25	10.29	4.83	897	11.15	20.57	10.72
848	15.60	14.00	2.92	898	15.11	19.02	5.93
849	10.99	13.63	7.45	899	15.77	14.47	5.99
850	24.34	21.42	2.94	900	18.78	17.37	7.22

  

MUESTRA	CHI-2 ENTEROS 1-5	CHI-2 ENTEROS 1-5 FRAC(100*X)	CHI-2 ENTEROS 1-10	MUESTRA	CHI-2 ENTEROS 1-5	CHI-2 ENTEROS 1-5 FRAC(100*X)	CHI-2 ENTEROS 1-10
901	12.64	11.23	13.70	951	16.22	15.88	11.71
902	16.80	12.60	2.93	952	5.82	12.83	9.31
903	16.17	26.37	5.55	953	14.45	10.94	3.03
904	17.99	24.86	3.85	954	16.22	19.08	7.39
905	10.96	16.00	7.37	955	4.96	7.08	7.98

906	18.93	5.42	7.01	956	9.38	9.29	3.56
907	8.41	13.86	4.27	957	15.88	12.23	11.30
908	15.16	15.03	6.22	958	8.47	6.26	2.44
909	27.84	21.86	0.97	959	19.94	14.44	3.80
910	11.56	10.27	2.35	960	23.28	13.61	8.51
911	13.39	10.43	9.44	961	11.79	7.91	1.46
912	14.85	16.14	3.70	962	10.07	21.02	3.54
913	11.69	14.61	4.24	963	17.45	10.78	5.03
914	12.14	18.33	2.02	964	19.51	19.04	3.56
915	16.99	15.28	7.55	965	19.14	22.46	2.76
916	9.98	17.14	5.74	966	8.41	13.32	2.21
917	14.49	19.72	2.70	967	13.53	13.01	4.52
918	15.53	11.07	5.04	968	9.01	12.68	10.53
919	17.67	12.21	14.91	969	12.54	24.07	3.87
920	12.21	18.54	2.60	970	7.64	9.60	2.82
921	12.35	27.67	13.89	971	13.04	6.58	7.21
922	10.86	14.73	5.01	972	17.91	18.15	4.00
923	9.71	17.25	7.05	973	23.23	16.17	7.53
924	15.45	16.52	5.72	974	6.43	14.39	9.15
925	21.68	10.75	4.54	975	19.95	12.87	2.62
926	41.58	7.51	5.29	976	15.26	12.33	2.62
927	29.80	21.06	1.58	977	15.49	17.10	5.70
928	12.13	15.14	1.06	978	17.54	15.28	5.63
929	8.20	17.51	7.06	979	5.01	10.81	13.05
930	15.93	20.19	9.99	980	7.12	10.23	9.79
931	7.85	18.57	6.97	981	13.74	13.79	2.45
932	8.27	6.99	0.42	982	19.68	14.64	2.70
933	9.19	12.75	0.51	983	10.73	12.87	6.24
934	17.95	9.57	9.68	984	7.60	12.35	8.68
935	12.17	16.58	5.85	985	9.75	16.19	3.43
936	11.47	16.99	6.60	986	31.65	20.08	1.91
937	17.54	11.33	12.53	987	17.19	15.62	7.41
938	12.55	13.88	4.71	988	7.02	14.14	5.61
939	13.36	19.88	5.08	989	12.77	9.35	6.85
940	13.02	12.72	11.60	990	11.10	12.60	5.10
941	12.71	12.61	8.43	991	4.87	6.53	0.71
942	9.44	9.08	19.35	992	12.72	10.69	4.38
943	17.77	15.62	2.31	993	8.40	11.24	5.36
944	23.84	9.40	5.59	994	15.38	14.87	6.06
945	22.51	16.42	0.46	995	17.49	9.58	3.36
946	20.99	16.34	4.73	996	5.14	8.50	4.34
947	10.91	18.46	3.52	997	16.02	11.35	5.35
948	8.43	13.10	11.07	998	10.60	8.95	3.65
949	17.84	9.57	4.38	999	12.62	12.82	3.07
950	13.10	16.21	6.94	1000	18.81	6.53	13.19

\*\*\*\*\*  
 VALORES CHI-2 PARA CADA MUESTRA DE NUMEROS ALEATORIOS DEL TEST GAP (TST05)  
 USANDO 1000 MUESTRAS DE 10000 NUMEROS ALEATORIOS CADA UNA.

MUESTRA	CHI-2 INTERVALO (0,0.5)	CHI-2 INTERVALO (0.5,1)	CHI-2 INTERVALO (.333,.667)	MUESTRA	CHI-2 INTERVALO (0,0.5)	CHI-2 INTERVALO (0.5,1)	CHI-2 INTERVALO (.333,.667)
1	13.59	7.02	8.18	51	12.28	6.36	6.68
2	12.04	5.24	11.99	52	10.89	19.91	12.51
3	12.60	6.88	7.88	53	6.72	8.61	10.79
4	4.84	12.28	9.43	54	5.96	8.34	14.97
5	10.61	13.43	12.53	55	4.82	8.33	8.56
6	8.34	10.83	16.42	56	14.12	10.69	4.46
7	9.06	10.96	23.82	57	10.93	8.72	12.02
8	9.56	12.07	34.65	58	20.64	4.82	8.42
9	11.78	25.73	7.25	59	6.48	7.58	11.95
10	9.97	8.34	11.99	60	20.79	9.14	17.27
11	10.10	7.73	10.48	61	10.31	17.10	10.50
12	9.73	6.76	12.86	62	8.57	10.42	13.87
13	17.32	8.67	10.95	63	6.24	7.57	9.10
14	12.31	27.75	13.83	64	11.42	6.41	6.49
15	6.24	12.52	11.28	65	10.67	13.41	10.78
16	7.29	9.42	15.85	66	7.23	10.22	17.82
17	6.14	7.45	19.06	67	6.64	7.77	11.85
18	9.79	21.27	11.75	68	8.83	7.58	11.96
19	7.59	3.94	16.39	69	5.96	6.33	20.37
20	14.95	9.35	13.84	70	11.37	7.37	7.98
21	3.89	2.71	14.89	71	6.62	7.76	13.18
22	7.33	6.25	16.19	72	3.05	5.34	14.04
23	9.91	7.98	20.05	73	12.30	13.49	15.44
24	10.82	6.68	12.27	74	30.99	8.67	25.48
25	11.74	5.40	8.76	75	7.34	13.79	8.28
26	14.93	25.35	14.85	76	10.00	10.20	10.88
27	7.07	14.71	14.65	77	14.53	7.17	21.45
28	13.07	9.53	15.37	78	7.45	9.18	10.51
29	6.60	9.66	16.04	79	4.22	13.01	15.64
30	9.62	12.73	5.04	80	18.00	8.56	6.21
31	12.66	8.36	11.47	81	6.25	5.91	5.92
32	3.59	9.34	10.19	82	5.83	10.87	27.74
33	10.49	18.97	14.59	83	9.49	9.69	9.26
34	19.47	14.41	14.50	84	9.40	13.67	14.02
35	10.59	10.87	15.92	85	3.09	8.74	17.45
36	14.15	19.77	11.47	86	5.56	3.00	11.83
37	9.02	13.17	15.66	87	6.95	17.61	9.26
38	12.22	7.16	7.32	88	14.00	13.09	20.54
39	20.52	10.27	21.07	89	7.66	11.16	16.83
40	9.46	7.99	22.50	90	6.27	4.67	10.21
41	10.74	9.32	12.00	91	8.20	4.55	12.82
42	9.73	6.92	6.14	92	16.75	12.28	8.99
43	4.10	6.70	8.43	93	8.75	10.22	7.03
44	8.63	11.27	7.72	94	15.97	11.39	10.55
45	19.43	10.26	13.07	95	14.31	12.24	8.40
46	16.84	14.66	7.87	96	10.40	15.75	13.84

47	8.97	11.13	19.32	97	7.10	7.02	9.55
48	3.81	8.52	10.52	98	7.80	8.94	15.48
49	4.86	12.92	17.73	99	9.38	10.61	9.01
50	7.63	9.26	9.90	100	15.03	12.42	11.21

- - - - siguen muestras N° 101...801 a 1000 - - - -

MUESTRA	CHI-2 INTERVALO (0,0,5)	CHI-2 INTERVALO (0,5,1)	CHI-2 INTERVALO (.333,.667)	MUESTRA	CHI-2 INTERVALO (0,0,5)	CHI-2 INTERVALO (0,5,1)	CHI-2 INTERVALO (.333,.667)
801	9.15	16.43	11.59	851	5.36	12.52	12.38
802	11.94	16.80	11.59	852	7.00	8.87	16.62
803	9.89	8.84	15.26	853	7.75	16.92	14.01
804	4.85	14.67	5.85	854	23.09	6.05	15.46
805	10.44	9.65	20.85	855	17.21	8.29	20.87
806	5.54	7.95	13.18	856	16.06	27.60	12.03
807	13.58	11.26	15.53	857	7.67	8.13	23.12
808	15.75	18.75	17.61	858	20.62	9.39	22.80
809	2.25	6.46	8.54	859	6.74	14.25	11.11
810	11.45	10.96	4.70	860	12.66	11.68	9.70
811	11.01	8.88	14.13	861	4.93	10.68	16.07
812	9.29	8.67	12.87	862	6.36	11.28	12.50
813	21.49	12.87	5.43	863	16.08	19.52	16.73
814	5.23	10.21	7.69	864	5.28	11.12	13.99
815	8.78	5.96	14.56	865	6.24	4.78	9.55
816	8.21	9.12	13.17	866	11.43	9.10	14.66
817	11.19	11.59	22.87	867	16.88	1.82	18.82
818	13.84	2.71	12.06	868	10.98	12.46	11.61
819	8.95	11.96	13.29	869	8.03	7.04	15.87
820	7.65	6.90	8.69	870	8.39	9.83	8.76
821	13.04	9.98	11.50	871	14.94	12.99	21.30
822	22.44	7.33	6.98	872	7.08	10.30	9.27
823	8.84	13.30	6.57	873	12.63	9.29	9.22
824	9.21	6.87	18.97	874	13.99	9.90	17.09
825	10.37	14.18	17.48	875	13.00	9.91	12.10
826	6.68	22.98	15.75	876	11.98	11.84	9.95
827	5.77	2.74	18.90	877	17.68	14.86	5.64
828	14.27	12.89	8.18	878	2.93	11.67	13.48
829	12.09	10.57	14.61	879	2.96	6.34	17.74
830	7.11	8.31	21.07	880	13.81	3.49	6.28
831	11.79	7.73	16.97	881	17.26	11.77	8.49
832	15.15	12.25	12.34	882	3.27	12.88	18.21
833	4.20	8.85	13.63	883	19.77	7.48	9.17
834	13.81	9.61	15.08	884	13.13	6.98	6.25
835	20.36	12.52	11.67	885	15.75	2.71	11.12
836	11.08	10.07	6.33	886	10.70	8.67	17.64
837	10.82	10.83	16.16	887	9.25	6.46	17.80
838	9.22	7.18	13.92	888	6.55	7.15	10.18
839	9.93	9.45	7.11	889	11.14	16.43	6.23
840	11.26	14.10	15.33	890	6.39	3.55	19.98
841	13.94	10.58	7.52	891	6.24	8.20	10.55
842	7.48	8.90	12.14	892	9.56	16.96	18.87
843	4.85	12.97	21.82	893	11.83	10.94	20.20
844	1.54	16.11	13.66	894	3.71	14.06	20.74
845	14.73	23.32	15.44	895	6.05	7.42	17.87
846	11.76	6.90	11.57	896	12.20	7.09	20.63
847	14.60	3.91	7.14	897	13.13	12.34	13.71
848	10.64	8.70	28.35	898	9.30	4.35	10.74
849	6.57	4.17	18.69	899	8.85	14.41	13.34
850	6.16	15.95	19.09	900	6.24	5.21	12.95

  

MUESTRA	CHI-2 INTERVALO (0,0,5)	CHI-2 INTERVALO (0,5,1)	CHI-2 INTERVALO (.333,.667)	MUESTRA	CHI-2 INTERVALO (0,0,5)	CHI-2 INTERVALO (0,5,1)	CHI-2 INTERVALO (.333,.667)
901	12.53	16.94	13.42	951	6.73	5.28	7.94
902	10.42	4.59	10.62	952	11.42	6.47	10.08
903	12.83	8.12	29.20	953	22.63	11.72	11.23
904	7.83	10.91	7.87	954	10.83	10.85	25.47
905	5.35	9.17	9.19	955	12.04	7.50	5.53
906	6.64	18.35	19.84	956	15.58	5.93	10.61
907	8.00	10.48	3.99	957	8.70	13.51	13.77
908	11.55	13.28	24.67	958	5.13	10.40	23.41
909	5.27	2.84	11.91	959	3.40	5.40	14.91
910	5.34	7.31	5.70	960	32.36	17.34	22.93
911	17.55	7.13	13.74	961	5.39	5.05	16.74
912	6.08	9.60	5.49	962	12.52	12.82	18.85
913	3.04	5.97	22.09	963	5.53	6.67	11.21
914	2.00	12.09	5.03	964	24.97	11.00	9.35
915	3.94	11.18	8.60	965	7.80	13.65	18.97
916	4.93	11.37	12.75	966	7.37	10.38	24.30
917	16.33	15.57	12.21	967	7.08	8.37	11.98
918	14.24	6.70	14.13	968	4.91	7.50	18.81
919	8.64	4.60	12.66	969	9.19	11.21	22.74
920	29.66	3.30	11.25	970	5.10	5.17	15.57
921	8.29	6.10	10.04	971	5.10	9.72	17.35
922	7.01	12.10	11.13	972	8.27	7.86	15.97
923	8.56	5.49	22.55	973	5.64	25.92	8.59
924	8.40	10.06	4.85	974	19.94	12.25	16.87
925	3.84	6.66	11.48	975	11.75	12.57	7.69
926	6.11	12.41	5.62	976	9.63	11.96	16.21
927	7.77	14.71	12.91	977	15.25	8.69	7.08
928	18.73	12.89	13.26	978	13.57	12.99	11.95
929	5.17	2.94	13.29	979	9.65	2.71	19.96
930	35.52	14.63	11.90	980	5.04	5.81	4.93
931	8.36	4.43	16.86	981	10.47	11.23	5.70
932	16.51	12.88	7.73	982	13.42	12.51	17.15
933	5.71	12.74	19.44	983	4.49	9.25	7.59
934	13.58	10.45	19.09	984	16.24	12.31	10.27
935	7.24	3.80	15.36	985	6.48	7.90	14.11
936	18.44	11.25	6.54	986	14.53	7.65	12.28
937	10.95	13.49	14.43	987	13.21	4.96	7.81



938	9.15	10.89	9.37	988	10.66	12.41	6.01
939	12.78	11.92	15.46	989	3.31	4.65	22.26
940	8.20	6.58	9.78	990	12.87	5.78	10.61
941	8.56	5.13	7.15	991	6.79	12.97	27.23
942	6.77	4.21	20.47	992	16.38	3.91	6.11
943	7.73	4.46	14.41	993	12.44	11.93	17.21
944	4.51	10.87	17.20	994	12.30	6.82	15.82
945	12.17	6.83	29.59	995	6.56	6.22	13.46
946	5.04	7.05	7.02	996	5.02	10.45	11.34
947	10.74	5.22	29.41	997	13.05	0.92	21.64
948	7.89	10.31	9.11	998	6.01	11.74	4.66
949	13.44	11.92	25.85	999	10.06	13.17	23.39
950	5.15	15.38	19.39	1000	4.07	6.08	17.15

\*\*\*\*\*  
 VALORES CHI-2 PARA CADA MUESTRA DE NUMEROS ALEATORIOS DEL TEST DE PERMUTACION (TST07) UTILIZANDO 1000 MUESTRAS DE 10000 NUMEROS ALEATORIOS CADA UNA.

MUESTRA	CHI-2 3 TUPLAS	CHI-2 4 TUPLAS	CHI-2 5 TUPLAS	MUESTRA	CHI-2 3 TUPLAS	CHI-2 4 TUPLAS	CHI-2 5 TUPLAS
1	1.67	27.57	115.56	51	4.70	23.23	110.44
2	12.34	18.96	154.46	52	8.75	17.96	118.48
3	4.21	19.29	137.92	53	10.30	17.68	112.72
4	3.87	29.73	115.00	54	1.26	32.15	168.16
5	8.20	14.51	128.68	55	1.28	26.24	116.40
6	1.94	13.99	101.10	56	3.59	18.46	117.16
7	5.99	28.47	108.83	57	3.13	19.98	156.16
8	0.91	19.16	152.18	58	10.35	20.19	141.94
9	2.29	27.60	126.88	59	1.89	27.55	121.36
10	2.41	32.25	108.76	60	5.78	28.97	130.48
11	4.90	11.18	129.96	61	4.41	17.47	113.64
12	3.69	34.54	115.00	62	1.71	29.85	124.60
13	4.48	20.27	133.00	63	3.87	24.01	134.08
14	7.71	22.75	145.12	64	3.88	37.22	108.40
15	3.20	32.31	136.21	65	0.88	21.52	141.40
16	6.77	10.51	100.36	66	4.84	19.77	114.96
17	2.85	22.42	130.36	67	4.95	13.28	131.49
18	6.02	23.46	123.48	68	1.26	19.87	83.68
19	0.77	17.95	150.85	69	1.14	12.16	130.85
20	2.53	17.56	129.76	70	9.12	14.74	130.24
21	5.37	16.54	124.44	71	1.86	23.36	124.16
22	8.66	27.20	119.88	72	5.62	14.12	111.16
23	8.96	19.29	119.84	73	0.94	22.59	114.28
24	4.48	45.29	98.75	74	0.76	21.00	106.36
25	4.11	32.40	106.72	75	8.55	19.81	134.01
26	3.86	18.10	124.96	76	13.32	17.85	109.19
27	7.79	27.93	107.80	77	6.58	27.59	106.43
28	6.77	23.36	112.84	78	5.11	15.58	98.51
29	3.62	28.62	112.27	79	5.06	23.46	119.32
30	7.45	8.81	141.04	80	2.65	14.76	114.40
31	1.64	33.75	134.56	81	5.00	10.36	113.47
32	1.90	17.64	120.68	82	2.19	14.64	96.60
33	3.15	32.15	133.09	83	1.59	27.24	110.87
34	6.73	14.16	110.83	84	1.72	20.73	147.40
35	6.82	25.29	132.29	85	8.59	27.17	144.46
36	1.66	20.88	128.88	86	17.28	21.75	123.32
37	2.83	16.24	82.48	87	2.85	27.47	107.08
38	6.62	20.90	114.64	88	8.02	24.75	141.91
39	3.37	20.94	110.63	89	6.26	32.35	133.98
40	12.91	22.53	119.08	90	1.66	30.62	102.23
41	9.72	18.29	95.92	91	6.32	27.51	110.92
42	9.28	25.53	112.80	92	7.44	46.25	142.72
43	4.61	24.84	108.04	93	5.91	14.11	101.56
44	4.18	22.69	158.44	94	5.67	20.33	114.96
45	6.88	20.15	115.80	95	6.84	19.39	153.98
46	3.02	23.10	133.77	96	4.19	13.07	121.56
47	2.69	12.86	129.96	97	4.76	19.41	94.13
48	1.23	14.72	114.16	98	2.75	20.56	124.96
49	6.37	26.07	110.87	99	4.27	24.97	125.44
50	3.71	33.74	117.68	100	2.39	26.59	123.56

- - - - siguen muestras N° 101...801 a 1000 - - - -

MUESTRA	CHI-2 3 TUPLAS	CHI-2 4 TUPLAS	CHI-2 5 TUPLAS	MUESTRA	CHI-2 3 TUPLAS	CHI-2 4 TUPLAS	CHI-2 5 TUPLAS
801	5.92	36.36	142.45	851	2.13	26.05	141.16
802	3.11	19.90	89.68	852	5.30	25.09	112.24
803	2.63	24.07	118.60	853	4.02	27.95	133.60
804	4.31	14.99	117.64	854	5.58	24.65	99.40
805	2.52	23.00	105.35	855	4.56	18.21	143.77
806	9.42	36.95	95.39	856	6.61	26.44	106.48
807	5.67	33.96	149.44	857	4.85	25.21	107.03
808	4.40	21.36	107.59	858	1.48	20.37	125.64
809	3.66	16.18	112.00	859	9.57	28.28	120.64
810	3.93	19.94	102.64	860	1.71	18.31	118.00
811	0.63	28.89	108.76	861	1.21	18.77	97.96
812	3.22	30.01	138.76	862	4.78	14.76	124.36
813	8.20	17.83	99.35	863	8.87	17.47	128.20
814	13.26	8.77	124.12	864	9.14	21.29	133.69
815	2.00	22.32	113.92	865	0.57	20.73	134.85
816	1.23	29.74	101.75	866	5.24	22.90	139.72
817	1.72	16.28	113.32	867	7.16	40.16	132.97
818	2.04	15.41	110.80	868	3.82	26.87	143.20
819	9.22	20.15	134.08	869	5.08	10.99	95.63
820	6.30	29.64	123.72	870	4.64	16.31	103.24

821	3.06	34.70	127.44	871	3.25	24.61	119.80
822	2.74	22.04	120.60	872	7.13	21.40	105.47
823	8.85	32.61	95.92	873	6.41	41.62	126.40
824	4.44	41.75	119.24	874	4.98	14.28	137.08
825	8.37	29.66	128.08	875	3.52	20.00	120.00
826	1.38	24.67	80.68	876	5.08	15.22	119.64
827	6.12	19.75	125.08	877	3.71	13.95	121.60
828	6.40	11.55	139.12	878	4.39	27.24	104.32
829	5.56	19.08	123.68	879	4.55	20.12	131.65
830	4.80	24.05	127.92	880	6.49	21.75	102.76
831	2.89	14.68	128.92	881	7.71	25.07	142.24
832	2.77	22.07	122.40	882	9.19	28.47	146.56
833	2.81	20.92	108.64	883	1.60	24.76	124.60
834	10.29	21.38	117.12	884	8.16	11.03	103.86
835	6.23	19.58	130.65	885	5.86	15.20	109.79
836	3.84	22.11	84.76	886	1.78	20.31	105.59
837	6.32	27.64	109.00	887	14.64	16.14	141.61
838	8.59	17.68	132.52	888	6.34	13.76	113.08
839	10.79	20.29	120.16	889	2.18	35.26	100.79
840	2.92	15.49	118.36	890	7.01	28.33	156.16
841	3.07	35.28	102.88	891	10.90	24.34	122.40
842	3.19	25.13	105.76	892	0.47	33.96	119.88
843	2.34	17.62	102.95	893	3.47	30.48	113.92
844	5.18	28.95	118.36	894	1.05	21.96	128.08
845	14.37	24.03	120.64	895	8.06	26.18	133.48
846	1.73	13.97	131.56	896	6.33	25.13	128.80
847	5.01	23.44	132.04	897	1.16	20.19	120.40
848	9.23	28.82	117.76	898	1.95	24.24	113.44
849	1.45	21.04	92.92	899	1.58	28.13	156.57
850	5.46	30.10	143.17	900	5.51	25.64	125.20

MUESTRA	CHI-2 3 TUPLAS	CHI-2 4 TUPLAS	CHI-2 5 TUPLAS	MUESTRA	CHI-2 3 TUPLAS	CHI-2 4 TUPLAS	CHI-2 5 TUPLAS
901	2.68	38.49	104.08	951	5.28	23.53	136.48
902	3.03	21.56	121.20	952	1.91	24.30	112.12
903	2.60	38.09	131.77	953	5.09	19.56	126.24
904	6.87	15.41	131.80	954	2.34	19.62	132.01
905	6.01	21.13	112.84	955	5.23	22.42	116.08
906	2.43	13.11	134.65	956	8.45	31.22	121.88
907	6.62	12.90	122.56	957	5.52	26.22	117.28
908	2.09	31.72	117.28	958	9.61	13.66	116.95
909	4.68	11.51	129.88	959	4.35	25.05	95.44
910	3.27	18.87	114.28	960	6.01	24.01	126.00
911	8.28	25.67	128.52	961	5.31	26.91	103.36
912	1.58	16.68	144.13	962	19.53	27.55	127.84
913	8.24	23.80	140.20	963	3.35	25.61	120.64
914	11.37	27.09	118.80	964	8.74	33.44	140.08
915	2.78	29.24	114.28	965	6.77	18.41	82.18
916	1.96	10.91	102.42	966	2.28	23.15	106.00
917	5.76	18.35	113.80	967	5.00	20.31	115.12
918	6.25	21.19	110.78	968	4.36	18.54	137.94
919	11.79	16.10	128.68	969	3.39	21.67	114.64
920	5.10	23.72	93.40	970	8.34	19.87	116.76
921	7.52	35.72	130.44	971	12.24	22.02	97.24
922	5.35	27.08	100.48	972	7.82	17.43	110.92
923	1.68	12.84	115.96	973	3.47	26.85	107.99
924	6.84	18.23	124.56	974	5.65	25.19	133.48
925	4.67	23.48	117.48	975	10.52	15.14	103.67
926	3.17	24.22	114.96	976	1.88	32.68	147.25
927	1.15	26.87	113.11	977	4.48	21.46	119.88
928	1.33	18.41	116.28	978	7.24	23.09	95.39
929	2.59	14.70	162.04	979	4.67	24.01	119.88
930	2.99	21.54	94.79	980	1.65	27.93	113.52
931	2.53	17.20	100.00	981	9.77	18.16	100.60
932	2.66	27.01	131.65	982	7.82	17.48	84.10
933	7.35	23.49	137.89	983	7.42	35.42	125.20
934	1.17	21.96	110.03	984	6.63	28.43	105.40
935	2.65	23.96	143.56	985	6.92	19.96	125.28
936	3.63	17.54	112.87	986	7.76	28.64	125.92
937	4.41	13.61	131.20	987	4.64	28.20	112.12
938	5.89	22.23	127.72	988	4.03	36.92	117.36
939	3.81	15.85	113.40	989	10.52	24.61	124.44
940	2.93	26.72	114.76	990	7.44	26.26	126.76
941	3.42	14.20	97.00	991	4.96	20.69	121.84
942	6.14	14.32	123.28	992	2.85	13.61	113.88
943	0.52	21.17	111.00	993	5.18	14.12	130.72
944	13.81	10.72	131.89	994	20.48	22.23	123.80
945	1.41	13.63	141.25	995	8.74	25.95	107.35
946	1.65	21.58	96.35	996	3.32	19.21	139.72
947	5.02	19.65	109.72	997	5.95	26.19	104.63
948	3.04	27.47	128.80	998	9.87	28.33	148.57
949	5.76	30.60	111.52	999	0.77	13.36	116.88
950	2.39	23.78	142.84	1000	5.92	12.76	135.85

\*\*\*\*\*  
 VALORES CHI-2 PARA CADA MUESTRA DE NUMEROS ALEATORIOS DEL TEST DEL  
 POKER (TST08) UTILIZANDO 1000 MUESTRAS DE 10000  
 NUMEROS ALEATORIOS CADA UNA.

MUESTRA	CHI-2 MANOS DE 4 4 REPARTIC.	CHI-2 MANOS DE 5 6 REPARTIC.	CHI-2 MANOS DE 5 4 REPARTIC.	MUESTRA	CHI-2 MANOS DE 4 4 REPARTIC.	CHI-2 MANOS DE 5 6 REPARTIC.	CHI-2 MANOS DE 5 4 REPARTIC.
1	0.10	0.63	0.48	51	4.47	3.70	2.81
2	2.36	4.97	4.86	52	2.73	3.94	2.63
3	3.55	2.03	1.97	53	3.30	1.40	0.49
4	1.53	7.03	4.33	54	7.44	3.91	1.73
5	0.34	3.39	2.04	55	0.35	5.95	3.32
6	11.52	1.18	0.95	56	2.46	7.42	6.35

7	2.37	0.75	0.23	57	5.06	2.11	1.06
8	1.11	5.67	2.81	58	3.64	5.77	2.00
9	5.30	3.81	1.40	59	2.88	4.82	3.44
10	2.76	1.72	0.83	60	4.88	4.71	1.10
11	1.61	3.21	2.68	61	2.94	5.04	3.44
12	4.70	4.50	2.82	62	1.84	7.00	6.40
13	14.76	4.67	4.66	63	0.98	7.96	5.13
14	4.54	4.25	3.66	64	2.03	4.44	0.51
15	0.27	2.08	1.60	65	0.07	0.44	0.26
16	1.07	5.31	0.41	66	3.60	7.52	6.51
17	3.96	3.65	0.80	67	2.83	6.35	5.36
18	4.47	4.88	4.86	68	2.83	1.46	0.80
19	2.13	5.85	4.91	69	2.38	4.21	1.33
20	3.24	3.46	3.25	70	9.17	11.22	8.81
21	1.90	5.93	4.70	71	3.67	7.06	3.59
22	3.30	4.18	1.11	72	6.56	3.70	3.20
23	2.34	2.22	0.66	73	2.81	11.32	5.19
24	3.44	2.70	2.51	74	3.99	1.70	1.48
25	6.70	3.49	1.56	75	3.86	7.30	4.51
26	1.59	5.45	4.37	76	2.28	9.03	1.59
27	5.52	2.67	0.84	77	1.73	26.39	14.31
28	4.73	2.99	1.73	78	4.70	13.85	7.94
29	3.94	3.00	1.49	79	1.79	5.39	1.81
30	0.43	2.89	2.69	80	1.69	7.55	6.66
31	2.07	8.52	5.65	81	9.52	8.35	5.43
32	3.30	5.66	3.31	82	2.13	15.56	14.01
33	0.43	5.57	1.64	83	0.99	11.47	3.13
34	0.91	3.87	0.78	84	2.01	5.65	2.97
35	4.97	3.90	2.13	85	3.20	3.98	2.70
36	0.48	2.51	1.98	86	5.69	6.60	4.82
37	0.16	9.56	6.58	87	3.08	6.28	5.30
38	3.05	15.44	14.01	88	1.66	7.15	4.72
39	0.71	7.67	5.84	89	0.52	4.98	1.88
40	4.61	4.37	2.60	90	0.48	0.92	0.40
41	0.76	2.98	2.79	91	1.38	5.21	1.87
42	1.61	4.15	1.78	92	5.48	1.65	0.29
43	5.11	4.49	0.28	93	1.89	5.25	3.40
44	0.35	7.55	6.10	94	0.60	5.45	4.66
45	0.75	5.98	1.42	95	1.20	1.64	0.33
46	2.89	3.26	2.32	96	2.13	7.04	3.65
47	6.17	3.34	0.48	97	2.27	0.96	0.68
48	11.46	3.16	2.72	98	1.12	0.95	0.74
49	7.84	4.14	3.08	99	4.24	4.51	3.59
50	0.28	4.26	2.13	100	1.33	3.37	2.46

- - - - siguen muestras N° 101...801 a 1000 - - - -

MUESTRA	CHI-2 MANOS DE 4 4 REPARTIC.	CHI-2 MANOS DE 5 6 REPARTIC.	CHI-2 MANOS DE 5 4 REPARTIC.	MUESTRA	CHI-2 MANOS DE 4 4 REPARTIC.	CHI-2 MANOS DE 5 6 REPARTIC.	CHI-2 MANOS DE 5 4 REPARTIC.
801	1.10	5.19	2.75	851	4.36	2.66	2.29
802	2.20	2.47	0.71	852	3.80	3.29	0.14
803	0.48	7.97	3.06	853	0.48	3.41	2.64
804	5.75	1.58	0.42	854	17.07	9.40	5.75
805	8.22	11.43	7.92	855	0.82	9.58	1.39
806	0.35	2.27	1.47	856	2.35	1.05	0.13
807	1.82	0.83	0.82	857	2.96	0.95	0.49
808	2.28	2.18	1.46	858	1.00	2.33	0.93
809	1.01	4.91	4.10	859	2.49	4.35	4.12
810	9.35	6.09	5.65	860	3.29	1.78	1.64
811	6.34	1.72	1.70	861	4.57	2.14	1.91
812	3.22	7.18	2.88	862	3.42	0.74	0.61
813	3.97	2.84	2.67	863	3.17	2.81	2.00
814	4.29	8.00	3.99	864	9.68	6.49	2.47
815	2.18	4.55	4.32	865	2.16	11.37	3.21
816	2.12	2.21	1.15	866	5.62	3.54	1.50
817	2.56	5.14	3.68	867	10.60	7.39	6.66
818	4.39	5.81	5.19	868	4.46	5.04	3.13
819	0.33	4.59	3.44	869	8.55	5.33	4.90
820	11.57	1.27	1.05	870	0.82	4.99	4.85
821	1.05	6.24	4.81	871	3.37	4.70	2.57
822	8.07	8.70	7.04	872	2.54	2.17	2.06
823	2.72	12.81	12.10	873	3.74	2.72	0.84
824	2.24	0.64	0.51	874	2.91	5.80	2.13
825	6.58	10.00	8.79	875	2.38	9.62	4.12
826	1.98	3.18	2.22	876	3.21	1.93	0.78
827	0.55	11.68	2.54	877	9.48	7.03	5.18
828	0.18	8.07	1.31	878	0.59	3.92	1.67
829	3.98	1.88	0.32	879	4.40	5.98	0.64
830	1.15	1.64	1.49	880	0.41	4.22	1.45
831	5.14	5.06	3.08	881	1.76	11.90	2.31
832	3.51	2.35	2.35	882	0.17	4.40	3.82
833	10.03	8.73	7.38	883	1.70	2.81	0.66
834	1.32	4.31	3.32	884	1.93	13.99	1.33
835	4.65	5.98	5.97	885	1.40	2.95	2.55
836	1.68	6.65	3.63	886	0.93	5.37	3.63
837	1.48	5.86	1.81	887	2.05	14.41	13.22
838	1.38	1.28	0.57	888	8.90	4.15	3.56
839	0.83	3.73	2.90	889	3.56	13.31	4.32
840	10.08	2.42	1.84	890	1.73	6.37	2.65
841	4.49	4.28	3.21	891	2.56	6.06	4.00
842	0.08	4.23	0.35	892	0.36	2.75	2.34
843	6.69	6.08	5.39	893	2.03	9.97	4.24
844	1.14	4.26	3.48	894	2.02	8.97	2.08
845	1.68	2.68	2.04	895	0.72	8.26	4.31
846	0.65	14.64	13.05	896	3.92	2.68	1.76
847	3.82	11.39	4.50	897	3.92	5.52	4.94
848	0.37	5.73	1.53	898	0.31	1.61	0.28
849	1.57	1.03	0.88	899	2.96	12.39	6.79
850	0.52	3.45	3.15	900	1.66	6.79	1.17

MUESTRA	CHI-2 MANOS DE 4 4 REPARTIC.	CHI-2 MANOS DE 5 6 REPARTIC.	CHI-2 MANOS DE 5 4 REPARTIC.	MUESTRA	CHI-2 MANOS DE 4 4 REPARTIC.	CHI-2 MANOS DE 5 6 REPARTIC.	CHI-2 MANOS DE 5 4 REPARTIC.
901	0.93	6.50	2.68	951	1.53	6.37	5.80
902	2.64	6.74	4.48	952	2.59	4.37	1.74
903	3.90	6.76	4.86	953	4.14	4.17	3.44
904	3.61	9.48	5.54	954	1.61	5.80	3.71
905	2.57	4.03	3.99	955	7.31	2.73	0.13
906	1.06	6.55	4.69	956	1.25	4.34	3.82
907	0.66	3.75	1.19	957	6.21	1.11	0.79
908	7.53	5.53	5.11	958	1.15	8.63	8.38
909	7.31	12.69	9.37	959	3.16	1.04	0.40
910	1.77	2.52	0.74	960	4.09	1.43	0.77
911	2.41	7.26	1.21	961	0.73	7.36	4.47
912	0.83	6.78	3.14	962	1.60	3.37	3.24
913	4.01	3.19	2.44	963	5.66	2.33	1.43
914	2.46	10.23	6.79	964	4.99	1.14	1.08
915	6.87	1.66	0.38	965	3.62	2.22	1.20
916	3.51	12.07	4.42	966	3.11	18.84	8.73
917	5.91	2.29	2.08	967	2.72	6.18	6.12
918	3.53	0.89	0.36	968	1.16	7.50	1.80
919	2.27	6.46	5.67	969	7.01	0.94	0.20
920	2.39	8.40	2.12	970	2.21	0.88	0.31
921	0.81	3.96	1.89	971	10.09	6.80	3.72
922	4.07	2.74	1.92	972	1.51	5.05	3.14
923	3.68	3.05	1.75	973	4.20	1.77	0.47
924	3.97	6.19	2.86	974	4.82	4.15	3.43
925	1.32	4.11	3.10	975	7.42	5.58	3.56
926	3.03	1.08	0.17	976	0.52	3.00	2.70
927	2.56	4.03	1.80	977	3.58	3.80	3.13
928	0.34	3.75	2.79	978	1.87	3.44	2.03
929	2.48	0.73	0.08	979	10.33	10.06	7.10
930	7.97	3.41	0.93	980	3.36	4.24	3.31
931	3.59	3.18	2.87	981	2.07	4.13	3.98
932	1.45	5.15	3.37	982	2.61	1.84	1.59
933	4.75	0.69	0.28	983	3.02	4.88	3.15
934	2.90	8.13	1.93	984	0.95	4.94	0.84
935	2.82	3.07	0.15	985	2.11	3.76	0.30
936	0.67	2.68	1.33	986	1.89	3.15	1.84
937	5.50	0.93	0.83	987	3.06	4.58	1.49
938	4.34	5.79	2.46	988	1.40	1.53	1.16
939	0.67	6.55	5.66	989	1.85	2.68	1.53
940	2.79	0.97	0.40	990	3.13	4.39	3.78
941	1.62	7.76	0.45	991	1.39	4.46	0.82
942	9.35	5.93	4.03	992	2.28	11.10	4.37
943	2.39	6.78	3.45	993	6.58	10.96	2.67
944	4.51	6.65	3.70	994	6.90	9.63	5.26
945	4.92	1.67	1.28	995	3.18	2.14	0.71
946	0.48	8.23	4.74	996	6.88	2.36	1.16
947	5.08	6.02	2.60	997	4.50	7.50	5.26
948	1.11	3.10	2.37	998	2.41	4.41	0.30
949	2.34	2.29	0.61	999	8.65	4.91	4.68
950	0.44	8.30	3.49	1000	0.36	4.37	4.17

\*\*\*\*\*

VALORES CHI-2 PARA CADA MUESTRA DE NUMEROS ALEATORIOS EL TEST RUNS UP  
(TST09) UTILIZANDO 1000 MUESTRAS DE 10000 NUMEROS ALEATORIOS CADA UNA.

MUESTRA	CHI-2 X	CHI-2 FRAC(10*X)	CHI-2 FRAC(100*X)	MUESTRA	CHI-2 X	CHI-2 FRAC(10*X)	CHI-2 FRAC(100*X)
1	8.49	2.09	2.21	51	8.14	11.39	4.20
2	11.04	0.40	7.32	52	7.12	4.38	11.10
3	8.21	2.90	6.06	53	11.71	2.89	8.78
4	14.03	14.21	17.19	54	4.38	4.62	5.28
5	1.72	4.20	7.38	55	6.67	3.35	4.31
6	3.38	5.03	2.13	56	1.52	10.15	4.94
7	10.06	6.42	15.08	57	5.53	3.97	2.98
8	9.14	1.79	8.43	58	4.65	4.50	8.89
9	5.04	3.04	5.53	59	8.83	6.27	2.60
10	5.05	7.12	9.37	60	11.00	4.30	1.77
11	4.79	7.26	2.35	61	3.44	6.58	2.06
12	4.91	7.05	7.04	62	5.30	8.19	10.05
13	10.25	6.15	5.30	63	4.24	1.24	5.41
14	5.14	16.45	4.46	64	5.03	6.34	7.32
15	3.40	7.19	4.12	65	1.34	10.89	10.28
16	2.56	8.50	2.83	66	3.31	10.81	7.98
17	5.51	3.52	3.67	67	3.96	6.43	3.13
18	2.08	6.41	4.45	68	1.78	3.54	4.73
19	7.03	4.87	7.64	69	8.09	3.18	5.02
20	2.54	4.53	5.25	70	4.40	9.00	2.06
21	3.59	3.21	3.20	71	4.83	2.32	6.14
22	6.76	3.69	11.18	72	3.07	9.20	4.26
23	10.64	3.34	12.53	73	9.64	8.78	4.64
24	14.24	4.32	2.82	74	6.36	2.58	6.07
25	8.07	6.33	13.64	75	7.90	1.74	2.60
26	6.80	5.33	5.79	76	5.61	5.76	5.31
27	7.96	6.20	2.91	77	4.18	2.68	1.48
28	10.33	5.08	6.22	78	13.56	12.10	8.07
29	6.44	9.09	5.74	79	16.47	1.65	7.21
30	5.65	2.15	3.40	80	1.79	11.22	3.56
31	4.01	3.94	3.71	81	5.87	5.43	8.13
32	13.18	3.83	2.08	82	6.69	0.80	7.92
33	9.72	2.72	6.00	83	11.04	16.35	4.44
34	2.34	3.57	8.73	84	2.40	5.29	3.93
35	8.60	3.83	4.69	85	8.07	1.28	0.54
36	6.36	4.91	2.84	86	3.71	4.21	10.22
37	6.94	7.63	1.07	87	10.24	2.25	5.43
38	3.13	3.86	6.01	88	10.73	4.46	7.90

39	6.64	4.25	7.66	89	6.35	5.02	8.52
40	7.68	5.68	2.18	90	2.71	2.56	8.20
41	3.97	3.93	5.52	91	6.30	3.37	5.36
42	7.98	3.06	6.87	92	14.36	1.57	4.66
43	2.22	2.76	5.28	93	5.01	12.54	13.61
44	1.06	3.47	7.25	94	14.92	11.48	2.59
45	2.51	3.22	3.02	95	1.61	3.81	9.18
46	7.36	14.36	4.91	96	6.56	3.82	9.78
47	4.19	4.87	4.29	97	7.54	8.47	8.31
48	1.74	5.53	13.41	98	6.38	9.71	8.52
49	7.56	5.14	15.04	99	40.37	5.52	4.95
50	6.68	3.19	3.33	100	5.10	4.26	6.47

- - - - siguen muestras N° 101...801 a 1000 - - - -

MUESTRA	CHI-2 X	CHI-2 FRAC(10*X)	CHI-2 FRAC(100*X)	MUESTRA	CHI-2 X	CHI-2 FRAC(10*X)	CHI-2 FRAC(100*X)
801	5.46	4.88	6.91	851	3.13	6.50	9.49
802	5.71	6.00	9.95	852	7.68	3.98	14.11
803	5.86	1.54	3.75	853	2.97	3.38	11.78
804	3.36	4.57	11.21	854	3.24	2.02	11.11
805	9.07	4.91	12.37	855	8.93	2.76	15.98
806	5.87	10.21	5.96	856	8.04	12.35	1.71
807	9.19	3.82	3.50	857	4.35	15.72	8.29
808	3.74	4.34	5.42	858	4.75	2.22	1.87
809	2.05	2.69	6.27	859	6.95	2.98	2.08
810	6.16	6.78	5.93	860	6.87	3.75	5.73
811	2.06	4.24	1.86	861	6.86	6.28	7.89
812	4.41	2.34	4.69	862	7.39	1.34	5.80
813	4.72	13.44	4.18	863	7.63	1.45	9.29
814	2.99	4.99	3.20	864	4.67	7.57	4.95
815	4.97	11.84	3.76	865	3.58	3.58	5.58
816	5.14	9.81	2.76	866	9.26	3.11	12.26
817	5.23	2.46	8.27	867	7.47	1.84	6.56
818	6.15	7.09	2.61	868	9.77	7.09	5.27
819	6.64	8.46	5.11	869	4.04	12.86	1.96
820	2.63	3.87	9.76	870	2.78	7.49	6.32
821	4.74	0.78	2.95	871	3.23	5.82	12.36
822	10.12	9.41	18.28	872	4.36	5.28	7.23
823	13.06	18.34	6.44	873	6.23	8.64	9.69
824	4.70	4.72	5.14	874	4.10	7.96	7.57
825	2.41	9.30	7.96	875	3.38	7.67	3.65
826	11.87	11.49	6.06	876	8.24	4.36	2.94
827	6.28	6.97	11.02	877	2.37	6.41	1.49
828	5.44	1.49	7.28	878	7.67	7.55	1.46
829	13.27	7.27	3.86	879	6.05	4.82	3.22
830	9.25	4.81	2.93	880	6.34	2.79	5.81
831	21.05	7.96	2.63	881	8.93	5.36	3.49
832	3.22	4.25	4.28	882	7.57	7.57	2.49
833	6.54	10.13	4.18	883	6.17	4.50	7.63
834	1.77	8.31	9.88	884	1.68	11.66	2.89
835	9.65	3.16	7.08	885	7.09	5.35	5.52
836	8.24	4.83	5.22	886	7.91	3.50	4.43
837	1.78	2.81	2.57	887	5.38	4.02	7.93
838	5.27	4.79	2.71	888	6.37	6.28	7.15
839	2.21	3.41	1.18	889	6.62	4.81	3.33
840	8.81	7.38	2.14	890	12.75	2.71	4.04
841	8.61	6.10	4.26	891	8.05	9.36	11.83
842	12.48	1.66	7.94	892	4.52	5.02	4.63
843	1.65	8.25	2.20	893	6.54	0.08	5.41
844	10.18	4.55	5.62	894	4.68	4.12	4.07
845	3.45	1.23	7.06	895	3.92	6.03	3.20
846	5.44	4.32	9.52	896	7.47	9.55	2.53
847	2.68	4.11	6.95	897	1.13	1.59	7.56
848	5.00	2.35	8.59	898	5.25	2.84	3.15
849	3.60	2.50	12.53	899	1.00	5.55	5.20
850	10.33	9.69	5.80	900	10.44	10.87	5.91
MUESTRA	CHI-2 X	CHI-2 FRAC(10*X)	CHI-2 FRAC(100*X)	MUESTRA	CHI-2 X	CHI-2 FRAC(10*X)	CHI-2 FRAC(100*X)
901	1.27	9.49	1.24	951	7.49	7.63	16.25
902	14.11	2.71	2.30	952	3.79	5.29	9.22
903	8.39	8.17	2.99	953	1.86	3.27	13.36
904	3.25	2.83	1.33	954	4.14	8.03	4.58
905	1.06	2.88	5.02	955	7.20	15.20	1.80
906	3.90	6.37	3.63	956	5.98	4.06	5.89
907	3.71	4.72	7.75	957	7.79	2.06	5.62
908	8.21	5.09	7.35	958	4.89	1.89	10.14
909	4.74	8.93	10.23	959	5.19	1.05	1.83
910	7.63	9.98	14.76	960	8.84	1.08	4.97
911	5.83	8.94	10.01	961	6.78	3.94	0.96
912	3.06	7.46	4.89	962	8.12	7.94	3.71
913	7.15	4.43	3.82	963	6.04	4.59	20.22
914	20.59	3.05	1.75	964	7.59	6.16	4.73
915	8.67	5.22	7.39	965	4.83	5.28	11.42
916	4.60	9.68	11.95	966	6.72	9.36	3.39
917	9.41	4.76	9.18	967	3.70	2.88	3.18
918	7.06	7.69	6.16	968	7.54	7.03	8.69
919	4.57	4.14	5.70	969	10.26	2.56	3.06
920	7.02	1.65	2.99	970	7.90	3.78	7.67
921	6.47	5.14	1.62	971	5.22	2.82	3.87
922	1.83	8.37	12.50	972	7.15	5.83	3.87
923	4.17	6.83	5.30	973	5.71	5.86	6.09
924	2.62	15.12	5.87	974	6.80	3.41	3.94
925	3.93	3.65	7.67	975	10.80	6.84	2.24
926	10.65	12.08	4.13	976	3.12	6.72	3.76
927	4.42	8.80	9.64	977	6.84	9.29	3.31
928	6.23	8.24	4.67	978	1.85	3.58	14.68
929	7.67	4.61	1.97	979	11.63	4.65	9.29
930	4.52	8.23	7.63	980	2.33	5.83	7.01
931	5.95	4.07	5.89	981	1.83	3.95	13.35

932	1.35	4.97	9.60	982	4.78	2.57	2.88
933	14.78	5.10	7.85	983	7.02	4.99	3.57
934	5.60	3.57	8.38	984	5.96	4.20	3.63
935	6.25	9.64	7.50	985	6.60	18.14	6.17
936	1.98	8.92	3.76	986	6.12	5.81	7.95
937	4.22	6.65	6.59	987	3.04	4.18	4.99
938	9.80	2.95	3.78	988	9.68	8.06	17.12
939	7.50	3.88	3.43	989	5.75	4.21	0.96
940	3.62	8.13	3.07	990	6.76	9.63	1.63
941	3.65	10.36	6.86	991	3.50	3.75	4.80
942	3.54	13.19	2.53	992	6.99	0.92	7.86
943	3.12	4.21	9.75	993	5.40	7.16	7.98
944	12.68	6.76	9.93	994	16.26	7.20	8.62
945	6.43	2.10	5.46	995	11.32	10.26	2.53
946	4.68	2.28	7.23	996	4.82	2.83	4.58
947	1.34	8.11	7.11	997	7.83	9.78	4.98
948	5.72	1.61	9.89	998	5.73	3.63	4.83
949	3.31	13.30	6.99	999	2.26	4.52	7.41
950	5.50	1.85	5.15	1000	3.97	8.39	2.54

\*\*\*\*\*  
 VALORES CHI-2 PARA CADA MUESTRA DE NUMEROS ALEATORIOS DEL TEST DE PARES  
 SERIALES (TST10) USANDO 1000 MUESTRAS DE 10000 NUM. ALEATORIOS CADA UNA.

MUESTRA	CHI-2 ENTEROS 1-3	CHI-2 ENTEROS 1-10	CHI-2 ENTEROS 1-20	MUESTRA	CHI-2 ENTEROS 1-3	CHI-2 ENTEROS 1-10	CHI-2 ENTEROS 1-20
1	1.99	129.52	432.16	51	7.13	88.88	387.68
2	8.68	102.64	427.84	52	5.61	108.40	466.40
3	5.12	90.04	419.84	53	12.85	57.24	362.40
4	8.53	115.04	404.32	54	9.63	125.32	421.44
5	2.30	96.44	400.80	55	4.23	110.84	442.72
6	7.80	105.92	396.80	56	9.35	101.04	446.40
7	6.53	104.08	365.44	57	1.99	79.80	394.88
8	15.81	100.80	406.08	58	5.29	90.84	384.16
9	4.78	112.08	422.08	59	4.34	99.48	399.20
10	11.92	95.08	418.08	60	17.62	114.48	424.96
11	7.69	82.36	420.80	61	9.31	113.08	420.96
12	7.33	89.84	379.36	62	7.86	84.24	351.20
13	3.89	93.88	378.88	63	9.66	98.72	416.00
14	3.63	93.00	383.20	64	3.50	91.72	381.12
15	14.84	112.28	379.36	65	10.95	83.00	375.52
16	10.45	108.28	404.64	66	4.54	87.36	408.48
17	7.17	99.40	416.64	67	3.15	81.84	406.08
18	4.56	78.16	374.88	68	5.99	92.32	446.24
19	7.00	87.84	372.00	69	9.60	112.64	402.72
20	4.48	117.24	364.16	70	7.28	110.04	426.40
21	3.84	92.92	417.44	71	8.38	85.04	353.76
22	7.54	108.44	421.28	72	5.35	101.76	368.00
23	2.66	96.80	430.72	73	9.55	67.56	436.64
24	5.79	114.68	393.60	74	7.71	107.24	456.48
25	7.70	94.40	434.08	75	13.37	113.60	393.44
26	13.57	91.00	422.40	76	7.72	102.76	348.00
27	9.15	106.08	418.88	77	21.61	126.60	484.00
28	3.80	80.84	372.16	78	3.13	89.60	391.84
29	3.85	92.72	410.88	79	5.99	96.96	427.84
30	3.49	79.72	395.20	80	6.90	82.80	385.60
31	6.52	104.32	417.92	81	6.05	89.48	432.32
32	4.36	82.20	363.04	82	11.11	115.80	411.04
33	5.28	96.40	364.96	83	7.49	132.96	455.52
34	8.31	93.28	388.96	84	14.81	96.76	380.80
35	10.36	95.44	432.16	85	4.66	95.76	398.56
36	6.50	78.32	373.60	86	4.93	91.68	409.92
37	11.77	101.16	399.84	87	14.61	109.36	443.20
38	6.08	78.80	387.52	88	5.32	81.08	399.04
39	9.84	116.48	389.12	89	5.73	104.84	428.48
40	3.56	92.24	366.72	90	6.54	96.16	365.76
41	10.03	91.12	405.60	91	8.04	102.00	399.36
42	9.55	91.40	390.40	92	8.43	124.28	422.88
43	3.98	99.04	396.16	93	8.05	102.04	379.20
44	4.69	67.84	403.04	94	11.79	84.92	431.20
45	10.15	133.88	422.56	95	7.63	81.76	380.00
46	7.29	84.96	388.80	96	12.05	94.52	378.56
47	10.04	94.04	380.48	97	5.72	89.68	384.00
48	6.51	95.20	433.28	98	6.65	71.44	367.36
49	8.93	107.96	392.64	99	5.48	79.28	408.16
50	8.85	123.80	451.36	100	9.40	93.56	399.68

- - - - siguen muestras N° 101...801 a 1000 - - - -

MUESTRA	CHI-2 ENTEROS 1-3	CHI-2 ENTEROS 1-10	CHI-2 ENTEROS 1-20	MUESTRA	CHI-2 ENTEROS 1-3	CHI-2 ENTEROS 1-10	CHI-2 ENTEROS 1-20
801	9.48	93.36	367.04	851	4.00	121.12	411.84
802	10.81	114.44	399.84	852	4.56	87.72	377.44
803	14.95	99.32	413.44	853	6.74	85.68	365.12
804	4.19	115.88	372.64	854	5.09	125.20	383.36
805	8.50	95.76	397.92	855	10.54	115.88	384.80
806	6.47	96.04	436.48	856	6.63	66.84	368.48
807	6.16	99.36	348.64	857	16.21	85.52	378.24
808	8.81	102.24	384.80	858	15.89	92.52	368.16
809	3.69	92.28	362.08	859	6.12	108.32	414.72
810	6.35	98.64	362.08	860	5.63	120.80	421.60
811	7.09	86.44	371.20	861	11.20	102.56	426.24
812	4.16	111.24	410.08	862	3.48	72.04	399.36
813	5.80	88.44	408.32	863	13.84	110.16	422.72
814	5.55	102.32	379.20	864	5.41	71.00	389.12
815	6.52	108.84	427.84	865	4.26	105.00	454.56
816	9.76	104.44	405.60	866	10.35	125.04	448.00

817	8.83	87.20	405.28	867	10.86	101.84	396.16
818	4.45	90.20	395.04	868	27.35	111.12	396.00
819	9.77	90.20	450.40	869	5.86	114.28	427.04
820	6.43	105.76	427.20	870	8.03	95.76	393.28
821	5.24	86.36	414.56	871	6.11	95.88	404.48
822	9.94	102.80	429.60	872	4.70	79.72	398.88
823	7.27	111.48	444.48	873	25.48	124.04	426.24
824	14.54	102.48	392.48	874	3.79	114.00	440.48
825	7.74	97.52	400.16	875	12.60	74.40	388.80
826	8.23	99.04	350.40	876	3.13	105.16	391.20
827	6.30	75.20	414.56	877	6.10	98.40	413.44
828	8.00	112.40	407.84	878	2.56	100.76	401.28
829	13.06	90.40	375.68	879	3.46	89.96	387.84
830	11.64	91.44	368.16	880	4.15	108.32	414.88
831	9.12	73.28	388.80	881	5.70	92.68	393.28
832	8.44	99.80	389.12	882	6.15	97.92	420.16
833	10.26	136.28	424.96	883	13.18	114.96	405.28
834	6.59	76.92	385.76	884	7.39	95.52	399.36
835	7.74	93.24	415.84	885	21.04	110.32	397.44
836	9.82	86.00	415.52	886	7.78	80.80	339.84
837	10.67	123.64	371.20	887	4.87	85.24	397.12
838	6.53	110.48	409.12	888	3.30	87.44	342.24
839	5.91	113.56	421.12	889	22.93	111.04	421.12
840	4.56	88.08	383.52	890	6.77	84.76	349.60
841	14.36	125.92	450.08	891	12.04	96.00	413.12
842	5.76	102.08	392.64	892	3.47	90.24	406.56
843	11.38	95.24	381.44	893	4.73	82.88	377.76
844	16.02	110.80	445.44	894	6.69	101.36	414.40
845	9.97	95.88	388.64	895	9.32	110.16	395.84
846	5.32	90.60	374.72	896	11.74	97.72	386.72
847	8.23	116.64	415.04	897	8.98	80.36	350.40
848	12.41	113.00	474.88	898	13.55	94.84	390.24
849	7.29	105.76	388.00	899	8.20	97.96	388.16
850	3.67	94.84	437.28	900	5.94	115.60	407.36

MUESTRA	CHI-2 ENTEROS 1-3	CHI-2 ENTEROS 1-10	CHI-2 ENTEROS 1-20	MUESTRA	CHI-2 ENTEROS 1-3	CHI-2 ENTEROS 1-10	CHI-2 ENTEROS 1-20
901	14.65	110.20	408.00	951	6.10	113.40	373.76
902	3.83	84.84	366.72	952	9.59	118.36	414.40
903	14.03	110.12	418.40	953	5.87	95.32	402.08
904	4.26	92.60	412.32	954	5.49	98.56	339.20
905	7.47	96.80	407.20	955	16.00	97.00	391.36
906	13.75	92.72	361.92	956	9.08	108.20	406.56
907	7.91	77.60	394.08	957	8.38	84.20	369.76
908	9.91	95.68	422.88	958	7.44	90.24	364.00
909	10.95	93.60	413.28	959	4.41	129.76	409.12
910	5.14	82.20	405.76	960	4.48	117.52	410.24
911	4.81	100.52	450.88	961	8.05	105.32	401.28
912	6.88	91.44	402.08	962	9.43	102.76	435.52
913	9.67	101.24	388.64	963	5.54	101.76	391.84
914	2.91	123.64	382.88	964	7.84	106.24	446.72
915	5.99	78.76	370.72	965	20.07	97.96	418.40
916	3.25	113.16	425.12	966	3.37	75.44	368.48
917	8.97	115.20	422.72	967	5.04	88.44	349.44
918	9.14	110.40	412.64	968	8.81	89.88	413.44
919	13.53	107.16	436.00	969	9.07	85.92	467.04
920	13.05	98.32	371.84	970	13.06	108.04	402.24
921	7.33	83.00	357.44	971	9.16	110.68	438.24
922	3.22	75.04	383.20	972	4.70	100.36	438.08
923	10.47	96.92	439.20	973	1.76	123.64	427.68
924	5.27	92.20	368.96	974	13.53	80.68	405.12
925	8.95	103.16	411.84	975	4.04	96.72	401.28
926	5.80	72.24	395.20	976	3.79	87.64	367.36
927	8.00	104.88	437.44	977	3.91	80.92	367.20
928	2.63	108.72	400.32	978	7.65	94.48	393.60
929	7.71	105.12	377.92	979	7.49	114.72	435.20
930	7.15	123.88	417.12	980	4.96	99.20	422.40
931	10.28	114.60	392.96	981	7.07	99.24	381.44
932	6.39	107.84	440.96	982	8.95	113.00	381.28
933	13.20	106.80	377.92	983	16.37	98.96	373.28
934	5.70	81.28	363.52	984	8.79	114.68	392.64
935	10.67	101.64	390.72	985	9.02	101.52	378.24
936	6.31	101.96	399.84	986	6.44	103.28	371.20
937	7.63	77.88	378.72	987	7.70	110.76	411.52
938	8.49	88.00	354.24	988	3.15	73.60	358.56
939	6.28	129.16	416.64	989	16.14	95.88	393.60
940	8.08	85.36	381.28	990	13.18	99.08	388.64
941	8.69	72.60	343.84	991	4.19	81.12	359.36
942	3.76	86.12	408.00	992	9.61	97.16	399.36
943	4.61	115.92	423.68	993	9.33	110.56	348.96
944	5.64	84.72	356.48	994	8.86	110.92	381.44
945	1.20	98.88	386.88	995	5.44	86.28	384.48
946	2.39	70.44	374.88	996	6.80	85.80	398.40
947	21.67	111.84	405.76	997	3.02	119.56	446.88
948	11.09	114.96	399.52	998	12.56	97.92	385.92
949	16.90	106.88	381.92	999	5.65	115.16	404.00
950	3.58	92.60	415.20	1000	5.33	111.72	431.04

\*\*\*\*\*  
 VALORES KOLMOGOROV-SMIRNOV DE CADA MUESTRA DE NUMEROS ALEATORIOS  
 TEST DE BONDAD DE AJUSTE K-S (TST06) USANDO 1000 MUESTRAS CON  
 10 SUBMUESTRAS DENTRO DE CADA MUESTRA DE 10000 NUMEROS ALEATORIOS.

MUESTRA	SUBMUESTRA	K+	K-	K	MUESTRA	SUBMUESTRA	K+	K-	K
1	1	31.016	29.987	31.016	6	1	0.628	0.677	0.677
1	2	29.872	29.591	29.872	6	2	0.256	1.098	1.098
1	3	31.086	30.694	31.086	6	3	0.541	0.318	0.541
1	4	29.919	30.027	30.027	6	4	1.145	0.134	1.145
1	5	29.602	30.823	30.823	6	5	0.528	0.579	0.579

1	6	30.258	30.916	30.916	6	6	0.696	0.276	0.696
1	7	29.629	29.127	29.629	6	7	0.832	0.435	0.832
1	8	31.253	30.718	31.253	6	8	0.465	0.697	0.697
1	9	30.333	29.915	30.333	6	9	0.659	0.460	0.659
1	10	30.216	30.439	30.439	6	10	0.051	1.392	1.392
2	1	0.862	0.195	0.862	7	1	1.088	0.246	1.088
2	2	0.347	0.840	0.840	7	2	0.493	0.431	0.493
2	3	0.587	0.638	0.638	7	3	0.335	0.611	0.611
2	4	0.906	0.523	0.906	7	4	0.189	1.192	1.192
2	5	0.653	0.865	0.865	7	5	0.559	0.842	0.842
2	6	0.946	0.274	0.946	7	6	0.929	0.381	0.929
2	7	0.695	0.645	0.695	7	7	0.569	0.418	0.569
2	8	0.534	0.367	0.534	7	8	0.933	0.428	0.933
2	9	0.964	0.395	0.964	7	9	0.755	0.393	0.755
2	10	0.222	1.026	1.026	7	10	0.372	0.683	0.683
3	1	0.329	0.828	0.828	8	1	0.710	0.291	0.710
3	2	0.878	0.657	0.878	8	2	0.552	0.593	0.593
3	3	0.672	0.586	0.672	8	3	0.288	1.014	1.014
3	4	0.595	0.341	0.595	8	4	0.984	0.503	0.984
3	5	0.354	1.318	1.318	8	5	0.304	0.932	0.932
3	6	0.550	0.991	0.991	8	6	0.534	1.226	1.226
3	7	1.694	0.183	1.694	8	7	0.357	0.325	0.357
3	8	1.520	0.095	1.520	8	8	0.500	0.941	0.941
3	9	0.563	0.608	0.608	8	9	0.257	0.926	0.926
3	10	0.047	0.877	0.877	8	10	0.639	0.126	0.639
4	1	1.187	0.326	1.187	9	1	0.719	0.711	0.719
- - - - - siguen muestras N° 10 ... 991 a 1000 - - - - -									
991	1	0.487	0.785	0.785	996	1	0.856	0.570	0.856
991	2	0.372	0.606	0.606	996	2	0.140	0.794	0.794
991	3	0.568	0.334	0.568	996	3	0.876	0.281	0.876
991	4	0.235	0.926	0.926	996	4	0.917	0.483	0.917
991	5	0.680	1.289	1.289	996	5	0.104	1.048	1.048
991	6	0.327	0.811	0.811	996	6	1.921	0.066	1.921
991	7	0.630	0.467	0.630	996	7	0.957	0.372	0.957
991	8	0.202	1.075	1.075	996	8	0.365	1.206	1.206
991	9	0.265	0.646	0.646	996	9	0.691	0.465	0.691
991	10	0.372	0.551	0.551	996	10	0.414	1.033	1.033
992	1	1.261	0.111	1.261	997	1	0.960	0.304	0.960
992	2	0.448	0.825	0.825	997	2	0.593	0.627	0.627
992	3	0.654	0.750	0.750	997	3	0.343	0.951	0.951
992	4	0.379	0.499	0.499	997	4	0.246	0.659	0.659
992	5	1.327	0.704	1.327	997	5	1.070	0.423	1.070
992	6	1.241	0.199	1.241	997	6	0.429	0.583	0.583
992	7	0.666	0.501	0.666	997	7	0.290	1.120	1.120
992	8	0.211	0.787	0.787	997	8	0.851	0.844	0.851
992	9	0.400	0.678	0.678	997	9	0.586	0.576	0.586
992	10	0.024	1.316	1.316	997	10	0.630	0.718	0.718
993	1	0.948	0.608	0.948	998	1	0.997	0.775	0.997
993	2	0.859	0.554	0.859	998	2	0.112	0.962	0.962
993	3	1.313	0.154	1.313	998	3	0.320	1.063	1.063
993	4	1.207	0.146	1.207	998	4	0.761	0.450	0.761
993	5	0.694	0.572	0.694	998	5	0.832	0.387	0.832
993	6	1.301	0.170	1.301	998	6	0.752	0.713	0.752
993	7	0.779	0.661	0.779	998	7	0.449	0.518	0.518
993	8	0.588	0.511	0.588	998	8	0.324	0.540	0.540
993	9	0.670	0.346	0.670	998	9	0.469	0.575	0.575
993	10	0.070	0.745	0.745	998	10	0.105	1.268	1.268
994	1	0.961	0.175	0.961	999	1	0.945	0.339	0.945
994	2	0.448	0.459	0.459	999	2	0.651	0.409	0.651
994	3	0.934	0.428	0.934	999	3	0.938	0.363	0.938
994	4	0.786	0.535	0.786	999	4	0.435	0.744	0.744
994	5	0.568	0.470	0.568	999	5	0.361	0.773	0.773
994	6	0.912	0.410	0.912	999	6	0.352	0.776	0.776
994	7	0.826	0.196	0.826	999	7	0.130	1.723	1.723
994	8	0.529	0.497	0.529	999	8	0.526	0.503	0.526
994	9	0.543	0.941	0.941	999	9	0.669	0.495	0.669
994	10	0.464	0.410	0.464	999	10	0.705	0.731	0.731
995	1	1.386	0.053	1.386	1000	1	0.273	0.819	0.819
995	2	0.968	0.322	0.968	1000	2	0.471	1.450	1.450
995	3	1.081	0.165	1.081	1000	3	0.771	0.693	0.771
995	4	0.634	1.039	1.039	1000	4	0.273	0.732	0.732
995	5	0.571	0.893	0.893	1000	5	0.887	0.514	0.887
995	6	1.002	0.518	1.002	1000	6	1.534	0.167	1.534
995	7	0.387	1.431	1.431	1000	7	0.480	0.487	0.487
995	8	0.700	0.625	0.700	1000	8	0.777	1.031	1.031
995	9	0.353	0.644	0.644	1000	9	0.754	0.397	0.754
995	10	0.322	1.056	1.056	1000	10	0.813	0.210	0.813

\*\*\*\*\*

VALORES KOLMOGOROV-SMIRNOV DE CADA MUESTRA DE NUMEROS ALEATORIOS  
DEL TEST K-S MAXIMO-DE-T (TST11) USANDO 1000 MUESTRAS CON 10  
SUBMUESTRAS DENTRO DE CADA MUESTRA DE 10000 NUMEROS ALEATORIOS.

MUESTRA	SUBMUESTRA	K+	K-	K	MUESTRA	SUBMUESTRA	K+	K-	K
1	1	21.761	21.165	21.761	6	1	0.577	0.496	0.577
1	2	20.286	20.083	20.286	6	2	0.378	0.933	0.933
1	3	21.406	21.323	21.406	6	3	0.603	0.549	0.603
1	4	21.598	20.741	21.598	6	4	0.831	0.302	0.831
1	5	20.385	21.331	21.331	6	5	0.756	0.369	0.756
1	6	21.354	21.589	21.589	6	6	0.644	0.494	0.644
1	7	20.572	20.342	20.572	6	7	0.673	0.132	0.673
1	8	21.783	21.185	21.783	6	8	0.558	0.822	0.822
1	9	20.448	20.420	20.448	6	9	1.086	0.201	1.086
1	10	20.927	21.055	21.055	6	10	0.245	0.906	0.906
2	1	0.810	0.313	0.810	7	1	0.913	0.088	0.913
2	2	0.471	0.670	0.670	7	2	0.816	0.363	0.816
2	3	0.784	0.704	0.784	7	3	0.332	0.543	0.543
2	4	1.060	0.326	1.060	7	4	0.174	1.062	1.062



2	5	0.151	0.996	0.996	7	5	0.291	0.886	0.886
2	6	0.743	0.397	0.743	7	6	1.106	0.546	1.106
2	7	0.694	0.353	0.694	7	7	0.605	0.318	0.605
2	8	0.653	0.302	0.653	7	8	1.004	0.460	1.004
2	9	0.864	0.548	0.864	7	9	1.094	0.430	1.094
2	10	0.221	1.373	1.373	7	10	0.529	0.199	0.529
3	1	0.459	0.991	0.991	8	1	0.675	0.479	0.675
3	2	0.721	0.429	0.721	8	2	0.563	0.448	0.563
3	3	0.370	1.012	1.012	8	3	0.402	0.947	0.947
3	4	0.463	0.515	0.515	8	4	0.822	0.731	0.822
3	5	0.307	1.130	1.130	8	5	0.372	0.865	0.865
3	6	0.151	0.991	0.991	8	6	0.499	0.865	0.865
3	7	1.207	0.348	1.207	8	7	0.546	0.447	0.546
3	8	1.268	0.110	1.268	8	8	0.349	0.696	0.696
3	9	0.718	0.637	0.718	8	9	0.386	0.468	0.468
3	10	0.113	1.054	1.054	8	10	1.012	0.093	1.012
4	1	1.230	0.164	1.230	9	1	0.631	0.608	0.631
4	2	0.607	0.450	0.607	9	2	0.972	0.400	0.972
4	3	0.824	0.903	0.903	9	3	0.729	0.449	0.729
4	4	0.210	1.670	1.670	9	4	1.333	0.232	1.333
4	5	0.606	0.491	0.606	9	5	0.800	0.448	0.800
4	6	0.653	0.361	0.653	9	6	0.394	0.797	0.797
4	7	0.626	0.175	0.626	9	7	0.798	0.564	0.798
4	8	0.259	1.030	1.030	9	8	0.626	0.862	0.862
4	9	0.536	0.455	0.536	9	9	0.977	0.346	0.977
4	10	1.148	0.148	1.148	9	10	0.664	1.034	1.034
5	1	0.634	0.689	0.689	10	1	0.022	1.576	1.576
5	2	0.100	0.933	0.933	10	2	1.185	0.179	1.185
5	3	0.612	0.389	0.612	10	3	1.331	0.025	1.331
5	4	0.329	0.877	0.877	10	4	0.600	0.413	0.600
5	5	1.186	0.303	1.186	10	5	0.758	0.609	0.758
5	6	0.934	0.351	0.934	10	6	1.108	0.134	1.108
5	7	0.932	0.198	0.932	10	7	0.271	1.708	1.708
5	8	0.722	0.416	0.722	10	8	0.472	0.429	0.472

- - - - siguen muestras N° 11 ... 981 a 1000 - - - -

MUESTRA	SUBMUESTRA	K+	K-	K	MUESTRA	SUBMUESTRA	K+	K-	K
981	1	0.994	0.339	0.994	986	1	1.020	0.110	1.020
981	2	1.220	0.519	1.220	986	2	0.059	0.944	0.944
981	3	0.557	0.696	0.696	986	3	1.133	0.373	1.133
981	4	0.635	0.632	0.635	986	4	0.914	0.338	0.914
981	5	0.330	1.146	1.146	986	5	0.661	0.216	0.661
981	6	0.915	0.575	0.915	986	6	0.874	0.342	0.874
981	7	0.434	0.643	0.643	986	7	0.483	0.841	0.841
981	8	0.687	0.842	0.842	986	8	0.760	0.526	0.760
981	9	0.500	0.560	0.560	986	9	0.423	0.488	0.488
981	10	0.554	0.247	0.554	986	10	0.906	0.487	0.906
982	1	0.783	0.476	0.783	987	1	1.153	0.190	1.153
982	2	0.304	0.543	0.543	987	2	0.109	0.679	0.679
982	3	0.296	0.729	0.729	987	3	0.693	0.597	0.693
982	4	0.592	0.434	0.592	987	4	1.167	0.305	1.167
982	5	0.614	0.332	0.614	987	5	0.143	1.004	1.004
982	6	0.908	0.224	0.908	987	6	0.651	0.329	0.651
982	7	0.628	0.451	0.628	987	7	0.466	0.719	0.719
982	8	0.774	0.585	0.774	987	8	1.045	0.315	1.045
982	9	0.364	0.730	0.730	987	9	0.174	1.014	1.014
982	10	1.296	0.393	1.296	987	10	0.249	0.655	0.655
983	1	0.837	0.239	0.837	988	1	0.141	1.268	1.268
983	2	0.392	0.503	0.503	988	2	0.252	1.428	1.428
983	3	0.555	0.508	0.555	988	3	0.818	0.638	0.818
983	4	0.731	0.646	0.731	988	4	0.486	0.568	0.568
983	5	0.780	0.334	0.780	988	5	0.481	0.538	0.538
983	6	0.754	0.183	0.754	988	6	1.155	0.175	1.155
983	7	0.202	0.660	0.660	988	7	0.657	0.806	0.806
983	8	0.424	0.637	0.637	988	8	0.565	0.779	0.779
983	9	0.319	1.273	1.273	988	9	0.807	0.665	0.807
983	10	0.492	0.919	0.919	988	10	0.368	0.406	0.406
984	1	0.412	0.830	0.830	989	1	1.016	0.034	1.016
984	2	0.136	1.193	1.193	989	2	0.345	0.797	0.797
984	3	0.274	0.911	0.911	989	3	0.705	0.454	0.705
984	4	0.535	0.713	0.713	989	4	0.130	1.187	1.187
984	5	0.632	0.587	0.632	989	5	1.008	0.508	1.008
984	6	0.910	0.134	0.910	989	6	0.849	0.227	0.849
984	7	0.935	0.592	0.935	989	7	0.862	0.690	0.862
984	8	0.403	1.171	1.171	989	8	0.487	1.193	1.193
984	9	0.322	1.350	1.350	989	9	0.753	0.292	0.753
984	10	0.900	0.219	0.900	989	10	0.248	0.837	0.837
985	1	0.325	1.139	1.139	990	1	0.612	0.558	0.612
985	2	0.540	0.484	0.540	990	2	1.058	0.453	1.058
985	3	0.437	0.437	0.437	990	3	1.624	0.208	1.624
985	4	0.473	1.310	1.310	990	4	0.208	0.885	0.885
985	5	0.474	0.690	0.690	990	5	1.097	0.161	1.097
985	6	0.877	0.243	0.877	990	6	0.836	0.148	0.836
985	7	0.732	0.514	0.732	990	7	1.476	0.126	1.476
985	8	0.672	0.278	0.672	990	8	0.961	0.372	0.961
985	9	0.330	0.737	0.737	990	9	0.873	0.322	0.873
985	10	0.542	1.056	1.056	990	10	0.487	0.489	0.489

  

MUESTRA	SUBMUESTRA	K+	K-	K	MUESTRA	SUBMUESTRA	K+	K-	K
991	1	0.567	0.795	0.795	996	1	0.820	0.569	0.820
991	2	0.289	0.791	0.791	996	2	0.164	0.794	0.794
991	3	0.627	0.235	0.627	996	3	0.951	0.265	0.951
991	4	0.634	0.639	0.639	996	4	0.892	0.545	0.892
991	5	0.550	1.114	1.114	996	5	0.126	0.772	0.772
991	6	0.450	0.747	0.747	996	6	1.732	0.155	1.732
991	7	0.864	0.422	0.864	996	7	0.577	0.466	0.577
991	8	0.200	1.115	1.115	996	8	0.509	1.105	1.105
991	9	0.396	0.664	0.664	996	9	0.424	0.365	0.424
991	10	0.289	0.675	0.675	996	10	0.658	0.920	0.920

992	1	1.111	0.076	1.111	997	1	0.714	0.377	0.714
992	2	0.387	0.672	0.672	997	2	0.510	0.870	0.870
992	3	0.408	0.684	0.684	997	3	0.387	0.495	0.495
992	4	0.747	0.282	0.747	997	4	0.292	0.788	0.788
992	5	0.811	1.194	1.194	997	5	1.398	0.275	1.398
992	6	0.998	0.286	0.998	997	6	0.397	0.681	0.681
992	7	0.735	0.662	0.735	997	7	0.430	1.091	1.091
992	8	0.501	0.530	0.530	997	8	1.106	1.054	1.106
992	9	0.120	0.909	0.909	997	9	0.730	0.640	0.730
992	10	0.080	1.550	1.550	997	10	0.711	0.970	0.970
993	1	0.668	0.328	0.668	998	1	1.158	0.226	1.158
993	2	0.493	0.465	0.493	998	2	0.153	0.634	0.634
993	3	1.033	0.149	1.033	998	3	0.381	1.059	1.059
993	4	1.135	0.105	1.135	998	4	0.354	0.669	0.669
993	5	0.763	0.490	0.763	998	5	0.591	0.368	0.591
993	6	1.558	0.241	1.558	998	6	0.892	0.840	0.892
993	7	0.721	0.662	0.721	998	7	0.393	0.504	0.504
993	8	0.811	0.586	0.811	998	8	0.312	0.701	0.701
993	9	0.805	0.497	0.805	998	9	0.561	0.537	0.561
993	10	0.096	0.973	0.973	998	10	0.055	1.380	1.380
994	1	0.787	0.290	0.787	999	1	1.325	0.502	1.325
994	2	0.724	0.729	0.729	999	2	0.505	0.489	0.505
994	3	0.999	0.607	0.999	999	3	1.100	0.176	1.100
994	4	0.693	0.318	0.693	999	4	0.529	0.404	0.529
994	5	0.525	0.497	0.525	999	5	0.313	0.770	0.770
994	6	1.147	0.279	1.147	999	6	0.398	0.606	0.606
994	7	0.792	0.267	0.792	999	7	0.090	1.413	1.413
994	8	0.605	0.658	0.658	999	8	0.619	0.143	0.619
994	9	0.820	0.738	0.820	999	9	0.524	0.469	0.524
994	10	0.315	0.683	0.683	999	10	0.378	0.451	0.451
995	1	0.960	0.111	0.960	1000	1	0.409	1.015	1.015
995	2	0.655	0.463	0.655	1000	2	0.573	1.101	1.101
995	3	1.282	0.407	1.282	1000	3	0.818	0.392	0.818
995	4	0.292	1.219	1.219	1000	4	0.183	0.797	0.797
995	5	0.733	0.407	0.733	1000	5	0.582	0.618	0.618
995	6	1.062	0.691	1.062	1000	6	1.305	0.236	1.305
995	7	0.258	1.298	1.298	1000	7	0.283	0.468	0.468
995	8	0.919	0.382	0.919	1000	8	1.026	0.973	1.026
995	9	0.506	0.578	0.578	1000	9	0.598	0.524	0.598
995	10	0.254	1.115	1.115	1000	10	0.755	0.358	0.755

\*\*\*\*\*  
TST01  
\*\*\*\*\*

EL TEST CHI-CUADRADO SOBRE CHI-CUADRADO HA SIDO APLICADO A 1000 GRUPOS DE VALORES CHI-CUADRADO GENERADOS POR EL TEST D.UNIF Y SE HAN OBTENIDO LOS SIGUIENTES RESULTADOS:

VALORES CHI-CUADRADO CALCULADOS CADA UNO CON 99GRADOS DE LIBERTAD

J	PROB. P(J)	CHI-2 P( 99)	FREC. ESPERADA	FREC. OBS.	CONTRIBUCION CHI-2
1	0.01	69.2299	10.00	8	0.40
2	0.02	72.2880	10.00	12	0.40
3	0.03	74.2754	10.00	12	0.40
4	0.04	75.7949	10.00	6	1.60
5	0.05	77.0463	10.00	18	6.40
6	0.06	78.1226	10.00	4	3.60
7	0.07	79.0746	10.00	12	0.40
8	0.08	79.9336	10.00	11	0.10
9	0.09	80.7204	10.00	9	0.10
10	0.10	81.4493	10.00	5	2.50
11	0.11	82.1306	10.00	12	0.40
12	0.12	82.7724	10.00	15	2.50
13	0.13	83.3805	10.00	17	4.90
14	0.14	83.9599	10.00	7	0.90
15	0.15	84.5143	10.00	11	0.10
16	0.16	85.0469	10.00	15	2.50
17	0.17	85.5603	10.00	8	0.40
18	0.18	86.0566	10.00	13	0.90
19	0.19	86.5377	10.00	9	0.10
20	0.20	87.0052	10.00	9	0.10
21	0.21	87.4605	10.00	8	0.40
22	0.22	87.9048	10.00	14	1.60
23	0.23	88.3390	10.00	13	0.90
24	0.24	88.7642	10.00	9	0.10

25	0.25	89.1812	10.00	7	0.90
26	0.26	89.5907	10.00	9	0.10
27	0.27	89.9934	10.00	10	0.00
28	0.28	90.3898	10.00	10	0.00
29	0.29	90.7806	10.00	3	4.90
30	0.30	91.1663	10.00	13	0.90
31	0.31	91.5472	10.00	11	0.10
32	0.32	91.9238	10.00	10	0.00
33	0.33	92.2966	10.00	11	0.10
34	0.34	92.6659	10.00	12	0.40
35	0.35	93.0320	10.00	7	0.90
36	0.36	93.3953	10.00	7	0.90
37	0.37	93.7560	10.00	16	3.60
38	0.38	94.1145	10.00	11	0.10
39	0.39	94.4710	10.00	6	1.60
40	0.40	94.8259	10.00	9	0.10
41	0.41	95.1793	10.00	10	0.00
42	0.42	95.5315	10.00	5	2.50
43	0.43	95.8828	10.00	8	0.40
44	0.44	96.2333	10.00	10	0.00
45	0.45	96.5834	10.00	11	0.10
46	0.46	96.9332	10.00	8	0.40
47	0.47	97.2829	10.00	14	1.60
48	0.48	97.6329	10.00	8	0.40
49	0.49	97.9832	10.00	16	3.60
50	0.50	98.3341	10.00	9	0.10

-----  
TOTAL 1000 1000 92.40  
SI X ES UNA VAR.ALEAT. CHI-2 CON 99 GRADOS DE LIBERTAD, ENTONCES  
P(X<= 92.30)=.33 y P(X<= 92.67)=.34

\*\*\*\*\*  
TST01  
\*\*\*\*\*

EL TEST CHI-CUADRADO SOBRE CHI-CUADRADO HA SIDO APLICADO A 1000 GRUPOS  
DE VALORES CHI-CUADRADO GENERADOS POR EL TEST CUPON Y SE HAN OBTENIDO  
LOS SIGUIENTES RESULTADOS:

VALORES CHI-CUADRADO CALCULADOS CADA UNO CON 15GRADOS DE LIBERTAD

TESTS CON SEGMENTOS COMPLETOS CONSISTENTES DE ENTEROS DE 1 A 5 USANDO  
LOS NROS. ALEAT. COMO FUERON GENERADOS

J	PROB. P(J)	CHI-2 P( 15)	FREC. ESPERADA	FREC. OBS.	CONTRIBUCION CHI-2
1	0.01	5.2293	10.00	14	1.60
2	0.02	5.9849	10.00	11	0.10
3	0.03	6.5032	10.00	9	0.10
4	0.04	6.9137	10.00	4	3.60
5	0.05	7.2609	10.00	6	1.60
6	0.06	7.5661	10.00	9	0.10
7	0.07	7.8410	10.00	9	0.10
8	0.08	8.0930	10.00	12	0.40
9	0.09	8.3271	10.00	16	3.60
10	0.10	8.5468	10.00	15	2.50
11	0.11	8.7545	10.00	8	0.40
12	0.12	8.9523	10.00	9	0.10
13	0.13	9.1416	10.00	7	0.90
14	0.14	9.3236	10.00	12	0.40
15	0.15	9.4993	10.00	11	0.10
16	0.16	9.6695	10.00	10	0.00

17	0.17	9.8348	10.00	14	1.60
18	0.18	9.9959	10.00	10	0.00
19	0.19	10.1531	10.00	8	0.40
20	0.20	10.3070	10.00	10	0.00
21	0.21	10.4578	10.00	8	0.40
22	0.22	10.6059	10.00	14	1.60
23	0.23	10.7515	10.00	10	0.00
24	0.24	10.8950	10.00	10	0.00
25	0.25	11.0365	10.00	9	0.10
26	0.26	11.1763	10.00	10	0.00
27	0.27	11.3145	10.00	8	0.40
28	0.28	11.4512	10.00	3	4.90
29	0.29	11.5868	10.00	12	0.40
30	0.30	11.7212	10.00	5	2.50
31	0.31	11.8546	10.00	11	0.10
32	0.32	11.9872	10.00	10	0.00
33	0.33	12.1190	10.00	9	0.10
34	0.34	12.2502	10.00	13	0.90
35	0.35	12.3809	10.00	8	0.40
36	0.36	12.5111	10.00	11	0.10
37	0.37	12.6411	10.00	13	0.90
38	0.38	12.7707	10.00	11	0.10
39	0.39	12.9003	10.00	3	4.90
40	0.40	13.0297	10.00	6	1.60
41	0.41	13.1592	10.00	9	0.10
42	0.42	13.2888	10.00	13	0.90
43	0.43	13.4186	10.00	8	0.40
44	0.44	13.5486	10.00	7	0.90
45	0.45	13.6790	10.00	10	0.00
46	0.46	13.8098	10.00	12	0.40
47	0.47	13.9411	10.00	4	3.60
48	0.48	14.0730	10.00	11	0.10
49	0.49	14.2056	10.00	6	1.60
50	0.50	14.3389	10.00	10	0.00

-----  
TOTAL 1000 1000 87.20  
SI X ES UNA VAR.ALEAT. CHI-2 CON 99 GRADOS DE LIBERTAD, ENTONCES  
 $P(X \leq 87.01) = .20$  y  $P(X \leq 87.46) = .21$

EL TEST CHI-CUADRADO SOBRE CHI-CUADRADO HA SIDO APLICADO A 1000 GRUPOS DE VALORES CHI-CUADRADO GENERADOS POR EL TEST CUPON Y SE HAN OBTENIDO LOS SIGUIENTES RESULTADOS:

VALORES CHI-CUADRADO CALCULADOS CADA UNO CON 15GRADOS DE LIBERTAD

TESTS CON SEGMENTOS COMPLETOS CONSISTENTES DE ENTEROS DE 1 A 5 USANDO LA PARTE FRACCIONARIA DE 100 \* NUM. ALEAT.

J	PROB. P(J)	CHI-2 P( 15)	FREC. ESPERADA	FREC. OBS.	CONTRIBUCION CHI-2
1	0.01	5.2293	10.00	8	0.40
2	0.02	5.9849	10.00	11	0.10
3	0.03	6.5032	10.00	14	1.60
4	0.04	6.9137	10.00	18	6.40
5	0.05	7.2609	10.00	12	0.40
6	0.06	7.5661	10.00	11	0.10
7	0.07	7.8410	10.00	11	0.10
8	0.08	8.0930	10.00	11	0.10
9	0.09	8.3271	10.00	5	2.50
10	0.10	8.5468	10.00	7	0.90
11	0.11	8.7545	10.00	9	0.10
12	0.12	8.9523	10.00	8	0.40
13	0.13	9.1416	10.00	14	1.60
14	0.14	9.3236	10.00	8	0.40
15	0.15	9.4993	10.00	8	0.40
16	0.16	9.6695	10.00	12	0.40
17	0.17	9.8348	10.00	5	2.50

18	0.18	9.9959	10.00	5	2.50
19	0.19	10.1531	10.00	9	0.10
20	0.20	10.3070	10.00	12	0.40
21	0.21	10.4578	10.00	9	0.10
22	0.22	10.6059	10.00	9	0.10
23	0.23	10.7515	10.00	14	1.60
24	0.24	10.8950	10.00	13	0.90
25	0.25	11.0365	10.00	11	0.10
26	0.26	11.1763	10.00	7	0.90
27	0.27	11.3145	10.00	18	6.40
28	0.28	11.4512	10.00	10	0.00
29	0.29	11.5868	10.00	14	1.60
30	0.30	11.7212	10.00	13	0.90
31	0.31	11.8546	10.00	7	0.90
32	0.32	11.9872	10.00	6	1.60
33	0.33	12.1190	10.00	14	1.60
34	0.34	12.2502	10.00	9	0.10
35	0.35	12.3809	10.00	8	0.40
36	0.36	12.5111	10.00	9	0.10
37	0.37	12.6411	10.00	12	0.40
38	0.38	12.7707	10.00	10	0.00
39	0.39	12.9003	10.00	18	6.40
40	0.40	13.0297	10.00	8	0.40
41	0.41	13.1592	10.00	9	0.10
42	0.42	13.2888	10.00	5	2.50
43	0.43	13.4186	10.00	14	1.60
44	0.44	13.5486	10.00	10	0.00
45	0.45	13.6790	10.00	11	0.10
46	0.46	13.8098	10.00	10	0.00
47	0.47	13.9411	10.00	13	0.90
48	0.48	14.0730	10.00	14	1.60
49	0.49	14.2056	10.00	10	0.00
50	0.50	14.3389	10.00	9	0.10

-----  
TOTAL 1000 1000 95.40  
SI X ES UNA VAR.ALEAT. CHI-2 CON 99 GRADOS DE LIBERTAD, ENTONCES  
 $P(X \leq 95.18) = .41$  y  $P(X \leq 95.53) = .42$

EL TEST CHI-CUADRADO SOBRE CHI-CUADRADO HA SIDO APLICADO A 1000 GRUPOS  
DE VALORES CHI-CUADRADO GENERADOS POR EL TEST CUPON Y SE HAN OBTENIDO  
LOS SIGUIENTES RESULTADOS:

VALORES CHI-CUADRADO CALCULADOS CADA UNO CON 5 GRADOS DE LIBERTAD

TESTS CON SEGMENTOS COMPLETOS CONSISTENTES DE ENTEROS DE 1 A 10 USANDO  
UNA PARTICION DE LONGITUD 40 O MENOR

J	PROB. P(J)	CHI-2 P( 5)	FREC. ESPERADA	FREC. OBS.	CONTRIBUCION CHI-2
1	0.01	0.5543	10.00	7	0.90
2	0.02	0.7519	10.00	9	0.10
3	0.03	0.9031	10.00	8	0.40
4	0.04	1.0313	10.00	13	0.90
5	0.05	1.1455	10.00	12	0.40
6	0.06	1.2499	10.00	8	0.40
7	0.07	1.3472	10.00	10	0.00
8	0.08	1.4390	10.00	9	0.10
9	0.09	1.5264	10.00	11	0.10
10	0.10	1.6103	10.00	8	0.40
11	0.11	1.6913	10.00	10	0.00
12	0.12	1.7697	10.00	10	0.00
13	0.13	1.8461	10.00	5	2.50
14	0.14	1.9207	10.00	9	0.10
15	0.15	1.9938	10.00	7	0.90
16	0.16	2.0656	10.00	9	0.10
17	0.17	2.1362	10.00	11	0.10
18	0.18	2.2058	10.00	10	0.00

19	0.19	2.2745	10.00	7	0.90
20	0.20	2.3425	10.00	14	1.60
21	0.21	2.4099	10.00	5	2.50
22	0.22	2.4767	10.00	13	0.90
23	0.23	2.5430	10.00	10	0.00
24	0.24	2.6090	10.00	11	0.10
25	0.25	2.6746	10.00	10	0.00
26	0.26	2.7400	10.00	6	1.60
27	0.27	2.8051	10.00	15	2.50
28	0.28	2.8701	10.00	8	0.40
29	0.29	2.9350	10.00	14	1.60
30	0.30	2.9999	10.00	7	0.90
31	0.31	3.0648	10.00	10	0.00
32	0.32	3.1297	10.00	10	0.00
33	0.33	3.1947	10.00	9	0.10
34	0.34	3.2598	10.00	7	0.90
35	0.35	3.3251	10.00	8	0.40
36	0.36	3.3906	10.00	5	2.50
37	0.37	3.4564	10.00	13	0.90
38	0.38	3.5224	10.00	8	0.40
39	0.39	3.5888	10.00	14	1.60
40	0.40	3.6555	10.00	15	2.50
41	0.41	3.7226	10.00	9	0.10
42	0.42	3.7902	10.00	12	0.40
43	0.43	3.8582	10.00	6	1.60
44	0.44	3.9268	10.00	12	0.40
45	0.45	3.9959	10.00	7	0.90
46	0.46	4.0657	10.00	8	0.40
47	0.47	4.1360	10.00	13	0.90
48	0.48	4.2071	10.00	10	0.00
49	0.49	4.2789	10.00	13	0.90
50	0.50	4.3515	10.00	13	0.90

-----  
TOTAL 1000 1000 98.40  
SI X ES UNA VAR.ALEAT. CHI-2 CON 99 GRADOS DE LIBERTAD, ENTONCES  
P(X<= 98.33)=.50 y P(X<= 98.69)=.51

\*\*\*\*\*  
TST01  
\*\*\*\*\*

EL TEST CHI-CUADRADO SOBRE CHI-CUADRADO HA SIDO APLICADO A 1000 GRUPOS  
DE VALORES CHI-CUADRADO GENERADOS POR EL TEST GAP Y SE HAN OBTENIDO  
LOS SIGUIENTES RESULTADOS:

VALORES CHI-CUADRADO CALCULADOS CADA UNO CON 10 GRADOS DE LIBERTAD

TESTS CON INTERVALOS DESDE 0.000 A 0.500 - CORRIDAS POR DEBAJO  
DE LA MEDIA

J	PROB. P(J)	CHI-2 P( 10)	FREC. ESPERADA	FREC. OBS.	CONTRIBUCION CHI-2
1	0.01	2.5582	10.00	13	0.90
2	0.02	3.0591	10.00	10	0.00
3	0.03	3.4121	10.00	10	0.00
4	0.04	3.6965	10.00	6	1.60
5	0.05	3.9403	10.00	17	4.90
6	0.06	4.1567	10.00	10	0.00
7	0.07	4.3534	10.00	11	0.10
8	0.08	4.5350	10.00	5	2.50
9	0.09	4.7049	10.00	9	0.10
10	0.10	4.8652	10.00	14	1.60
11	0.11	5.0176	10.00	11	0.10
12	0.12	5.1634	10.00	15	2.50

13	0.13	5.3036	10.00	11	0.10
14	0.14	5.4389	10.00	11	0.10
15	0.15	5.5701	10.00	16	3.60
16	0.16	5.6976	10.00	9	0.10
17	0.17	5.8219	10.00	8	0.40
18	0.18	5.9434	10.00	8	0.40
19	0.19	6.0623	10.00	9	0.10
20	0.20	6.1791	10.00	13	0.90
21	0.21	6.2939	10.00	10	0.00
22	0.22	6.4069	10.00	8	0.40
23	0.23	6.5183	10.00	9	0.10
24	0.24	6.6284	10.00	13	0.90
25	0.25	6.7372	10.00	12	0.40
26	0.26	6.8449	10.00	13	0.90
27	0.27	6.9517	10.00	9	0.10
28	0.28	7.0576	10.00	12	0.40
29	0.29	7.1627	10.00	13	0.90
30	0.30	7.2672	10.00	8	0.40
31	0.31	7.3712	10.00	12	0.40
32	0.32	7.4747	10.00	7	0.90
33	0.33	7.5778	10.00	9	0.10
34	0.34	7.6806	10.00	9	0.10
35	0.35	7.7832	10.00	10	0.00
36	0.36	7.8857	10.00	11	0.10
37	0.37	7.9881	10.00	5	2.50
38	0.38	8.0905	10.00	8	0.40
39	0.39	8.1929	10.00	8	0.40
40	0.40	8.2955	10.00	11	0.10
41	0.41	8.3982	10.00	10	0.00
42	0.42	8.5012	10.00	8	0.40
43	0.43	8.6045	10.00	11	0.10
44	0.44	8.7082	10.00	12	0.40
45	0.45	8.8124	10.00	11	0.10
46	0.46	8.9170	10.00	13	0.90
47	0.47	9.0222	10.00	5	2.50
48	0.48	9.1280	10.00	11	0.10
49	0.49	9.2345	10.00	12	0.40
50	0.50	9.3418	10.00	10	0.00

-----  
TOTAL 1000 1000 82.60  
SI X ES UNA VAR.ALEAT. CHI-2 CON 99 GRADOS DE LIBERTAD, ENTONCES  
 $P(X \leq 82.13) = .11$  y  $P(X \leq 82.77) = .12$

EL TEST CHI-CUADRADO SOBRE CHI-CUADRADO HA SIDO APLICADO A 1000 GRUPOS DE VALORES CHI-CUADRADO GENERADOS POR EL TEST GAP Y SE HAN OBTENIDO LOS SIGUIENTES RESULTADOS:

VALORES CHI-CUADRADO CALCULADOS CADA UNO CON 10GRADOS DE LIBERTAD

TESTS CON INTERVALOS DESDE 0.500 A 1.000 - CORRIDAS POR ARRIBA DE LA MEDIA

J	PROB. P(J)	CHI-2 P( 10)	FREC. ESPERADA	FREC. OBS.	CONTRIBUCION CHI-2
1	0.01	2.5582	10.00	7	0.90
2	0.02	3.0591	10.00	17	4.90
3	0.03	3.4121	10.00	8	0.40
4	0.04	3.6965	10.00	10	0.00
5	0.05	3.9403	10.00	9	0.10
6	0.06	4.1567	10.00	9	0.10
7	0.07	4.3534	10.00	7	0.90
8	0.08	4.5350	10.00	7	0.90
9	0.09	4.7049	10.00	11	0.10
10	0.10	4.8652	10.00	6	1.60
11	0.11	5.0176	10.00	2	6.40
12	0.12	5.1634	10.00	14	1.60
13	0.13	5.3036	10.00	13	0.90

14	0.14	5.4389	10.00	9	0.10
15	0.15	5.5701	10.00	10	0.00
16	0.16	5.6976	10.00	6	1.60
17	0.17	5.8219	10.00	9	0.10
18	0.18	5.9434	10.00	13	0.90
19	0.19	6.0623	10.00	7	0.90
20	0.20	6.1791	10.00	6	1.60
21	0.21	6.2939	10.00	8	0.40
22	0.22	6.4069	10.00	5	2.50
23	0.23	6.5183	10.00	13	0.90
24	0.24	6.6284	10.00	9	0.10
25	0.25	6.7372	10.00	9	0.10
26	0.26	6.8449	10.00	9	0.10
27	0.27	6.9517	10.00	13	0.90
28	0.28	7.0576	10.00	12	0.40
29	0.29	7.1627	10.00	9	0.10
30	0.30	7.2672	10.00	13	0.90
31	0.31	7.3712	10.00	9	0.10
32	0.32	7.4747	10.00	10	0.00
33	0.33	7.5778	10.00	15	2.50
34	0.34	7.6806	10.00	15	2.50
35	0.35	7.7832	10.00	10	0.00
36	0.36	7.8857	10.00	11	0.10
37	0.37	7.9881	10.00	8	0.40
38	0.38	8.0905	10.00	8	0.40
39	0.39	8.1929	10.00	11	0.10
40	0.40	8.2955	10.00	6	1.60
41	0.41	8.3982	10.00	16	3.60
42	0.42	8.5012	10.00	8	0.40
43	0.43	8.6045	10.00	5	2.50
44	0.44	8.7082	10.00	17	4.90
45	0.45	8.8124	10.00	14	1.60
46	0.46	8.9170	10.00	13	0.90
47	0.47	9.0222	10.00	9	0.10
48	0.48	9.1280	10.00	8	0.40
49	0.49	9.2345	10.00	6	1.60
50	0.50	9.3418	10.00	11	0.10

-----  
TOTAL 1000 1000 112.60  
SI X ES UNA VAR.ALEAT. CHI-2 CON 99 GRADOS DE LIBERTAD, ENTONCES  
 $P(X \leq 112.32) = .83$  y  $P(X \leq 112.94) = .84$

EL TEST CHI-CUADRADO SOBRE CHI-CUADRADO HA SIDO APLICADO A 1000 GRUPOS  
DE VALORES CHI-CUADRADO GENERADOS POR EL TEST GAP Y SE HAN OBTENIDO  
LOS SIGUIENTES RESULTADOS:

VALORES CHI-CUADRADO CALCULADOS CADA UNO CON 13 GRADOS DE LIBERTAD

TESTS CON INTERVALOS DE

.333 A .667

J	PROB. P(J)	CHI-2 P( 13)	FREC. ESPERADA	FREC. OBS.	CONTRIBUCION CHI-2
1	0.01	4.1069	10.00	10	0.00
2	0.02	4.7654	10.00	13	0.90
3	0.03	5.2210	10.00	10	0.00
4	0.04	5.5838	10.00	15	2.50
5	0.05	5.8919	10.00	15	2.50
6	0.06	6.1635	10.00	13	0.90
7	0.07	6.4088	10.00	10	0.00
8	0.08	6.6343	10.00	10	0.00
9	0.09	6.8442	10.00	9	0.10
10	0.10	7.0415	10.00	11	0.10
11	0.11	7.2284	10.00	9	0.10
12	0.12	7.4066	10.00	9	0.10
13	0.13	7.5774	10.00	10	0.00



14	0.14	7.7419	10.00	15	2.50
15	0.15	7.9008	10.00	18	6.40
16	0.16	8.0550	10.00	9	0.10
17	0.17	8.2049	10.00	7	0.90
18	0.18	8.3511	10.00	6	1.60
19	0.19	8.4939	10.00	15	2.50
20	0.20	8.6339	10.00	8	0.40
21	0.21	8.7711	10.00	6	1.60
22	0.22	8.9061	10.00	7	0.90
23	0.23	9.0389	10.00	13	0.90
24	0.24	9.1698	10.00	10	0.00
25	0.25	9.2991	10.00	15	2.50
26	0.26	9.4268	10.00	7	0.90
27	0.27	9.5532	10.00	7	0.90
28	0.28	9.6784	10.00	9	0.10
29	0.29	9.8025	10.00	9	0.10
30	0.30	9.9257	10.00	4	3.60
31	0.31	10.0481	10.00	6	1.60
32	0.32	10.1697	10.00	10	0.00
33	0.33	10.2908	10.00	11	0.10
34	0.34	10.4114	10.00	3	4.90
35	0.35	10.5315	10.00	10	0.00
36	0.36	10.6513	10.00	14	1.60
37	0.37	10.7709	10.00	9	0.10
38	0.38	10.8904	10.00	5	2.50
39	0.39	11.0098	10.00	9	0.10
40	0.40	11.1291	10.00	12	0.40
41	0.41	11.2486	10.00	13	0.90
42	0.42	11.3682	10.00	10	0.00
43	0.43	11.4881	10.00	6	1.60
44	0.44	11.6082	10.00	11	0.10
45	0.45	11.7288	10.00	12	0.40
46	0.46	11.8498	10.00	7	0.90
47	0.47	11.9713	10.00	11	0.10
48	0.48	12.0934	10.00	12	0.40
49	0.49	12.2162	10.00	6	1.60
50	0.50	12.3398	10.00	8	0.40

-----  
TOTAL 1000 1000 102.40  
SI X ES UNA VAR.ALEAT. CHI-2 CON 99 GRADOS DE LIBERTAD, ENTONCES  
 $P(X \leq 102.30) = .61$  y  $P(X \leq 102.68) = .62$

\*\*\*\*\*  
TST01  
\*\*\*\*\*

EL TEST CHI-CUADRADO SOBRE CHI-CUADRADO HA SIDO APLICADO A 1000 GRUPOS  
DE VALORES CHI-CUADRADO GENERADOS POR EL TEST PERMUT Y SE HAN OBTENIDO  
LOS SIGUIENTES RESULTADOS:

VALORES CHI-CUADRADO CALCULADOS CADA UNO CON 5 GRADOS DE LIBERTAD

TESTS CON PERMUTACIONES DE 3 OBJETOS

J	PROB. P(J)	CHI-2 P( 5)	FREC. ESPERADA	FREC. OBS.	CONTRIBUCION CHI-2
1	0.01	0.5543	10.00	8	0.40
2	0.02	0.7519	10.00	10	0.00
3	0.03	0.9031	10.00	7	0.90
4	0.04	1.0313	10.00	8	0.40
5	0.05	1.1455	10.00	10	0.00
6	0.06	1.2499	10.00	15	2.50
7	0.07	1.3472	10.00	14	1.60



9	0.09	14.5510	10.00	7	0.90
10	0.10	14.8480	10.00	14	1.60
11	0.11	15.1277	10.00	5	2.50
12	0.12	15.3929	10.00	7	0.90
13	0.13	15.6460	10.00	9	0.10
14	0.14	15.8885	10.00	6	1.60
15	0.15	16.1219	10.00	5	2.50
16	0.16	16.3474	10.00	11	0.10
17	0.17	16.5658	10.00	10	0.00
18	0.18	16.7780	10.00	12	0.40
19	0.19	16.9847	10.00	11	0.10
20	0.20	17.1865	10.00	6	1.60
21	0.21	17.3839	10.00	7	0.90
22	0.22	17.5772	10.00	17	4.90
23	0.23	17.7670	10.00	9	0.10
24	0.24	17.9536	10.00	9	0.10
25	0.25	18.1373	10.00	13	0.90
26	0.26	18.3183	10.00	12	0.40
27	0.27	18.4970	10.00	7	0.90
28	0.28	18.6736	10.00	7	0.90
29	0.29	18.8482	10.00	8	0.40
30	0.30	19.0211	10.00	11	0.10
31	0.31	19.1924	10.00	6	1.60
32	0.32	19.3624	10.00	11	0.10
33	0.33	19.5312	10.00	8	0.40
34	0.34	19.6989	10.00	13	0.90
35	0.35	19.8657	10.00	11	0.10
36	0.36	20.0317	10.00	10	0.00
37	0.37	20.1970	10.00	15	2.50
38	0.38	20.3618	10.00	10	0.00
39	0.39	20.5262	10.00	10	0.00
40	0.40	20.6902	10.00	8	0.40
41	0.41	20.8541	10.00	10	0.00
42	0.42	21.0178	10.00	10	0.00
43	0.43	21.1816	10.00	15	2.50
44	0.44	21.3455	10.00	5	2.50
45	0.45	21.5096	10.00	11	0.10
46	0.46	21.6740	10.00	15	2.50
47	0.47	21.8388	10.00	6	1.60
48	0.48	22.0042	10.00	11	0.10
49	0.49	22.1702	10.00	13	0.90
50	0.50	22.3369	10.00	9	0.10

-----  
TOTAL 1000 1000 99.40  
SI X ES UNA VAR.ALEAT. CHI-2 CON 99 GRADOS DE LIBERTAD, ENTONCES  
 $P(X \leq 99.39) = .53$  y  $P(X \leq 99.75) = .54$

EL TEST CHI-CUADRADO SOBRE CHI-CUADRADO HA SIDO APLICADO A 1000 GRUPOS DE VALORES CHI-CUADRADO GENERADOS POR EL TEST PERMUT Y SE HAN OBTENIDO LOS SIGUIENTES RESULTADOS:

VALORES CHI-CUADRADO CALCULADOS CADA UNO CON 119 GRADOS DE LIBERTAD

TESTS CON PERMUTACIONES DE 5 OBJETOS

J	PROB. P(J)	CHI-2 P( 119)	FREC. ESPERADA	FREC. OBS.	CONTRIBUCION CHI-2
1	0.01	86.0738	10.00	12	0.40
2	0.02	89.4997	10.00	2	6.40
3	0.03	91.7210	10.00	4	3.60
4	0.04	93.4165	10.00	9	0.10
5	0.05	94.8112	10.00	12	0.40
6	0.06	96.0095	10.00	13	0.90
7	0.07	97.0686	10.00	12	0.40
8	0.08	98.0235	10.00	7	0.90
9	0.09	98.8976	10.00	11	0.10

10	0.10	99.7067	10.00	10	0.00
11	0.11	100.4628	10.00	8	0.40
12	0.12	101.1745	10.00	11	0.10
13	0.13	101.8487	10.00	8	0.40
14	0.14	102.4906	10.00	4	3.60
15	0.15	103.1046	10.00	13	0.90
16	0.16	103.6942	10.00	6	1.60
17	0.17	104.2623	10.00	9	0.10
18	0.18	104.8113	10.00	11	0.10
19	0.19	105.3434	10.00	4	3.60
20	0.20	105.8602	10.00	10	0.00
21	0.21	106.3633	10.00	11	0.10
22	0.22	106.8541	10.00	9	0.10
23	0.23	107.3337	10.00	10	0.00
24	0.24	107.8032	10.00	9	0.10
25	0.25	108.2634	10.00	8	0.40
26	0.26	108.7153	10.00	5	2.50
27	0.27	109.1595	10.00	7	0.90
28	0.28	109.5967	10.00	6	1.60
29	0.29	110.0276	10.00	8	0.40
30	0.30	110.4526	10.00	11	0.10
31	0.31	110.8724	10.00	13	0.90
32	0.32	111.2873	10.00	14	1.60
33	0.33	111.6979	10.00	9	0.10
34	0.34	112.1045	10.00	8	0.40
35	0.35	112.5075	10.00	11	0.10
36	0.36	112.9073	10.00	12	0.40
37	0.37	113.3043	10.00	6	1.60
38	0.38	113.6987	10.00	13	0.90
39	0.39	114.0908	10.00	14	1.60
40	0.40	114.4810	10.00	15	2.50
41	0.41	114.8695	10.00	7	0.90
42	0.42	115.2566	10.00	11	0.10
43	0.43	115.6426	10.00	6	1.60
44	0.44	116.0277	10.00	9	0.10
45	0.45	116.4122	10.00	8	0.40
46	0.46	116.7964	10.00	5	2.50
47	0.47	117.1804	10.00	12	0.40
48	0.48	117.5645	10.00	14	1.60
49	0.49	117.9489	10.00	10	0.00
50	0.50	118.3340	10.00	8	0.40

-----  
TOTAL 1000 1000 113.40  
SI X ES UNA VAR.ALEAT. CHI-2 CON 99 GRADOS DE LIBERTAD, ENTONCES  
 $P(X \leq 112.94) = .84$  y  $P(X \leq 113.59) = .85$

\*\*\*\*\*  
TST01  
\*\*\*\*\*

EL TEST CHI-CUADRADO SOBRE CHI-CUADRADO HA SIDO APLICADO A 1000 GRUPOS  
DE VALORES CHI-CUADRADO GENERADOS POR EL TEST POKER Y SE HAN OBTENIDO  
LOS SIGUIENTES RESULTADOS:

VALORES CHI-CUADRADO CALCULADOS CADA UNO CON 3GRADOS DE LIBERTAD

TESTS CON MANOS DE 4 CON 4 REPARTICIONES

J	PROB. P(J)	CHI-2 P( 3)	FREC. ESPERADA	FREC. OBS.	CONTRIBUCION CHI-2
1	0.01	0.1148	10.00	14	1.60
2	0.02	0.1848	10.00	10	0.00
3	0.03	0.2451	10.00	10	0.00

4	0.04	0.3002	10.00	7	0.90
5	0.05	0.3518	10.00	15	2.50
6	0.06	0.4012	10.00	11	0.10
7	0.07	0.4487	10.00	4	3.60
8	0.08	0.4949	10.00	8	0.40
9	0.09	0.5401	10.00	9	0.10
10	0.10	0.5844	10.00	11	0.10
11	0.11	0.6280	10.00	6	1.60
12	0.12	0.6710	10.00	7	0.90
13	0.13	0.7136	10.00	10	0.00
14	0.14	0.7558	10.00	7	0.90
15	0.15	0.7978	10.00	6	1.60
16	0.16	0.8395	10.00	10	0.00
17	0.17	0.8810	10.00	13	0.90
18	0.18	0.9225	10.00	11	0.10
19	0.19	0.9638	10.00	10	0.00
20	0.20	1.0052	10.00	11	0.10
21	0.21	1.0465	10.00	6	1.60
22	0.22	1.0879	10.00	9	0.10
23	0.23	1.1293	10.00	6	1.60
24	0.24	1.1709	10.00	13	0.90
25	0.25	1.2125	10.00	10	0.00
26	0.26	1.2544	10.00	11	0.10
27	0.27	1.2964	10.00	5	2.50
28	0.28	1.3386	10.00	9	0.10
29	0.29	1.3810	10.00	9	0.10
30	0.30	1.4237	10.00	6	1.60
31	0.31	1.4666	10.00	9	0.10
32	0.32	1.5098	10.00	7	0.90
33	0.33	1.5534	10.00	8	0.40
34	0.34	1.5973	10.00	10	0.00
35	0.35	1.6416	10.00	13	0.90
36	0.36	1.6862	10.00	8	0.40
37	0.37	1.7313	10.00	7	0.90
38	0.38	1.7768	10.00	11	0.10
39	0.39	1.8227	10.00	10	0.00
40	0.40	1.8692	10.00	10	0.00
41	0.41	1.9161	10.00	14	1.60
42	0.42	1.9636	10.00	7	0.90
43	0.43	2.0116	10.00	8	0.40
44	0.44	2.0602	10.00	11	0.10
45	0.45	2.1095	10.00	13	0.90
46	0.46	2.1593	10.00	8	0.40
47	0.47	2.2099	10.00	9	0.10
48	0.48	2.2612	10.00	14	1.60
49	0.49	2.3132	10.00	15	2.50
50	0.50	2.3660	10.00	10	0.00

-----  
TOTAL 1000 1000 107.80  
SI X ES UNA VAR.ALEAT. CHI-2 CON 99 GRADOS DE LIBERTAD, ENTONCES  
 $P(X \leq 107.63) = .74$  y  $P(X \leq 108.09) = .75$

EL TEST CHI-CUADRADO SOBRE CHI-CUADRADO HA SIDO APLICADO A 1000 GRUPOS  
DE VALORES CHI-CUADRADO GENERADOS POR EL TEST POKER Y SE HAN OBTENIDO  
LOS SIGUIENTES RESULTADOS:

VALORES CHI-CUADRADO CALCULADOS CADA UNO CON 5 GRADOS DE LIBERTAD

TESTS CON MANOS DE 5 CON 6 REPARTICIONES

J	PROB. P(J)	CHI-2 P( 5)	FREC. ESPERADA	FREC. OBS.	CONTRIBUCION CHI-2
1	0.01	0.5543	10.00	8	0.40
2	0.02	0.7519	10.00	6	1.60
3	0.03	0.9031	10.00	6	1.60
4	0.04	1.0313	10.00	15	2.50

5	0.05	1.1455	10.00	15	2.50
6	0.06	1.2499	10.00	9	0.10
7	0.07	1.3472	10.00	4	3.60
8	0.08	1.4390	10.00	9	0.10
9	0.09	1.5264	10.00	8	0.40
10	0.10	1.6103	10.00	15	2.50
11	0.11	1.6913	10.00	11	0.10
12	0.12	1.7697	10.00	12	0.40
13	0.13	1.8461	10.00	7	0.90
14	0.14	1.9207	10.00	13	0.90
15	0.15	1.9938	10.00	10	0.00
16	0.16	2.0656	10.00	6	1.60
17	0.17	2.1362	10.00	6	1.60
18	0.18	2.2058	10.00	8	0.40
19	0.19	2.2745	10.00	8	0.40
20	0.20	2.3425	10.00	13	0.90
21	0.21	2.4099	10.00	7	0.90
22	0.22	2.4767	10.00	6	1.60
23	0.23	2.5430	10.00	7	0.90
24	0.24	2.6090	10.00	13	0.90
25	0.25	2.6746	10.00	8	0.40
26	0.26	2.7400	10.00	12	0.40
27	0.27	2.8051	10.00	13	0.90
28	0.28	2.8701	10.00	13	0.90
29	0.29	2.9350	10.00	9	0.10
30	0.30	2.9999	10.00	15	2.50
31	0.31	3.0648	10.00	9	0.10
32	0.32	3.1297	10.00	10	0.00
33	0.33	3.1947	10.00	17	4.90
34	0.34	3.2598	10.00	12	0.40
35	0.35	3.3251	10.00	10	0.00
36	0.36	3.3906	10.00	10	0.00
37	0.37	3.4564	10.00	8	0.40
38	0.38	3.5224	10.00	10	0.00
39	0.39	3.5888	10.00	13	0.90
40	0.40	3.6555	10.00	11	0.10
41	0.41	3.7226	10.00	7	0.90
42	0.42	3.7902	10.00	13	0.90
43	0.43	3.8582	10.00	8	0.40
44	0.44	3.9268	10.00	9	0.10
45	0.45	3.9959	10.00	17	4.90
46	0.46	4.0657	10.00	9	0.10
47	0.47	4.1360	10.00	11	0.10
48	0.48	4.2071	10.00	13	0.90
49	0.49	4.2789	10.00	13	0.90
50	0.50	4.3515	10.00	9	0.10

-----  
TOTAL 1000 1000 96.00  
SI X ES UNA VAR.ALEAT. CHI-2 CON 99 GRADOS DE LIBERTAD, ENTONCES  
 $P(X \leq 95.88) = .43$  y  $P(X \leq 96.23) = .44$

EL TEST CHI-CUADRADO SOBRE CHI-CUADRADO HA SIDO APLICADO A 1000 GRUPOS DE VALORES CHI-CUADRADO GENERADOS POR EL TEST POKER Y SE HAN OBTENIDO LOS SIGUIENTES RESULTADOS:

VALORES CHI-CUADRADO CALCULADOS CADA UNO CON 3 GRADOS DE LIBERTAD

TESTS CON MANOS DE 5 CON 4 REPARTICIONES

J	PROB. P(J)	CHI-2 P( 3)	FREC. ESPERADA	FREC. OBS.	CONTRIBUCION CHI-2
1	0.01	0.1148	10.00	9	0.10
2	0.02	0.1848	10.00	14	1.60
3	0.03	0.2451	10.00	11	0.10
4	0.04	0.3002	10.00	11	0.10
5	0.05	0.3518	10.00	9	0.10

6	0.06	0.4012	10.00	12	0.40
7	0.07	0.4487	10.00	10	0.00
8	0.08	0.4949	10.00	12	0.40
9	0.09	0.5401	10.00	9	0.10
10	0.10	0.5844	10.00	6	1.60
11	0.11	0.6280	10.00	5	2.50
12	0.12	0.6710	10.00	7	0.90
13	0.13	0.7136	10.00	9	0.10
14	0.14	0.7558	10.00	9	0.10
15	0.15	0.7978	10.00	10	0.00
16	0.16	0.8395	10.00	13	0.90
17	0.17	0.8810	10.00	12	0.40
18	0.18	0.9225	10.00	8	0.40
19	0.19	0.9638	10.00	13	0.90
20	0.20	1.0052	10.00	11	0.10
21	0.21	1.0465	10.00	10	0.00
22	0.22	1.0879	10.00	12	0.40
23	0.23	1.1293	10.00	9	0.10
24	0.24	1.1709	10.00	12	0.40
25	0.25	1.2125	10.00	11	0.10
26	0.26	1.2544	10.00	6	1.60
27	0.27	1.2964	10.00	10	0.00
28	0.28	1.3386	10.00	11	0.10
29	0.29	1.3810	10.00	11	0.10
30	0.30	1.4237	10.00	11	0.10
31	0.31	1.4666	10.00	9	0.10
32	0.32	1.5098	10.00	18	6.40
33	0.33	1.5534	10.00	9	0.10
34	0.34	1.5973	10.00	8	0.40
35	0.35	1.6416	10.00	9	0.10
36	0.36	1.6862	10.00	11	0.10
37	0.37	1.7313	10.00	9	0.10
38	0.38	1.7768	10.00	12	0.40
39	0.39	1.8227	10.00	8	0.40
40	0.40	1.8692	10.00	8	0.40
41	0.41	1.9161	10.00	7	0.90
42	0.42	1.9636	10.00	9	0.10
43	0.43	2.0116	10.00	8	0.40
44	0.44	2.0602	10.00	8	0.40
45	0.45	2.1095	10.00	7	0.90
46	0.46	2.1593	10.00	12	0.40
47	0.47	2.2099	10.00	7	0.90
48	0.48	2.2612	10.00	10	0.00
49	0.49	2.3132	10.00	12	0.40
50	0.50	2.3660	10.00	9	0.10

-----  
TOTAL 1000 1000 90.40  
SI X ES UNA VAR.ALEAT. CHI-2 CON 99 GRADOS DE LIBERTAD, ENTONCES  
P(X<= 90.39)=.28 y P(X<= 90.78)=.29

\*\*\*\*\*  
TST01  
\*\*\*\*\*

EL TEST CHI-CUADRADO SOBRE CHI-CUADRADO HA SIDO APLICADO A 1000 GRUPOS  
DE VALORES CHI-CUADRADO GENERADOS POR EL TEST CORRID Y SE HAN OBTENIDO  
LOS SIGUIENTES RESULTADOS:

VALORES CHI-CUADRADO CALCULADOS CADA UNO CON 6GRADOS DE LIBERTAD

TESTS USANDO LOS NUMEROS ALEATORIOS COMO FUERON GENERADOS

J	PROB. P(J)	CHI-2 P( 6)	FREC. ESPERADA	FREC. OBS.	CONTRIBUCION CHI-2
---	---------------	----------------	-------------------	---------------	-----------------------

1	0.01	0.8721	10.00	10	0.00
2	0.02	1.1344	10.00	10	0.00
3	0.03	1.3296	10.00	4	3.60
4	0.04	1.4924	10.00	9	0.10
5	0.05	1.6354	10.00	6	1.60
6	0.06	1.7649	10.00	9	0.10
7	0.07	1.8846	10.00	11	0.10
8	0.08	1.9967	10.00	5	2.50
9	0.09	2.1029	10.00	8	0.40
10	0.10	2.2041	10.00	12	0.40
11	0.11	2.3014	10.00	10	0.00
12	0.12	2.3953	10.00	9	0.10
13	0.13	2.4863	10.00	6	1.60
14	0.14	2.5748	10.00	12	0.40
15	0.15	2.6613	10.00	8	0.40
16	0.16	2.7459	10.00	9	0.10
17	0.17	2.8289	10.00	14	1.60
18	0.18	2.9104	10.00	6	1.60
19	0.19	2.9908	10.00	8	0.40
20	0.20	3.0701	10.00	9	0.10
21	0.21	3.1484	10.00	16	3.60
22	0.22	3.2260	10.00	8	0.40
23	0.23	3.3028	10.00	11	0.10
24	0.24	3.3789	10.00	12	0.40
25	0.25	3.4546	10.00	10	0.00
26	0.26	3.5298	10.00	10	0.00
27	0.27	3.6046	10.00	12	0.40
28	0.28	3.6792	10.00	8	0.40
29	0.29	3.7534	10.00	10	0.00
30	0.30	3.8276	10.00	9	0.10
31	0.31	3.9015	10.00	7	0.90
32	0.32	3.9754	10.00	10	0.00
33	0.33	4.0493	10.00	5	2.50
34	0.34	4.1233	10.00	7	0.90
35	0.35	4.1973	10.00	10	0.00
36	0.36	4.2714	10.00	15	2.50
37	0.37	4.3457	10.00	9	0.10
38	0.38	4.4203	10.00	11	0.10
39	0.39	4.4950	10.00	10	0.00
40	0.40	4.5702	10.00	10	0.00
41	0.41	4.6456	10.00	7	0.90
42	0.42	4.7215	10.00	14	1.60
43	0.43	4.7978	10.00	9	0.10
44	0.44	4.8746	10.00	10	0.00
45	0.45	4.9519	10.00	6	1.60
46	0.46	5.0298	10.00	11	0.10
47	0.47	5.1083	10.00	12	0.40
48	0.48	5.1875	10.00	12	0.40
49	0.49	5.2674	10.00	12	0.40
50	0.50	5.3481	10.00	8	0.40

TOTAL 1000 1000 102.60  
SI X ES UNA VAR.ALEAT. CHI-2 CON 99 GRADOS DE LIBERTAD, ENTONCES  
 $P(X \leq 102.30) = .61$  y  $P(X \leq 102.68) = .62$

EL TEST CHI-CUADRADO SOBRE CHI-CUADRADO HA SIDO APLICADO A 1000 GRUPOS DE VALORES CHI-CUADRADO GENERADOS POR EL TEST CORRID Y SE HAN OBTENIDO LOS SIGUIENTES RESULTADOS:

VALORES CHI-CUADRADO CALCULADOS CADA UNO CON 6GRADOS DE LIBERTAD

TESTS USANDO LA PARTE FRACCIONARIA DE 10 \* NUMERO ALEATORIO

J	PROB. P(J)	CHI-2 P(6)	FREC. ESPERADA	FREC. OBS.	CONTRIBUCION CHI-2
---	---------------	---------------	-------------------	---------------	-----------------------



1	0.01	0.8721	10.00	10	0.00
2	0.02	1.1344	10.00	9	0.10
3	0.03	1.3296	10.00	8	0.40
4	0.04	1.4924	10.00	9	0.10
5	0.05	1.6354	10.00	13	0.90
6	0.06	1.7649	10.00	14	1.60
7	0.07	1.8846	10.00	9	0.10
8	0.08	1.9967	10.00	8	0.40
9	0.09	2.1029	10.00	13	0.90
10	0.10	2.2041	10.00	8	0.40
11	0.11	2.3014	10.00	5	2.50
12	0.12	2.3953	10.00	8	0.40
13	0.13	2.4863	10.00	7	0.90
14	0.14	2.5748	10.00	12	0.40
15	0.15	2.6613	10.00	8	0.40
16	0.16	2.7459	10.00	11	0.10
17	0.17	2.8289	10.00	8	0.40
18	0.18	2.9104	10.00	11	0.10
19	0.19	2.9908	10.00	9	0.10
20	0.20	3.0701	10.00	10	0.00
21	0.21	3.1484	10.00	7	0.90
22	0.22	3.2260	10.00	9	0.10
23	0.23	3.3028	10.00	5	2.50
24	0.24	3.3789	10.00	16	3.60
25	0.25	3.4546	10.00	14	1.60
26	0.26	3.5298	10.00	12	0.40
27	0.27	3.6046	10.00	11	0.10
28	0.28	3.6792	10.00	14	1.60
29	0.29	3.7534	10.00	12	0.40
30	0.30	3.8276	10.00	13	0.90
31	0.31	3.9015	10.00	10	0.00
32	0.32	3.9754	10.00	14	1.60
33	0.33	4.0493	10.00	10	0.00
34	0.34	4.1233	10.00	11	0.10
35	0.35	4.1973	10.00	11	0.10
36	0.36	4.2714	10.00	18	6.40
37	0.37	4.3457	10.00	12	0.40
38	0.38	4.4203	10.00	10	0.00
39	0.39	4.4950	10.00	16	3.60
40	0.40	4.5702	10.00	13	0.90
41	0.41	4.6456	10.00	5	2.50
42	0.42	4.7215	10.00	6	1.60
43	0.43	4.7978	10.00	6	1.60
44	0.44	4.8746	10.00	14	1.60
45	0.45	4.9519	10.00	8	0.40
46	0.46	5.0298	10.00	18	6.40
47	0.47	5.1083	10.00	12	0.40
48	0.48	5.1875	10.00	6	1.60
49	0.49	5.2674	10.00	9	0.10
50	0.50	5.3481	10.00	16	3.60

-----  
TOTAL 1000 1000 93.80  
SI X ES UNA VAR.ALEAT. CHI-2 CON 99 GRADOS DE LIBERTAD, ENTONCES  
P(X<= 93.76)=.37 y P(X<= 94.11)=.38

EL TEST CHI-CUADRADO SOBRE CHI-CUADRADO HA SIDO APLICADO A 1000 GRUPOS DE VALORES CHI-CUADRADO GENERADOS POR EL TEST CORRID Y SE HAN OBTENIDO LOS SIGUIENTES RESULTADOS:

VALORES CHI-CUADRADO CALCULADOS CADA UNO CON 6GRADOS DE LIBERTAD

TESTS USANDO LA PARTE FRACCIONARIA DE 100 \* NUMERO ALEATORIO

J	PROB. P(J)	CHI-2 P( 6)	FREC. ESPERADA	FREC. OBS.	CONTRIBUCION CHI-2
1	0.01	0.8721	10.00	7	0.90

2	0.02	1.1344	10.00	8	0.40
3	0.03	1.3296	10.00	6	1.60
4	0.04	1.4924	10.00	7	0.90
5	0.05	1.6354	10.00	8	0.40
6	0.06	1.7649	10.00	8	0.40
7	0.07	1.8846	10.00	13	0.90
8	0.08	1.9967	10.00	13	0.90
9	0.09	2.1029	10.00	12	0.40
10	0.10	2.2041	10.00	13	0.90
11	0.11	2.3014	10.00	8	0.40
12	0.12	2.3953	10.00	8	0.40
13	0.13	2.4863	10.00	2	6.40
14	0.14	2.5748	10.00	13	0.90
15	0.15	2.6613	10.00	11	0.10
16	0.16	2.7459	10.00	7	0.90
17	0.17	2.8289	10.00	12	0.40
18	0.18	2.9104	10.00	8	0.40
19	0.19	2.9908	10.00	13	0.90
20	0.20	3.0701	10.00	12	0.40
21	0.21	3.1484	10.00	12	0.40
22	0.22	3.2260	10.00	14	1.60
23	0.23	3.3028	10.00	5	2.50
24	0.24	3.3789	10.00	10	0.00
25	0.25	3.4546	10.00	4	3.60
26	0.26	3.5298	10.00	12	0.40
27	0.27	3.6046	10.00	14	1.60
28	0.28	3.6792	10.00	9	0.10
29	0.29	3.7534	10.00	10	0.00
30	0.30	3.8276	10.00	12	0.40
31	0.31	3.9015	10.00	8	0.40
32	0.32	3.9754	10.00	18	6.40
33	0.33	4.0493	10.00	10	0.00
34	0.34	4.1233	10.00	7	0.90
35	0.35	4.1973	10.00	12	0.40
36	0.36	4.2714	10.00	14	1.60
37	0.37	4.3457	10.00	11	0.10
38	0.38	4.4203	10.00	9	0.10
39	0.39	4.4950	10.00	9	0.10
40	0.40	4.5702	10.00	8	0.40
41	0.41	4.6456	10.00	10	0.00
42	0.42	4.7215	10.00	11	0.10
43	0.43	4.7978	10.00	10	0.00
44	0.44	4.8746	10.00	11	0.10
45	0.45	4.9519	10.00	11	0.10
46	0.46	5.0298	10.00	12	0.40
47	0.47	5.1083	10.00	3	4.90
48	0.48	5.1875	10.00	14	1.60
49	0.49	5.2674	10.00	12	0.40
50	0.50	5.3481	10.00	15	2.50

-----  
TOTAL 1000 1000 119.00  
SI X ES UNA VAR.ALEAT. CHI-2 CON 99 GRADOS DE LIBERTAD, ENTONCES  
 $P(X \leq 118.34) = .91$  y  $P(X \leq 119.36) = .92$

\*\*\*\*\*  
TST01  
\*\*\*\*\*

EL TEST CHI-CUADRADO SOBRE CHI-CUADRADO HA SIDO APLICADO A 1000 GRUPOS  
DE VALORES CHI-CUADRADO GENERADOS POR EL TEST PARSER Y SE HAN OBTENIDO  
LOS SIGUIENTES RESULTADOS:

VALORES CHI-CUADRADO CALCULADOS CADA UNO CON 8 GRADOS DE LIBERTAD

TESTS USANDO PARES (Q,R) CON Q y R ENTRE 1 y 3

J	PROB. P(J)	CHI-2 P( 8)	FREC. ESPERADA	FREC. OBS.	CONTRIBUCION CHI-2
1	0.01	1.6465	10.00	9	0.10
2	0.02	2.0325	10.00	7	0.90
3	0.03	2.3101	10.00	8	0.40
4	0.04	2.5366	10.00	9	0.10
5	0.05	2.7326	10.00	12	0.40
6	0.06	2.9080	10.00	11	0.10
7	0.07	3.0683	10.00	9	0.10
8	0.08	3.2172	10.00	9	0.10
9	0.09	3.3570	10.00	11	0.10
10	0.10	3.4895	10.00	10	0.00
11	0.11	3.6160	10.00	16	3.60
12	0.12	3.7375	10.00	10	0.00
13	0.13	3.8546	10.00	13	0.90
14	0.14	3.9680	10.00	8	0.40
15	0.15	4.0782	10.00	7	0.90
16	0.16	4.1856	10.00	10	0.00
17	0.17	4.2906	10.00	10	0.00
18	0.18	4.3934	10.00	4	3.60
19	0.19	4.4943	10.00	10	0.00
20	0.20	4.5936	10.00	11	0.10
21	0.21	4.6913	10.00	12	0.40
22	0.22	4.7878	10.00	14	1.60
23	0.23	4.8830	10.00	7	0.90
24	0.24	4.9773	10.00	13	0.90
25	0.25	5.0706	10.00	9	0.10
26	0.26	5.1632	10.00	9	0.10
27	0.27	5.2551	10.00	12	0.40
28	0.28	5.3463	10.00	12	0.40
29	0.29	5.4371	10.00	14	1.60
30	0.30	5.5274	10.00	10	0.00
31	0.31	5.6174	10.00	18	6.40
32	0.32	5.7071	10.00	12	0.40
33	0.33	5.7966	10.00	14	1.60
34	0.34	5.8860	10.00	12	0.40
35	0.35	5.9753	10.00	13	0.90
36	0.36	6.0646	10.00	13	0.90
37	0.37	6.1539	10.00	9	0.10
38	0.38	6.2433	10.00	8	0.40
39	0.39	6.3329	10.00	9	0.10
40	0.40	6.4226	10.00	6	1.60
41	0.41	6.5127	10.00	9	0.10
42	0.42	6.6031	10.00	16	3.60
43	0.43	6.6938	10.00	9	0.10
44	0.44	6.7850	10.00	12	0.40
45	0.45	6.8766	10.00	9	0.10
46	0.46	6.9688	10.00	9	0.10
47	0.47	7.0616	10.00	7	0.90
48	0.48	7.1551	10.00	11	0.10
49	0.49	7.2492	10.00	6	1.60
50	0.50	7.3441	10.00	11	0.10

-----  
TOTAL 1000 1000 77.00  
SI X ES UNA VAR.ALEAT. CHI-2 CON 99 GRADOS DE LIBERTAD, ENTONCES  
 $P(X \leq 75.79) = .04$  y  $P(X \leq 77.05) = .05$

EL TEST CHI-CUADRADO SOBRE CHI-CUADRADO HA SIDO APLICADO A 1000 GRUPOS  
DE VALORES CHI-CUADRADO GENERADOS POR EL TEST PARSEY Y SE HAN OBTENIDO  
LOS SIGUIENTES RESULTADOS:

VALORES CHI-CUADRADO CALCULADOS CADA UNO CON 99 GRADOS DE LIBERTAD

TESTS USANDO PARES (Q,R) CON Q y R ENTRE 1 y 10

J	PROB. P(J)	CHI-2 P( 99)	FREC. ESPERADA	FREC. OBS.	CONTRIBUCION CHI-2
1	0.01	69.2299	10.00	15	2.50
2	0.02	72.2880	10.00	13	0.90
3	0.03	74.2754	10.00	13	0.90
4	0.04	75.7949	10.00	10	0.00
5	0.05	77.0463	10.00	13	0.90
6	0.06	78.1226	10.00	11	0.10
7	0.07	79.0746	10.00	13	0.90
8	0.08	79.9336	10.00	11	0.10
9	0.09	80.7204	10.00	13	0.90
10	0.10	81.4493	10.00	16	3.60
11	0.11	82.1306	10.00	8	0.40
12	0.12	82.7724	10.00	16	3.60
13	0.13	83.3805	10.00	14	1.60
14	0.14	83.9599	10.00	9	0.10
15	0.15	84.5143	10.00	11	0.10
16	0.16	85.0469	10.00	10	0.00
17	0.17	85.5603	10.00	11	0.10
18	0.18	86.0566	10.00	7	0.90
19	0.19	86.5377	10.00	7	0.90
20	0.20	87.0052	10.00	4	3.60
21	0.21	87.4605	10.00	10	0.00
22	0.22	87.9048	10.00	14	1.60
23	0.23	88.3390	10.00	9	0.10
24	0.24	88.7642	10.00	5	2.50
25	0.25	89.1812	10.00	8	0.40
26	0.26	89.5907	10.00	8	0.40
27	0.27	89.9934	10.00	15	2.50
28	0.28	90.3898	10.00	10	0.00
29	0.29	90.7806	10.00	10	0.00
30	0.30	91.1663	10.00	8	0.40
31	0.31	91.5472	10.00	6	1.60
32	0.32	91.9238	10.00	7	0.90
33	0.33	92.2966	10.00	8	0.40
34	0.34	92.6659	10.00	12	0.40
35	0.35	93.0320	10.00	8	0.40
36	0.36	93.3953	10.00	10	0.00
37	0.37	93.7560	10.00	6	1.60
38	0.38	94.1145	10.00	10	0.00
39	0.39	94.4710	10.00	6	1.60
40	0.40	94.8259	10.00	7	0.90
41	0.41	95.1793	10.00	15	2.50
42	0.42	95.5315	10.00	9	0.10
43	0.43	95.8828	10.00	17	4.90
44	0.44	96.2333	10.00	8	0.40
45	0.45	96.5834	10.00	10	0.00
46	0.46	96.9332	10.00	17	4.90
47	0.47	97.2829	10.00	6	1.60
48	0.48	97.6329	10.00	8	0.40
49	0.49	97.9832	10.00	7	0.90
50	0.50	98.3341	10.00	5	2.50

TOTAL 1000 1000 103.00  
SI X ES UNA VAR.ALEAT. CHI-2 CON 99 GRADOS DE LIBERTAD, ENTONCES  
 $P(X \leq 102.68) = .62$  y  $P(X \leq 103.06) = .63$

EL TEST CHI-CUADRADO SOBRE CHI-CUADRADO HA SIDO APLICADO A 1000 GRUPOS  
DE VALORES CHI-CUADRADO GENERADOS POR EL TEST PARSER Y SE HAN OBTENIDO  
LOS SIGUIENTES RESULTADOS:

VALORES CHI-CUADRADO CALCULADOS CADA UNO CON 399GRADOS DE LIBERTAD

TESTS USANDO PARES (Q,R) CON Q y R ENTRE 1 y 20

J	PROB. P(J)	CHI-2 P( 399)	FREC. ESPERADA	FREC. OBS.	CONTRIBUCION CHI-2
1	0.01	336.2376	10.00	9	0.10
2	0.02	343.1507	10.00	9	0.10
3	0.03	347.5863	10.00	10	0.00
4	0.04	350.9483	10.00	13	0.90
5	0.05	353.6992	10.00	9	0.10
6	0.06	356.0520	10.00	7	0.90
7	0.07	358.1237	10.00	11	0.10
8	0.08	359.9854	10.00	5	2.50
9	0.09	361.6843	10.00	9	0.10
10	0.10	363.2528	10.00	15	2.50
11	0.11	364.7147	10.00	6	1.60
12	0.12	366.0876	10.00	13	0.90
13	0.13	367.3853	10.00	14	1.60
14	0.14	368.6184	10.00	9	0.10
15	0.15	369.7956	10.00	10	0.00
16	0.16	370.9239	10.00	9	0.10
17	0.17	372.0091	10.00	10	0.00
18	0.18	373.0562	10.00	9	0.10
19	0.19	374.0692	10.00	7	0.90
20	0.20	375.0516	10.00	14	1.60
21	0.21	376.0066	10.00	10	0.00
22	0.22	376.9368	10.00	4	3.60
23	0.23	377.8446	10.00	10	0.00
24	0.24	378.7319	10.00	14	1.60
25	0.25	379.6006	10.00	7	0.90
26	0.26	380.4523	10.00	8	0.40
27	0.27	381.2885	10.00	22	14.40
28	0.28	382.1106	10.00	15	2.50
29	0.29	382.9197	10.00	6	1.60
30	0.30	383.7170	10.00	8	0.40
31	0.31	384.5034	10.00	8	0.40
32	0.32	385.2798	10.00	11	0.10
33	0.33	386.0472	10.00	9	0.10
34	0.34	386.8064	10.00	4	3.60
35	0.35	387.5580	10.00	11	0.10
36	0.36	388.3028	10.00	4	3.60
37	0.37	389.0414	10.00	17	4.90
38	0.38	389.7745	10.00	9	0.10
39	0.39	390.5027	10.00	7	0.90
40	0.40	391.2265	10.00	7	0.90
41	0.41	391.9464	10.00	8	0.40
42	0.42	392.6630	10.00	15	2.50
43	0.43	393.3767	10.00	8	0.40
44	0.44	394.0882	10.00	18	6.40
45	0.45	394.7977	10.00	9	0.10
46	0.46	395.5059	10.00	11	0.10
47	0.47	396.2131	10.00	9	0.10
48	0.48	396.9198	10.00	10	0.00
49	0.49	397.6265	10.00	6	1.60
50	0.50	398.3335	10.00	7	0.90

-----  
TOTAL 1000 1000 117.00  
SI X ES UNA VAR.ALEAT. CHI-2 CON 99 GRADOS DE LIBERTAD, ENTONCES  
P(X<= 116.54)=.89 y P(X<= 117.41)=.90

\*\*\*\*\*  
TST01  
\*\*\*\*\*

EL TEST KOLMOGOROV-SMIRNOV SOBRE KOLMOGOROV-SMIRNOV FUE APLICADO A 10000 VALORES K-S GENERADOS POR LOS VALORES DE K-S EN EL TEST DE BONDAD DE AJUSTE Y SE HAN PRODUCIDO LOS SIGUIENTES RESULTADOS:

	K+	K-	K
VALORES DE K-S SOBRE K+:	0.855	0.500	0.855
VALORES DE K-S SOBRE K-:	1.39	0.209	1.39
VALORES DE K-S SOBRE K:	1.48	0.314	1.48

AQUI LA SIGNIFICACION PUEDE CALCULARSE TENIENDO EN CUENTA LAS DISTRIBUCIONES DE K+, K-, K, Y USANDO LOS VALORES:

DE K-S SOBRE K+ :	DE K-S SOBRE K-:	DE K-S SOBRE K :
P(K+ <=.855 )=.768	, P(K- <=.500 ) =.393	, P(K<=.855 )=.542
P(K+ <=1.39 )=.978	, P(K- <=.209 ) =.838E-01,	P(K<=1.39 )=.957
P(K+ <=1.48 )=.988	, P(K- <=.314 ) =.179	, P(K<=1.48 )=.975

\*\*\*\*\*  
TST01  
\*\*\*\*\*

EL TEST KOLMOGOROV-SMIRNOV SOBRE KOLMOGOROV-SMIRNOV FUE APLICADO A 10000 VALORES GENERADOS POR LOS VALORES K-S EN EL TEST DE LOS MAXIMOS Y SE HAN PRODUCIDO LOS SIGUIENTES RESULTADOS:

	K+	K-	K
VALORES DE K-S SOBRE K+:	0.878	0.148	0.878
VALORES DE K-S SOBRE K-:	1.80	0.144	1.80
VALORES DE K-S SOBRE K:	1.95	0.119	1.95

AQUI LA SIGNIFICACION PUEDE CALCULARSE TENIENDO EN CUENTA LAS DISTRIBUCIONES DE K+, K-, K, Y USANDO LOS VALORES:

DE K-S SOBRE K+ :	DE K-S SOBRE K-:	DE K-S SOBRE K :
P(K+ <=.878 )=.786	, P(K- <=.148 ) =.428E-01,	P(K<=.878 )=.576
P(K+ <=1.80 )=.998	, P(K- <=.144 ) =.409E-01,	P(K<=1.80 )=.997
P(K+ <=1.95 )=.999	, P(K- <=.119 ) =.278E-01,	P(K<=1.95 )=.999

\*\*\*\*\*

LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO (EL SIGUIENTE A LOS 10000000 YA GENERADOS) ES:

INTIN(1)= 524287 INTIN(2)= 38888  
 INTIN(3)= 5631 INTIN(4)= 20251  
 INTIN(5)= 24173

SE HAN LEIDO Y PROCESADO TODOS LOS DATOS SUMINISTRADOS POR LOS ARCHIVOS DE ENTRADA Y, POR LO TANTO, LA EJECUCION HA FINALIZADO.

\*\*\*\*\*  
 G E N E R A D O R: URN13  
 \*\*\*\*\*

T E S T	T(I)	P[CHI-CUAD<=T(I)]
DISTR.UNIF.(TST02)	92.40	0.33
CUPONES D=5(TST04)	87.20	0.20
CUPONES FR(100R)	95.40	0.41
CUPONES D=10	98.40	0.50
GAP ARRIBA MEDIA(TST05)	82.60	0.11
GAP BAJO MEDIA	112.60	0.83
GAP (0.33,0.667)	102.40	0.61
PERMUTACIONES 3S(TST07)	91.40	0.30
PERMUTACIONES 4S	99.40	0.53
PERMUTACIONES 5S	113.40	0.84
POKER (4,4) (TST08)	107.80	0.74
POKER (5,6)	96.00	0.43
POKER (5,4)	90.40	0.28
CORRIDAS R(TST09)	102.60	0.61
CORRIDAS FR(10R)	93.80	0.37
CORRIDAS FR(100R)	119.00	0.92
PARES SERIALES 3X3(TST10)	77.00	0.04
PARES SERIALES 10X10	103.00	0.62
PARES SERIALES 20X20	117.00	0.89
<hr/>		
CANTIDAD DE G`= P[CHI-CUAD<= T(I)]>= 0.99		0
CANTIDAD DE G`= P[CHI-CUAD<= T(I)]>= 0.95		0
CANTIDAD DE G`= P[CHI-CUAD<= T(I)]>= 0.50		10

**PAQUETE DE PROGRAMAS GENTSTPC**

**CODIGOS FUENTE FORTRAN77/90**

**AUTOR: ANGEL ADOLFO OLMOS CERVANTES**

**Tesis de Doctorado en Ingeniería (Orientación  
Investigación de Operaciones) - F.I.- UNAM -2005**

**"Generación de números pseudoaleatorios eficientes  
en microcomputadores"(ANEXO II)**



## Capítulo 7

### CODIGOS FUENTE GENTSTPC1 Y GENTSTPC2

```
PROGRAM GENTSTPC1
C   Gentstpc1 corre en PC con sistemas operativos WINDOWS 9*
C   COMPILADOR:Micro Soft Developer Studio-FORTRAN POWER STATION 4.0
C   Autor: ANGEL ADOLFO OLMOS CERVANTES - 2005

C

      CHARACTER*5 GNOMBRE
      CHARACTER*4 TNOMBRE,RESTO4
      CHARACTER*3 RTN
      CHARACTER*2 RESTO2
      CHARACTER*1 RTN1,TSTM,TEST01,TST0
      CHARACTER*6 RESTO6
      CHARACTER*7 RESTO7
      CHARACTER*41 RESTO41

C
      OPEN(UNIT=20,FILE='ARCH',ACCESS='SEQUENTIAL',RECL=80)
      OPEN(UNIT=60,ACCESS='DIRECT',RECL=1,FORM='FORMATTED',FILE=
* 'OPCION')
      OPEN(UNIT=90,RECL=11,FORM='FORMATTED',FILE='ARCHDAT')
C      OPEN(UNIT=90,ACCESS='DIRECT',RECL=11,FORM='FORMATTED',FILE=
C      * 'ARCHDAT')
      OPEN(UNIT=45,ACCESS='DIRECT',RECL=2,FORM='FORMATTED',FILE=
* 'INDICE')
      READ(45,9088)ISUPE
9088  FORMAT(I2)
      CLOSE(45)
      TSTM='0'
      TEST01='0'
      READ(20,900)GNOMBRE
      900  FORMAT(A5)
      WRITE(*,9001)GNOMBRE
9001  FORMAT(1X,A5)
      IF(ISUPE.GT.1)THEN
      DO 9099 I=1,ISUPE-1
      READ(20,89)TNOMBRE,RTN1,RTN,RESTO41,RESTO7,RESTO4,RESTO2,RESTO6
      IF(TNOMBRE.EQ.'TSTM')TSTM='1'
      IF((TNOMBRE.EQ.'TST0').AND.(RTN1.EQ.'1')) THEN
      IF((RESTO7.EQ.'111111').AND.(RESTO2.EQ.'11')) THEN
      TEST01 ='1'
      ENDIF
      ENDIF
9099  CONTINUE
      ENDIF
89   FORMAT(A4,A1,A3,A41,A7,A4,A2,A6)
      CLOSE(20)
      OPEN(UNIT=20,FILE='ARCH',ACCESS='SEQUENTIAL',RECL=80)
      WRITE(60,11)'N'
      WRITE(60,11)'1'
      CLOSE(60)
11   FORMAT(1A)
C
C*****
```

```

C
    CALL TESTER
C
C*****
C
    CLOSE(20)
    CLOSE(90)
    IF(TEST01.EQ.'0')THEN
    CLOSE(96,STATUS='DELETE')
    OPEN(UNIT=65,FILE='ENTROPIA.DAT',ACCESS='SEQUENTIAL',
*FORM='FORMATTED')
    READ(65,12) ENTRVAL
12  FORMAT(A1)
    IF(ENTRVAL.EQ.'1')THEN
    CALL TSTENTR
    ENDIF
    CLOSE(65)
C
    WRITE(*,*)' '
    WRITE(6,516)
516  FORMAT(1X,' SE HAN LEIDO Y PROCESADO TODOS LOS DATOS ',
* 'SUMINISTRADOS POR LOS ARCHIVOS DE')
    WRITE(6,5162)
5162  FORMAT(1X,' ENTRADA Y, POR LO TANTO, LA EJECUCION HA
FINALIZADO.')
    STOP
    ENDIF
C
    CALL PRINTSTS(GNOMBRE)
    OPEN(UNIT=65,FILE='ENTROPIA.DAT',ACCESS='SEQUENTIAL',FORM=
*'FORMATTED')
    READ(65,12) ENTRVAL
    IF(ENTRVAL.EQ.'1')THEN
    CALL TSTENTR
    ENDIF
    CLOSE(65)
C
    WRITE(*,*)' '
    WRITE(6,516)
    END

C
C*****
C
    SUBROUTINA TESTER
C*****
C "TESTER" ES LA RUTINA USADA POR GENTSTPC PARA VERIFICAR LA CALIDAD
C ESTADISTICA DE LOS NUMEROS PSEUDOALEATORIOS GENERADOS.
C TESTER PUEDE SER APLICADA A UNO DE LOS GENERADORES PROVISTOS POR
C GENTSTPC (URN01-URN40) O AL PROPIO GENERADOR (SUBROUTINA "URNWN")
C LAS RUTINAS PUEDEN SER LLAMADAS EN FORMA INDIVIDUAL, POR LO QUE EL
C USUARIO PUEDE USARLAS DE DISTINTAS FORMAS NO ESTANDAR CON UN MINIMO
C DE REPROGRAMACION. PARA ESTE PROPOSITO, LOS BLOQUES "COMMON" DEBEN
C USARSE COMO SE ESPECIFICA.
C PROPOSITO:
C   GENERACION Y VERIFICACION DE NUMEROS ALEATORIOS
C USO:
C   PARA USAR LAS FACILIDADES PROVISTAS POR LA SUBROUTINA "TESTER", EL
C   PROGRAMA PRINCIPAL DEBE CONTENER LA INSTRUCCION "CALL TESTER".
C NOTAS:
C   ESTA SUBROUTINA, ENTRE OTRAS, UTILIZA EL BLOQUE "COMMON" PARA

```

```

C PASAR INFORMACION ENTRE RUTINAS. LOS NOMBRES DE LOS BLOQUES
USADOS
C NO DEBEN SER UTILIZADOS POR EL USUARIO EN SU PROGRAMA PRINCIPAL
C A NO SER QUE LLAME A ESTAS RUTINAS DIRECTAMENTE PARA OBTENER
C TESTS NO ESTANDAR, EN CUYO CASO LA RUTINA QUE LLAMA DEBE USAR
C BLOQUES "COMMON" PARA PASAR INFORMACION. LOS NOMBRES DE BLOQUES
COMMON RESERVADOS Y LAS RUTINAS EN QUE SE EMPLEAN SON:
C MESS TESTER, REPORT, TST02, TST03, TST04, TST05, TST06,
C TST07, TST08, TST09, TST10, TST11, TSTM1, TSTM2,TSTM3
C FIRST TST01, PNTCHI, POINTS, TST05
C SECOND TESTER, DBLPRE
C THIRD TESTER, EXTPRE
C FOURTH TESTER, REPORT, TST02, TST03, TST04, TST05, TST06,
C TST07, TST08, TST09, TST10, TST11, DBLPRE, EXTPRE
C FIFTH TESTER, REPORT, RRNWN, RRN01, RRN02, RRN03, RRN04,
C RRN05, RRN06, RRN07, RRN08, RRN09, RRN10, RRN11,
C RRN12, RRN13, RRN14, RRN15, RRN16, RRN17, RRN18,
C RRN19, RRN20, URN05
C SIXTH REPORT, TST01, TST02, TST04, TST05, TST06, TST07,
C TST08, TST09, TST10, TST11
C URNPRM TESTER, REPORT, TST01, TST02, TST03, TST04, TST05,
C TST06, TST07, TST08, TST09, TST10, TST11, RRNWN,
C RRN01, RRN02, RRN03, RRN04, RRN05, RRN06, RRN07,
C RRN08, RRN09, RRN10, RRN11, RRN12, RRN13, RRN14,
C RRN15, RRN16, RRN17, RRN18, RRN19, RRN20, URNWN1,
C URN01I, URN02I, URN03I, URN04I, URN05,URN05I,URN12I
C T01PRM TESTER, REPORT, TST01, TST02, TST04, TST05, TST06,
C TST07, TST08, TST09, TST10, TST11
C OWNPRM TESTER, REPORT, RRNWN, URNWN, URNWN1
C COUNTS TESTER, REPORT, TST01, TST02, TST04, TST05, TST06,
C TST07, TST08, TST09, TST10, TST11, RRNWN, RRN01,
C RRN02, RRN03, RRN04, RRN05, RRN06, RRN07, RRN08,
C RRN09, RRN10, RRN11, RRN12, RRN13, RRN14, RRN15,
C RRN16, RRN17, RRN18, RRN19, RRN20
C RPRMS TESTER, REPORT, RRN11, URN06I
C RPYMDBL TESTER, REPORT
C IPRMS TESTER, REPORT, RRN01, RRN03, RRN04, RRN06, RRN07,
C RRN08, RRN09, RRN10, RRN11, RRN13, RRN14, RRN15,
C RRN16, RRN17, RRN18, RRN19, RRN20, URN01I, URN03I,
C URN04I
C GENARR TESTER, REPORT, URN04I
C CODE TESTER, REPORT, TST01, TST02, TST03, TST04, TST05,
C TST06, TST07, TST08, TST09, TST10, TST11
C TSTPRM TESTER, REPORT, TST03, TST04, TST05, TST06, TST07,
C TST08, TST09, TST10, TST11, TSTM1, TSTM2
C T01VAL REPORT, TST01, TST04, TST05, TST06, TST07,TST08,TST11
C SUBRUTINAS Y FUNCIONES REQUERIDAS:
C TST01, TST02, TST03, TST04, TST05, TST06, TST07, TST08, TST09,
C TST10, TST11, TSTM1, TSTM2, TSTM3, TSTM4, RRN01, RRN02, RRN03,
C RRN04, RRN05, RRN06, RRN07, RRN08, RRN09, RRN10, RRN11, RRN12,
C RRN13, RRN14, RRN15, RRN16, RRN17, RRN18, RRN19, RRN20, RRNWN,
C DBLPRE, EXTPRE, REPORT
C REFERENCIA:
C 1. DUDEWICZ, E.J. AND RALLEY, T.: "THE HANDBOOK OF RANDOM NUMBER
C GENERATION AND TESTING", AMERICAN SCIENCES PRESS,
C COLUMBUS, OHIO, 1981.
C*****
C
C SUBROUTINE TESTER
CHARACTER*8 NAME, MESSG, VARIAB, BL, LASTGN, URNM(41), TNAME(15),
*TITLE, CKNAME

```

```

CHARACTER*2 TIME
CHARACTER*3 TIMES,YES,TNO
REAL*8 XEXT
REAL*8 XDBL
REAL*8 C2,ZZ2,SEED
INTEGER DBLPRT,EXTPRT,SWCHI,GCODE
COMMON /MESS/MESSG(9),VARIAB(9)
COMMON /SECOND/XDBL(2500)
COMMON /THIRD/XEXT(1250)
COMMON /FOURTH/Y(10000)
COMMON /FIFTH/X(10000)
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /T01PRM/NCHI,NPRNT
COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /CODE/GCODE
COMMON /RPRMS/C,P,Q,ZZ
COMMON /RPMDBL/C2,ZZ2
COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKK,L,ML,MK,MMM
COMMON /TSTPRM/IPARM(5)
COMMON /GENARR/MFSR(98),NN(128),ITABLE(128)
COMMON /COMUN30/ISTRM,IZ
COMMON /COMUN3132/IXX
COMMON /COMUN33/XI
COMMON /COMUN3435/KW1,KY1,KZ1,XR1
COMMON /COMUN39/SEED,ITOTAL39
DIMENSION INTDIV(31,2)

C
C      EL VECTOR TNAME(15) CONTIENE ALFANUMERICOS QUE SON LOS
NOMBRES
C      LOS TESTS. LOS TESTS ASI IDENTIFICADOS SON:
C      TST01...CHI-CUADRADO SOBRE CHI-CUADRADO
C      TST02...DISTRIBUCION UNIFORME
C      TST03...TEST DEL PRODUCTO REZAGADO (LAGGED CORRELATION)
C      TST04...RECOLECTOR DE CUPONES
C      TST05...GAP Y CORRIDAS DEBAJO/SOBRE DE LA MEDIA
C      TST06...KOLMOGOROV-SMIRNOV
C      TST07...PERMUTACION
C      TST08...POKER
C      TST09...TEST DE LAS CORRIDAS (RUNS UP)
C      TST10...PARES SERIALES
C      TST11...MAX DE T
C      TSTM1...CORRELACION SERIAL
C      TSTM2...POTENCIA
C      TSTM3...FACTORES PRIMOS
C      TSTM4...ESPECTRAL
C
DATA TNAME /'TST01','TST02','TST03','TST04','TST05','TST06',
*'TST07','TST08','TST09','TST10','TST11','TSTM1','TSTM2','TSTM3',
*'TSTM4'/
DATA TITLE/'TITLE'/BL/'      '/

C
C      EL VECTOR URNM(41) CONTIENE CARACTERES ALFANUMERICOS QUE SON
NOMBRES
C      DE GENERADORES. LOS GENERADORES SE DENOMINAN ASI:
C      URN01...GENERADOR BASICO LLRANDOM ("RANDOM" EN IRCRAND)
C      URN02...GENERADOR MARSAGLIA-BRAY ("RN3" EN IRCRAND)
C      URN03...GENERADOR SUPER-DUPER DE MCGILL ("UNI" EN IRCRAND)
C      URN04...GENERADOR SHUFFLED LLRANDOM ("SRAND" EN IRCRAND)
C      URN05...GENERADOR GFSR DE LEWIS-PAYNE (NO ESTA EN IRCRAND)

```

```

C      URN06...GENERADOR LURIE-MASON ("RN1" EN IRCCRAND)
C      URN07...RANDU EN ASSEMBLER ("KERAND" EN IRCCRAND)
C      URN08...RANDU COMO EN SSP DE IBM ("RN2" EN IRCCRAND)
C      URN09...RANDU COMO EN SPSS ("RN4" EN IRCCRAND)
C      URN10...GENERADOR OMNITAB 2 DE KRUSKAL ("RN5" EN IRCCRAND)
C      URN11...GENERADOR CONGRUENCIAL MIXTO CON PARAMETROS DEL
C      USUARIO ("RNCG" EN IRCCRAND)
C      URN12...UNO MAS PRECISO, DOBLE PRECISION INTERNA, URN11
C      (NO ESTA EN IRCCRAND)
C      URN13...GENERADOR AHRENS-DIETER (NO ESTA EN IRCCRAND)
C      URN14...GENERADOR ZARLING
C      URN15...GENERADOR HOAGLIN
C      URN16...GENERADOR HOAGLIN
C      URN17...GENERADOR HOAGLIN
C      URN18...GENERADOR HOAGLIN
C      URN19...GENERADOR HOAGLIN
C      URN20...GENERADOR HOAGLIN
C      URN21...GENERADOR G. Y M. SWAIN (1980)
C      URN22...GENERADOR VAX-11/780
C      URN23...GENERADOR BRODY (1984)
C      URN24...GENERADOR THESEN (1985)
C      URN25...GENERADOR GAIT (1977)
C      URN26...GENERADOR MATHEWS (1986)
C      URN27...GENERADOR HENRIKSEN-CRANE (1983)
C      URN28...GENERADOR ANALOGO URN01
C      URN29...GENERADOR ANALOGO URN01
C      URN30...GENERADOR SIMSCRIPT ,WEST-JOHNSON (1984)
C      URN31...GENERADOR MIYAZAKI (1987)
C      URN32...GENERADOR MIYAZAKI (1987)
C      URN33...GENERADOR MIYAZAKI (1987)
C      URN34...GENERADOR SEZGIN (1987)
C      URN35...GENERADOR L' ECUYER (1987)
C      URN36...GENERADOR JENNERGREN (1984)
C      URN37...GENERADOR DUDEWICZ-KARIAN-MARSHALL (1985)
C      URN38...GENERADOR JENNERGREN
C      URN39...GENERADOR REY (1990)
C      URN40...GENERADOR MARSAGLIA Y ZAMAN
C      URNWN...UN NOMBRE RESERVADO PARA UN GENERADOR SUMINISTRADO
C      POR EL USUARIO

```

```

C      DATA URNM/'URN01','URN02','URN03','URN04','URN05','URN06',
C      *'URN07','URN08','URN09','URN10','URN11','URN12','URN13','URN14',
C      *'URN15','URN16','URN17','URN18','URN19','URN20','URN21','URN22',
C      *'URN23','URN24','URN25','URN26','URN27','URN28','URN29','URN30',
C      *'URN31','URN32','URN33','URN34','URN35','URN36','URN37','URN38',
C      *'URN39','URN40','URNWN'/
C      DATA YES/'YES'/TNO/' NO'/

```

```

C
C      EL SIGUIENTE ENCABEZAMIENTO SIRVE PARA LA GENERACION
C      DE NUMEROS Y RUTINAS DE TESTS CADA VEZ QUE SE HACE UNA NUEVA
C      LLAMADA A TESTER

```

```

C      WRITE(6,70)
70      FORMAT(1X,
C      '*****
C      *****')
C      WRITE(6,73)
73      FORMAT(1X,'*',2X,'LAS SIGUIENTES SALIDAS HAN SIDO PRODUCIDAS
POR
*EL SISTEMA: " GENTSTPC "' ,2X,'*')
C      WRITE(6,75)

```

```

75     FORMAT(1X,'*',2X,'EL PAQUETE ESTA CONSTITUIDO POR RUTINAS PARA
LA
      * GENERACION DE NUMEROS      ','*')
      WRITE(6,74)
74     FORMAT(1X,'*',2X,'PSEUDOALEATORIOS Y TESTS ESTADISTICOS
PROGRAMADO
      *S PARA MICROCOMPUTADORES.','*')
      WRITE(6,87)
      WRITE(6,87)
87
FORMAT(1X,'*****
*****')
C     EL SIGUIENTE CODIGO INICIALIZA EL VECTOR DE MENSAJES CON BLANCOS
C     CADA VEZ QUE LA RUTINA TESTER ES LLAMADA E INICIALIZA EL
CONTADOR
C     GENCODE
C     COMIENZO SUBRUTINA TESTER:
      DO 10 I=1,9
      MESSG(I)=BL
10    CONTINUE
      GCODE=0
C
      OPEN(UNIT=45,ACCESS='DIRECT',RECL=2,FORM='FORMATTED',FILE=
*'INDICE')
      READ(45,9088)INDEXSUP
9088  FORMAT(I2)
      CLOSE(45)
      DO 631 INDEX=1,INDEXSUP
7899  FORMAT(1X,'INDEX= ',I3)
C
      READ(20,640)NAME,VARIAB
640   FORMAT(10A8)
      WRITE(*,641)NAME
641   FORMAT(1X,A8)
      DO 20 KSUB=1,41
      IF(NAME.EQ.URNM(KSUB))GO TO 800
20    CONTINUE
C
C     SE EVALUA SI ES TST01 O TSTM4. ESTAS DOS SUBRUTINAS HACEN SU PASE
C     DE INFORMACION EN "VARIAB" Y POR LO TANTO NECESITAN SER TRATADAS
C     SEPARADAMENTE DE LOS OTROS TESTS.
C
      IF(NAME.EQ.TNAME(1)) GOTO 910
      IF(NAME.EQ.TNAME(15)) GOTO 559
C
      IF(NAME.NE.TITLE)GO TO 40
      DO 30 I=1,9
      MESSG(I)=VARIAB(I)
30    CONTINUE
      GO TO 631
C
40    DO 45 J=2,14
      IF(NAME.EQ.TNAME(J))GO TO 902
45    CONTINUE
C
C     SI LLEGAMOS A ESTE PUNTO HA OCURRIDO UN ERROR EN LOS DATOS DE
ENTRADA
C     (COLUMNAS 1-8)
C
C     WRITE(*,539)NAME

```

```

C 539  FORMAT(1X,A8)
      WRITE(6,537)
      537 FORMAT(1X,' ERROR: LA SIGUIENTE ENTRADA ES INCORRECTA...')
C      WRITE(6,538)NAME,VARIAB
C 538  FORMAT(' ',2A8)
      WRITE(6,525)
      525 FORMAT(' ',/1X,' LA EJECUCION CONTINUA A PARTIR DEL SIGUIENTE
PUNTO
      *EN LA CORRIDA')
      GO TO 631

C
C  EL SIGUIENTE CODIGO PROCESA LOS VALORES DE ENTRADA PARA TODOS LOS
C  TESTS EXCEPTO TST01 Y TSTM4 (LOS CUALES HAN SIDO MANEJADO EN FORMA
C  SEPARADA ANTERIORMENTE)
C
      902 CONTINUE
      540 CONTINUE
      542 GO
TO(910,546,547,548,549,550,551,552,553,554,555,556,557,558,559)
      * ,J
      546 CALL TST02
      GO TO 560
      547 CALL TST03
      GO TO 560
      548 CALL TST04
      GO TO 560
      549 CALL TST05
      GO TO 560
      550 CALL TST06
      DO 801 J1=1,10000
      801 Y(J1)=X(J1)
      GO TO 560
      551 CALL TST07
      GO TO 560
      552 CALL TST08
      GO TO 560
      553 CALL TST09
      GO TO 560
      554 CALL TST10
      GO TO 560
      555 CALL TST11
      DO 802 J1=1,10000
      802 Y(J1)=X(J1)
      GOTO 560
      556 CALL TSTM1
      GO TO 560
      557 CALL TSTM2
      GO TO 560
      558 CALL TSTM3(IPARM(1),INTDIV,INDX,ISW)
      GO TO 560
      559 CALL TSTM4
      560 GOTO 631

C
C  EL SIGUIENTE CODIGO PROCESA DATOS DE ENTRADA PARA TST01 Y VERIFICA
C  QUE TST01 NO SE INTENTE REALIZAR MAS DE UNA VEZ
C
      910 IF(SWCHI.EQ.0) GOTO 911
      WRITE(6,903)
      903 FORMAT(1X,'TST01 PUEDE USARSE SOLO UNA VEZ CON UN GENERADOR.',
*/1X,' TST01 SE IGNORA.')
      GOTO 631

```

```

C
  911 SWCHI=1
C*****
  CALL REPORT(ISUB)
C
C   IMPRIME LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO
DESPUES
C   DE GENERAR NUMEROS PARA TST01
C
  NEXTRN=NBATCH*NCHI
  WRITE(*,*)'NEXTRN= NBATCH*NCHI',NEXTRN
C
  IF(LASTGN.EQ.URNM(1)
*.OR.LASTGN.EQ.URNM(7).OR.LASTGN.EQ.URNM(8).OR.LASTGN.EQ.URNM(9)
*.OR.LASTGN.EQ.URNM(10).OR.LASTGN.EQ.URNM(15).OR.LASTGN.EQ.URNM(16)
*.OR.LASTGN.EQ.URNM(17).OR.LASTGN.EQ.URNM(18).OR.LASTGN.EQ.URNM(19)
*.OR.LASTGN.EQ.URNM(20).OR.LASTGN.EQ.URNM(28)) THEN
  WRITE(*,*)' '
  WRITE(6,430)NEXTRN
  430 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO
(EL
  * SIGUIENTE A LOS ',I11)
  WRITE(6,4431)IX
  4431 FORMAT(1X,'YA GENERADOS) ES:', ' IX=',I11)
  ENDIF
C
  WRITE(*,87)
  IF(LASTGN.EQ.URNM(2))WRITE(6,434)NEXTRN,(I,NN(I),I=1,128),ML,MM,
*MK,L,MMM,KKK
  434 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO ',
*' (EL SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES:',
*/,/,25(5(' NN(',I3,')=' ,Z8),/),3(' NN(',I3,')=' ,Z8),
*/, ' ML=',I11, ' , MM=',I11,
*' , MK=',I11, ' , L=',I11, ' , MMM=',I11, ' , KKK=',I11)
C
  IF(LASTGN.EQ.URNM(3).OR.LASTGN.EQ.URNM(6).OR.LASTGN.EQ.URNM(14))
*WRITE(6,431)NEXTRN,IX,JX
  431 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO ',
*' (EL SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES:',
*/, ' IX=',I11,5X, ' , JX=',I11)
C
  IF(LASTGN.EQ.URNM(4))WRITE(6,437)NEXTRN,IX,(I,ITABLE(I),I=1,128)
  437 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO ',
*' (EL SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES:',
*/,/, ' IX=',I11,
*/,25(5(' ITABLE(',I3,')=' ,Z8),/),3(' ITABLE(',I3,')=' ,Z8))
C
  IF(LASTGN.EQ.URNM(5))WRITE(6,435)NEXTRN,(I,MFSR(I),I=1,98),P,Q,
*INTSIZ,JJ
  435 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO ',
*' (EL SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES:',
*/,/,16(6(' MFSR(',I2,')=' ,I11),/),2(' MFSR(',I2,')=' ,I11),
*/, ' P=' ,F11.0, ' , Q=' ,F11.0, ' , INTSIZ=' ,I11, ' , JJ=' ,I11)
C
  IF(LASTGN.EQ.URNM(11)) WRITE(6,432)NEXTRN,IX,L,C,ZZ
  432 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO ',
*' (EL SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES:',
*/, ' IX=' ,I11, ' , L=' ,I11, ' , C=' ,F16.1, ' , ZZ=' ,F16.1)
C

```



```

      IF(LASTGN.EQ.URNM(12)) WRITE(6,433)NEXTRN,IX,L,C2,ZZ2
433 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO ',
      *(EL SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES:',
      */, ' IX=',I11,' , L=',I11,' , C=',F16.1,' , ZZ=',F16.1)
C
      IF(LASTGN.EQ.URNM(21))WRITE(6,421)NEXTRN,M1,M2,M3
421 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO ',
      *(EL SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES:',
      */, ' M1 =',I11,2X,' , M2 =',I11,2X,' , M3 =',I11)
C
      IF(LASTGN.EQ.URNM(22))WRITE(6,422)NEXTRN,M1
422 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO ',
      *(EL SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES:',
      */, ' M1 =',I11)
C
      IF(LASTGN.EQ.URNM(30))WRITE(*,530)NEXTRN,ISTRM,IZ
530 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO
(EL
      * SIGUIENTE A LOS ',I11/1X,'YA GENERADOS) ES:', ' ISTRM=',I11,5X,
      * ' IZ= ',I11)
C
IF(LASTGN.EQ.URNM(31).OR.LASTGN.EQ.URNM(32))WRITE(6,531)NEXTRN,IXX
531 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO
(EL
      * SIGUIENTE A LOS ',I11/1X,'YA GENERADOS) ES:', ' IXX=',I11)
C
      IF(LASTGN.EQ.URNM(33))WRITE(6,533)NEXTRN,XI
533 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO
(EL
      * SIGUIENTE A LOS ',I11/1X,'YA GENERADOS) ES:', ' XI=',G15.10)
C
      IF(LASTGN.EQ.URNM(34).OR.LASTGN.EQ.URNM(35))WRITE(6,534)NEXTRN,
      *KW1,KY1,KZ1,XR1
534 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO ',
      *(EL SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES:',
      */, ' KW1=',I15,3X,' , KY1=',I15,3X,' , KZ1=',I15,/1X,'XR1=',F12.10)
C
IF(LASTGN.EQ.URNM(36).OR.LASTGN.EQ.URNM(37).OR.LASTGN.EQ.URNM(38).
      *OR.LASTGN.EQ.URNM(13))WRITE(*,536)NEXTRN,(INTIN(I),I=1,5)
536 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO
(EL
      * SIGUIENTE A LOS ',I11/1X,'YA GENERADOS) ES:', '
INTIN(1)=' ,I11,2X,
      * ' INTIN(2)=' ,I11,/1X,' INTIN(3)=' ,I11,2X,' INTIN(4)=' ,I11,/1X,' INTIN(
      *5)=' ,I11)
C
      IF(LASTGN.EQ.URNM(39))WRITE(*,539)NEXTRN,SEED,ITOTAL39
539 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO
(EL
      * SIGUIENTE A LOS ',I11/1X,'YA GENERADOS) ES:', '
SEED=' ,D20.10,/1X,
      * ' ITOTAL39=' ,I20)
C
      IF(LASTGN.NE.URNM(41)) GOTO 445
      WRITE(6,441) NEXTRN
441 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO ',
      *(EL SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES:')
      IF(IA.EQ.0) GOTO 439

```

```

        WRITE(6,438)
438  FORMAT(' ','PARA VARIABLES ENTERAS:')
        WRITE(6,442) (INTIN(K),K=1,IA)
442  FORMAT(10(2X,I11))
439  IF(IR.EQ.0) GOTO 445
        WRITE(6,444)
444  FORMAT(' ','PARA VARIABLES REALES PUNTO FLOT.:')
        WRITE(6,440) (REALIN(K),K=1,IR)
440  FORMAT(5(2X,F18.7))
445  GO TO 631
C
C   PROCESA LOS DATOS DE ENTRADA DEL GENERADOR
C
800  LASTGN=NAME
C   write(*,*)'LASTGN=NAME'
C   write(*,8999)LASTGN
C8999  format(1x,a8)
        TIMLOW=10000
        LOWSAM=0
        SWCHI=0
        ICNT=1
        ICNTKS=0
        ICNTMT=0
        ISUB=KSUB
        NEWGEN=0
        GCODE=GCODE+1
        OPEN(UNIT=22,FILE='ARCH2',ACCESS='SEQUENTIAL',FORM='FORMATTED')
        READ(22,4000)NBATCH
4000  FORMAT(I7)
        READ(22,5000)TIME
5000  FORMAT(A2)
        TIMES=TIME
        READ(22,5010)IPRINT
        READ(22,5010)IDBL
        READ(22,5010)IEXT
        READ(22,5010)DBLPRT
        READ(22,5010)EXTPRT
5010  FORMAT(I6)
        READ(22,5020)IOPT
5020  FORMAT(I1)
        CLOSE (22)
C   WRITE(*,5011)IPRINT,IDBL,IEXT,DBLPRT,EXTPRT,IOPT
C5011  FORMAT(1X,5(I6,2X))
        WRITE(*,*)
C
C   IF(TIMES.NE.YES)TIMES=TNO
C
C   SE LEEN NUEVE DATOS DE ENTRADA PARA UN GENERADOR, Y ESTOS TIENEN
C   LOS SIGUIENTES SIGNIFICADOS:
C   NAME...EL NOMBRE DEL GENERADOR A UTILIZARSE
C   NBATCH...CANTIDAD DE NUMEROS ALEATORIOS A USARSE EN CADA TEST
C   TIMES...SI SE CALCULAN LOS TIEMPOS O NO (YES O NO)
C   IPRINT...CANTIDAD DE NUMEROS ALEATORIOS A IMPRIMIRSE
C   IDBL...CANTIDAD DE NUMEROS EN DOBLE PRECISION A SER GENERADOS
C   IEXT...CANTIDAD DE NUMEROS EN PRECISION EXTENDIDA A SER
GENERADOS
C   DBLPRT...CANTIDAD DE NUMEROS EN DOBLE PRECISION A SER IMPRESOS
C   EXTPRT...CANTIDAD DE NUMEROS EN PRECISION EXTENDIDA A SER
IMPRESOS
C   IOPT...OPCIONES A UTILIZAR EN LA INICIALIZACION DE VARIABLES
C

```

```

C          ANTES DE GENERAR LOS NUMEROS ALETORIOS
C
C          LOS 7 PARAMETROS NUMERICOS DEBEN ESTAR JUSTIFICADOS A DERECHA EN
C          LAS COLUMNAS COMO SE ESPECIFICA ABAJO; LOS PARAMETROS CHARACTER
C          DEBEN SER JUSTIFICADOS A IZQUIERDA.
C
C          VARIABLE          LONG.          NUM./CAR.
C          NAME              8              CAR
C          NBATCH            7              NUM
C          TIMES             3              CAR
C          IPRINT            6              NUM
C          IOPT              1              NUM
C
C          TITULO DE PAGINA PARA EL NUEVO GENERADOR
C
C          WRITE(6,300)
300 FORMAT('
', '*****
', '*****')
C          WRITE(6,304)
304 FORMAT(1X,'*',1X,'SE HA ENCONTRADO UN GENERADOR EN LOS DATOS DE
EN
*TRADA.TODOS LOS TESTS      ','*')
C          WRITE(6,305)
305 FORMAT(1X,'*',1X,'DISPONIBLES SERAN APLICADOS A ESTE GENERADOR
HAS
*TA EL FIN DE LA EJECUCION.','*')
C          WRITE(6,306)
306 FORMAT(' ','*',75X,'*')
C          WRITE(6,307)
C          307 FORMAT(' ','*',1X,'SE ASIGNARA UN NUMERO DE CODIGO AL PRESENTE
GEN
*ERADOR',21X,'*')
C          WRITE(6,310)GCODE
C          310 FORMAT(' ','*',1X,'EL CODIGO ASIGNADO ES GENCODE:',I2,'
C          *          ',5X,'*')
C          WRITE(6,360)
360
FORMAT(1X,'*****
*****')
C          WRITE(6,364)
364 FORMAT(' ','LA INICIALIZACION ESPECIFICADA ANTES DE QUE ALGUN
NUME
*RO SEA GENERADO ES:')
C          WRITE(*,*)' '
C
C          IF(DBLPRT.LE.IDBL)GO TO 820
C          DBLPRT=IDBL
C          WRITE(6,832) DBLPRT
832 FORMAT('SE IMPRIMIRAN MAS NUMEROS EN DOBLE PRECISION QUE LOS ',
*'CALCULADOS',/, 'POR LO TANTO, SOLO ',I6,' NUMEROS SERAN
IMPRESOS')
C          820 IF(EXTPRT.LE.IEXT)GO TO 835
C          EXTPRT=IEXT
C          WRITE(6,837) EXTPRT
837 FORMAT('SE IMPRIMIRAN MAS NUMEROS EN PRECISION EXTENDIDA QUE
LOS',
*'CALCULADOS',/, 'POR LO TANTO, SOLO ',I6,' NUMEROS SERAN
IMPRESOS')
C
C          REALIZA UNA LLAMADA AL GENERADOR APROPIADO

```

```

C
835 DO 25 KK=1,10000
25  Y(KK)=0
    GO TO (110,120,130,140,150,160,170,180,190,200,210,215,220,161,
    *171,181,191,201,211,212,2021,2022,2023,2024,2025,2026,2027,
    *2028,2029,2030,2031,2032,2033,2034,2035,2036,2037,2038,2039,
    *2040,2041),ISUB
110 CALL RRN01
    GO TO 630
120 CALL RRN02(NN,ML,MM,MK,L,MMM,KKK)
    GO TO 630
130 CALL RRN03
    GO TO 630
c 140 CALL RRN04(ITABLE)
140 GO TO 6303
150 CALL RRN05(MFSR,IP,IQ,INTSIZ,JJ)
    GO TO 630
160 CALL RRN06
    GO TO 630
c 170 CALL RRN07
170 GO TO 6303
180 CALL RRN08
    GO TO 630
190 CALL RRN09
    GO TO 630
200 CALL RRN10
    GO TO 630
210 CALL RRN11
    GO TO 630
215 CALL RRN12(IX,L,C2,ZZ2)
    GO TO 630
220 CALL RRN13(X,NBATCH,TIMES)
    GO TO 630
161 CALL RRN14
    GO TO 630
c 171 CALL RRN15
171 GO TO 6303
c 181 CALL RRN16
181 GO TO 6303
c 191 CALL RRN17
191 GO TO 6303
c 201 CALL RRN18
201 GO TO 6303
c 211 CALL RRN19
211 GO TO 6303
212 CALL RRN20(X,NBATCH,TIMES)
    GO TO 630
2021 CALL RRN21(NBATCH)
    GO TO 630
2022 CALL RRN22(NBATCH)
    GO TO 630
2023 GOTO 6303
2024 GOTO 6303
2025 GOTO 6303
2026 GOTO 6303
2027 GOTO 6303
2028 CALL RRN28(X,NBATCH,TIMES)
    GO TO 630
2029 CONTINUE
2030 CALL RRN30(X,NBATCH,TIMES)
    GO TO 630

```

```

2031 CALL RRN31(X,NBATCH,TIMES)
      GO TO 630
2032 CALL RRN32(X,NBATCH,TIMES)
      GO TO 630
2033 CALL RRN33(X,NBATCH,TIMES)
      GO TO 630
2034 CALL RRN34(X,NBATCH,TIMES)
      GOTO 630
2035 CALL RRN35(X,NBATCH,TIMES)
      GOTO 630
2036 CALL RRN36(X,NBATCH,TIMES)
      GOTO 630
2037 CALL RRN37(X,NBATCH,TIMES)
      GOTO 630
2038 CALL RRN38(X,NBATCH,TIMES)
      GOTO 630
2039 CALL RRN39(X,NBATCH,TIMES)
      GOTO 630
2040 CALL RRN40(X,NBATCH,TIMES)
      GOTO 630
2041 CALL RRNWN
      GOTO 630
6303 WRITE(*,6304)NAME
6304 FORMAT(1X,'EL GENERADOR:',A8,' NO ESTA IMPLEMENTADO
      *EN ESTA VERSION DE GENTSTPC')
      PAUSE
      STOP
630 NEWGEN=1
C
C REGISTRA EL TIEMPO SI FUE REQUERIDO
C
      IF(TIMES.NE.YES) GOTO 997
      CTIME(1)=KTIME/FLOAT(NBATCH)
      IF(CTIME(1).LT.0) CTIME(1)=99999
      IF(CTIME(1).LT.TIMLOW) LOWSAM=1
      IF(CTIME(1).LT.TIMLOW) TIMLOW=CTIME(1)
C
C A CONTINUACION SE IMPRIMEN LOS NUMEROS DE LA PRIMER MUESTRA SI
FUE
C REQUERIDO
C
      997 IF(IPRINT.LE.0) GOTO 940
          IF(IPRINT.LE.NBATCH) GOTO 515
          WRITE(6,745) IPRINT,NBATCH
      745 FORMAT(' ',' LA CANTIDAD DE NUM. ALEATORIOS A SER IMPRESOS
EXCEDE
          *LA CANT. GENERADA. ',/, ' EL VALOR DE IPRINT','I6,' COMO HA SIDO
          *ESPECIFICADO, FUE CAMBIADO A: ',I6)
          IPRINT=NBATCH
      515 WRITE(*,*)' '
          WRITE(6,1)IPRINT,NAME
      1 FORMAT(' ',' ESTOS SON LOS',I7,' NUMEROS A SER IMPRESOS POR EL
GENE
          *RADOR ',A8,':')
          WRITE(*,*)' '
          L1=1
          DO 4 I=1,100
          M1=L1+9
          IF(M1.GT.IPRINT)M1=IPRINT
          WRITE(6,2)(X(L2),L2=L1,M1)
      2 FORMAT(' ',5F11.8)

```

```

L1=M1+1
IF(L1.GT.IPRINT)GO TO 940
IF(I.EQ.50) THEN
WRITE(*,*)' '
WRITE(6,400)IPRINT,NAME
400 FORMAT(' ','PRIMEROS',I7,' NUMEROS ALEATORIOS PARA EL GENERADOR
',
*A8,'(CONTINUACION)',///)
ENDIF
4 CONTINUE
C
C GUARDA LOS NUMEROS
C
940 DO 941 K1=1,NBATCH
941 Y(K1)=X(K1)
C
C VERIFICACION DOBLE PRECISION
C
999 IF(IDBL.EQ.0)GO TO 632
IF(NBATCH.GE.IDBL*4)GO TO 633
WRITE(6,636)
636 FORMAT('EL VALOR DE NBATCH DEBE SER AL MENOS 4 VECES MAYOR QUE
',
*'EL VALOR DE IDBL. LA EJECUCION CONTINUA.')
GO TO 631
633 CALL DBLPRE(IDBL)
WRITE(6,751)DBLPRT,NAME
751 FORMAT(1X,'ESTOS SON LOS ',I6,' NUMEROS EN DOBLE PRECISION A SER
I
*MPRESOS POR EL GENERADOR ',A8)
WRITE(6,732)(XDBL(I),I=1,DBLPRT)
732 FORMAT(5(1X,D22.16))
C
C VERIFICACION PRECISION EXTENDIDA
C
632 IF(IEXT.EQ.0)GO TO 631
IF(NBATCH.GE.IEXT*8)GO TO 634
WRITE(6,635)
635 FORMAT('EL VALOR DE NBATCH DEBE SER AL MENOS 4 VECES MAYOR QUE
',
*'EL VALOR DE IEXT. LA EJECUCION CONTINUA.')
GO TO 631
634 CALL EXTPRE(IEXT)
WRITE(6,742)EXTPRT,NAME
742 FORMAT(1X,'ESTOS SON LOS ',I6,' NUMEROS EN PRECISION EXTENDIDA A
S
*ER IMPRESOS POR EL GENERADOR ',A8)
WRITE(6,734)(XEXT(I),I=1,EXTPRT)
C 734 FORMAT(3(1X,Q41.35))
734 FORMAT(3(1X,G41.35))
C
631 CONTINUE
CLOSE(20)
C
C CICLO FINAL DE PROCESAMIENTO DE DATOS DE ENTRADA
626 RETURN
END
C ***** FIN TESTER
*****
C
C

```

```

C*****
C      SUBROUTINE TSTM1
C*****
C TSTM1 CALCULA LA CORRELACION SERIAL DE UN GENERADOR DE NUMEROS
C ALEATORIOS CONGRUENCIAL (MIXTO) ESPECIFICADO POR EL USUARIO.
C
C PROPOSITO:
C
C DADO UN METODO PARA GENERAR NUMEROS REALES EN PUNTO FLOTANTE ENTRE
C 0 Y 1 (INCLUSIVE), QUE SE SUPONEN ALEATORIOS, ES DESEABLE QUE SU
C CORRELACION SERIAL LAG1 ESTE CERCA DE CERO. TST03 VERIFICA ESTO
C EMPIRICAMENTE.
C SIN EMBARGO PARA CIERTOS TIPOS DE GENERADORES, LA CORRELACION SE-
C RIAL LAG1 PUEDE SER CALCULADA A TRAVES DE FORMULAS TEORICAS. ESTA
C SUBRUTINA REALIZA DICHO CALCULO PARA GENERADORES DE NUMEROS
ALEATO-
C RIOS DE LA FORMA:
C
C  $X(I+1) = L*X(I)+C \text{ MOD } ZZ, I=0,1,2,\dots$ 
C
C USO:
C
C CUANDO SE EJECUTA EL COMANDO "CALL TESTER" EN EL PROG.PRINCIPAL,
UN
C ARCHIVO DE ENTRADA CON LOS DATOS NECESARIOS PRODUCE UNA LLAMADA
C DE LA RUTINA TSTM1. LOS PARAMETROS DE ESTA SUBRUTINA SON:
C
C VARIABLE          LONG.          NUM./CAR.
C TNAME              8              CAR
C IPARM(1)           11             NUM
C IPARM(2)           11             NUM
C IPARM(3)           11             NUM
C
C DESCRIPCION DE PARAMETROS:
C
C IPARM(1)...ES LA CONSTANTE MULTIPLICATIVA L=A DEL GENERADOR.
C IPARM(2)...CONSTANTE ADITIVA DEL GENERADOR.(EL GENERADOR ES
LLAMA-
C DO "CONGRUENCIAL MIXTO" SI ESTA CONSTANTE C ES
DISTINTA
C DO 0, Y "CONGRUENCIAL MULTIPLICATIVO" SI C=0)
C IPARM(3)...MODULO ZZ DEL GENERADOR.
C
C SUBRUTINAS Y FUNCIONES REQUERIDAS:
C GCD
C
C METODO:
C VER REFERENCIA 3 PARA EL ALGORITMO EXACTO USADO CUANDO L Y ZZ SON
C RELATIVAMENTE PRIMOS. REFERENCIA 1 PARA LA APROXIMACION USADA DE
C OTRA FORMA Y REFERENCIA 2 PARA DISCUSION Y EJEMPLOS
C 1. COVEYOU,R.R.: "SERIAL CORRELATION IN THE GENERATION OF PSEUDO-
C RANDOM NUMBERS",JOURNAL OF THE ACM, VOL.7(1960), PAG.72-74.
C 2. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C COLUMBUS, OHIO, 1981.
C 3. KNUTH,D.E.: "THE ART OF COMPUTER PROGRAMMING, VOLUME 2/
C SEMINUMERICAL ALGORITHMS",ADDISON-WESLEY PUBLISHING COMPANY,
C INC., READING, MASSACHUSETTS, 1969, PAG.72,75-76.
C*****
C
C      SUBROUTINE TSTM1

```

```

COMMON /MESS/MESSG(9),VARIAB(9)
COMMON /TSTPRM/IPARM(5)
INTEGER*4 L,C,ZZ,C1,Z1
CHARACTER*8 MESSG,VARIAB
REAL*8 RL,RC,RZZ,DSERCOR,DERR

C
C      ENCABEZAMIENTOS
C
WRITE(*,*)' '
WRITE(*,*)' '

WRITE(*,*)'*****'
WRITE(*,*)'*****'

WRITE(*,*)'*****'
WRITE(*,*)'*****'
WRITE(*,*)' '
WRITE(*,*)' TEST DE CORRELACION SERIAL:'
WRITE(*,*)' -----'
WRITE(*,*)' '
WRITE(6,2)
2 FORMAT(3X,'EL TEST DE CORRELACION SERIAL (RUTINA TSTM1) HA SIDO
AP *LICADO')
WRITE(6,3)
3 FORMAT(3X,'Y HA PRODUCIDO LOS SIGUIENTES RESULTADOS: ',/)

C
C      VERIFICAR SI LOS PARAMETROS SON REALES O ENTEROS
C
C*****ARCHIVO DE
DATOS*****

OPEN(UNIT=32,FILE='ARCHT1',ACCESS='SEQUENTIAL',FORM='FORMATTED')
READ(32,2349)IBAN
2349 FORMAT(I11)
IF (IBAN.EQ.-1)GO TO 30

C
C      PARAMETROS ENTEROS
C
READ(32,2349)IPARM(1)
READ(32,2349)IPARM(2)
READ(32,2349)IPARM(3)
L=IPARM(1)
C=IPARM(2)
ZZ=IPARM(3)
C WRITE(*,*)C,ZZ
CALL GCD(L,ZZ,IGCD,INV)
IF(IGCD.EQ.1) GO TO 70
GO TO 50

C
C      PARAMETROS REALES
C
30 READ(32,2359)RL
READ(32,2359)RC
READ(32,2359)RZZ
2359 FORMAT(D22.0)
DSERCOR=(1-6*RC/RZZ*(1-RC/RZZ))/RL
DERR=(RL+6)/RZZ

C
C      IMPRESIONES CORRESPONDIENTES A UNA APROXIMACION REAL
C

```



```

        WRITE(6,4)
    4  FORMAT(3X,'EL GENERADOR CONGRUENCIAL LINEAL DE NROS.
ALEATORIOS',/
        *1X,T10, ' X(N+1) = A * X(N) + C (MODULO M)  CON:',/)
        WRITE(6,5) RL
    5  FORMAT(3X,'MULTIPLICADOR A:      ',D22.10)
        WRITE(6,6) RC
    6  FORMAT(3X,'CONSTANTE C:          ',G22.10)
        WRITE(6,7) RZZ
    7  FORMAT(3X,'MODULO M:              ',D22.10)
        WRITE(6,8) DSERCOR,DERR
    8  FORMAT(3X,'TIENE UN COEFICIENTE DE CORRELACION APROXIMADO: ',
        *G22.10,/3X, 'CON UN ERROR MENOR QUE: ',G22.10)
C
    GO TO 999
C
C      PROCESAMIENTO DE PARAMETROS ENTEROS QUE SON PRIMOS ENTRE SI
C
    50 ERR=(L+6.)/FLOAT(ZZ)
        SERCOR=(1-6*FLOAT(C)/ZZ*(1-FLOAT(C)/ZZ))/L
C
        WRITE(6,9)
    9  FORMAT(3X,'EL GENERADOR CONGRUENCIAL LINEAL DE NUMEROS
ALEATORIOS:
        *',/3X)
        WRITE(6,11) L,C,ZZ
    11 FORMAT(3X,'X(N+1) = ',I11,' * X(N) + ',I11,' (MODULO ',I11,')
        *')
        WRITE(6,12)
    12 FORMAT(3X,'TIENE MULTIPLICADOR Y MODULO QUE NO SON PRIMOS ENTRE
SI
        *.')
        WRITE(6,13) SERCOR,ERR
    13 FORMAT(3X,'EL COEFICIENTE DE CORRELACION SERIAL APROXIMADO ES:',
        *G22.10,/3X,'CON UN ERROR MENOR QUE:',G22.10)
        GO TO 999
C
C      PROCESAMIENTO DE ENTEROS QUE SON PRIMOS ENTRE SI
C
    70 L1=L
        Z1=ZZ
        C1=C
        DEDSUM=0
        ISGN=1
    505 L1=MOD(L1,Z1)
        C1=MOD(C1,Z1)
        IF(Z1.LE.2) GO TO 550
        IF(C1.LE.L1) GO TO 530
        IR=C1/L1
        IREM=MOD(C1,L1)
        IF(IREM.EQ.0) ID=3
        IF(IREM.NE.0) ID=0
    520 DEDSUM=DEDSUM+ID+ISGN*(6*IR*(C1+IREM-Z1))/FLOAT(Z1)
        C1=IREM
    530 FZ1=FLOAT(Z1)
        FL1=FLOAT(L1)
        DEDSUM=DEDSUM+ISGN*(-3+FL1/FZ1+FZ1/FL1+(1+6*C1**2)/(FL1*FZ1))
    40 ISGN=-ISGN
        ITEMP=L1
        L1=Z1
        Z1=ITEMP

```

```

GO TO 505
C
550 CALL GCD(L,ZZ,IGCD,IP1)
      INV=MOD(IP1*C,ZZ)
      FZZ=FLOAT(ZZ)
      SERCOR=(FZZ*DEDSUM-3+6*(FZZ-INV-C))/(FZZ**2-1)
C
      WRITE(6,60)
60 FORMAT(3X,'EL GENERADOR CONGRUENCIAL LINEAL DE NUMEROS
ALEATORIOS:
      ',/3X)
      WRITE(6,61) L,C,ZZ
61 FORMAT(3X,'X(N+1) = ',I11,' * X(N) + ',I11,' (MODULO ',I11,')
      *')
      WRITE(6,63) SERCOR
63 FORMAT(/3X,'TIENE UN COEFICIENTE DE CORRELACION SERIAL IGUAL
A: ',
      *//T20,G22.10)
999 CLOSE (32)
      RETURN
      END
C
C
C*****
C      SUBROUTINE TSTM2
C*****
C TSTM2 CALCULA EL PERIODO/POTENCIA P(X(I+1)<X(I)) DE UN GENERADOR DE
C NUMEROS ALEATORIOS CONGRUENCIAL (MIXTO) ESPECIFICADO POR EL USUARIO
C
C PROPOSITO
C DADO UN METODO DE GENERACION DE NUMEROS REALES EN PTO. FLOTANTE EN-
C TRE 0 Y 1 (INCLUSIVE) EL CUAL SE SUPONE ALEATORIO, EL PROPOSITO ES
C QUE NO "CICLE" PRONTO, Y QUE P(X(I+1)<X(I)) ESTE "CERCA" A 0.5.
C ESTA SUBROUTINA REALIZA CALCULOS RELACIONADOS CON ESTE OBJETIVO PARA
C GENERADORES DE NUMEROS ALEATORIOS QUE SON DE ESTA FORMA
C  $X(I+1) = A*X(I)+C \text{ MOD } ZZ, I=0,1,2,\dots$ 
C
C USO
C LOS PARAMETROS DE TSTM2 SON:
C
C VARIABLE          LONG.          NUM./CAR.
C TNAME              8              CAR
C IPARM(1)           11             NUM
C IPARM(2)           11             NUM
C IPARM(3)           11             NUM
C
C DESCRIPCION DE PARAMETROS
C IPARM(1)...ES EL MULTIPLICADOR ENTERO A DEL GENERADOR.
C IPARM(2)...ES LA CONSTANTE ADITIVA DEL GENERADOR. (EL GENERADOR
C ES DENOMINADO "CONGRUENCIAL MIXTO" SI ESTA CONSTANTE C
C NO ES IGUAL A CERO, Y "CONGRUENCIAL MULTIPLICATIVO" SI
C C=0)
C IPARM(3)...ES EL MODULO ZZ POR EL CUAL LOS NUMEROS SON REDUCIDOS.
C
C OBSERVACIONES:
C SI UNO O MAS DE LOS A,C,ZZ CONSIDERADOS EXCEDEN EL VALOR ENTERO
C MAXIMO DISPONIBLE DE UN MICROCOMPUTADOR (2147483647 = 2**31-1)DE
C 32 BITS, ENTONCES ESTA SUBROUTINA NO PUEDE USARSE
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS

```

```

C   GCD, TSTM3, ORDER
C
C METODO
C   VER REFERENCIA 1. PARA DISCUSION Y EJEMPLOS, Y REFERENCIA 2. PARA
C   UN RESUMEN DE LOS RESULTADOS BASICOS CODIFICADOS EN ESTE
PROGRAMA:
C   1. DUDEWICZ,E.J. Y RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C   GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C   COLUMBUS, OHIO, 1981.
C   2. KNUTH,D.E.: "THE ART OF COMPUTER PROGRAMMING, VOLUME 2/
C   SEMINUMERICAL ALGORITHMS",ADDISON-WESLEY PUBLISHING COMPANY,
C   INC., READING, MASSACHUSETTS, 1969, PAG.9-10,15-19,21-23,69,
C   294-313
C*****
C
C   SUBROUTINE TSTM2
C
C   COMMON /MESS/MESSG(9),VARIAB(9)
C   COMMON /TSTPRM/IPARM(5)
C
C   INTEGER*4 L,C,ZZ
C   CHARACTER*8 MESSG,VARIAB
C   REAL*8 MESSG,VARIAB
C   DIMENSION
INTZZ(31,2),INTL1(31,2),INTORD(31,2),INTB(31,2),NTRK(31)
C
C   INICIALIZACIONES
C
C*****ARCHIVO DE
DATOS*****
OPEN(UNIT=33,FILE='ARCHIM2',ACCESS='SEQUENTIAL',FORM='FORMATTED')
  READ(33,2349)IPARM(1)
  READ(33,2349)IPARM(2)
  READ(33,2349)IPARM(3)
2349  FORMAT(I11)
      L=IPARM(1)
      C=IPARM(2)
      ZZ=IPARM(3)
      CLOSE(33)
      L=MOD(L,ZZ)
      IF(L.NE.0) GOTO 8
      WRITE(6,98) IPARM(1),ZZ
  98  FORMAT(3X,'EL MULTIPLICADOR ',I11,' EN TSTM2 ES CONGRUENTE CON
CER
      *O SEGUN MODULO ',I11,'./,' FIN TEST TSTM2.')
      GOTO 999
  8  NSTOR=0
      L1=L-1
C
C   ISW=1 SUPRIME LA IMPRESION DE LOS RESULTADOS DE TSTM3 CUANDO
C   ESA SUBRUTINA ES LLAMADA INTERNAMENTE
C
C   ISW=1
C
C   ENCABEZADOS
C
C   WRITE(*,*)' '
C   WRITE(*,*)' '
C
WRITE(*,*)'*****'

```

```

*****
WRITE(*,*)'*****'
*****
WRITE(*,*)' '
WRITE(*,*)' TEST DE POTENCIA:'
WRITE(*,*)' -----'
WRITE(*,*)' '
WRITE(6,100)
100 FORMAT(3X,'EL TEST DE POTENCIA FUE APLICADO (RUTINA TSTM2) Y SE
HA
*N OBTENIDO')
WRITE(6,103)
103 FORMAT(3X,'LOS SIGUIENTES RESULTADOS:')
WRITE(6,104) MESSG
104 FORMAT(30X,9A8)
C WRITE(6,110)
C 110 FORMAT(1X,'*****')
WRITE(6,105)
105 FORMAT(3X,'EL GENERADOR DE NUMEROS ALEATORIOS PROBADO ES
CONGRUENC
*IAL MULTIPLICATIVO',/3X,'CON:',/3X)
WRITE(6,106) IPARM(1),C,ZZ
106 FORMAT(3X,'EL MULTIPLICADOR A = ',I11,/,
* 3X,'EL TERMINO CONSTANTE C = ',I11,/,
* 3X,'EL MODULO M = ',I11)
C
C FACTOR ZZ
C
C CALL TSTM3(ZZ,INTZZ,NDXZZ,ISW)
C
C VERIFICA QUE GCD(C,M)=1
C
IF(C.EQ.0) GOTO 88
CALL GCD(C,ZZ,IGCDC,INVERS)
IF(IGCDC.EQ.1) GOTO 97
88 WRITE(6,205) C,ZZ
205 FORMAT(/,3X,'EL TERMINO CONSTANTE ',I11,' Y EL MODULO ',I11,
*' NO SON RELATIVA-',/3X,'MENTE PRIMOS Y ASI EL GENERADOR NO
TIENE
* UN PERIODO MAXIMO.LA POTENCIA',/3X,'NO ESTA ENTONCES
DEFINIDA. ')
GOTO 95
C
C CALCULA LA PROBABILIDAD P(X(I+1)<X(I))
C
97 CALL GCD(L1,ZZ,IGCD,INVERS)
IC=MOD(C,IGCD)
PROB=(2*IC-IGCD)/(2.0*FLOAT(ZZ))+0.5
C
C VERIFICA LA CONDICION DE DIVISIBILIDAD POR 4
C
90 IF(MOD(ZZ,4).NE.0) GOTO 92
IF(MOD(L1,4).EQ.0) GOTO 92
WRITE(6,225) ZZ,L1
225 FORMAT(/,3X,'4 ES UN FACTOR DEL MODULO:',I11,' PERO NO ES UN
FA
*CTOR DE:',/3X,'L-1 = ',I11,/3X,'ASI EL GENERADOR NO TIENE UN
*PERIODO MAXIMO.', /3X,'LA POTENCIA NO ESTA ENTONCES DEFINIDA')
GOTO 95
C

```

```

C          FACTOR L1
C
92 CALL TSTM3(L1,INTL1,NDXL1,ISW)
C
C   VERIFICA QUE CADA FACTOR PRIMO DE M SEA TAMBIEN FACTOR DE L1
C
      DO 20 I=1,NDXZZ
      INT=INTZZ(I,1)
      DO 10 J=1,NDXL1
      IF(INTL1(J,1).GT.INTZZ(I,1)) GO TO 210
      IF(INTZZ(I,1).EQ.INTL1(J,1)) GOTO 15
10  CONTINUE
      GOTO 210
15  NTRK(I)=J
20  CONTINUE
C
C   CALCULA POTENCIA
C
      DO 60 I=1,NDXZZ
      NN=NTRK(I)
      S=FLOAT(INTZZ(I,2))/INTL1(NN,2)
      SS=S-IFIX(S)
      IF(SS.EQ.0.) GO TO 40
      NS=IFIX(S)+1
      GO TO 50
40  NS=IFIX(S)
50  IF(NSTOR.GE.NS) GO TO 60
      NSTOR=NS
60  CONTINUE
      GOTO 94
C
210 WRITE(6,215) ZZ,INT,L1
215 FORMAT(///,3X,'EL MODULO ',I11,' TIENE UN FACTOR PRIMO,',I11,' ,
N
      *O OCURRENTE EN LA FACTORIZACION DE L-1 = ',I11,/,3X,
      *'Y ASI EL GENERADOR NO TIENE MAXIMO PERIODO. LA POTENCIA NO ESTA
E
      *NTONCES DEFINIDA.')
      GOTO 95
C
C   IMPRIME POTENCIA, SI ESTA DEFINIDA
C
94  WRITE(6,170) NSTOR
170 FORMAT(3X,'ESTE GENERADOR TIENE POTENCIA ',I5,' .')
C
C   IMPRIME EL PERIODO, SI ES MAXIMO
C
      WRITE(6,172) ZZ
172 FORMAT(3X,'ESTE GENERADOR ES DE PERIODO MAXIMO, ES DECIR,
PERIODO
      *= M = ',I11,' .')
      GOTO 96
C
C   CALCULA EL PERIODO, SI NO ES MAXIMO
C
95  CALL GCD(L,ZZ,IGCDL,INVL)
      IF(IGCDL.EQ.1) GOTO 295
      WRITE(6,800)
800  FORMAT(3X,'EL MULTIPLICADOR NO ES RELATIVAMENTE PRIMO CON EL
MODUL
      *O. ',/,3X,'EL PERIODO NO PUEDE SER CALCULADO.')

```

```

        GOTO 96
295 CALL ORDER(L,INTZZ,NDXZZ,IORDER)
    CALL TSTM3(IORDER,INTORD,NDXORD,ISW)
    IB=(L-1)+C
    CALL TSTM3(IB,INTB,NDXB,ISW)
    IF(C.NE.0) GOTO 130
    WRITE(6,810) IORDER
810  FORMAT('0',23X,'EL PERIODO ES ',I11)
    GOTO 96
C
C    CALCULA EL PERIODO CUANDO C>0 Y EL PERIODO NO ES MAXIMO.
C
130  IPER=1
    DO 520 I=1,NDXZZ
    IF(INTZZ(I,1).GT.2) GOTO 420
    IF(INTZZ(I,2).LT.2) GOTO 520
    IF(MOD(L,4).NE.1) GOTO 350
C
C    EL CASO P=2 Y A CONGRUENTE CON 1 MODULO 4 Y
C    EL CASO P>2 Y A CONGRUENTE CON 1 MODULO P.
C
300  DO 310 K=1,NDXB
    IF(INTB(K,1).GT.INTZZ(I,1)) GOTO 330
    IF(INTB(K,1).EQ.INTZZ(I,1)) GOTO 320
310  CONTINUE
    GOTO 320
C
320  ID=INTZZ(I,2)-INTB(K,2)
    IF(ID.LT.0) ID=0
    GOTO 340
C
330  ID=INTZZ(I,2)
C
340  IPERP=INTZZ(I,1)**ID
    CALL GCD(IPERP,IPER,IGCDP,INVP)
    IPER=IPER*(IPERP/IGCDP)
    GOTO 520
C
C    EL CASO P=2 Y A CONGRUENTE CON -1 MODULO 4.
C
350  MOD2=INTZZ(I,1)**INTZZ(1,2)
    N2=1
    L2=MOD(L,MOD2)
    CALL ORDER(L2,INTZZ(I,1),N2,IORD2)
    CALL TSTM3(IORD2,INTORD,NDXORD,ISW)
    IF(INTORD(NDXORD,1).EQ.INTZZ(I,1)) GOTO 370
400  WRITE(6,820) INTZZ(I,2)
820  FORMAT(3X,'HA OCURRIDO UN ERROR EN EL CALCULO DEL PERIODO PARA
EL
    *CASO P=2 Y MULTIPLICADOR CONGRUENTE A -1 MODULO 4.',/,
    *'EL CALCULO INDICA QUE 2 NO DIVIDE EL ORDEN EN EL GRUPO DE
UNIDADE
    *S MODULO 2**',I5)
    GOTO 96
C
370  DO 380 K=1,NDXB
    IF(INTB(K,1).GT.INTZZ(I,1)) GOTO 395
    IF(INTB(K,1).EQ.INTZZ(I,1)) GOTO 390
380  CONTINUE
    GOTO 395
C

```

```

390 ID=INTORD(NDXORD,2)-INTB(K,2)
    IF(ID.LT.0) ID=0
    GOTO 410
C
395 ID=INTORD(NDXORD,2)
    GOTO 410
C
410 IPER=IPER*(2**(ID+1))
    GOTO 520
C
C    EL CASO P>2
C
420 IF(MOD(L,INTZZ(I,1)).EQ.1) GOTO 300
C
C    EL CASO P>2 Y A NO CONGRUENTE CON 1 MODULO P
C
430 MODP=INTZZ(I,1)**INTZZ(I,2)
    NP=1
    LP=MOD(L,MODP)
    CALL ORDER(LP,INTZZ(I,1),NP,IORDP)
    IPPORD=IORDP
    CALL TSTM3(IORDP,INTORD,NDXORD,ISW)
    IF(INTORD(NDXORD,1).NE.INTZZ(I,1)) GOTO 470
    IPPORD=IPCORD/(INTORD(NDXORD,1)**INTORD(NDXORD,2))
C
    DO 440 K=1,NDXB
    IF(INTB(K,1).GT.INTZZ(I,1)) GOTO 460
    IF(INTB(K,1).EQ.INTZZ(I,1)) GOTO 450
440 CONTINUE
    GOTO 460
C
450 ID=INTORD(NDXORD,2)-INTB(K,2)
    IF(ID.LT.0) ID=0
    GOTO 500
C
460 ID=INTORD(NDXORD,2)
    GOTO 500
C
470 ID=0
C
500 IPERP=IPCORD*(INTZZ(I,1)**ID)
    CALL GCD(IPER,IPERP,IGCDP,INVP)
    IPER=IPER*(IPERP/IGCDP)
520 CONTINUE
C
    WRITE(6,830) IPER
830 FORMAT(3X,'EL VALOR DEL PERIODO DEPENDE EN PARTE, DEL VALOR DE
LA
    *SEMILLA INICIAL X0.',/,3X,'SI X0=1, EL PERIODO ES: ',I11)
C
C    IMPRIME LA PROBABILIDAD DE QUE X(I+1) < X(I)
C
96 IF(C.EQ.0) GOTO 999
    WRITE(6,180) PROB
180 FORMAT(3X,'LA PROBABILIDAD DE QUE X(I+1) < X(I) ES: ',F7.4)
999 RETURN
    END
C
C
C*****
C    SUBROUTINE TSTM3(N,INTDIV,INDX,ISW)

```

```

C*****
C TSTM3 CALCULA LOS FACTORES PRIMOS DE UN ENTERO ESPECIFICADO.
C EL ENTERO MENCIONADO ES FRECUENTEMENTE USADO COMO LA CONSTANTE
C MULTIPLICATIVA DE UN GENERADOR CONGRUENCIAL DE NUMEROS ALEATORIOS
C (MULTIPLICATIVO O MIXTO)
C
C PROPOSITO:
C
C CALCULAR E IMPRIMIR LA FACTORIZACION PRIMA DE UN ENTERO DADO
C
C USO:
C LA SUBROUTINA TSTM3 UTILIZA LOS SIGUIENTES PARAMETROS:
C
C VARIABLE          LONG.      NUM./CAR.
C   TNAME            8          CAR
C   N                11         NUM
C
C DESCRIPCION DE PARAMETROS:
C   N.....ES EL ENTERO A SER FACTORIZADO EN FACTORES PRIMOS.
C   INTDIV...ARREGLO DE 31 X 2 DONDE SE ALMACENARA LA FACTORIZACION
C           DE N.
C           INTDIV(K,1) ES EL K-ESIMO PRIMO DISTINTO QUE OCURRE EN
LA
C           FACTORIZACION DE N.
C           INTDIV(K,2) ES LA POTENCIA CON LA CUAL APARECE ESTE
PRIMO
C           EN N.
C   INDX.....CANTIDAD DE FACTORES PRIMOS DISTINTOS DE N.
C   ISW.....SE USA PARA IMPRIMIR LOS RESULTADOS CUANDO ISW=1, O NO
C           IMPRIMIRLOS CUANDO ISW=0.
C
C NOTA:
C EL ENTERO A SER FACTORIZADO PUEDE NO EXCEDER EL VALOR ENTERO
MAXIMO
C DISPONIBLE DE UN MICROMPUTADOR (2147483647=2**31-1) DE 32 BITS.
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS:
C NINGUNA
C
C REFERENCIAS:
C 1. DUDEWICZ, E.J. AND RALLEY, T.: "THE HANDBOOK OF RANDOM NUMBER
C   GENERATION AND TESTING", AMERICAN SCIENCES PRESS,
C   COLUMBUS, OHIO, 1981.
C 2. KNUTH, D.E.: "THE ART OF COMPUTER PROGRAMMING, VOLUME 2/
C   SEMINUMERICAL ALGORITHMS", ADDISON-WESLEY PUBLISHING COMPANY,
C   INC., READING, MASSACHUSETTS, 1969, PAG.338-340.
C*****
C
C   SUBROUTINE TSTM3(N, INTDIV, INDX, ISW)
C   COMMON /MESS/MESSG(9), VARIAB(9)
C
C   CHARACTER*8 MESSG, VARIAB
C   INTEGER PRIME(200)
C   INTEGER*4 N
C   REAL*8 MESSG, VARIAB
C   DIMENSION INTDIV(31,2)
C
C   LOS VALORES DE UN ARREGLO DE ENTEROS "PRIME" SON INICIALIZADOS
C   CON LOS PRIMEROS DOSCIENTOS NUMEROS PRIMOS.
C

```



```

DATA PRIME/2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,
*59,61,67,71,73,79,83,89,97,101,103,107,109,113,127,131,
*137,139,149,151,157,163,167,173,179,181,191,193,197,
*199,211,223,227,229,233,239,241,251,257,263,
*269,271,277,281,283,293,307,311,313,317,331,337,347,349,
*353,359,367,373,379,383,389,397,401,409,419,421,431,433,
*439,443,449,457,461,463,467,479,487,491,499,503,509,521,
*523,541,547,557,563,569,571,577,587,593,599,601,607,613,
*617,619,631,641,643,647,653,659,661,673,677,683,691,701,
*709,719,727,733,739,743,751,757,761,769,773,787,797,809,
*811,821,823,827,829,839,853,857,859,863,877,881,883,887,
*907,911,919,929,937,941,947,953,967,971,977,983,991,997,
*1009,1013,1019,1021,1031,1033,1039,1049,1051,1061,1063,
*1069,1087,1091,1093,1097,1103,1109,1117,1123,1129,1151,
*1153,1163,1171,1181,1187,1193,1201,1213,1217,1223/

```

C

C\*\*\*\*\*ARCHIVO DE

DATOS\*\*\*\*\*

OPEN(UNIT=34, FILE='ARCHTM3', ACCESS='SEQUENTIAL', FORM='FORMATTED')

READ(34,2349) N

2349 FORMAT(I11)

CLOSE(34)

DO120 I=1,31

DO120 J=1,2

INTDIV(I,J)=0

120 CONTINUE

C

C INICIALIZACIONES

NI=N

INDX=0

C

C BUSCA FACTORES PRIMOS ENTRE 2 Y 1223

C

130 DO 170 I=1,200

IFLAG=0

IF(SQRT(FLOAT(NI)).LT.FLOAT(PRIME(I))) GO TO 145

C

140 IREM=MOD(NI,PRIME(I))

IF(IREM.NE.0) GO TO 170

150 IF(IFLAG.NE.0) GO TO 160

INDX=INDX+1

IFLAG=1

INTDIV(INDX,1)=PRIME(I)

160 INTDIV(INDX,2)=INTDIV(INDX,2)+1

NI=NI/PRIME(I)

GO TO 140

170 CONTINUE

C

IF(SQRT(FLOAT(NI)).GT.1223) GO TO 210

145 IF(NI.EQ.1) GOTO 155

INDX=INDX+1

INTDIV(INDX,1)=NI

INTDIV(INDX,2)=INTDIV(INDX,2)+1

155 IF(ISW.EQ.0) GOTO 300

RETURN

C

C VERIFICA LOS FACTORES PRIMOS ENTRE 6\*204 Y 6\*7724. CADA PRIMO

DEBE

C SER CONGRUENTE A 1 O 5 MODULO 6, ENTONCES SOLO LOS ENTEROS DE

ESTA

```

C      FORMA SON VERIFICADOS. YA QUE LOS FACTORES PRIMOS SE REMUEVEN
EN
C      EN ORDEN DE MEDIDA CRECIENTE, CUALQUIER NUMERO DE ESTA FORMA QUE
C      DIVIDA EL ENTERO DEL TEST NO PODRA SER DIVISIBLE POR CUALQUIER
C      PRIMO MENOR Y POR LO TANTO DEBE SER POR SI MISMO UN DIVISOR
PRIMO.
C
  210 DO 260 I=204,7724
      IDIV=I*6+1
      IFLAG=0
      IF(SQRT(FLOAT(NI)).LT.FLOAT(IDIV)) GO TO 270
  220 IREM=MOD(NI, IDIV)
      IF(IREM.NE.0) GO TO 240
      IF(IFLAG.NE.0) GO TO 230
      INDX=INDX+1
      IFLAG=1
      INTDIV(INDX,1)=IDIV
  230 INTDIV(INDX,2)=INTDIV(INDX,2)+1
      NI=NI/IDIV
      GO TO 220
C
  240 IDIV=I*6+5
      IF(SQRT(FLOAT(NI)).LT.FLOAT(IDIV)) GO TO 270
      IFLAG=0
  245 IREM=MOD(NI, IDIV)
      IF(IREM.NE.0) GO TO 260
      IF(IFLAG.NE.0) GOTO 250
      INDX=INDX+1
      IFLAG=1
      INTDIV(INDX,1)=IDIV
  250 INTDIV(INDX,2)=INTDIV(INDX,2)+1
      NI=NI/IDIV
      GO TO 245
  260 CONTINUE
C
  270 IF(NI.EQ.1) GOTO 280
      INDX=INDX+1
      INTDIV(INDX,1)=NI
      INTDIV(INDX,2)=INTDIV(INDX,2)+1
  280 IF(ISW.EQ.1) GOTO 999
C
  300 WRITE(*,*)' '
      WRITE(*,*)' '

WRITE(*,*)'*****'
*****'

WRITE(*,*)'*****'
*****'
      WRITE(*,*)' '
      WRITE(*,*)' TEST DE FACTORES PRIMOS:'
      WRITE(*,*)' -----'
      WRITE(*,*)' '
      WRITE(6,310)
  310 FORMAT(3X,'EL TEST DE FACTORES PRIMOS FUE APLICADO (RUTINA
TSTM3)
      *Y HA PRODUCIDO')
      WRITE(6,312)
  312 FORMAT(3X,'LOS SIGUIENTES RESULTADOS:')
      WRITE(6,315) MESSG

```

```

315 FORMAT(3X,9A8)
      IF(NI.EQ.N) GO TO 360
      WRITE(6,340) N
340  FORMAT(3X,'EN SU FACTORIZACION PRIMA, EL ENTERO ',I11,' ES EL
PROD
      *UCTO DE: ',//)
      DO 50 I=1,INDX
      WRITE(6,345) INTDIV(I,1),INTDIV(I,2)
345  FORMAT(3X,I11,' **',I3)
      50 CONTINUE
      GOTO 999
360  WRITE(6,370) N
370  FORMAT(3X,'EL ENTERO ',I11,' ES PRIMO.')
999  RETURN
      END

C
C
C      SUBROUTINE TSTM4(A,M)
C*****
C TSTM4 CALCULA LAS CANTIDADES D Y R PARA ESTABLECER UNA MEDIDA DE LA
C UNIFORMIDAD EN LAS DIMENSIONES T=2,3,4,5,6 PARA LOS NUMEROS
C ALEATORIOS
C GENERADOS POR UN ALGORITMO CONGRUENCIAL MULTIPLICATIVO EN UN
C PERIODO.
C
C LA CANTIDAD D, EN LA DIMENSION n=T, SE DEFINE COMO LA MINIMA
C DISTANCIA
C ENTRE LOS HIPERPLANOS PARALELOS GENERADOS A PARTIR DE {Ui} POR LAS
C N-UPLAS (Ui,...,Un+i-1)
C LA CANTIDAD R SE DEFINE COMO EL COCIENTE ENTRE D Y B, DONDE B ES UNA
C COTA
C SUPERIOR DE D EN LA DIMENSION T PROPUESTA POR MARSAGLIA, QUE ALCANZA
C UN VALOR 1 COMO MINIMO.
C
C PROPOSITO:
C CALCULAR E IMPRIMIR LAS MEDIDAS D Y R
C
C USO:
C LA SUBROUTINA TSTM4 UTILIZA LOS SIGUIENTES PARAMETROS:
C
C A ... FACTOR MUTIPLICATIVO DEL GENERADOR CONGRUENCIAL
C M ... MODULO DEL GENERADOR
C
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS:
C NINGUNA
C
C REFERENCIAS:
C 1. COVEYOU, R.R. AND MACPHERSON, R.D.: "FOURIER ANALYSIS OF
C UNIFORM RANDOM NUMBER GENERATORS," JOURNAL OF THE ACM, VOL.14
C (1967), PAG.100-119
C 2. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C GENERATION AND TESTING," AMERICAN SCIENCES PRESS,
C COLUMBUS, OHIO, 1981.
C 3. HOAGLIN D.C.: "THEORETICAL PROPERTIES OF CONGRUENTIAL RANDOM
C NUMBER GENERATORS:UN EMPIRICAL VIEW" ,MEMORANDUM NS-340,
C DEPARTMENT OF STATISTICS, HARVARD UNIVERSITY,CAMBRIDGE,MASS.
C 4. KNUTH,D.E.: "THE ART OF COMPUTER PROGRAMMING, VOLUME 2/
C SEMINUMERICAL ALGORITHMS," ADDISON-WESLEY PUBLISHING COMPANY,
C INC., READING, MASSACHUSETTS, 1969.
C*****

```

```

C
SUBROUTINE TSTM4
REAL*8 M,A,H,HP,P,PP,R,S,Q,U,V,MU,NU,COTA,RR,AUXILIAR,AUXILIAR2
REAL*8 R6,R7,R8
INTEGER I,II,J,K,FLAG57,FLAG911,L,TT
REAL*8 UU(10,10),VV(10,10),QV(10),AUXDOT(10),X(10),Y(10),Z(10)
REAL*8 DOTPROD,DOTPROD2

OPEN(UNIT=35,FILE='ARCHTM4',ACCESS='SEQUENTIAL',FORM='FORMATTED')
WRITE(*,*)' '

WRITE(*,*)'*****'
*****'

WRITE(*,*)'*****'
*****'

WRITE(*,*)' '
WRITE(*,*)'TEST ESPECTRAL:'
READ(35,1490)A
READ(35,1490)M
1490 FORMAT(D26.0)
WRITE(*,*)'----- '
WRITE(*,*)''
WRITE(*,*)''
WRITE(*,*)'LOS VALORES DE R QUE SEAN PROXIMOS A 1 SON
SATISFACTOR
*IOS'
WRITE(*,*)'LA DIMENSION: 2<= T <=6 ES ADECUADA EN MUCHAS
APLICACIO
*NES(REF.KNUTH D.,P.101)'
WRITE(*,*)'LA DISTANCIA INTERPLANAR D DEBE VERIFICAR D<=COTA'
WRITE(*,*)''
WRITE(*,*)'EL TEST ESPECTRAL HA SIDO APLICADO (RUTINA TSTM4) Y
HA'
WRITE(*,*)'PRODUCIDO LOS SIGUIENTES RESULTADOS:'
WRITE(*,*)''

C
IT=2
IPRUEBA=0
C
INICIALIZACION DE LAS ESTRUCTURAS DE DATOS.
DO 10 I=1,10
X(I)=0.0D0
Y(I)=0.0D0
Z(I)=0.0D0
QV(I)=0.0D0
AUXDOT(I)=0.0D0
10 CONTINUE
DO 20 I=1,10
DO 30 J=1,10
UU(I,J)=0.0D0
VV(I,J)=0.0D0
30 CONTINUE
20 CONTINUE
TT=6
C
H = A
HP = M
P = 1.0D0
PP =0.0D0
RR = A
S =1.0D0+A**2

```

```

IT = 2
Q =DINT(HP/H)
U = HP-Q*H
V = PP-Q*P
1000 CONTINUE
IF (U**2+V**2 .LT. S) THEN
S = U**2+V**2
HP = H
H = U
PP = P
P = V
Q =DINT(HP/H)
U = HP-Q*H
V = PP-Q*P
GOTO 1000
END IF
U = U-H
V = V-P
IF (U**2+V**2 .LT. S) THEN
S = U**2+V**2
HP = U
PP = V
END IF
C      IMPRIMIR RESULTADOS
WRITE(*,*)'-----'
--
*-----'
WRITE(*,19)IT
19  FORMAT(1X,T10,'DIMENSION T =',I3)
    NU=SQRT(S)
    WRITE(*,21)NU
21  FORMAT(1X,T10,'NU=',D20.10)
    WRITE(*,22) 1.D0/NU
22  FORMAT(1X,T10,'1/NU=',D20.10)
    D= 1/ DSQRT(S)
    WRITE(*,26)D
26  FORMAT(1X,T10,'D=',D20.10)
    COTA=2.0**(-30.0/FLOAT(IT))
    WRITE(*,25)COTA
    IF(D.LE.COTA) THEN
    IPRUEBA=0
    ELSE
    IPRUEBA=1
    END IF
25  FORMAT(1X,T10,'COTA=',D20.10)
    MU=(3.141593*S)/M
    R8=DSQRT(2.0D0*M)
    R= D*(R8 - 1.0D0)
    WRITE(*,24)R
24  FORMAT(1X,T10,'R=',D20.10)
    IF(IPRUEBA.EQ.1)THEN
    WRITE(*,27)IT
27  FORMAT(1X,'PARA T=',I2,' EL TEST NO SATISFACE: D <= COTA')
    END IF
29  UU(1,1)=-H
    UU(2,1)=P
    UU(1,2)=-HP
    UU(2,2)=PP
    IF(PP .GT. 0) THEN
    VV(1,1)=-PP
    VV(2,1)=-HP

```

```

VV(1,2)=P
VV(2,2)=H
ELSE
VV(1,1)=PP
VV(2,1)=HP
VV(1,2)=-P
VV(2,2)=-H
END IF
C PASO 4
3000 IF (IT .GT. TT) GOTO 44
FLAG911=1
IT=IT+1
AUXILIAR=A*RR
RR=DMOD(AUXILIAR,M)
UU(1,IT)=-RR
IIT=IT-1
DO 1100 I=1,IIT
UU(IT,I)=0.0D0
1100 CONTINUE
DO 1101 I=2,IIT
UU(I,IT)=0.0D0
1101 CONTINUE
UU(IT,IT)=1.0D0
DO 1102 I=1,IIT
AUXILIAR=VV(1,I)*RR/M
QV(I)=DNINT(AUXILIAR)
1102 CONTINUE
DO 1103 I=1,IIT
VV(IT,I)=VV(1,I)*RR-QV(I)*M
1103 CONTINUE
DO 1104 J=1,IIT
VV(J,IT)=0.0D0
1104 CONTINUE
VV(IT,IT)=M
DO 1105 L=1,IIT
DO 16 I=1,IT
UU(I,IT)=UU(I,IT)+QV(L)*UU(I,L)
16 CONTINUE
1105 CONTINUE
DOTPROD=0.0D0
DO 1107 I=1,IT
DOTPROD=DOTPROD+UU(I,IT)*UU(I,IT)
1107 CONTINUE
IF(S .GT. DOTPROD) THEN
S=DOTPROD
END IF
K=IT
J=1
FLAG57=1
3001 IF(FLAG57 .EQ. 1 ) THEN
DOTPROD=0.0D0
DO 1108 I=1,IT
DOTPROD=DOTPROD+VV(I,J)*VV(I,J)
1108 CONTINUE
DO 1109 L=1,IT
DOTPROD2=0.0D0
DO 1110 I=1,IT
DOTPROD2=DOTPROD2+VV(I,L)*VV(I,J)
1110 CONTINUE
AUXILIAR=DOTPROD2/DOTPROD
QV(L)=DNINT(AUXILIAR)

```

```

AUXDOT(L)=DOTPROD2
1109 CONTINUE
DO 1111 I=1,IT
AUXILIAR=2.0D0*DABS(AUXDOT(I))
IF((I .NE. J) .AND. (AUXILIAR .GT. DOTPROD)) THEN
DO 1112 II=1,IT
VV(II,I)=VV(II,I)-QV(I)*VV(II,J)
1112 CONTINUE
DO 1113 II=1,IT
UU(II,J)=UU(II,J)+QV(I)*UU(II,I)
1113 CONTINUE
K=J
END IF
1111 CONTINUE
C PASO 6
IF(K .EQ. J) THEN
DOTPROD=0.0D0
DO 1114 I=1,IT
DOTPROD=DOTPROD+UU(I,J)*UU(I,J)
1114 CONTINUE
IF(S .GT. DOTPROD) THEN
S=DOTPROD
END IF
END IF
C PASO 7
IF(J .EQ. IT) THEN
J=1
ELSE
J=J+1
END IF
IF(K .EQ. J) THEN
FLAG57=0
END IF
GOTO 3001
END IF
C PASO 8
K=IT
DO 1115 I=1,10
X(I)=0.0D0
Y(I)=0.0D0
1115 CONTINUE
DO 1116 L=1,IT
DOTPROD=0.0D0
DO 1117 I=1,IT
DOTPROD=DOTPROD+VV(I,L)*VV(I,L)
1117 CONTINUE
AUXILIAR2=DINT(DOTPROD)
AUXILIAR2=AUXILIAR2*S/(M**2)
AUXILIAR=DSQRT(AUXILIAR2)
Z(L)=DINT(AUXILIAR)
1116 CONTINUE

C PASO 9
FLAG911=1
3002 IF(FLAG911 .EQ. 1) THEN
IF(X(K) .NE. Z(K)) THEN
X(K)=X(K)+1.0D0
DO 1118 I=1,IT
Y(I)=Y(I)+UU(I,K)
1118 CONTINUE
K=K+1

```

```

3003 IF(K .LE. IT) THEN
      X(K)=-1.0D0*Z(K)
      DO 1119 I=1,IT
      Y(I)=Y(I)-2.0D0*Z(K)*UU(K,I)
1119 CONTINUE
      K=K+1
      GOTO 3003
      END IF
      DOTPROD=0.0D0
      DO 1120 I=1,IT
      DOTPROD=DOTPROD+Y(I)*Y(I)
1120 CONTINUE
      IF(S .GT. DOTPROD) THEN
      S=DOTPROD
      END IF
      END IF
      K=K-1
      IF(K .LT. 1) THEN
      FLAG911=0
      END IF

C
      GOTO 3002
      END IF

C      IMPRESION DE LOS DATOS
      WRITE(*,*)'-----'
--
      *-----'
      WRITE(*,19)IT
      NU=SQRT(S)
      WRITE(*,210)NU
210  FORMAT(1X,T10,'NU=',D20.10)
      WRITE(*,220) 1.D0/NU
220  FORMAT(1X,T10,'1/NU=',D20.10)
      D= 1/ DSQRT(S)
      WRITE(*,26)D
260  FORMAT(1X,T10,'D=',D20.10)
      COTA=2.0**(-30.0/FLOAT(IT))
      WRITE(*,25)COTA
      IF(D.LE.COTA)THEN
      IPRUEBA=0
      ELSE
      IPRUEBA=1
      END IF
      R6=DFLOAT(IT)
      R7=R6
      R8=R7
37   IF(R8 .GT. 1.0D0) THEN
      R7=R7*(R8-1.0D0)
      R8=R8-1.0D0
      GOTO 37
      END IF
      R=((R7*M)**(1.0D0/R6)-1.0D0)*D
      WRITE(*,24)R
      IF ((IPRUEBA.EQ.1).AND.(IT.LT.4)) THEN
      WRITE(*,27)IT
      END IF
      GOTO 3000
44   RETURN
      END

C
C

```



```

C*****
C SUBROUTINE TST01
C*****
C TST01 APLICA EL TEST CHI-CUADRADO A VARIABLES SUPUESTAS CHI-CUADRADO
C Y EL TEST KOLMOGOROV-SMIRNOV A VARIABLES CON DISTRIBUCION SUPUESTA
C K-S.LAS VARIABLES SUPUESTAS COMO CHI-CUADRADO Y K-S SURGEN DE
C APLICAR
C DISTINTOS TESTS AL GENERADOR DE NUMEROS ALEATORIOS PROPUESTO POR EL
C USUARIO.(TST01 PUEDE SER LLAMADO SOLO UNA VEZ POR GENERADOR)
C
C PROPOSITO
C PRUEBA CHI-CUADRADO A VARIABLES SUPUESTAS CHI-CUADRADO Y PRUEBA
C K-S A VARIABLES CON DISTRIBUCION SUPUESTA K-S
C
C USO
C PARTE DEL PROCESO DE TST01 SE HACE EN LA RUTINA "REPORT" POR
C CONVE-
C NIENCIA DE PROGRAMACION. ANTES DE SER LLAMADA ESTA RUTINA (TST01)
C LEE LOS SIGUIENTES 6 PARAMETROS:
C
C VARIABLE LONG. NUM./CAR.
C TNAME 8 CAR
C NCHI 11 NUM
C NPRNT 11 NUM
C TCODE 8 CAR
C CSCODE 8 NUM
C KSCODE 8 NUM
C
C DESCRIPCION DE PARAMETROS
C TNAME..DEBE SER "TST01" PARA QUE SE INVOQUE LA RUTINA
C NCHI...CANTIDAD (ENTRE 1 Y 1000) DE LOTES DE NUMEROS ALEAT.(CADA
C UNO
C CONTENIENDO NBATCH NUMEROS) A LOS QUE SE APLICARA EL TEST
C ESPECIFICADO EN CSCODE O KSCODE.
C NPRNT..CANTIDAD DE LOTES PARA LOS QUE SE IMPRIMIRAN LOS
C RESULTADOS.
C (SI NCHI=NPRNT=1000 Y SE ESPECIFICAN LOS 7 TESTS, SE
C IMPRI-
C MIRAN 7000 PAGINAS). NPRNT DEBE SER RELATIVAMENTE
C PEQUE.Y.O.
C (NPRNT=10)
C PARA TENER UNA MUESTRA IMPRESA DE LOS RESULTADOS
C OBTENIDOS.
C NPRNT PUEDE SER CUALQUIER ENTERO DE 0 A NCHI
C TCODE..ES "CSCS" SI SE DESEA APLICAR EL TEST CHI-CUADRADO A LAS
C VARIABLES SUPUESTAS CHI-CUAD. "KSKS" SI SE DESEA APLICAR
C EL TEST K-S A VARIABLES CON DISTRIBUCION SUPUESTA K-S Y
C "CS2KS2" SI SE DESEA APLICAR AMBOS.
C CSCODE..INDICA CUAL DE LOS TESTS
C TST02,TST04,TST05,TST07,TST08,TST09,
C TST10 SE DESEA APLICAR.
C RESPECTIVAMENTE, "1" INDICA QUE SE SELECCIONA Y "0" QUE
C NO.
C (EJEMPLO:CSCODE=1111011 SELECCIONARA TODOS SALVO
C TST08,MIEN-
C TRAS QUE CSCODE=1111111 SELECCIONARA TODOS)
C KSCODE..INDICA SI SE APLICA TST06 Y/O TST11 PARA LAS VARIABLES
C CON
C DISTRIBUCION CONSIDERADA COMO K-S
C

```

```

C OBSERVACIONES:
C   TST01 ES UN TEST MUY IMPORTANTE, CUYO USO DEBE SER ESTUDIADO CON
CUI-
C   DADO ANTES DE SER EMPLEADO.
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   POINTS, XTRMS, CDFKS, SORT
C
C METODO:
C   REFERENCIAS:
C   1. DUDEWICZ, E.J.: "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM
C      NUMBER GENERATOR PACKAGE", TECHNICAL REPORT, DEPARTMENT OF
C      STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
C   2. DUDEWICZ, E.J. AND RALLEY, T.: "THE HANDBOOK OF RANDOM NUMBER
C      GENERATION AND TESTING", AMERICAN SCIENCES PRESS,
C      COLUMBUS, OHIO, 1981.
C*****
C
C   SUBROUTINE TST01(K1)
C     COMMON /FIRST/CHIVAL(100)
C     COMMON /SIXTH/XX(7,1000,3),YY(10000,2,3)
C     COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C     COMMON /T01PRM/NCHI,NPRNT
C     COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C     COMMON /CODE/GCODE
C     COMMON /T01VAL/INF01(10,6),NDEGF(3)
C
C     INTEGER V,TSTVAL
C     REAL*8 CHIVAL
C
C     character*8 name,messg,variab,t01nm,tstnm,cknm(10)
C     character*3 times,yes,tno
C     character*8 CHI22,HEAD11(14),HEAD21(14),HEAD31(14),
C
C     *HEAD12(14),HEAD22(14),HEAD32(14),HEAD13(14),HEAD23(14),HEAD33(14),
C
C     *HEAD14(14),HEAD24(14),HEAD34(14),HEAD15(14),HEAD25(14),HEAD35(14),
C     *HEAD16(14),HEAD26(14),HEAD36(14),HD(14,3,6)
C
C     REAL*8 CHIVAL,CHI22,HEAD11(14),HEAD21(14),HEAD31(14),
C
C     *HEAD12(14),HEAD22(14),HEAD32(14),HEAD13(14),HEAD23(14),HEAD33(14),
C
C     *HEAD14(14),HEAD24(14),HEAD34(14),HEAD15(14),HEAD25(14),HEAD35(14),
C     *HEAD16(14),HEAD26(14),HEAD36(14),HD(14,3,6)
C
C     DIMENSION KFREQ(100),CHICON(100),TSTVAL(7,3),CHI22(7)
C     DIMENSION VPLUS(3),VMINUS(3),VAL(3),PROBP(3),PROBM(3),PROB(3)
C     DIMENSION VALCK(3)
C
C     EQUIVALENCE
C
C     *(HD(1,1,1),HEAD11(1)),(HD(1,2,1),HEAD21(1)),(HD(1,3,1),HEAD31(1)),
C
C     *(HD(1,1,2),HEAD12(1)),(HD(1,2,2),HEAD22(1)),(HD(1,3,2),HEAD32(1)),
C
C     *(HD(1,1,3),HEAD13(1)),(HD(1,2,3),HEAD23(1)),(HD(1,3,3),HEAD33(1)),
C
C     *(HD(1,1,4),HEAD14(1)),(HD(1,2,4),HEAD24(1)),(HD(1,3,4),HEAD34(1)),
C
C     *(HD(1,1,5),HEAD15(1)),(HD(1,2,5),HEAD25(1)),(HD(1,3,5),HEAD35(1)),

```

```

*(HD(1,1,6),HEAD16(1)),(HD(1,2,6),HEAD26(1)),(HD(1,3,6),HEAD36(1))
C
  DATA TSTVAL/99,15,0, 5,3,6,8,
*          0,15,0, 23,5,6,99,
*          0, 5,0,119,3,6,399/
  DATA CHI22 /'D.UNIF','CUPON ',' GAP ',' PERMUT','POKER',
*'CORRID','PARSER'/
  DATA HEAD11 /
*'TESTS CO','N SEGMENT','TOS COMP','LETOS CO','NSISTENT','ES DE
EN',
*'TEROS DE',' 1 A 5 U','SANDO ','LOS NROS','. ALEAT.',' COMO
FU',
*'ERON GEN','ERADOS '/
  DATA HEAD21 /
*'TESTS CO','N SEGMENT','TOS COMP','LETOS CO','NSISTENT','ES DE
EN',
*'TEROS DE',' 1 A 5 U','SANDO LA','PARTE FR','ACCIONAR','IA DE
10',
*'0 * NUM.',' ALEAT. '/
  DATA HEAD31 /
*'TESTS CO','N SEGMENT','TOS COMP','LETOS CO','NSISTENT','ES DE
EN',
*'TEROS DE',' 1 A 10 ','USANDO ','UNA PART','ICION DE','
LONGITU',
*'D 40 O M','ENOR '/
  DATA HEAD12 /
*'TESTS CO','N INTERV','ALOS ',' DE','SDE 0.00','0 A
0.50',
*'0 - CORR','IDAS POR',' DEBAJO ','DE LA ME','DIA ','3*' '/
  DATA HEAD22 /
*'TESTS CO','N INTERV','ALOS ',' DE','SDE 0.50','0 A
1.00',
*'0 - CORR','IDAS POR',' ARRIBA ','DE LA ME','DIA ','3*' '/
  DATA HEAD32 /
*'TESTS CO','N INTERV','ALOS DE ',' ','10*' '/
  DATA HEAD13 /
*'TESTS CO','N PERMUT','ACIONES ','DE 3 OBJ','ETOS ','9*' '/
  DATA HEAD23 /
*'TESTS CO','N PERMUT','ACIONES ','DE 4 OBJ','ETOS ','9*' '/
  DATA HEAD33 /
*'TESTS CO','N PERMUT','ACIONES ','DE 5 OBJ','ETOS ','9*' '/
  DATA HEAD14 /
*'TESTS CO','N MANOS ','DE 4 CON',' 4 REPAR','TICIONES',9*' '/
  DATA HEAD24 /
*'TESTS CO','N MANOS ','DE 5 CON',' 6 REPAR','TICIONES',9*' '/
  DATA HEAD34 /
*'TESTS CO','N MANOS ','DE 5 CON',' 4 REPAR','TICIONES',9*' '/
  DATA HEAD15 /
*'TESTS US','ANDO LOS',' NUMEROS',' ALEATOR','IOS COMO',' FUERON
',
*'GENERADO','S ','6*' '/
  DATA HEAD25 /
*'TESTS US','ANDO LA ','PARTE FR','ACCIONAR','IA DE 10',' *
NUMER',
*'O ALEATO','RIO ','6*' '/
  DATA HEAD35 /
*'TESTS US','ANDO LA ','PARTE FR','ACCIONAR','IA DE 10','0 *
NUME',
*'RO ALEAT','ORIO ','6*' '/
  DATA HEAD16 /

```

```

      *'TESTS US', 'ANDO PAR', 'ES (Q,R)', ' CON Q Y', ' R ENTRE', ' 1 Y 3
      ,
      *8* ' /
      DATA HEAD26 /
      *'TESTS US', 'ANDO PAR', 'ES (Q,R)', ' CON Q Y', ' R ENTRE', ' 1 Y 10
      ,
      *8* ' /
      DATA HEAD36 /
      *'TESTS US', 'ANDO PAR', 'ES (Q,R)', ' CON Q Y', ' R ENTRE', ' 1 Y 20
      ,
      *8* ' /
      WRITE(*,*)' '
      WRITE(*,*)' '
      WRITE(*,*)' '

WRITE(*,*)'*****'
*****'
      WRITE(*,1234)
1234 FORMAT(1X,T30,'TST01 ')

WRITE(*,*)'*****'
*****'
      WRITE(*,*)' '
C
C   SI K1=9 O K1=10 LA EJECUCION BIFURCA A LA SECCION DE
PROCESAMIENTO
C   DEL TEST  KS
C
      IF(K1.GE.9) GOTO 500
C
      IF(K1.NE.3) GOTO 1
      DO 2 I=1,3
2 TSTVAL(3,I)=NDEGF(I)
      A=INF01(3,1)/1000.
      B=INF01(3,2)/1000.
C
C   TST02 TIENE SOLO UNA FORMA; LOS OTROS TESTS TIENEN 3
C   SE GENERA UNA PAGINA SEPARADA DE SALIDA PARA CADA FORMA
C
1 IF(K1.EQ.1) LIM=1
      IF(K1.GT.1) LIM=3
      KTST=K1-1
      TSTNM=CHI22(K1)
C
4 DO 400 N1=1,LIM
      V=TSTVAL(K1,N1)
      CALL POINTS(V)
      KTFREQ=0
      SUMCHI=0.
C
      DO 100 KK=1,100
100 KFREQ(KK)=0
C
C   ENCABEZADOS PARA IMPRESION DE LOS RESULTADOS DEL TEST CHI2-CHI2
C   *****
C
      WRITE(6,155)NCHI
155 FORMAT(' ',1X,'EL TEST CHI-CUADRADO SOBRE CHI-CUADRADO HA SIDO
APL
      *ICADO A ',I6, ' GRUPOS')
      WRITE(6,157) TSTNM

```

```

157 FORMAT(' ',1X,'DE VALORES CHI-CUADRADO GENERADOS POR EL TEST
',A8,
      *'Y SE HAN OBTENIDO')
      WRITE(6,158)
158 FORMAT(' ',1X,'LOS SIGUIENTES RESULTADOS:')
      WRITE(*,*)' '
      WRITE(6,160) V
160 FORMAT(' ',1X,'VALORES CHI-CUADRADO CALCULADOS CADA UNO
CON',I5,'
      *GRADOS DE LIBERTAD')
      WRITE(*,*)' '
      IF(K1.NE.1) WRITE(6,165) (HD(J,N1,KTST),J=1,14)
165 FORMAT(2X,9A8,/2X,5A8)
      IF((K1.EQ.3).AND.(N1.EQ.3)) WRITE(6,170) A,B
170 FORMAT(' ',2X,F4.3,' A ',F4.3)
C
      WRITE(*,*)' '
      WRITE(6,190)
190 FORMAT(8X,'PROB.',6X,'CHI-
2',6X,'FREC.',3X,'FREC.',2X,'CONTRIBUCIO
      *N')
      WRITE(6,200)V
200 FORMAT(4X,'J',3X,'P(J)',4X,'P(
',I5,')',3X,'ESPERADA',2X,'OBS.',6X
      *,'CHI-2',/3X,'-----')
--
      *---')
C
      EXPFQ=NCHI/100.
C
C ASIGNA VALORES XX(K1,*, N1) A INTERVALOS DETERMINADOS POR CHIVAL
C
      DO 240 J=1,NCHI
      KTFREQ=KTFREQ+1
      IF(CHIVAL(99).GE.XX(K1,J,N1))GO TO 300
      KFREQ(100)=KFREQ(100)+1
      GO TO 240
300 DO 245 JJ=1,99
      IF(CHIVAL(JJ).LT.XX(K1,J,N1))GO TO 245
      KFREQ(JJ)=KFREQ(JJ)+1
      GO TO 240
245 CONTINUE
240 CONTINUE
C
      S=0.00
C
      DO 220 J=1,50
      S=S+.01
      SS=S+.50
      JJ=J+50
C
C CHICON(J) Y CHICON(JJ) SON CONTRIBUCIONES CHI-CUADRADO DE
C INTERVALOS DETERMINADOS POR LOS VALORES CHIVAL(J) Y CHIVAL(JJ)
C
      CHICON(J)=((KFREQ(J)-EXPFQ)**2)/EXPFQ
      SUMCHI=SUMCHI+CHICON(J)
      CHICON(JJ)=((KFREQ(JJ)-EXPFQ)**2)/EXPFQ
      SUMCHI=SUMCHI+CHICON(JJ)
C
      WRITE(6,230)J,S,CHIVAL(J),EXPFQ,KFREQ(J),CHICON(J)
230 FORMAT(1X,I4,3X,F6.2,3X,F9.4,2X,F8.2,3X,I3,7X,F6.2)

```

```

220 CONTINUE
C
  WRITE(*,*)'-----'
--
  *-----'
7777
FORMAT('*****')
  WRITE(6,270)NCHI,KTFREQ,SUMCHI
270 FORMAT(5X,'TOTAL',14X,I10,5X,I7,8X,F8.2)
C ***** LLAMADO A GAMA *****
  INDGAM=INDGAM+1
  CALL GAMA(SUMCHI,INDGAM)
C
C   TEST SOBRE SUMCHI
C
  CALL POINTS(99)
  IF(SUMCHI.LT.CHIVAL(1)) GOTO 110
  IF(SUMCHI.LE.CHIVAL(99)) GOTO 104
  WRITE(6,105) CHIVAL(99)
105 FORMAT(' ','SI X ES UNA VAR.ALEAT. CHI-2 CON 99 GRADOS DE
LIBERTAD
  *, ENTONCES P(X <= ',F7.2,')=.99')
  GOTO 400
110 WRITE(6,111) CHIVAL(1)
111 FORMAT(' ','SI X ES UNA VAR.ALEAT. CHI-2 CON 99 GRADOS DE
LIBERTAD
  *, ENTONCES P(X<= ',F7.2,')=.01')
  GOTO 400
104 DO 102 K2=1,99
  IF(CHIVAL(K2).GE.SUMCHI) GOTO 109
102 CONTINUE
109 K3=K2-1
  PLOW=K3
  PLOW=PLOW/100
  PHIGH=K2
  PHIGH=PHIGH/100
  WRITE(6,103)
103 FORMAT(' ','SI X ES UNA VAR.ALEAT. CHI-2 CON 99 GRADOS DE
LIBERTAD
  *, ENTONCES')
  WRITE(6,998)CHIVAL(K3),PLOW,CHIVAL(K2),PHIGH
998 FORMAT(' ','P(X<= ',F7.2,')=' ,F3.2, ' Y P(X<= ',F7.2,')=' ,F3.2)
C
  WRITE(*,*)' '
400 CONTINUE
  GOTO 999
C
C   REALIZA CALCULOS PARA EL TESTS KS SOBRE KS
C*****
C
500 IF(K1.EQ.9) N=ICNTKS
  IF(K1.EQ.10) N=ICNTMT
  INDX=K1-8
  LBND=-180.218
C
  DO 59 I=1,3
  CALL SORT(YY(1,INDX,I),N)
C
  DO 58 J=1,N

```

```

        GOTO(52,52,54),I
52 EDSTB=-2*YY(J,INDX,I)**2
   IF(EDSTB.LT.LBND) DSTB=1.0
   IF(EDSTB.GE.LBND) DSTB=1.0-EXP(EDSTB)
   GOTO 56
54 DSTB=CDFKS(YY(J,INDX,I))
56 YY(J,INDX,I)=J/FLOAT(N)-DSTB
58 CONTINUE

C
   CALL XTRMS(YY(1,INDX,I),1,N,AMIN,AMAX)

C
   VPLUS(I)=SQRT(FLOAT(N))*AMAX
   VMINUS(I)=SQRT(FLOAT(N))*(-AMIN+1/FLOAT(N))
   VAL(I)=AMAX1(VPLUS(I),VMINUS(I))
   VALCK(I)=VAL(I)
   IF(VALCK(I).GT.50.0) VALCK(I)=50.0000

C
   IF(N.LT.100) GOTO 60

C
   EPLUS=-2*VPLUS(I)**2
   EMINUS=-2*VMINUS(I)**2

   IF(EPLUS.GT.LBND) PROBP(I)=1.-EXP(EPLUS)
   IF(EPLUS.LE.LBND) PROBP(I)=1.0000
   IF(EMINUS.GT.LBND) PROBM(I)=1.-EXP(EMINUS)
   IF(EMINUS.LE.LBND) PROBM(I)=1.0000
   PROB(I)=CDFKS(VALCK(I))

C
   IF(PROBP(I).GT.1.0) PROBP(I)=1.0000
   IF(PROBP(I).LT.0.0) PROBP(I)=.0000
   IF(PROBM(I).GT.1.0) PROBM(I)=1.0000
   IF(PROBM(I).LT.0.0) PROBM(I)=.0000
   IF(PROB(I).GT.1.0) PROB(I)=1.0000
   IF(PROB(I).LT.0.0) PROB(I)=.0000

C
59 CONTINUE

C
60 WRITE(6,502)N
502 FORMAT(2X,'EL TEST KOLMOGOROV-SMIRNOV SOBRE KOLMOGOROV-SMIRNOV
FUE
   * APLICADO A ',I5)
   GOTO(503,505),INDX
503 WRITE(6,504)
504 FORMAT(2X,'VALORES K-S GENERADOS POR LOS VALORES DE K-S EN EL
TEST
   * DE BONDAD DE AJUSTE')
   GOTO 61
505 WRITE(6,506)
506 FORMAT(2X,'VALORES GENERADOS POR LOS VALORES K-S EN EL EL TEST
DE
   *LOS MAXIMOS')
61 WRITE(6,512)
512 FORMAT(2X,'Y SE HAN PRODUCIDO LOS SIGUIENTES RESULTADOS:')
   WRITE(6,514)
514 FORMAT(/33X,'K+',11X,'K-',11X,'K')
   WRITE(6,516) VPLUS(1),VMINUS(1),VAL(1)
516 FORMAT(2X,'VALORES DE K-S SOBRE K+:',4X,G9.3,2(4X,G9.3))
   WRITE(6,518) VPLUS(2),VMINUS(2),VAL(2)
518 FORMAT(2X,'VALORES DE K-S SOBRE K-:',4X,G9.3,2(4X,G9.3))
   WRITE(6,520) VPLUS(3),VMINUS(3),VAL(3)
520 FORMAT(2X,'VALORES DE K-S SOBRE K:',5X,G9.3,2(4X,G9.3),/)

```

```

C
    IF(N.LT.100) GOTO 62
C
    WRITE(6,522)
    522 FORMAT(1X,'AQUI LA SIGNIFICACION PUEDE CALCULARSE TENIENDO EN
CUEN
    *TA LAS DISTRIBUCIONES DE',/1X,' K+, K-, K, Y USANDO LOS
VALORES:',
    */1X)
    WRITE(6,525)
    525 FORMAT(1X,'DE K-S SOBRE K+ :',T30,'DE K-S SOBRE K-:',T59,'DE K-S
S
    *OBRE K :')
C
WRITE(6,524)VPLUS(1),PROBP(1),VMINUS(1),PROBM(1),VALCK(1),PROB(1)
    524 FORMAT('0',/1X,'P(K+ <=' ,G8.3,')=' ,G8.3,',' ,2X,'P(K- <=' ,G8.3,')
=
    *',G8.3,',' ,2X,'P(K<=' ,G8.3,')=' ,G8.3)
WRITE(6,526)VPLUS(2),PROBP(2),VMINUS(2),PROBM(2),VALCK(2),PROB(2)
    526 FORMAT('0',/1X,'P(K+ <=' ,G8.3,')=' ,G8.3,',' ,2X,'P(K- <=' ,G8.3,')
=
    *',G8.3,',' ,2X,'P(K<=' ,G8.3,')=' ,G8.3)
WRITE(6,528)VPLUS(3),PROBP(3),VMINUS(3),PROBM(3),VALCK(3),PROB(3)
    528 FORMAT('0',/1X,'P(K+ <=' ,G8.3,')=' ,G8.3,',' ,2X,'P(K- <=' ,G8.3,')
=
    *',G8.3,',' ,2X,'P(K<=' ,G8.3,')=' ,G8.3)
    GOTO 999
C
    62 WRITE(6,530)
    530 FORMAT(2X,'AQUI SE PUEDE ASEGURAR LA SIGNIFICACION USANDO LA
TABLA
    * 1 DE L.H. MILLER, '/2X, '"TABLE OF PERCENTAGE POINTS OF
KOLMOGOROV
    * STATISTICS" JOURNAL OF THE',/2X, 'AMERICAN STATISTICAL
ASSOCIATION
    *, VOL. 51 (1956). (NOTAR QUE ALLI SE USA LA',/2X, 'NOTACION D,
D*.'
    *LA CORRESPONDENCIA ES K+ = SQRT(N)*D, K = SQRT(N)*(D*)',/)
C
C    FIN DE TST01
C
    999 RETURN
    END
C
C*****
C    SUBROUTINE GAMA
C*****
C
C    PROPOSITO:
C    CALCULAR LAS PROBABILIDADES  $G'=P[CHI-CUAD \leq T(I)]$ ; CHI-CUAD DIS-
C    TRIB. CON 99 GRADOS LIBERTAD; ESTADISTICO  $T(I)=CHI-CUAD.TOTAL$ 
C    ( $I=1,1000$ ) APLICADO SOBRE 100 INTERVALOS.
C    GAMA=0.01(0.01)0.99,VARIACION DE 0.01 A 0.99 POR PASO DE 0.01
C
C    USO :
C    " CALL GAMA(SUMCHI,INDGAM)"
C
C    METODO:

```



```

C
C   CALCULO VALOR APROXIMADO G' ,DE LA PROBABILIDAD GAMA, EN TABLA
C   (G,Y) , G' TAL QUE  $y \leq T(I)$ , MEDIANTE INTERPOLACION.
C   SE INTRODUCE TABLA K&D(P.430) PARA INTERPOLAR LOS VALORES
C    $y = T(I) = \text{SUMCHI}$ , SUMAS DE 100 CONTRIBUCIONES AL TEST CHI-
CUADRA-
C   DO EN CADA UNO DE LOS 19 TESTS 'CHI-CUADRADO SOBRE CHI-CUADRADO'
C   (CSCS)
C   CADA TEST RECIBE 1000 VALORES T(I) ,CADA UNO PROVENIENTE DE
10000
C   NUMEROS PSEUDOALEATORIOS OBTENIDOS POR UN GENERADOR URN**.
C   SE CONSIDERA QUE UN BUEN GENERADOR DEBE TENER ALREDEDOR DE LA
MI-
C   TAD DE LOS 19 VALORES DE G' POR ENCIMA DE 0.50. SI G'  $\geq 0.99$  EL
C   TEST FALLA Y SI  $0.95 \leq G' \leq 0.99$ , EL GENERADOR ES SOSPECHOSO.
C
C*****
C   SUBROUTINE GAMA(SUMCHI,INDGAM)
C     COMMON/GAMCHI2/TT(19,2)
C     DIMENSION TABKD(99)
C     DATA TABKD/69.2299,72.2880,74.2754,75.7949,77.0463,
*78.1226,79.0746,79.9336,80.7204,81.4493,82.1306,82.7724,83.3805,
*83.9599,84.5143,85.0469,85.5603,86.0566,86.5377,87.0052,
*87.4605,87.9048,88.3390,88.7642,89.1812,89.5907,89.9934,
*90.3898,90.7806,91.1663,91.5472,91.9238,92.2966,92.6659,93.0320,
*93.3953,93.7560,94.1145,94.4710,94.8259,95.1793,95.5315,95.8828,
*96.2333,96.5834,96.9332,97.2829,97.6329,97.9832,98.3341,98.686,
*99.039,99.393,99.749,100.106,100.465,100.827,101.191,101.558,
*101.928,102.301,102.678,103.059,103.444,103.834,104.229,104.630,
*105.036,105.449,105.868,106.296,106.731,107.175,107.629,108.093,
*108.369,109.057,109.558,110.074,110.607,111.157,111.728,112.321,
*112.939,113.585,114.263,114.978,115.735,116.542,117.407,118.342,
*119.364,120.495,121.765,123.225,124.955,127.103,129.996,134.642/
C
C   IF(SUMCHI.GT.134.642) THEN
C     WRITE(*,*)'          ERROR: VALOR SUPERIOR TABLA'
C     GAMA2=1.0
C     GOTO 44
C   ENDIF
C   IF(SUMCHI.LT.69.2299) THEN
C     WRITE (*,*)'          ERROR: VALOR INFERIOR MINIMO TABLA'
C     GAMA2=0.0
C     GOTO 44
C   ENDIF
C   XI=0.0
C   DO 100 J = 1,99
C     XI=XI+0.01
C     TA=TABKD(J)
C     IF (SUMCHI.EQ.TA) THEN
C       GAMA2 =TA
C       GOTO 44
C     ENDIF
C     IF (SUMCHI.GT.TA) THEN
C       GO TO 100
C     ELSE
C       XI=XI-0.01
C       TTKD=TABKD(J-1)
C       GAMA2 = (XI * SUMCHI) / TTKD
C       GO TO 66
C     ENDIF
100  CONTINUE

```

```

C CASO VALORES GAMA MAYORES QUE 1.0
  GAMA2=(0.99*SUMCHI)/134.642
  GOTO 44
C REDONDEO
66 PFR= GAMA2-INT(GAMA2)
  PENT=INT(PFR*1000.)
  PENT=PENT/10.
  DIFE = PENT - INT(PENT)
  IF (DIFE.EQ.0) THEN
    GAMA2 = PENT/100.
    GO TO 44
  ENDIF
  IF (10*DIFE.GE.5.0) THEN
    PENT = PENT+1.0
    GAMA2=INT(PENT)/100.
  ELSE
    GAMA2=INT(PENT)/100.
  ENDIF
C EN TT(I,1)SE GUARDAN VALORES "SUMCHI", Y EN TT(I,2) "GAMA
  APROX."
44 WRITE(*,*)'VALORES SUMCHI Y GAMA EN LINEA 2290'
  WRITE(*,*)SUMCHI,GAMA2
  TT(INDGAM,1)= SUMCHI
  TT(INDGAM,2)= GAMA2
  RETURN
  END
C
C*****
C SUBROUTINE TST02
C*****
C TST02 APLICA EL TEST CHI-CUADRADO PARA UNA DISTRIBUCION UNIFORME
C
C PROPOSITO
C DADA UNA LISTA DE N NUMEROS REALES EN PUNTO FLOTANTE ENTRE 0 Y 1
C QUE SE SUPONEN ALEATORIOS, SE DETERMINAN CUANTOS CAEN EN CADA UNO E
C DE LOS 100 INTERVALOS
C (0.00,0.01),(0.01,0.02),..., (0.98,0.99), (0.99,1.00).
C EL TEST CHI-CUADRADO (CON 99 GRADOS DE LIBERTAD) CONSISTE EN
C DETERMINAR
C SI LOS NUMEROS EN LOS INTERVALOS CONCUERDAN CON EL MODELO
C ESPECIFICADO,
C EN ESTE CASO EL MODELO DE ALEATORIEDAD BAJO EL CUAL UNO "ESPERA"
C ENCONTRAR
C N/100 NUMEROS EN CADA UNO DE LOS 100 INTERVALOS.
C
C USO
C LA CANTIDAD DE NUMEROS QUE SE ALMACENARAN ES SIEMPRE "NBATCH", EL
C ARREGLO
C ENTERO. TST02 UTILIZA LOS SIGUIENTES PARAMETROS:
C
C VARIABLE COLUMNAS LONGITUD NUMERICA/CHAR
C TNAME 1-8 8 CHAR
C
C DESCRIPCION DE PARAMETROS
C TNAME...TST02
C
C OBSERVACIONES:
C
C EL TEST SE REALIZA A LOS "NBATCH" NUMEROS.
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS:

```

```

C NINGUNA
C
C REFERENCIAS
C 1. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C GENERATION AND TESTING," AMERICAN SCIENCES PRESS,
C COLUMBUS, OHIO, 1981.
C*****
C
C SUBROUTINE TST02
C DIMENSION M(100),R(100)
C character*8 name,messg,variab,t01nm,tstnm,ckname
C REAL*8 CKNAME,MESSG,NAME,VARIAB
C COMMON /MESS/MESSG(9),VARIAB(9)
C COMMON /FOURTH/Y(10000)
C COMMON /SIXTH/XX(7,1000,3),YY(10000,2,3)
C COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C COMMON /T01PRM/NCHI,NPRNT
C COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C COMMON /CODE/GCODE
C DATA CKNAME/'TST01 '/
C
C MTOT=0
C DO 800 K1=1,100
C M(K1)=0
800 CONTINUE
C
C DO 400 K1=1,NBATCH
C K=100*(Y(K1))
C IF(K.EQ.100) K=K-1
C M(K+1)=M(K+1)+1
C MTOT=MTOT+1
400 CONTINUE
C
C EXFRE=NBATCH/100.
C XX(1,ICNT,1)=0.0
C DO 5 I=1,100
C
C R(I) IS THE CHI-SQUARE VALUE FOR THE I-TH INTERVAL
C
C R(I)=((M(I)-EXFRE)**2)/EXFRE
C XX(1,ICNT,1)=XX(1,ICNT,1)+R(I)
5 CONTINUE
C
C IF((NAME.EQ.CKNAME).AND.(ICNT.GT.NPRNT)) GOTO 170
C
C HEADINGS FOR THE PRINTOUT OF THE TEST RESULTS
C
C WRITE(*,*)' '
C WRITE(*,*)' '
C WRITE(*,*)' '
C
C WRITE(*,*)'*****'
C *****'
C
C WRITE(*,*)'*****'
C *****'
C WRITE(*,*)' '
C WRITE(*,1234)
1234 FORMAT(1X,'TST02: TEST DE DISTRIBUCION UNIFORME')
C WRITE(*,*)' '
C WRITE(6,500)

```

```

500 FORMAT(' ',1X,'EL TEST CHI-CUADRADO FUE APLICADO (RUTINA TST02)
Y
* HA PRODUCIDO')
WRITE(6,501)
501 FORMAT(' ',1X,'LOS SIGUIENTES RESULTADOS:')
WRITE(*,*)' '
WRITE(6,110) ICNT
110 FORMAT(' ',1X,'NUMERO DE MUESTRA: ',I5)
WRITE(*,2345)
2345 FORMAT(' ',1X,'-----')
WRITE(*,*)' '
WRITE(6,115)
115 FORMAT(' ', 'INDICE DE COTA SUPERIOR FRECUENCIA FRECUENCIA
CONT
*RIBUCION DE')
WRITE(6,1152)
1152 FORMAT(' ', 'INTERVALO DEL INTERVALO ESPERADA OBSERVADA
CHI-
*CUADRADO')
S=0.00
C
DO 125 I=1,100
S=S+.01
SS=S+.50
II=I+50
WRITE(6,130)I,S,EXFRE,M(I),R(I)
130 FORMAT(' ',I5,10X,F4.2,6X,F8.2,6X,I5,7X,F7.2)
125 CONTINUE
C
WRITE(*,*)'-----'
--
*----'
WRITE(6,140)NBATCH,MTOT,XX(1,ICNT,1)
140 FORMAT(' ',7X,'TOTAL:',8X,I10,6X,I7,7X,F7.2)
170 RETURN
END
C
SUBROUTINE TST03
C*****
C TST03 APLICA EL TEST DEL PRODUCTO REZAGADO (LAGGED CORRELATION)
C
C PROPOSITO
C DADA UNA LISTA DE N NUMEROS REALES EN PUNTO FLOTANTE ENTRE 0 Y 1 QUE
C SE SUPONEN ALEATORIOS, SE ESPERA QUE EL LAG L DE CORRELACION
C SERIAL SEA PEQUENIO O UNA REGLA HEURISTICA (AUN NO JUSTIFICADA
C TEORICAMENTE EN
C LA LITERATURA) ES QUE EL 95% DE LAS VECES SE PUEDE ENCONTRAR, BAJO
C ALEATORIEDAD, UN LAG L DE CORRELACION SERIAL ENTRE (2*SQRT(N)+1)/N
C Y (2*SQRT(N)-1)/N. SI ESTO NO SE VERIFICA, EL GENERADOR ES
C SOSPECHOSO.
C COMO SE HAN PROPUESTO DISTINTAS FORMULAS DE CORRELACION SERIAL
C PARA UN LAG L, CALCULAMOS 4 DE ELLAS PARA QUE EL USUARIO PUEDA
C VER COMO
C AFECTA AL RESULTADO LA FORMULA ELEGIDA; SE ESPERA QUE LA
C DIFERENCIA
C POR CAMBIO DE FORMULA SEA MINIMA.
C USO
C PARAMETROS UTILIZADOS POR TST03:
C
C VARIABLE COLUMNAS LONGITUD NUMERICA/CHAR
C TNAME 1-8 8 CHAR

```

```

C   IPARM(1)      9-19          11          NUM
C   IPARM(2)     20-30         11          NUM
C   IPARM(3)     31-80
C DESCRIPCION DE PARAMETROS
C   IPARM(1)...CANTIDAD (ENTRE IPARM(2)+2 Y NBATCH, INCLUSIVE) DE
NUMEROS
C   ALEATORIOS A SER USADOS EN EL TEST DE CORRELACION SERIAL
C   IPARM(2)...CANTIDAD DE LAGS PARA LOS QUE EL LAG DE CORRELACION
C   SERIAL L SERA CALCULADO. SE DEBE TENER 1<=IPARM(2)<=20. EL VALOR
C   POR DEFECTO ES IPARM(2)=20.
C OBSERVACIONES:
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C METODO:
C VER REFERENCIAS
C   1. DUDEWICZ, E.J. AND RALLEY, T.: "THE HANDBOOK OF RANDOM NUMBER
C   GENERATION AND TESTING," AMERICAN SCIENCES PRESS,
C   COLUMBUS, OHIO, 1981.
C   2. KNUTH, D.E.: "THE ART OF COMPUTER PROGRAMMING, VOLUME 2/
C   SEMINUMERICAL ALGORITHMS," ADDISON-WESLEY PUBLISHING COMPANY,
C   INC., READING, MASSACHUSETTS, 1969, PP. 64-65.
C*****
C
C   SUBROUTINE TST03
C
C   COMMON /MESS/MESSG(9),VARIAB(9)
C   COMMON /FOURTH/Y(10000)
C   COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C   COMMON /CODE/GCODE
C   COMMON /TSTPRM/IPARM(5)
C
C   DIMENSION CORR1(20),CORR2(20),CORR3(20),CORR4(20)
C
C   character*8 name,messg,variab,t01nm,tstnm,ckname
C   ,accept,reject,stmt
C   REAL*8 MESSG,NAME,VARIAB,ACCEPT,REJECT,STMT
C
C   DATA ACCEPT/' SI' /,REJECT/'NO' /
C
C   OPEN(UNIT=23,FILE='ARCHT03',ACCESS='SEQUENTIAL',FORM='FORMATTED')
C   READ(23,2349)IPARM(1)
C   READ(23,2349)IPARM(2)
2349  FORMAT(I11)
C   CLOSE(23)
C   N=IPARM(1)
C   LAG=IPARM(2)
C   IF((LAG.LE.0).OR.(LAG.GT.20)) LAG=20
C
C   VERIFICACION DE ERRORES
C
C   IF(N.GT.20) GOTO 2
C   WRITE(6,100)
100  FORMAT(1X,'EL VALOR DEL PARAMETRO 1 DE TST03 ES DEMASIADO CHICO
PA
*RA RESULTADOS SIGNIFICATIVOS',/1X,' TST03 SE IGNORA.')
C   GOTO 999
2   IF(N.LE.NBATCH) GOTO 3
C   WRITE(6,102) N,NBATCH

```

```

102 FORMAT(1X,'EL VALOR ',I11,' DEL PARAMETRO 1 DEL TST03 EXCEDE LA
CA
      *NTIDAD DE NUMEROS ALEATORIOS',1X/,'DISPONIBLES PARA EL
TEST.',/1X,
      *'EL VALOR HA SIDO CAMBIADO A: ',I11)
      N=NBATCH
C
3 BND=(2*SQRT(FLOAT(N))-1)/N
C
      SUM=0
      SUMSQ=0
C
      DO 4 I=1,N
      SUM=SUM+Y(I)
4 SUMSQ=SUMSQ+Y(I)**2
C
      AVE=SUM/N
C
C      FORMA CIRCULAR
C
      DO 10 L=1,LAG
      T=0
      IEND=N-L
C
      DO 12 I=1,N
      IF(I.LE.IEND) T=T+Y(I)*Y(I+L)
      IF(I.GT.IEND) T=T+Y(I)*Y(I-IEND)
12 CONTINUE
C
      CORR1(L)=(N*T-SUM**2)/(N*SUMSQ-SUM**2)
C
10 CONTINUE
C
C      FORMULA DE JENKINS Y WATTS CON UNA ALTERNATIVA CONVENIENTE
C
      DO 20 L=1,LAG
      IEND=N-L
      T=0
C
      DO 22 I=1,IEND
22 T=T+(Y(I)-AVE)*(Y(I+L)-AVE)
C
      CORR2(L)=T/(SUMSQ-N*AVE**2)
      CORR3(L)=(N/(N-L))*CORR2(L)
C
20 CONTINUE
C
C      DEFINICION BASICA
C
      DO 30 L=1,LAG
      IEND=N-L
C
      HEAD=0
      TAIL=0
C
      DO 32 I=1,L
      HEAD=HEAD+Y(I)
32 TAIL=TAIL+Y(IEND+I)
C
      SUMLH=SUM-HEAD
      SUMLT=SUM-TAIL

```

```

C
  T=0
  TSQ1=0
  TSQ2=0
C
  DO 34 I=1, IEND
  T=T+(Y(I)-SUMLT/(N-L))*(Y(I+L)-SUMLH/(N-L))
  TSQ1=TSQ1+(Y(I)-SUMLT/(N-L))**2
34 TSQ2=TSQ2+(Y(I+L)-SUMLH/(N-L))**2
C
  CORR4(L)=T/SQRT(TSQ1*TSQ2)
C
30 CONTINUE
C
  SALIDAS DE RESULTADOS
C
  WRITE(*,*)' '
  WRITE(*,*)' '
  WRITE(*,*)' '

WRITE(*,*)'*****
*****'

WRITE(*,*)'*****
*****'
  WRITE(*,*)' '
  WRITE(*,1234)
1234 FORMAT(1X,'TST03: TEST DE CORRELACION SERIAL')
  WRITE(*,*)' '
  WRITE(6,200)
  200 FORMAT(' ','EL TEST DE CORRELACION SERIAL FUE APLICADO (RUTINA
TST
  *03) ')
  WRITE(6,204)
  204 FORMAT(' Y SE HAN OBTENIDO LOS SIGUIENTES RESULTADOS:')
  WRITE(*,*)' '
  WRITE(6,208) N
  208 FORMAT(' ','LA MUESTRA CONSISTE DE ',I5,' NUMEROS ALEATORIOS')
  WRITE(*,*)' '
  WRITE(6,212)
  212
FORMAT(2X,'LAG',4X,'CORRELACION',4X,'CORRELACION',4X,'CORRELACION'
*,4X,'CORRELACION',4X,'ALEATORIEDAD')
  WRITE(6,214)
  214 FORMAT(1X,8X,'(CIRCULAR)',4X,'(JEN. Y W.)',3X,'(CONV.FORMULA)'
*,2X,'(BASIC.DEF.)')
C
  DO 40 L=1,LAG
  STMT=ACCEPT
  IF(ABS(CORR4(L)).GT.BND) STMT=REJECT
  WRITE(6,216) L,CORR1(L),CORR2(L),CORR3(L),CORR4(L),STMT
216 FORMAT(' ',1X,I2,6X,4(F7.4,8X),A8)
  40 CONTINUE
C
  WRITE(*,*)' '
  WRITE(6,218)
  218 FORMAT(' ','NOTAR QUE UNA CORRELACION ES JUZGADA "CONSISTENTE
CON
  *ALEATORIEDAD" SI Y SOLO SI')
  B1=-(2*SQRT(FLOAT(N)+1))/FLOAT(N)
  B2=(2*SQRT(FLOAT(N)-1))/FLOAT(N)

```

```

        WRITE(6,219) B1
219  FORMAT(' ','LA CORRELACION "DEFINICION BASICA" ESTA ENTRE -
(2*SQRT
      *(N)+1)/N= ',F7.4)
        WRITE(6,220) B2
220  FORMAT(1X,'Y (2*SQRT(N)-1)/N= ',F7.4)
999  RETURN
      END

C
C      SUBROUTINE TST04
C*****
C TST04 APLICA EL TEST DEL COLECCIONISTA DE CUPONES PARA PROBAR
C ALEATORIEDAD
C
C PROPOSITO:
C DADA UNA LISTA DE N NUMEROS REALES EN PUNTO FLOTANTE ENTRE 0 Y 1 QUE
C SE SUPONEN ALEATORIOS, SE CONVIERTEN A NUMEROS ENTEROS 1,2,...,D DE
C LA SIGUIENTE FORMA:
C REEMPLAZAR X POR 1 SI  $0.0 \leq X < 1/D$ 
C REEMPLAZAR X POR 2 SI  $1/D \leq X < 2/D$ 
C REEMPLAZAR X POR 3 SI  $2/D \leq X < 3/D$ 
C .
C .
C REEMPLAZAR X POR D SI  $(D-1)/D \leq X \leq 1.0$ .
C
C LUEGO LA NUEVA SUCESION {Yi} DE NUMEROS ENTEROS DEBE SER
C ALEATORIA.
C PARA COMPROBARLO APLICAMOS EL TEST DEL COLECCIONISTA DE CUPONES
C QUE
C CONSISTE EN BUSCAR SUB-SUCESIONES O SEGMENTOS DE NUMEROS ENTEROS
C DONDE
C SE PUEDA IDENTIFICAR A LOS ELEMENTOS 1,2,...,D QUE APARECEN EN
C CUALQUIER
C ORDEN EN {Yi}, MEDIANTE UNA BUSQUEDA PROGRESIVA.
C UNA VEZ COMPLETADO UN CONJUNTO {1,2,...,D} SE CUENTAN LOS
C ELEMENTOS QUE
C FUE NECESARIO OBSERVAR PARA OBTENER ESE CONJUNTO. TAL CANTIDAD SE
C LLAMA
C LONGITUD DE LA SUBSUCESION O SEGMENTO. SE RECOMIENZA CON EL
C ELEMENTO
C QUE SIGUE INMEDIATAMENTE EN {Yi} PARA FORMAR UN NUEVO CONJUNTO
C COMPLETO
C SEA Q LA CANTIDAD DE TERMINOS DE LA SUCESION QUE DEBEN EXAMINARSE
C PARA ENCONTRAR UN CONJUNTO COMPLETO 1,2,...,D. ENTONCES LOS
C POSIBLES
C VALORES DE Q SON D,D+1,D+2,... CUANDO LA VERDADERA ALEATORIEDAD
C PREVALECE, LAS PROBABILIDADES SIGUIENTES SON CONOCIDAS.
C P(Q=D)
C P(Q=D+1)
C P(Q=D+2)
C .
C .
C .
C SI SE CALCULAN TALES VALORES Q REPETIDAMENTE, UN TEST CHI-CUADRADO
C PUEDE REALIZARSE PARA VER SI LOS NUMEROS OBSERVADOS EN CADA
C CATEGORIA
C CONCUERDAN CON LAS PROBABILIDADES TEORICAS. ASI SE PRODUCE UN TEST
C DE ALEATORIEDAD DE LA SUCESION BASICA DE NUMEROS.
C SE REALIZA ESTE TEST PARA TRES CASOS:
C D=5 CON CATEGORIAS

```



```

C   Q=5
C   Q=6
C   .
C   .
C   .
C   Q=19
C   Q>=20;
C   D=5 APLICADO A FR(100X), DONDE FR(Z) DENOTA LA PARTE FRACCIONARIA
C     DE Z, CON CATEGORIAS
C   Q=5
C   Q=6
C   .
C   .
C   .
C   Q=19
C   Q>=20;
C   D=10 APLICADO A LOS X CON CATEGORIAS
C   10<=Q<=19
C   20<=Q<=23
C   24<=Q<=27
C   28<=Q<=32
C   33<=Q<=39
C     Q>=40
C
C  USO:
C  TST04 EMPLEA LOS SIGUIENTES PARAMETROS:
C    VARIABLE   COLUMNAS      LONGITUD   NUMERICA/CHAR
C    TNAME      1-8           8          CHAR
C    IPARM(1)   9-19         11         NUM
C    IPARM(2)   20-30        11         NUM
C              20-80
C
C  DESCRIPCION DE PARAMETROS
C  PARM(1)...INDICE PARA COMIENZO DE LA LISTA DE NBATCH NUMEROS ALEAT.
C  IPARM(2)...INDICE PARA EL FINAL DE LA LISTA DE NBATCH NUMEROS
C  ALEATORIOS.
C
C  SUBROUTINAS Y FUNCIONES REQUERIDAS
C    PNTCHI
C
C  METODO:
C    VER REFERENCIAS
C    1. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C      GENERATION AND TESTING," AMERICAN SCIENCES PRESS,
C      COLUMBUS, OHIO, 1981.
C    2. GREENWOOD,R.E.: "COUPON COLLECTOR'S TEST FOR RANDOM DIGITS,"
C      MATHEMATICAL TABLES AND OTHER AIDS TO COMPUTATION,
C      VOL.9(1955),
C      PAG. 1-5.
C*****
C
C    SUBROUTINE TST04
C
C    DIMENSION LSEG(16),LSEGFR(16),LSEGPR(6),ICOUPN(10)
C    DIMENSION PROB(16),PROBPR(6)
C    DIMENSION EXP(16),EXPFR(16),EXPPR(6)
C    DIMENSION DF15(11),DF5(11)
C
C    COMMON /MESS/MESSG(9),VARIAB(9)
C    COMMON /FOURTH/Y(10000)
C    COMMON /SIXTH/XX(7,1000,3),YY(10000,2,3)

```

```

COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /T01PRM/NCHI,NPRNT
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON /TSTPRM/IPARM(5)
COMMON /CODE/GCODE
COMMON /T01VAL/INF01(10,6),NDEGF(3)
C
character*8 name,messg,variab,t01nm,tstnm,ckname
*,accept,reject,stmt
c REAL*8 MESSG,VARIAB,NAME,CKNAME
C
DATA CKNAME /'TST01'/
DATA PROBPR /.17321155,.17492380,.17150818,.17134210,.15216056,
* .15685380/
DATA PROB/.03840000,.07680000,.09984000,.10752000,.10450944,
* .09547776,.08381645,.07163904,.06011299,.04979157,
* .04086200,.03331007,.02702163,.02184196,.01760857,
* .07144851/
C
OPEN(UNIT=24,FILE='ARCHT04',ACCESS='SEQUENTIAL',FORM='FORMATTED')
READ(24,2349)IPARM(1)
READ(24,2349)IPARM(2)
2349 FORMAT(I11)
CLOSE(24)
ISTART=IPARM(1)
IEND=IPARM(2)
IF(ISTART.LE.0) ISTART=1
N=IEND-ISTART+1
C
C VERIFICACION DE ERRORES
C
IF(IEND.LE.NBATCH) GOTO 2
WRITE(6,200)
200 FORMAT(1X,'EL VALOR DEL SEGUNDO PARAMETRO DE TST04 EXCEDE LA
CANTI
*DAD DE NUMEROS')
WRITE(*,*)'ALEATORIOS DISPONIBLES PARA EL TEST.'
WRITE(6,201) NBATCH
201 FORMAT(' ','LOS VALORES DEL PRIMER Y SEGUNDO PARAMETRO HA SIDO
CAM
*BIADOS A 1 Y ',I6,' RESPECTIVAMENTE.')
IEND=NBATCH
N=NBATCH
INF01(2,2)=NBATCH
2 IF(N.GT.0) GOTO 1
WRITE(6,199)
199 FORMAT(1X,'LOS PARAMETROS DE TST04 NO PROVEEN SUFICIENTES
NUMEROS
*PARA RESULTADOS ')
WRITE(*,*)'SIGNIFICATIVOS. TST04 ES IGNORADO.'
IF(NAME.EQ.CKNAME) INF01(2,6)=0
GOTO 999
C
C INICIALIZACIONES PARA CADA NUEVA APLICACION DEL TEST DEL
RECOLECTOR
C DE CUPONES
C
1 DO 10 I=1,16
LSEG(I)=0
LSEFR(I)=0

```

```

10 CONTINUE
C
DO 11 I=1,6
LSEGPR(I)=0
11 CONTINUE
C
LST=1
LSTFR=1
LSTPR=1
C
NSEG=0
NSEGFR=0
NSEGPR=0
C
MAXLEN=0
MAXFR=0
MAXPR=0
C
C
C TEST TRADICIONAL DEL RECOLECTOR DE CUPONES
C
I=ISTART-1
C
C INICIALIZACIONES PARA CADA NUEVO SEGMENTO
C
12 NFIND=0
LEN=0
DO 13 J=1,5
ICOUPN(J)=0
13 CONTINUE
C
14 I=I+1
IF(I.GT.IEND) GOTO 23
LEN=LEN+1
C
C RECOLECTAR UN CUPON DEL ARREGLO DE NUMEROS ALEATORIOS
C
IND=Y(I)*5+1
IF(IND.GT.5) IND=5
C
IF(ICOUPN(IND).NE.0) GOTO 14
C
C SE HA OBTENIDO UN NUEVO CUPON. SE DEBE REGISTRAR Y VERIFICAR
C PARA DETERMINAR SI TODOS LOS CUPONES HAN SIDO RECOLECTADOS
C
ICOUPN(IND)=1
NFIND=NFIND+1
IF(NFIND.LT.5) GOTO 14
C
C INCOMPLETO. RECOLECTAR MAS CUPONES
C RECOLECCION COMPLETA; REGISTRAR INFORMACION
C
LST=I
IF(LEN.GT.MAXLEN) MAXLEN=LEN
IF(LEN.GT.20) LEN=20
K=LEN-4
LSEG(K)=LSEG(K)+1
NSEG=NSEG+1
GOTO 12
C
C TEST DEL RECOLECTOR DE CUPONES USANDO LA PARTE FRACCIONAL DE
C 100*NUMERO ALEATORIO

```

```

C
23 I=ISTART-1
C
C   INICIALIZACIONES PARA CADA NUEVO SEGMENTO
C
24 NFIND=0
   LEN=0
   DO 25 J=1,5
   ICUPN(J)=0
25 CONTINUE
C
26 I=I+1
   IF(I.GT.IEND) GOTO 29
   LEN=LEN+1
C
C   RECOLECTAR UN CUPON DEL ARREGLO DE NUMEROS ALEATORIOS
C
   INT=100*Y(I)
   FR=100*Y(I)-INT
   IND=FR*5+1
   IF(IND.GT.5) IND=5
C
   IF(ICUPN(IND).NE.0) GOTO 26
C
C   SE HA OBTENIDO UN NUEVO CUPON. SE DEBE REGISTRAR Y VERIFICAR
C   PARA DETERMINAR SI TODOS LOS CUPONES HAN SIDO RECOLECTADOS
C
   ICUPN(IND)=1
   NFIND=NFIND+1
   IF(NFIND.LT.5) GOTO 26
C
C   INCOMPLETO. RECOLECTAR MAS CUPONES
C   RECOLECCION COMPLETA; REGISTRAR INFORMACION
C
   LSTFR=I
   IF(LEN.GT.MAXFR) MAXFR=LEN
   IF(LEN.GT.20) LEN=20
   K=LEN-4
   LSEGFR(K)=LSEGFR(K)+1
   NSEGFR=NSEGFR+1
   GOTO 24
C
C   TEST DEL RECOLECTOR DE CUPONES USANDO D=10 Y UNA PARTICION
C   DE POSIBLES LONGITUDES MENOR QUE 40
C
29 I=ISTART-1
C
C   INICIALIZACIONES PARA CADA NUEVO SEGMENTO
C
30 NFIND=0
   LEN=0
   DO 31 J=1,10
   ICUPN(J)=0
31 CONTINUE
C
32 I=I+1
   IF(I.GT.IEND) GOTO 34
   LEN=LEN+1
C
C   RECOLECTAR UN CUPON DEL ARREGLO DE NUMEROS ALEATORIOS
C

```

```

IND=10*Y(I)+1
IF(IND.GT.10) IND=10
C
IF(ICOUPN(IND).NE.0) GOTO 32
C
C SE HA OBTENIDO UN NUEVO CUPON. SE DEBE REGISTRAR Y VERIFICAR
C PARA DETERMINAR SI TODOS LOS CUPONES HAN SIDO RECOLECTADOS
C
ICOUPN(IND)=1
NFIND=NFIND+1
IF(NFIND.LT.10) GOTO 32
C
C INCOMPLETO. RECOLECTAR MAS CUPONES
C RECOLECCION COMPLETA; REGISTRAR INFORMACION
C
LSTPR=I
IF(LEN.GT.MAXPR) MAXPR=LEN
IF((LEN.GE.10).AND.(LEN.LE.19)) LSEGPR(1)=LSEGPR(1)+1
IF((LEN.GE.20).AND.(LEN.LE.23)) LSEGPR(2)=LSEGPR(2)+1
IF((LEN.GE.24).AND.(LEN.LE.27)) LSEGPR(3)=LSEGPR(3)+1
IF((LEN.GE.28).AND.(LEN.LE.32)) LSEGPR(4)=LSEGPR(4)+1
IF((LEN.GE.33).AND.(LEN.LE.39)) LSEGPR(5)=LSEGPR(5)+1
IF(LEN.GE.40) LSEGPR(6)=LSEGPR(6)+1
NSEGPR=NSEGPR+1
GOTO 30
C
C CICLO DE PROCESAMIENTO COMPLETO
C CALCULAR FRECUENCIAS ESPERADAS
C
34 DO 35 I=1,16
EXP(I)=PROB(I)*NSEG
EXPFR(I)=PROB(I)*NSEGFR
35 CONTINUE
C
DO 36 I=1,6
EXPPR(I)=PROBPR(I)*NSEGPR
36 CONTINUE
C
C CALCULAR VALORES CHI-CUADRADO
C
XX(2,ICNT,1)=0.0
XX(2,ICNT,2)=0.0
XX(2,ICNT,3)=0.0
C
DO 37 I=1,16
XX(2,ICNT,1)=XX(2,ICNT,1)+((EXP(I)-LSEG(I))**2/EXP(I))
XX(2,ICNT,2)=XX(2,ICNT,2)+((EXPFR(I)-LSEGFR(I))**2/EXPFR(I))
37 CONTINUE
C
DO 38 I=1,6
XX(2,ICNT,3)=XX(2,ICNT,3)+((EXPPR(I)-LSEGPR(I))**2/EXPPR(I))
38 CONTINUE
C
C IMPRIMIR RESULTADOS. OMITIR IMPRESION CUANDO LOS VALORES
C GENERADOS
C POR TST01 Y EL NUMERO DE MUESTRA ACTUAL EXCEDEN LA CANTIDAD DE
C TABLAS REQUERIDAS.
C
IF((CKNAME.EQ.NAME).AND.(ICNT.GT.NPRNT)) GOTO 999
C
ENCABEZADOS

```

```

C
WRITE(*,*)' '
WRITE(*,*)' '
WRITE(*,*)' '

WRITE(*,*)'*****'
*****'

WRITE(*,*)'*****'
*****'
WRITE(*,*)' '
WRITE(*,1234)
1234 FORMAT(1X,'TST04: RECOLECTOR DE CUPONES')
WRITE(*,*)' '
WRITE(6,100)
100 FORMAT(' ',1X,'EL TEST DEL RECOLECTOR DE CUPONES FUE APLICADO
(RUT
*INA TST04)')
WRITE(6,101)
101 FORMAT(' ',1X,'Y SE HAN OBTENIDO LOS SIGUIENTES RESULTADOS:')
WRITE(*,*)' '
WRITE(6,102) ICNT
102 FORMAT(' ',1X,'NUMERO DE MUESTRA: ',I5)
WRITE(*,2345)
2345 FORMAT(' ',1X,'-----')
WRITE(*,*)' '
WRITE(6,104) N
104 FORMAT(' ',1X,'LA CANTIDAD DE NUMEROS ALEATORIOS USADOS EN EL
TEST
* ES',I6)
WRITE(6,105)
105 FORMAT(' ',1X,'SOLO SEGMENTOS COMPLETOS CONSISTENTES DE ENTEROS
1
*A D HAN SIDO UTILIZADOS',/)

C
WRITE(6,*)'_____
C * _____'
WRITE(6,106)
106 FORMAT(' ',1X,'USANDO LOS NUMEROS ALEATORIOS COMO HAN SIDO
GENERAD
*OS, CON D=5: ')
WRITE(*,*)' '
WRITE(6,107) NSEG
107 FORMAT(' ',1X,'EL NUMERO DE SEGMENTOS ENCONTRADOS ES: ',I5)
WRITE(6,108) LST
108 FORMAT(' ',1X,'EL ULTIMO SEGMENTO COMPLETO TERMINA EN: ',I5)
WRITE(6,109) MAXLEN
109 FORMAT(' ',1X,'LA MAXIMA LONGITUD DE UN SEGMENTO ES: ',I5)
WRITE(*,*)' '
WRITE(6,110)
110 FORMAT(5X,'LONG. SEGMENTO',3X,'No. ESPERADO',5X,'No. OBSERVADO')
DO 50 I=1,15
J=I+4
WRITE(6,111) J,EXP(I),LSEG(I)
111 FORMAT(' ',9X,I2,10X,F9.3,10X,I5)
50 CONTINUE
WRITE(6,300) EXP(16),LSEG(16)
300 FORMAT(3X,' 20 O MAYOR ',6X,F9.3,10X,I5)
WRITE(*,*)' '
WRITE(6,122) XX(2,ICNT,1)

```

```

122 FORMAT(' ',5X,'CHI-CUADRADO PARA 15 GRADOS DE LIBERTAD
ES:',F9.3)
WRITE(6,112)
112 FORMAT(' ',5X,'PROBABILIDAD P DE QUE UNA VAR. ALEATORIA CHI-
CUADRA
*DO CON 15 GRADOS DE')
WRITE(6,113)
113 FORMAT(' ',5X,'LIBERTAD SEA <= X:')
WRITE(*,*)' '
WRITE(6,114)
114 FORMAT(' ',2X,'P|',1X,'.0001',2X,'.001',2X,'.01',2X,'.05',2X,'
.
*30',2X,'.50',2X,'.70',2X,'.95',2X,'
.99',2X,'.999',2X,'.9999')
WRITE(6,115)
115 FORMAT(' ',1X,'-----')
--
*-----')
WRITE(6,116)
116 FORMAT(' ',2X,'X|',1X,'
2.14',2X,'3.48',2X,'5.23',2X,'7.26',2X,'11
*.7',2X,'14.3',2X,'17.3',2X,'25.0',2X,'30.6',2X,'37.7',2X,'
44.3')
WRITE(*,*)' '
WRITE(6,117)
117 FORMAT(' ',1X,'RESULTADOS DEL TEST DEL RECOLECTOR DE
CUPONES
* (CONTINUACION)')
WRITE(6,119)
119 FORMAT(' ',1X,'UTILIZANDO LA PARTE FRACCIONARIA DE 100*(NROS.
ALEA
*TORIOS) CON D = 5')
WRITE(*,*)' '
WRITE(6,120) NSEGFR
120 FORMAT(' ',1X,'EL NUMERO DE SEGMENTOS ENCONTRADOS ES: ',I5)
WRITE(6,121) LSTFR
121 FORMAT(' ',1X,'EL ULTIMO SEGMENTO COMPLETO TERMINA EN: ',I5)
WRITE(6,123) MAXFR
123 FORMAT(' ',1X,'LA MAXIMA LONGITUD DE UN SEGMENTO ES: ',I5)
WRITE(*,*)' '
WRITE(6,1124)
1124 FORMAT(' ',7X,'LONG. SEGMENTO',4X,'NRO. ESPERADO',5X,'NRO.
OBSERVA
*DO')
DO 51 I=1,15
J=I+4
WRITE(6,1125) J,EXPFR(I),LSEGFR(I)
1125 FORMAT(' ',12X,I2,10X,F9.3,10X,I5)
51 CONTINUE
WRITE(6,301) EXPFR(16),LSEGFR(16)
301 FORMAT(5X,' 20 O MAYOR ',7X,F9.3,10X,I5)
WRITE(*,*)' '
WRITE(6,124) XX(2,ICNT,2)
124 FORMAT(' ',7X,'CHI-CUADRADO PARA 15 GRADOS DE LIBERTAD ES:
',F9.3)
WRITE(6,125)
125 FORMAT(' ',7X,'PROBABILIDAD P DE QUE UNA VAR. ALEATORIA CHI-
CUADRA
*DO CON 15 GRADOS')
WRITE(6,126)
126 FORMAT(' ',7X,'DE LIBERTAD SEA <= X:')

```

```

        WRITE(*,*)' '
        WRITE(6,127)
127 FORMAT(' ',2X,'P|',1X,'.0001',2X,'.001',2X,'.01',2X,'.05',2X,'
.
    *30',2X,'.50',2X,'.70',2X,'.95',2X,'
.99',2X,'.999',2X,'.9999')
        WRITE(6,128)
128 FORMAT(' ',1X,'-----')
--
    *-----')
        WRITE(6,129)
129 FORMAT(' ',2X,'X|',1X,'
2.14',2X,'3.48',2X,'5.23',2X,'7.26',2X,'11
    *.7',2X,'14.3',2X,'17.3',2X,'25.0',2X,'30.6',2X,'37.7',2X,'
44.3')
        WRITE(*,*)' '
        WRITE(6,130)
130 FORMAT(' ',1X,'RESULTADOS DEL TEST DEL RECOLECTOR DE
CUPONES
    * (CONTINUACION)')
        WRITE(6,132)
132 FORMAT(' ',1X,'USANDO UNA REPARTICION DE LONGITUD 40 O MENOR,
CON
    *D = 10')
        WRITE(*,*)' '
        WRITE(6,133) NSEGPR
133 FORMAT(' ',1X,'EL NUMERO DE SEGMENTOS ENCONTRADOS ES: ',I5)
        WRITE(6,135) LSTPR
135 FORMAT(' ',1X,'EL ULTIMO SEGMENTO COMPLETO FINALIZA EN: ',I5)
        WRITE(6,148) MAXPR
148 FORMAT(' ',1X,'LA LONGITUD MAXIMA DE UN SEGMENTO ES: ',I5)
        WRITE(*,*)' '
        WRITE(6,134)
134 FORMAT(' ',3X,'LONG. SEGMENTO',6X,'No. ESPERADO',6X,'No.
OBSERVADO
    *')
        WRITE(6,136) EXPPR(1),LSEGPR(1)
136 FORMAT(' ',4X,'10 A 19',10X,F9.3,12X,I5)
        WRITE(6,137) EXPPR(2),LSEGPR(2)
137 FORMAT(' ',4X,'20 A 23',10X,F9.3,12X,I5)
        WRITE(6,138) EXPPR(3),LSEGPR(3)
138 FORMAT(' ',4X,'24 A 27',10X,F9.3,12X,I5)
        WRITE(6,139) EXPPR(4),LSEGPR(4)
139 FORMAT(' ',4X,'28 A 32',10X,F9.3,12X,I5)
        WRITE(6,140) EXPPR(5),LSEGPR(5)
140 FORMAT(' ',4X,'33 A 39',10X,F9.3,12X,I5)
        WRITE(6,141) EXPPR(6),LSEGPR(6)
141 FORMAT(' ',3X,'40 O MAYOR',8X,F9.3,12X,I5)

        WRITE(*,*)' '
        WRITE(6,142) XX(2,ICNT,3)
142 FORMAT(' ',7X,'CHI-CUADRADO PARA 5 GRADOS DE LIBERTAD ES:
',F9.3)
        WRITE(6,143)
143 FORMAT(' ',7X,'PROBABILIDAD P DE QUE UNA VAR. ALEATORIA CHI-
CUADRA
    *DO CON 5 GRADOS')
        WRITE(6,144)
144 FORMAT(' ',7X,'DE LIBERTAD SEA <= X:')
        WRITE(*,*)' '
        WRITE(6,145)

```



```

145 FORMAT(' ',2X,'P|',1X,'.0001',2X,'.001',2X,'.01',2X,'.05',2X,'
.0
      *30',2X,'.50',2X,'.70',2X,'.95',2X,'
.99',2X,'.999',2X,'.9999')
      WRITE(6,146)
146 FORMAT(' ',1X,'-----')
--
      *-----')
      WRITE(6,147)
147 FORMAT(' ',2X,'X|',1X,'
.082',2X,'.210',2X,'.554',2X,'1.15',2X,'3.
      *00',2X,'4.35',2X,'6.06',2X,'11.1',2X,'15.1',2X,'20.5',2X,'
25.7')
      WRITE(*,*)' '
999 RETURN
      END
C
C      SUBROUTINE TST05
C*****
C TST05 APLICA EL TEST DE INTERVALOS (GAPS) O HUECOS PARA ALEATORIEDAD
C
C PROPOSITO:
C DADA UNA LISTA DE N NUMEROS REALES EN PUNTO FLOTANTE ENTRE 0 Y 1 QUE
C SE SUPONEN ALEATORIOS, SE EXAMINA CADA UNO PARA VER SI SE ENCUENTRA
C O NO ENTRE LOS NUMEROS ALFA<BETA . CADA NUMERO QUE ESTA ENTRE ALFA Y
C BETA ES REEMPLAZADO POR UN 1 Y LOS QUE NO, POR UN 0. ESTO
C CONVIERTE LOS
C N NUMEROS EN UNA SUCESION DE 0 Y 1. SEA P=BETA-ALFA. SI LOS N
C NUMEROS
C SON ALEATORIOS, LA PROBABILIDAD DE QUE OCURRAN J CEROS DESPUES DE
C UN 1 Y ANTES DEL SIGUIENTE 1 ES:
C  $P(J) = P*((1-P)**J)$ , J=0,1,2,...
C
C LUEGO DE OBSERVAR LA FRECUENCIA EN QUE OCURREN LAS CATEGORIAS, SE
C REALIZA UN TEST CHI-CUADRADO PARA VER SI LOS NUMEROS OBSERVADOS EN
C CADA CATEGORIA CONCUERDAN CON LAS PROBABILIDADES TEORICAS. ASI
C PRODUCE UN TEST DE ALEATORIEDAD DE LA SUCESION BASICA DE NUMEROS.
C KENDALL Y BABINGTON SMITH PROPUSIERON ESTE TEST EN UN CONTEXTO
C SUCECIONES
C DE LOS DIGITOS 0,1,2,3,4,5,6,7,8,9) QUE ES EQUIVALENTE A ELEGIR
C ALFA=0.0 Y BETA=0.1 .EL PROCEDIMIENTO SE LLAMA TEST "GAP". EL TEST
HA
C SIDO CODIFICADO PARA SER REALIZADO CON TRES VALORES DE ALFA,BETA):
C (ALFA,BETA) = (0.0,0.5)...LLAMADO TEST "CORRIDAS POR DEBAJO DE LA
C MEDIA"
C (ALFA,BETA) = (0.5,1.0)...LLAMADO TEST "CORRIDAS POR ARRIBA DE LA
C MEDIA"
C (ALFA,BETA) = (0.333,0.667 ...ESPECIFICADO POR EL USUARIO
C
C USO
C LOS PARAMETROS DE LA RUTINA TST05 SON:
C
C VARIABLE      COLUMNAS      LONGITUD      NUMERICA/CHAR
C TNAME         1-8             8             CHAR
C IPARM(1)      9-19            11            NUM
C IPARM(2)      20-30           11            NUM
C IPARM(3)      31-41           11            NUM
C IPARM(4)      42-52           11            NUM
C              53-80
C DESCRIPCION DE PARAMETROS

```

```

C   IPARM(1)...ENTERO 1000*ALFA, DONDE ALFA TIENE A LO SUMO 3
DECIMALES
C   IPARM(2)...ENTERO 1000*BETA, DONDE BETA TIENE A LO SUMO 3
DECIMALES
C   IPARM(3)...INDICE PARA COMENZAR EN LA LISTA DE NUMEROS ALEATORIOS
C   IPARM(4)...INDICE PARA TEMINAR EN LA LISTA DE NUMEROS ALEATORIOS
C
C OBSERVACIONES:
C EL NUMERO GN DE GAPS USADOS ES EL NUMERO (ALEATORIO) ENCONTRADO
ENTRE
C LOS (IPARM(4)-IPARM(3)+1) NUMEROS ANALIZADOS. TODAS LAS CATEGORIAS
J DONDE
C GN*P(J)>=4 SON USADAS EN EL TEST CHI-CUADRADO COMO CATEGORIAS
INDIVIDUALES
C MIENTRAS QUE EL RESTO DE LAS CATEGORIAS SON AGRUPADAS EN UNA SOLA
PARA EL
C EL TEST CHI-CUADRADO.
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   PNTCHI, POINTS
C
C REFERENCIA:
C   1. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C     GENERATION AND TESTING," AMERICAN SCIENCES PRESS,
C     COLUMBUS, OHIO, 1981.
C   2. KENDALL,M.G. AND BABINGTON SMITH, B.: "RANDOMNESS AND RANDOM
C     SAMPLING NUMBERS," JOURNAL OF THE ROYAL STATISTICAL SOCIETY,
C     VOL.101(1938), PP. 147-166.
C*****
C
C   SUBROUTINE TST05
C
C   DIMENSION LENBLW(14),LENABV(14),LEN(14),PROBMN(14),PROB(14),
*EXPBLW(14),EXPABV(14),EXP(14),HARTER(13,4)
C
C   character*8 name,messg,variab,t01nm,tstnm,ckname
* ,accept,reject,stmt
COMMON /MESS/MESSG(9),VARIAB(9)
COMMON /FIRST/CHIVAL(100)
COMMON /FOURTH/Y(10000)
COMMON /SIXTH/XX(7,1000,3),YY(10000,2,3)
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /T01PRM/NCHI,NPRNT
COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
COMMON /TSTPRM/IPARM(5)
COMMON /CODE/GCODE
COMMON /T01VAL/INF01(10,6),NDEGF(3)
REAL*8 CHIVAL
C
C   REAL*8 MESSG,VARIAB,NAME,CKNAME,CHIVAL,HARTER
C
C   DATA CKNAME/'TST01'/
DATA HARTER/
* 0.00000001, 0.00020001, 0.00521483, 0.0284185, 0.0821774,
* 0.172352, 0.299967, 0.463594, 0.660809, 0.888920,
* 1.14529, 1.42749, 1.73332,
* 0.00000157, 0.00200100, 0.0242976, 0.0908040, 0.210213,
* 0.381067, 0.598494, 0.857105, 1.15195, 1.47874,
* 1.83385, 2.21421, 2.61722,
* 10.8276, 13.8155, 16.2662, 18.4668, 20.5150,
* 22.4577, 24.3219, 26.1245, 27.8772, 29.5883,

```

```

* 31.2641,    32.9095,    34.5282,
* 15.1367,    18.4207,    21.1075,    23.5127,    25.7448,
* 27.8563,    29.8775,    31.8276,    33.7199,    35.5640,
* 37.3670,    39.1344,    40.8706/
C
C   ASIGNAR VALORES DE ENTRADA
C

OPEN(UNIT=25,FILE='ARCHT05',ACCESS='SEQUENTIAL',FORM='FORMATTED')
  READ(25,2349)IPARM(1)
  READ(25,2349)IPARM(2)
  READ(25,2349)IPARM(3)
  READ(25,2349)IPARM(4)
2349 FORMAT(I11)
  CLOSE(25)
  A=IPARM(1)/1000.
  B=IPARM(2)/1000.
  ISTART=IPARM(3)
  IEND=IPARM(4)
C
C   VERIFICACION DE ERRORES
C
  IF(ISTART.LE.0) ISTART=1
  IF(IPARM(4).LE.NBATCH) GOTO 30
  IEND=NBATCH
  INF01(3,4)=NBATCH
  WRITE(6,100) ISTART,IEND
100 FORMAT(1X,'EL VALOR DE IPARM(4) ESPECIFICADO EN TST05 EXCEDE LA
CA
*NTIDAD DE NUMEROS ALEATORIOS',/1X,'DISPONIBLES PARA EL TEST.LOS
VA
*LORES DE IPARM(3) E IPARM(4) UTILIZADOS PARA ESTE TEST SON:'
,I4,
*' E ',I5,/1X, 'RESPECTIVAMENTE. LA EJECUCION CONTINUA.')
```

```

C
30 N=IEND-ISTART+1
  LBND=N*ABS(B-A)
  IF((N.LT.8).OR.(N.LT.LBND)) GOTO 990
C
  IF(A.LE.B) GOTO 31
  TEMP=B
  B=A
  A=TEMP
C
C   INICIALIZACIONES PARA CADA NUEVA APLICACION DEL TEST GAP
C
31 NGAP=0
  NGPBLW=0
  NGPABV=0
C
  MAXGAP=0
  MAXBLW=0
  MAXABV=0
C
  LSTGAP=0
  LSTBLW=0
  LSTABV=0
C
  DO 1 I=1,14
  LEN(I)=0
  LENBLW(I)=0
```

```

1 LENABV(I)=0
C
C   INICIALIZACIONES PARA PROBABILIDADES
C
   PROBMN(1)=0.5
   PROB(1)=B-A
C
   DO 2 I=2,13
   PROBMN(I)=(0.5)**I
2  PROB(I)=PROB(1)*((1-PROB(1))**(I-1))
C
   PROBMN(14)=PROBMN(13)
   PROB(14)=(1-PROB(1))**13
C
C   ENCONTRAR LA CANTIDAD DE GAPS PARA LOS TRES TIPOS DE INTERVALOS:
C   CORRIDAS DEBAJO DE LA MEDIA
C   CORRIDAS ARRIBA DE LA MEDIA
C   UN INTERVALO SUMINISTRADO POR EL USUARIO
C
C   CICLOS PARA CORRIDAS DEBAJO DE LA MEDIA
   K=1
   DO 3 I=ISTART,IEND
   T=Y(I)
C
C   VERIFICAR PARA VER SI  $0 \leq T \leq 0.5$ . SI NO, INCREMENTAR EL CONTADOR
C   DE LONGITUD DEL GAP Y REPETIR CON EL PROXIMO NUMERO ALEATORIO.
C
   IF(T.GT.0.5) GOTO 3
C
C   SI  $0 \leq T \leq 0.5$ , SE HA ENCONTRADO UN NUEVO GAP. PROCESAR EL GAP.
C
   LSTBLW=I
   NGPBLW=NGPBLW+1
   IF(K.GT.MAXBLW) MAXBLW=K
   IF(K.GT.14) K=14
   LENBLW(K)=LENBLW(K)+1
   K=0
3  K=K+1
C
C   COMPLETA EL CICLO DE PROCESAMIENTO PARA CORRIDAS DEBAJO DE LA
MEDIA
C   CICLO PARA CORRIDAS SOBRE LA MEDIA
C
   K=1
   DO 4 I=ISTART,IEND
   T=Y(I)
C
C   VERIFICA PARA VER SI  $0.5 \leq T \leq 1$ . SI NO, INCREMENTAR EL CONTADOR
DE
C   LONGITUD DEL GAP Y REPETIR CON EL PROXIMO NUMERO ALEATORIO
C
   IF(T.LT.0.5) GOTO 4
C
C   SI  $0.5 \leq T \leq 1$ , SE HA ENCONTRADO UN NUEVO GAP. PROCESAR EL GAP.
C
   LSTABV=I
   NGPABV=NGPABV+1
   IF(K.GT.MAXABV) MAXABV=K
   IF(K.GT.14) K=14
   LENABV(K)=LENABV(K)+1
   K=0

```

```

4 K=K+1
C
C   COMPLETA EL CICLO DE PROCESAMIENTO PARA CORRIDAS SOBRE LA MEDIA
C   CICLO PARA UN INTERVALO ESPECIFICADO POR EL USUARIO
C
C   K=1
C
C   DO 5 I=ISTART,IEND
C   T=Y(I)
C
C   VERIFICA PARA VER SI A<=T<=B. SI NO, INCREMENTAR EL CONTADOR DE
C   LONGITUD DEL GAP Y REPETIR CON EL SIGUIENTE NUMERO ALEATORIO.
C
C   IF((T.LT.A).OR.(T.GT.B)) GOTO 5
C
C   SI A<=T<=B, SE HA ENCONTRADO UN NUEVO GAP. PROCESAR EL GAP.
C
C   LSTGAP=I
C   NGAP=NGAP+1
C   IF(K.GT.MAXGAP) MAXGAP=K
C   IF(K.GT.14) K=14
C   LEN(K)=LEN(K)+1
C   K=0
5 K=K+1
C
C   COMPLETA EL CICLO DE PROCESAMIENTO PARA UN INTERVALO
C   ESPECIFICADO
C   POR EL USUARIO. CALCULA FRECUENCIAS ESPERADAS PARA LONGITUDES DE
C   GAPS.
C
C   DO 6 I=1,14
C   EXPBLW(I)=NGPBLW*PROBMN(I)
C   EXPABV(I)=NGPABV*PROBMN(I)
6 EXP(I)=NGAP*PROB(I)
C
C   IF(ICNT.GT.1) GOTO 8
C
C   DESPUES DE LA PRIMER LLAMADA A TST05, SE HACE UNA VERIFICACION
C   USANDO
C   EL VALOR DE NBATCH ESPECIFICADO EN EL GENERADOR, PARA DETERMINAR LA
C   LONGITUD DEL SEGMENTO MAS PEQUE.Y.O CON FRECUENCIA ESPERADA MENOR
C   QUE 4.
C   ESTA LONGITUD SE GRABA Y SE USA EN TODAS LAS SUBSECUENTES LLAMADAS
C   A
C   TST05 EN UN TEST CHI-CUADRADO SOBRE CHI-CUADRADO. ASI, TODOS LOS
C   VALORES CHI-CUADRADO GENERADOS POR TST05 TIENEN EL MISMO NUMERO DE
C   GRADOS DE LIBERTAD. LOS GRADOS DE LIBERTAD SON ALMACENADOS EN UN
C   ARRE-
C   GLO EN COMUN, NDEGF, Y USADOS PARA ESTIPULAR LIMITES EN LAS
C   LONGITUDES
C   DE SEGMENTOS CONSIDERADOS EN TODAS LAS LLAMADAS POSTERIORES CON EL
C   MISMO GENERADOR.
C
C   DO 20 I=2,13
C   IF(EXP(I).GE.4) GOTO 20
C   LIM=I+1
C   DO 21 J=LIM,14
C   EXP(I)=EXP(I)+EXP(J)
21 LEN(I)=LEN(I)+LEN(J)
C   GOTO 22
20 CONTINUE

```

```

22 NDF=I-1
   NDEGF(3)=NDF
C
   DO 23 I=2,13
   IF(EXPBLW(I).GE.4) GOTO 23
   LIM=I+1
   DO 24 J=LIM,14
   EXPBLW(I)=EXPBLW(I)+EXPBLW(J)
24 LENBLW(I)=LENBLW(I)+LENBLW(J)
   GOTO 25
23 CONTINUE
25 NDFBLW=I-1
   NDEGF(1)=NDFBLW
C
   DO 26 I=2,13
   IF(EXPABV(I).GE.4) GOTO 26
   LIM=I+1
   DO 27 J=LIM,14
   EXPABV(I)=EXPABV(I)+EXPABV(J)
27 LENABV(I)=LENABV(I)+LENABV(J)
   GOTO 28
26 CONTINUE
28 NDFABV=I-1
   NDEGF(2)=NDFABV
   GOTO 37
C
8 LIM=NDEGF(1)+1
  IF(LIM.GT.14) GOTO 33
  DO 32 I=LIM,14
  EXP(NDEGF(1))=EXP(NDEGF(1))+EXP(I)
32 LEN(NDEGF(1))=LEN(NDEGF(1))+LEN(I)
33 LIM=NDEGF(2)+1
  IF(LIM.GT.14) GOTO 35
  DO 34 I=LIM,14
  EXPBLW(NDEGF(2))=EXPBLW(NDEGF(2))+EXPBLW(I)
34 LENBLW(NDEGF(2))=LENBLW(NDEGF(2))+LENBLW(I)
35 LIM=NDEGF(3)+1
  IF(LIM.GT.14) GOTO 37
  DO 36 I=LIM,14
  EXPABV(NDEGF(3))=EXPABV(NDEGF(3))+EXPABV(I)
36 LENABV(NDEGF(3))=LENABV(NDEGF(3))+LENABV(I)
C
C   CALCULA VALORES CHI-CUADRADO
C
37 XX(3,ICNT,1)=0.0
   XX(3,ICNT,2)=0.0
   XX(3,ICNT,3)=0.0
C
   LIM=NDEGF(1)+1
   DO 50 I=1,LIM
50 XX(3,ICNT,1)=XX(3,ICNT,1)+((EXPBLW(I)-LENBLW(I))**2/EXPBLW(I))
C
   LIM=NDEGF(2)+1
   DO 51 I=1,LIM
51 XX(3,ICNT,2)=XX(3,ICNT,2)+((EXPABV(I)-LENABV(I))**2/EXPABV(I))
C
   LIM=NDEGF(3)+1
   DO 52 I=1,LIM
52 XX(3,ICNT,3)=XX(3,ICNT,3)+((EXP(I)-LEN(I))**2/EXP(I))
C

```

```

C   IMPRIME RESULTADOS. LA IMPRESION SE OMITE CUANDO LOS VALORES
GENE-
C   RADOS PARA TST01 Y EL NUMERO DE MUESTRAS CONSIDERADAS EXCEDE LA
C   CANTIDAD DE TABLAS REQUERIDAS POR EL USUARIO.
C
C       IF((CKNAME.EQ.NAME).AND.(ICNT.GT.NPRNT)) GOTO 999
C
C   E N C A B E Z A D O S
C
C       WRITE(*,*)' '
C       WRITE(*,*)' '
C       WRITE(*,*)' '
C
WRITE(*,*)'*****'
*****'

WRITE(*,*)'*****'
*****'
C       WRITE(*,*)' '
C       WRITE(*,1234)
1234  FORMAT(1X,'TST05: TEST GAP (TEST DE LOS INTERVALOS O HUECOS)')
C
C       WRITE(*,*)' '
C       WRITE(6,200)
200  FORMAT(' ',1X,'EL TEST GAP FUE APLICADO (RUTINA TST05) Y SE HAN
OB
*TENIDO')
C       WRITE(6,201)
201  FORMAT(' ',1X,'LOS SIGUIENTES RESULTADOS:')
C       WRITE(*,*)' '
C       WRITE(6,119) ICNT
119  FORMAT(' ',1X,'NUMERO DE MUESTRA: ',I4)
C       WRITE(*,2345)
2345 FORMAT(' ',1X,'-----')
C       WRITE(*,*)' '
C
C       WRITE(6,204)
204  FORMAT(' ',3X,'INTERVALO DEL TEST 0.0 A 0.5 - CORRIDAS POR
DEBAJO
* DE LA MEDIA')
C       WRITE(*,*)' '
C       WRITE(6,205) NGPBLW
205  FORMAT(' ',7X,'LA CANTIDAD DE "GAPS" ENCONTRADOS ES: ',I5)
C       WRITE(6,231) LSTBLW
231  FORMAT(' ',7X,'EL ULTIMO GAP FINALIZA EN: ',I5)
C       WRITE(6,241) MAXBLW
241  FORMAT(' ',7X,'LA MAXIMA LONGITUD DE GAP ENCONTRADA ES: ',I5)
C       WRITE(*,*)' '
C       WRITE(6,206)
206  FORMAT(' ',3X,'LONG. GAP ',9X,'GAPS ESPERADOS',9X,'GAPS
OBSERVADOS
*')
C
C   CALL POINTS(NDFBLW)
C
C   LIM=NDFBLW+1
C   DO 40 I=1,LIM
C   J=I-1
C   WRITE(6,207) J,EXPBLW(I),LENBLW(I)
207  FORMAT(' ',6X,I2,16X,F9.3,13X,I6)
40  CONTINUE

```

```

WRITE(*,*)' '
WRITE(6,209) NDFBLW,XX(3,ICNT,1)
209 FORMAT(' ',7X,'CHI-CUADRADO PARA ',I2,' GRADOS DE LIBERTAD ES:',
*F9.3)
WRITE(6,210)
210 FORMAT(' ',7X,'PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-
CU
*ADRADO')
WRITE(6,211) NDFBLW
211 FORMAT(' ',7X,'CON ',I2,' GRADOS DE LIBERTAD SEA <=X:')
WRITE(*,*)' '
WRITE(6,212)
212 FORMAT(' ',2X,'P| ',1X,'.0001',2X,'.001',4X,'.01',3X,'.05',3X,
*'.30',4X,'.50',4X,'.70',4X,'.95',3X,'.99',3X,'.999',2X,'.9999')
WRITE(6,213)
213 FORMAT(' ',1X,'-----
--
*-----')
WRITE(6,214) (HARTER(NDFBLW,I),I=1,2),CHIVAL(1),CHIVAL(5),
*CHIVAL(30),CHIVAL(50),CHIVAL(70),CHIVAL(95),CHIVAL(99),
*(HARTER(NDFBLW,I),I=3,4)
214 FORMAT(' ',2X,'X| ',1X,F5.3,2X,F5.2,2X,3(2X,F4.2),2X,2(F5.2,2X),
*2(F4.1,2X),1X,F4.1,1X,F4.1)
C
WRITE(*,*)' '
WRITE(*,*)' '
WRITE(6,219)
219 FORMAT(' ',3X,'INTERVALO DEL TEST 0.5 A 1.0 - CORRIDAS POR
DEBAJO
* DE LA MEDIA')
WRITE(*,*)' '
WRITE(6,220) NGPABV
220 FORMAT(' ',7X,'EL NUMERO DE GAPS ENCONTRADOS ES: ',I5)
WRITE(6,48) LSTABV
48 FORMAT(' ',7X,'EL ULTIMO GAP FINALIZA EN: ',I5)
WRITE(6,49) MAXABV
49 FORMAT(' ',7X,'LA MAXIMA LONGITUD DE UN GAP ENCONTRADA ES: ',I5)
WRITE(*,*)' '
WRITE(6,221)
221 FORMAT(' ',3X,'LONG. GAP ',7X,'GAPS ESPERADOS',5X,'GAPS
OBSERVADOS
*')
CALL POINTS(NDFABV)
C
LIM=NDFABV+1
DO 41 I=1,LIM
J=I-1
WRITE(6,222) J,EXPABV(I),LENABV(I)
222 FORMAT(' ',6X,I2,14X,F9.3,10X,I6)
41 CONTINUE
WRITE(*,*)' '
WRITE(6,224) NDFABV,XX(3,ICNT,2)
224 FORMAT(' ',1X,'CHI-CUADRADO PARA ',I2,' GRADOS DE LIBERTAD ES:
',
*F9.3)
WRITE(6,225)
225 FORMAT(' ',1X,'PROBABILIDAD P DE QUE UN VARIABLE ALEATORIA CHI-
CUA
*DRADO')
WRITE(6,226) NDFABV
226 FORMAT(' ',1X,'CON ',I2,' GRADOS DE LIBERTAD SEA <=X:')

```



```

        WRITE(*,*)' '
        WRITE(6,227)
227 FORMAT(' ',1X,'P|',2X,'.0001',2X,'.001',3X,'.01',3X,'.05',3X,
        *.30',3X,'.50',3X,'.70',3X,'.95',3X,'.99',3X,'.999',3X,'.9999')
        WRITE(6,228)
228 FORMAT(' ',1X,'-----')
--
*-----')
        WRITE(6,229) (HARTER(NDFABV,I),I=1,2),CHIVAL(1),CHIVAL(5),
        *CHIVAL(30),CHIVAL(50),CHIVAL(70),CHIVAL(95),CHIVAL(99),
        *(HARTER(NDFABV,I),I=3,4)
229 FORMAT(' ',1X,'X|',1X,F5.3,2X,F5.2,1X,3(2X,F4.2),1X,2(F5.2,2X),
        *2(F4.1,2X),1X,F4.1,3X,F4.1)
C
        WRITE(*,*)' '
        WRITE(*,*)' '
        WRITE(6,234) A,B
234 FORMAT(' ',3X,'INTERVALO DEL TEST ',F5.3,' A ',F5.3)
        WRITE(*,*)' '
        WRITE(6,235) NGAP
235 FORMAT(' ',7X,'LA CANTIDAD DE GAPS ENCONTRADOS ES: ',I5)
        WRITE(6,511)LSTGAP
511 FORMAT(' ',7X,'EL ULTIMO GAP FINALIZA EN: ',I5)
        WRITE(6,521)MAXGAP
521 FORMAT(' ',7X,'LA MAXIMA LONGITUD ENCONTRADA ES: ',I5)
        WRITE(*,*)' '
        WRITE(6,236)
236 FORMAT(' ',3X,'LONG. GAP ',7X,'GAPS ESPERADOS',5X,'GAPS
OBSERVADOS
        *')
        CALL POINTS(NDF)
C
        LIM=NDF+1
        DO 42 I=1,LIM
        J=I-1
        WRITE(6,237) J,EXP(I),LEN(I)
237 FORMAT(' ',6X,I2,14X,F9.3,10X,I6)
        42 CONTINUE
        WRITE(*,*)' '
        WRITE(6,239) NDF,XX(3,ICNT,3)
239 FORMAT(' ',7X,'CHI-CUADRADO PARA ',I2,' GRADOS DE LIBERTAD ES:',
        *F9.3)
        WRITE(6,240)
240 FORMAT(' ',7X,'PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-
CU
        *ADRADO')
        WRITE(6,741) NDF
741 FORMAT(' ',7X,'CON ',I2,' GRADOS DE LIBERTAD SEA <=X:')
        WRITE(*,*)' '
        WRITE(6,242)
242 FORMAT(' ',1X,'P|',2X,'.0001',2X,'.001',3X,'.01',3X,'.05',3X,
        *.30',3X,'.50',3X,'.70',3X,'.95',3X,'.99',3X,'.999',3X,'.9999')
        WRITE(6,243)
243 FORMAT(' ',1X,'-----')
--
*-----')
        WRITE(6,244) (HARTER(NDF,I),I=1,2),CHIVAL(1),CHIVAL(5),
        *CHIVAL(30),CHIVAL(50),CHIVAL(70),CHIVAL(95),CHIVAL(99),
        *(HARTER(NDF,I),I=3,4)
244 FORMAT(' ',1X,'X|',1X,F5.3,2X,F5.2,1X,3(2X,F4.2),1X,2(F5.2,1X),
        *2(F4.1,2X),1X,F4.1,4X,F4.1)

```

```

GOTO 999
C
990 WRITE(6,900)
900 FORMAT(' ','LA CANTIDAD DE NUMEROS ALEATORIOS ESPECIFICADOS PARA
T
*ESTING ES DEMASIADO',/1X,'EXIGUA PARA RESULTADOS
SIGNIFICATIVOS.LA
* SENTENCIA QUE LLAMA A TST05 ES IGNORADA.')
C      SI LOS VALORS FUERON GENERADOS POR TST01,
C      PONER OFF EL SWITCH QUE LLAMA CHI2 SOBRE CHI2 PARA TST05
      IF(NAME.EQ.CKNAME) INF01(3,6)=0
999 RETURN
      END
C
C
C*****
C      IV
C*****
C PROPOSITO:
C      DADA UNA LISTA DE N NUMEROS REALES EN PUNTO FLOTANTE ENTRE 0 Y 1
C      QUE SE SUPONEN ALEATORIOS, EL TEST KOLMOGOROV-SMIRNOV EVALUA SU
C      ALEATORIEDAD COMPARANDO LAS DESVIACIONES DE SU FUNCION DE DISTRIBU-
C      CION, EN ESTE CASO, SUPUESTA UNIFORME EN 0.0 A 1.0, CON LA FUNCION
C      DE DISTRIBUCION EMPIRICA. ESTAS DESVIACIONES PUEDEN EVALUARSE EMPLE-
C      ANDO LAS MEDIDAS:
C
C      K+ = SQRT(N)*MAX(EMPIRICA(X)-ESTIMADA(X))
C      K- = SQRT(N)*MAX(ESTIMADA(X)-EMPIRICA(X))
C      K* = MAX(K+,K-)
C
C      TST06 CALCULA E IMPRIME K+,K-,K* CON LOS PUNTOS PORCENTUALES DE SU
C      RESPECTIVO A DISTRIBUCION
C
C USO
C      EL ARREGLO DE Nbatch NUMEROS ALEATORIOS ALMACENADOS ES USADO PARA
RE-
C      ASIGNAR VALORES AL ARREGLO ORIGINAL LUEGO DE HABER EJECUTADO TST06.
C      LOS PARAMETROS DE LA RUTINA TST06 SON:
C
C      VARIABLE          LONG.          NUM./CAR.
C      TNAME              8              CAR
C      IPARM(1)           11             NUM
C      IPARM(2)           11             NUM
C      IPARM(3)           11             NUM
C
C
C DESCRIPCION DE PARAMETROS
C      IPARM(1)...CANTIDAD (ENTRE 1 Y Nbatch) DE NUMEROS ALEAT. A SER
C      USADOS EN CADA TEST K-S
C      IPARM(2)...CANTIDAD DE CONJUNTOS DE IPARM(1) NUMEROS CADA UNO QUE
C      SERAN TESTEADOS. DEBE SER IPARM(1)*IPARM(2)<=Nbatch.
C      IPARM(3)...INDICE PARA COMENZAR EN LA LISTA DE NUMEROS ALEATORIOS
C      DEBE VERIFICARSE:IPARM(3)-1+IPARM(1)*IPARM(2) <= Nbatch.
C
C SUBRUTINAS Y FUNCIONES REQUERIDAS
C      KSTPL,CDFKS,XTRMS, SORT
C
C REFERENCIAS:
C      1. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C      GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C      COLUMBUS, OHIO, 1981.

```

```

C*****
C
SUBROUTINE TST06
COMMON /MESS/MESSG(9),VARIAB(9)
COMMON /FOURTH/Y(10000)
COMMON /SIXTH/XX(7,1000,3),YY(10000,2,3)
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /T01PRM/NCHI,NPRNT
COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
COMMON /CODE/GCODE
COMMON /TSTPRM/IPARM(5)
COMMON /T01VAL/INF01(10,6),NDEGF(3)
C
C      REAL*8 MESSG,VARIAB,NAME,CKNAME
C      character*8 name,messg,variab,t01nm,tstnm,ckname
C      *,accept,reject,stmt
C
C      DATA CKNAME/'TST01'/
C
C*****ARCHIVO DE
DATOS*****

OPEN(UNIT=26,FILE='ARCHT06',ACCESS='SEQUENTIAL',FORM='FORMATTED')
  READ(26,2349)IPARM(1)
  READ(26,2349)IPARM(2)
  READ(26,2349)IPARM(3)
2349 FORMAT(I11)
  CLOSE(26)
  N=IPARM(1)
  NREP=IPARM(2)
  NTOT=N*NREP
  ISTART=IPARM(3)
C
C      VERIFICACION DE ERRORES
C
C      IF(ISTART.LE.0) ISTART=1
C      IF(NTOT.LE.NBATCH) GOTO 4
C      WRITE(6,92) N,NREP
92 FORMAT(1X,'EL PRODUCTO DEL PRIMER PARAMETRO (LA CANTIDAD DE
NUMERO
  *S ALEATORIOS USADOS EN CADA',/1X,'TEST K-S)',I7,'POR EL 2DO
PARAME
  *TRO (LA CANT. DE TESTS INDIVIDUALES A SER
REALIZADOS)',/1X,'EX
  *CEDE LA CANTIDAD DE NUMEROS ALEATORIOS DISPONIBLES',/1X,'TST06
ES
  *IGNORADO.')
```

```

        6 IF(NAME.EQ.CKNAME) INF01(9,6)=0
          GOTO 999
C
    30 IF((NAME.EQ.CKNAME).AND.(ICNT.GT.NPRNT)) GOTO 163
C
    WRITE(*,*)' '
    WRITE(*,*)' '
    WRITE(*,*)' '

WRITE(*,*)'*****
*****'

WRITE(*,*)'*****
*****'
    WRITE(*,*)' '
    WRITE(*,1234)
1234 FORMAT(1X,'TEST06: KOLMOGOROV-SMIRNOV.')
    WRITE(*,*)' '
    WRITE(6,500)
    500 FORMAT(' ',1X,'EL TEST KOLMOGOROV-SMIRNOV FUE APLICADO (RUTINA
TST
    *06)')
    WRITE(6,401)
    401 FORMAT(' ',1X,'Y SE HAN OBTENIDO LOS SIGUIENTES RESULTADOS:')
    WRITE(*,*)' '
    WRITE(6,345)
    345 FORMAT(' ',9X,'ESPECIFICACIONES DEL USUARIO:')
    WRITE(6,856)N
    856 FORMAT(' ',1X,'LA CANTIDAD DE NUMEROS EN CADA TEST KS ES: ',I8)
    WRITE(6,344)NREP
    344 FORMAT(' ',1X,'LA CANTIDAD DE TESTS POR MUESTRA A SER EJECUTADOS
E
    *S: ',I6)
    WRITE(*,*)' '
C
C
C    AVANZAR A TRAVES DE ESTOS TESTS N VECES
C
    163 DO 20 I=1,NREP
C
C        KF ES EL INDICE DEL PRIMER NUMERO DEL I-ESIMO INTERVALO
C        KL ES EL INDICE DEL ULTIMO NUMERO DEL I-ESIMO INTERVALO
C
    KF=(I-1)*N+ISTART
    KL=KF+N-1
C
    IF((CKNAME.EQ.NAME).AND.(ICNT.GT.NPRNT)) GOTO 10
C
C
    WRITE(*,*)' '
    WRITE(*,7777)
    7777 FORMAT('-----
--
    *-----')
    WRITE(6,120) I,NREP,ICNT
    120 FORMAT(' ', 'SUBMUESTRA',I4,' (DE',I4,',') EN LA MUESTRA',I4,' DE
NUM
    *EROS ALEATORIOS')
C
    GOTO 12
C
    10 CALL SORT(Y(KF),N)

```

```

C
C           DETERMINE J/N - F(K(J))
C
12 DO 11 J=KF,KL
   Y(J)=(J-KF+1.0)/N - Y(J)
11 CONTINUE
   CALL XTRMS(Y,KF,KL,AMIN,AMAX)
C
C           MEDIANTE ECUACIONES DE KNUTH DETERMINAR K+ , K-
C
   VALPOS=SQRT(FLOAT(N))*AMAX
   VALMIN=SQRT(FLOAT(N))*(-AMIN+1.0/N)
   VAL=AMAX1(VALPOS,VALMIN)
   VALCK=VAL
   IF(VALCK.GT.50.0) VALCK=50.0
C
   EXPPOS=2*(VALPOS**2)
   EXPMIN=2*(VALMIN**2)
   IF(EXPPOS.GT.20) EXPPOS=20
   IF(EXPMIN.GT.20) EXPMIN=20
C
   PROBP=1-EXP(-EXPPOS)
   PROBM=1-EXP(-EXPMIN)
   PROB=CDFKS(VALCK)
C
   IF(PROBP.GT.1.0) PROBP=1.0000
   IF(PROBM.GT.1.0) PROBM=1.0000
   IF(PROB.GT.1.0) PROB=1.0000
C
   IF((CKNAME.EQ.NAME).AND.(ICNT.GT.NPRNT)) GOTO 18
C
   WRITE(*,*)' '
   WRITE(6,112) N,VALPOS,VALMIN,VAL
112 FORMAT(' ','EL TEST K-S CON N=',I6,' PRODUCE: K+=',F9.4,1X,'K-
='
   *,F9.4,1X,'K = ',F9.4)
   IF(N.LT.100) GOTO 16
C
   WRITE(*,*)' '
   WRITE(6,114)
114 FORMAT(' ','AQUI, LA SIGNIFICANCIA PUEDE EVALUARSE UTILIZANDO
LOS
   *HECHOS (OBTENIDOS DE',/)
   WRITE(6,115)
115 FORMAT(' ','DISTRIBUCIONES LIMITES DE K+, K-, K) DE QUE:',/)
   WRITE(6,116) VALPOS,PROBP,VALMIN,PROBM,VALCK,PROB
116 FORMAT(' ','P(K+ <= ',F8.4,' )=',G10.4,/1X ,'P(K- <= ',F8.4,'
)=',
   *G10.4,/1X,'P(K >= ',F8.4,' )=',G10.4)
   WRITE(*,*)' '
   GOTO 18
C
16 WRITE(*,*)' '
   WRITE(6,118)
118 FORMAT(' ','AQUI, LA SIGNIFICANCIA PUEDE EVALUARSE USANDO LA
TABLA
   *1 DE L.H.MILLER,"TABLE OF',/1X,'PERCENTAGE POINTS OF KOLMOGOROV
ST
   *ATISTICS"', '- JOURNAL OF THE AMERICAN
STATISTICAL',/1X,'ASSOCIATIO

```

```

      *N,VOL.51(1956),PAG.111-112.',/1X,'NOTAR QUE ALLI SE USA LA
NOTACIO
      *N D,D*. LA CORRESPONDENCIA ES:',/1X,'  K+ = SQRT(N/T)*D,  K =
SQR
      *T(N/T)*(D*).' )
C
      18 IF(I.GT.10) GOTO 20
          ICNTKS=ICNTKS+1
          YY(ICNTKS,1,1)=VALPOS
          YY(ICNTKS,1,2)=VALMIN
          YY(ICNTKS,1,3)=VAL
C
      20 CONTINUE
      999 RETURN
          END
C
C      SUBROUTINE TST07
C*****
C TST07 APLICA EL TEST DE PERMUTACION PARA PROBAR ALEATORIEDAD
C
C PROPOSITO:
C DADA UNA LISTA DE N NUMEROS REALES EN PUNTO FLOTANTE ENTRE 0 Y 1 QUE
C SE SUPONEN ALEATORIOS, SE TOMAN SUCEIVOS CONJUNTOS DE T NUMEROS.
C CADA UNO DE ESOS CONJUNTOS TIENE T-FACTORIAL ORDENES POSIBLES
C (CUANDO
C CLASIFICAMOS LOS T NUMEROS COMO MAYOR, SEGUNDO MAYOR,..., MENOR),
C Y CUANDO PREVALECE ALEATORIEDAD VERDADERA, LAS PROBABILIDADES DE
C LOS
C T-FACTORIAL ORDENES SON IGUALES A 1/(T-FACTORIAL). LUEGO DE
C OBSERVAR
C LA FRECUENCIA EN QUE OCURRE CADA ORDENAMIENTO CUANDO SE EXAMINAN
C MUCHOS GRUPOS DE T ENTEROS, SE REALIZA UN TEST CHI-CUAD. PARA VER
C SI LOS NUMEROS OBSERVADOS EN CADA CATEGORIA ESTAN DE ACUERDO CON
C IGUALES
C PROBABILIDADES PARA LAS CATEGORIAS, ASI SE PRODUCE UN TEST DE
C ALEATORIEDAD DE LA SUCESION DE NUMEROS.
C ESTE TEST ES REALIZADO PARA T=3,4,5.
C
C LOS PARAMETROS DE TST07 SON:
C   VARIABLE   COLUMNAS   LONGITUD NUMERICA/CHAR
C   TNAME      1-8         8           CHAR
C   IPARM(1)   9-19      11          NUM
C   IPARM(2)   20-30     11          NUM
C               31-80
C
C DESCRIPCION DE PARAMETROS
C   IPARM(1)...INDICE DEL PRIMER NUMERO EN LA LISTA DE NUMALEA
NUMEROS
C           ALEATORIOS.
C   IPARM(2)...INDICE DEL ULTIMO NUMERO EN LA LISTA DE NUMALEA
NUMEROS
C           ALEATORIOS.
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS:
C PNTCHI
C
C REFERENCIAS:
C   1. DUDEWICZ,E.J. Y RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C     GENERATION AND TESTING," AMERICAN SCIENCES PRESS,
C     COLUMBUS, OHIO, 1981.
C*****

```

```

C
SUBROUTINE TST07
C
DIMENSION NTYP3(6),NTYP4(24),NTYP5(120),PERM(5),MATRIX(120)
C
COMMON /MESS/MESSG(9),VARIAB(9)
COMMON /FOURTH/Y(10000)
COMMON /SIXTH/XX(7,1000,3),YY(10000,2,3)
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /T01PRM/NCHI,NPRNT
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON /TSTPRM/IPARM(5)
COMMON /CODE/GCODE
COMMON /T01VAL/INF01(10,6),NDEGF(3)
C
REAL*8 MESSG,VARIAB,NAME,CKNAME
character*8 name,messg,variab,t01nm,tstnm,ckname,accept,
*reject,stmt
C
DATA CKNAME/'TST01'/
C
OPEN(UNIT=27,FILE='ARCHT07',ACCESS='SEQUENTIAL',FORM='FORMATTED')
READ(27,2349)IPARM(1)
READ(27,2349)IPARM(2)
2349 FORMAT(I11)
CLOSE(27)
ISTART=IPARM(1)
IEND=IPARM(2)
IF(ISTART.LE.0) ISTART=1
IF(IEND.LE.NBATCH) GOTO 7
WRITE(6,790) IPARM(2),NBATCH
790 FORMAT(1X,'EL SEGUNDO PARAMETRO DE TST07 ',I6,' EXCEDE EL LIMITE
E
*N EL ARREGLO DE NUMEROS ALEATORIOS.',/1X,'EL VALOR HA SIDO
CAMBIAD
*O A: ',I6,'. LA EJECUCION CONTINUA.')
C
IEND=NBATCH
INF01(4,2)=NBATCH
C
7 NUM=IEND-ISTART+1
C
IF(NUM.LE.NBATCH) GOTO 8
WRITE(6,800)
800 FORMAT(1X,'LA CANTIDAD DE NUMEROS ALEATORIOS REQUERIDOS PARA EL
TE
*ST DE PERMUTACION EXCEDE EL NUMERO ',/1X,'DISPONIBLE PARA EL
TEST.
*',/1X,'TST07 ES IGNORADO. LA EJECUCION CONTINUA.')
C
SI LOS VALORES FUERON GENERADOS PARA TST01,
C
"APAGAR" EL SWITCH QUE LLAMA A CS SOBRE CS PARA TST07
C
IF(NAME.EQ.CKNAME) INF01(4,6)=0
GOTO 999
C
8 N3TOT=NUM/3
N4TOT=NUM/4
N5TOT=NUM/5
C

```

```

C      INICIALIZACIONES PARA CADA NUEVA APLICACION DEL TEST
C      DE PERMUTACION
C
      DO 10 I=1,6
10     NTYP3(I)=0
C
      DO 11 I=1,24
11     NTYP4(I)=0
C
      DO 12 I=1,120
12     NTYP5(I)=0
C
      NONPM3=0
      NONPM4=0
      NONPM5=0
C
C      PERMUTACIONES DE GRADO 3
C
      DO 18 I=1,N3TOT
C
      DO 13 J=1,3
      K=(I-1)*3+J
13     PERM(J)=Y(K)
C
C      VERIFICAR QUE LA 3-TUPLA NO TIENE ENTRADAS REPETIDAS Y ENCONTRAR
C      EL INDICE DE LA MAXIMA ENTRADA DE LA TUPLA
C
      LIM=3
14     RMAX=PERM(1)
      INDX=1
C
      DO 16 J=2,LIM
C
C      BIFURCAR SI LA ENTRADA EN PERM NO ES IGUAL AL ACTUAL MAXIMO
C
      IF(PERM(J).LT.RMAX) GOTO 16
      IF(PERM(J).GT.RMAX) GOTO 15
      NONPM3=NONPM3+1
      GOTO 18
15     RMAX=PERM(J)
      INDX=J
16     CONTINUE
C
C      REGISTRAR EL INDICE DE LA ENTRADA DONDE OCURRE EL MAXIMO (RESTAR
1
C      PARA PROPOSITOS COMPUTACIONALES) Y CAMBIAR EL MAXIMO POR LA
ENTRA-
C      DA EN LA ULTIMA POSICION ACTUAL, PERM(LIM).
C
      MATRIX(LIM)=INDX-1
      IF(LIM.EQ.INDX) GOTO 35
      PERM(INDX)=PERM(LIM)
      PERM(LIM)=RMAX
35     LIM=LIM-1
      IF(LIM.GT.1) GOTO 14
      MATRIX(1)=0
C
C      CALCULAR EL VALOR DE LA FUNCION PARA LA 3-TUPLA ACTUAL Y
REGISTRARLO
C
      IVAL=MATRIX(1)

```



```

DO 17 J=2,3
17 IVAL=IVAL*J+MATRIX(J)
C
    IVAL=6-IVAL
    NTYP3(IVAL)=NTYP3(IVAL)+1
18 CONTINUE
C
C   PERMUTACIONES DE GRADO 4
C
    DO 24 I=1,N4TOT
C
    DO 19 J=1,4
    K=(I-1)*4+J
19 PERM(J)=Y(K)
C
C   VERIFICAR QUE LA 4-TUPLA NO TIENE ENTRADAS REPETIDAS Y ENCONTRAR
C   EL INDICE DE LA MAXIMA ENTRADA DE LA TUPLA
C
    LIM=4
20 RMAX=PERM(1)
    INDX=1
C
    DO 22 J=2,LIM
C   BIFURCAR SI LA ENTRADA EN PERM NO ES IGUAL AL ACTUAL MAXIMO
    IF(PERM(J).LT.RMAX) GOTO 22
    IF(PERM(J).GT.RMAX) GOTO 21
    NONPM4=NONPM4+1
    GOTO 24
21 RMAX=PERM(J)
    INDX=J
22 CONTINUE
C
C   REGISTRAR EL INDICE DE LA ENTRADA DONDE OCURRE EL MAXIMO (RESTAR
1
C   PARA PROPOSITOS COMPUTACIONALES) Y CAMBIAR EL MAXIMO POR LA
ENTRA-
C   DA EN LA ULTIMA POSICION ACTUAL, PERM(LIM).
C
    MATRIX(LIM)=INDX-1
    IF(LIM.EQ.INDX) GOTO 37
    PERM(INDX)=PERM(LIM)
    PERM(LIM)=RMAX
37 LIM=LIM-1
    IF(LIM.GT.1) GOTO 20
    MATRIX(1)=0
C
C   CALCULAR EL VALOR DE LA FUNCION PARA LA 4-TUPLA ACTUAL Y
REGISTRARLO
C
    IVAL=MATRIX(1)
    DO 23 J=2,4
23 IVAL=IVAL*J+MATRIX(J)
C
    IVAL=24-IVAL
    NTYP4(IVAL)=NTYP4(IVAL)+1
24 CONTINUE
C
C   PERMUTACIONES DE GRADO 5
C
    DO 30 I=1,N5TOT
C

```

```

DO 25 J=1,5
K=(I-1)*5+J
25 PERM(J)=Y(K)
C
C VERIFICAR QUE LA 5-TUPLA NO TIENE ENTRADAS REPETIDAS Y ENCONTRAR
C EL INDICE DE LA MAXIMA ENTRADA DE LA TUPLA
C
LIM=5
26 RMAX=PERM(1)
INDX=1
C
DO 28 J=2,LIM
C
C BIFURCAR SI LA ENTRADA EN PERM NO ES IGUAL AL ACTUAL MAXIMO
C
IF(PERM(J).LT.RMAX) GOTO 28
IF(PERM(J).GT.RMAX) GOTO 27
NONPM5=NONPM5+1
GOTO 30
27 RMAX=PERM(J)
INDX=J
28 CONTINUE
C
C REGISTRAR EL INDICE DE LA ENTRADA DONDE OCURRE EL MAXIMO (RESTAR
1
C PARA PROPOSITOS COMPUTACIONALES) Y CAMBIAR EL MAXIMO POR LA
ENTRA-
C DA EN LA ULTIMA POSICION ACTUAL, PERM(LIM).
C
MATRIX(LIM)=INDX-1
IF(LIM.EQ.INDX) GOTO 39
TEMP=PERM(LIM)
PERM(LIM)=PERM(INDX)
PERM(INDX)=TEMP
39 LIM=LIM-1
IF(LIM.GT.1) GOTO 26
MATRIX(1)=0
C
C CALCULAR EL VALOR DE LA FUNCION PARA LA 5-TUPLA ACTUAL Y
REGISTRARLO
C
IVAL=MATRIX(1)
DO 29 J=2,5
29 IVAL=IVAL*J+MATRIX(J)
C
IVAL=120-IVAL
NTYP5(IVAL)=NTYP5(IVAL)+1
30 CONTINUE
C
C NUMERO TOTAL DE PERMUTACIONES ACTUALES
C
N3=N3TOT-NONPM3
N4=N4TOT-NONPM4
N5=N5TOT-NONPM5
C
C CALCULAR VALORES CHI-CUADRADO
C
EXP3=FLOAT(N3)/6.
EXP4=FLOAT(N4)/24.
EXP5=FLOAT(N5)/120.
C

```

```

        XX(4,ICNT,1)=0.0
        XX(4,ICNT,2)=0.0
        XX(4,ICNT,3)=0.0
C
        DO 31 I=1,6
31 XX(4,ICNT,1)=XX(4,ICNT,1)+((EXP3-NTYP3(I))**2/EXP3)
C
        DO 32 I=1,24
32 XX(4,ICNT,2)=XX(4,ICNT,2)+((EXP4-NTYP4(I))**2/EXP4)
C
        DO 33 I=1,120
33 XX(4,ICNT,3)=XX(4,ICNT,3)+((EXP5-NTYP5(I))**2/EXP5)
C
C      IMPRIMIR RESULTADOS. OMITIR IMPRESION CUANDO LOS VALORES SEAN
C      GENERADOS PARA TST01 Y EL NUMERO DE MUESTRA ACTUAL EXCEDE EL
C      NUMERO DE TABLAS REQUERIDAS
C
        IF((CKNAME.EQ.NAME).AND.(ICNT.GT.NPRNT)) GOTO 999
C
        ENCABEZAMIENTOS
C
        WRITE(*,*)' '
        WRITE(*,*)' '
        WRITE(*,*)' '

WRITE(*,*)'*****'
        *****'

WRITE(*,*)'*****'
        *****'
        WRITE(*,*)' '
        WRITE(*,1234)
1234 FORMAT(1X,'TST07: TEST DE PERMUTACION')
        WRITE(*,*)' '
        WRITE(6,100)
        100 FORMAT(' ',1X,'EL TEST DE PERMUTACION FUE APLICADO (RUTINA
TST07)
        *Y HA PRODUCIDO')
        WRITE(6,101)
101 FORMAT(' ',1X,'LOS SIGUIENTES RESULTADOS:')
        WRITE(*,*)' '
        WRITE(6,102) ICNT
102 FORMAT(' ',5X,'NUMERO DE MUESTRA: ',I5)
        WRITE(*,2345)
2345 FORMAT(' ',5X,'-----')
        WRITE(6,103) MESSG
103 FORMAT(3X,9A8)
        WRITE(6,104) NUM
104 FORMAT(' ',1X,'LA CANTIDAD DE NUMEROS ALEATORIOS UTILIZADOS EN
EL
        *TESTING ES:',I6
        *)
        WRITE(*,*)' '
        WRITE(*,*)' '
        WRITE(6,106)
106 FORMAT(' ',7X,'PERMUTACIONES DE 3 OBJETOS')
        WRITE(*,*)' '
        WRITE(6,107) N3TOT
107 FORMAT(' ',1X,'LA CANTIDAD DE GRUPOS DE 3 NUMEROS EXAMINADOS ES:
        ',I5)

```

```

        WRITE(6,117)EXP3
117 FORMAT(' ',1X,'LA FRECUENCIA ESPERADA DE CADA TIPO DE
PERMUTACION
        *ES: ',F9.3)
        WRITE(6,99)
119 FORMAT(' ',1X,'LA CANTIDAD DE TUPLAS DE 3 QUE FALLAN PARA SER
PERM
        *UTACIONES (Y ENTONCES ')
        WRITE(6,119)NONPM3
119 FORMAT(' ',1X,'SON EXCLUIDAS DE CONSIDERACION) ES: ',I3)
        WRITE(*,*)' '
        WRITE(6,108)
108 FORMAT(1X,'PERMUTACION:',1X,'(1 2 3)',2X,'(1 3 2)',2X,'(3 2
1)',2X
        *,'(2 1 3)',2X,'(2 3 1)',2X,'(3 1 2)')
        WRITE(6,109) (NTYP3(I),I=1,6)
109 FORMAT(1X,'OCURRENCIAS:',3X,I4,5(5X,I4))
        WRITE(*,*)' '
        WRITE(6,110) XX(4,ICNT,1)
110 FORMAT(1X,'CHI-CUADRADO CON 5 GRADOS DE LIBERTAD ES: ',F9.3)
        WRITE(6,111)
111 FORMAT(2X,'PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-
CUADRA
        *DO')
        WRITE(6,112)
112 FORMAT(2X,'CON 5 GRADOS DE LIBERTAD SEA <= X:')
        WRITE(*,*)' '
        WRITE(6,113)
113 FORMAT(' ',1X,'P|',1X,'.0001',2X,'.001',2X,'.01',2X,'.05',2X,'
.
        *30',2X,'.50',2X,'.70',2X,'.95',2X,'
.99',2X,'.999',2X,'.9999')
        WRITE(6,114)
114 FORMAT(' ',1X,'-----
--
        *-----')
        WRITE(6,115)
115 FORMAT(' ',1X,'X|',1X,'
.082',2X,'.210',2X,'.554',2X,'1.15',2X,'3.
        *00',2X,'4.35',2X,'6.06',2X,'11.1',2X,'15.1',2X,'20.5',2X,'
25.7')
00003420
        WRITE(*,*)' '
        WRITE(*,*)' '
        WRITE(6,118)
118 FORMAT(' ',7X,'PERMUTACIONES DE 4 OBJETOS')
        WRITE(*,*)' '
        WRITE(6,120) N4TOT
120 FORMAT(' ',1X,'LA CANTIDAD DE GRUPOS DE 4 NUMEROS EXAMINADOS ES:
,
        *,I5)
        WRITE(6,121)EXP4
121 FORMAT(' ',1X,'LA FRECUENCIA ESPERADA DE CADA TIPO DE
PERMUTACION
        *ES: ',F9.3)
        WRITE(6,122)
122 FORMAT(' ',1X,'LA CANTIDAD DE TUPLAS DE 4 QUE FALLAN PARA SER
PERM
        *UTACIONES (Y ENTONCES ')
        WRITE(6,123)NONPM4
123 FORMAT(' ',1X,'SON EXCLUIDAS DE CONSIDERACION) ES: ',I3)

```

```

        WRITE(*,*)' '
        WRITE(6,124)
124 FORMAT(1X,'PERMUTACION:',1X,'(1 2 3 4)',1X,'(1 2 4 3)',1X,'(1 4
3
*2)',1X,'(4 2 3 1)',1X,'(1 3 2 4)',1X,'(1 3 4 2)')
        WRITE(6,126) (NTYP4(I),I=1,6)
126 FORMAT(1X,'OCURRENCIAS:',4X,I4,5(6X,I4))
        WRITE(*,*)' '
        WRITE(6,128)
128 FORMAT(1X,'PERMUTACION:',1X,'(1 4 2 3)',1X,'(4 3 2 1)',1X,'(3 2
1
*4)',1X,'(3 2 4 1)',1X,'(3 4 1 2)',1X,'(4 2 1 3)')
        WRITE(6,130) (NTYP4(I),I=7,12)
130 FORMAT(1X,'OCURRENCIAS:',4X,I4,5(6X,I4))
        WRITE(*,*)' '
        WRITE(6,132)
132 FORMAT(1X,'PERMUTACION:',1X,'(2 1 3 4)',1X,'(2 1 4 3)',1X,'(2 4
3
*1)',1X,'(4 1 3 2)',1X,'(2 3 1 4)',1X,'(2 3 4 1)')
        WRITE(6,134) (NTYP4(I),I=13,18)
134 FORMAT(1X,'OCURRENCIAS:',4X,I4,5(6X,I4))
        WRITE(*,*)' '
        WRITE(6,136)
136 FORMAT(1X,'PERMUTACION:',1X,'(2 4 1 3)',1X,'(4 3 1 2)',1X,'(3 1
2
*4)',1X,'(3 1 4 2)',1X,'(3 4 2 1)',1X,'(4 1 2 3)')
        WRITE(6,138) (NTYP4(I),I=19,24)
138 FORMAT(1X,'OCURRENCIAS:',4X,I4,5(6X,I4))
        WRITE(*,*)' '
        WRITE(6,140) XX(4,ICNT,2)
140 FORMAT(' ',3X,'CHI-CUADRADO CON 23 GRADOS DE LIBERTAD ES:
',F9.3)
        WRITE(6,142)
142 FORMAT(' ',3X,'PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-
CU
*ADRADO')
        WRITE(6,144)
144 FORMAT(' ',3X,'CON 23 GRADOS DE LIBERTAD SEA <= X:')
        WRITE(*,*)' '
        WRITE(6,146)
146 FORMAT(' ',1X,'P |',2X,'.0001',2X,'.001',3X,'.01',3X,'.05',3X,
* '.30',3X,'.50',3X,'.70',3X,'.95',3X,'.99',3X,'.999',3X,'.9999')
        WRITE(6,148)
148 FORMAT(' ',1X,'-----
--
*-----')
        WRITE(6,150)
150 FORMAT(' ',1X,'X |',2X,'5.75',2X,'7.53',3X,'10.2',2X,'13.1',2X,
* '19.1',2X,'22.3',2X,'26.0',2X,'35.2',2X,'41.6',3X,'49.7',3X,
* '57.1')
        WRITE(*,*)' '
        WRITE(*,*)' '
        WRITE(6,162)
162 FORMAT(' ',7X,'PERMUTACIONES DE 5 OBJETOS')
        WRITE(*,*)' '
        WRITE(6,164) N5TOT
164 FORMAT(' ',1X,'LA CANTIDAD DE GRUPOS DE 5 NUMEROS EXAMINADOS ES:
,
*,I5)
        WRITE(6,165)EXP5

```

```

165 FORMAT(' ',1X,'LA FRECUENCIA ESPERADA DE CADA TIPO DE
PERMUTACION
      *ES: ',F9.3)
      WRITE(6,166)
166 FORMAT(' ',1X,'LA CANTIDAD DE TUPLAS DE 5 QUE FALLAN PARA SER
PERM
      *UTACIONES (Y ENTONCES ')
      WRITE(6,167)NONPM5
167 FORMAT(' ',1X,'SON EXCLUIDAS DE CONSIDERACION) ES: ',I3)
      WRITE(*,*)' '
      WRITE(6,170)
170 FORMAT(1X,'PERMUT.:(1 2 3 4 5)',1X,'(1 2 3 5 4)',1X,'(1 2 5 4
3)',
      *1X,'(1 5 3 4 2)',1X,'(5 2 3 4 1)',1X,'(1 2 4 3 5)')
      WRITE(6,172) (NTYP5(I),I=1,6)
172 FORMAT(1X,'OCUR.: ',2X,I5,5(7X,I5))
      WRITE(*,*)' '
      WRITE(6,174)
174 FORMAT(1X,'PERMUT.:(1 2 4 5 3)',1X,'(1 2 5 3 4)',1X,'(1 5 4 3
2)',
      *1X,'(5 2 4 3 1)',1X,'(1 4 3 2 5)',1X,'(1 4 3 5 2)')
      WRITE(6,176) (NTYP5(I),I=7,12)
176 FORMAT(1X,'OCUR.: ',2X,I5,5(7X,I5))
      WRITE(*,*)' '
      WRITE(6,177)
177 FORMAT(1X,'PERMUT.:(1 4 5 2 3)',1X,'(1 5 3 2 4)',1X,'(5 4 3 2
1)',
      *1X,'(4 2 3 1 5)',1X,'(4 2 3 5 1)',1X,'(4 2 5 1 3)')
      WRITE(6,178) (NTYP5(I),I=13,18)
178 FORMAT(1X,'OCUR.: ',2X,I5,5(7X,I5))
      WRITE(*,*)' '
      WRITE(6,180)
180 FORMAT(1X,'PERMUT.:(4 5 3 1 2)',1X,'(5 2 3 1 4)',1X,'(1 3 2 4
5)',
      *1X,'(1 3 2 5 4)',1X,'(1 3 5 4 2)',1X,'(1 5 2 4 3)')
      WRITE(6,182) (NTYP5(I),I=19,24)
182 FORMAT(1X,'OCUR.: ',2X,I5,5(7X,I5))
      WRITE(*,*)' '
      WRITE(6,184)
184 FORMAT(1X,'PERMUT.:(5 3 2 4 1)',1X,'(1 3 4 2 5)',1X,'(1 3 4 5
2)',
      *1X,'(1 3 5 2 4)',1X,'(1 5 4 2 4)',1X,'(5 3 4 2 1)')
      WRITE(6,186) (NTYP5(I),I=25,30)
186 FORMAT(1X,'OCUR.: ',2X,I5,5(7X,I5))
      WRITE(*,*)' '
      WRITE(6,188)
188 FORMAT(1X,'PERMUT.:(1 4 3 2 5)',1X,'(1 4 3 2 5)',1X,'(1 4 5 3
2)',
      *1X,'(1 5 2 3 4)',1X,'(5 4 2 3 1)',1X,'(4 3 2 1 5)')
      WRITE(6,190) (NTYP5(I),I=31,36)
190 FORMAT(1X,'OCUR.: ',2X,I5,5(7X,I5))
      WRITE(*,*)' '
      WRITE(6,192)
192 FORMAT(1X,'PERMUT.:(4 3 2 5 1)',1X,'(4 3 5 1 2)',1X,'(4 5 2 1
3)',
      *1X,'(5 3 2 1 4)',1X,'(3 2 1 4 5)',1X,'(3 2 1 5 4)')
      WRITE(6,194) (NTYP5(I),I=37,42)
194 FORMAT(1X,'OCUR.: ',2X,I5,5(7X,I5))
      WRITE(*,*)' '
      WRITE(6,196)

```

```

196 FORMAT(1X,'PERMUT.:(3 2 5 4 1)',1X,'(3 5 1 4 2)',1X,'(5 2 1 4
3)'),
    *1X,'(3 2 4 1 5)',1X,'(3 2 4 5 1)',1X,'(3 2 5 1 4)')
    WRITE(6,198) (NTYP5(I),I=43,48)
198 FORMAT(1X,'OCUR.:',2X,I5,5(7X,I5))
    WRITE(*,*)' '
    WRITE(6,200)
200 FORMAT(1X,'PERMUT.:(3 5 4 1 2)',1X,'(5 2 4 1 3)',1X,'(3 4 1 2
5)'),
    *1X,'(3 4 1 5 2)',1X,'(3 4 5 2 1)',1X,'(3 5 1 2 4)')
    WRITE(6,202) (NTYP5(I),I=49,54)
202 FORMAT(1X,'OCUR.:',2X,I5,5(7X,I5))
    WRITE(*,*)' '
    WRITE(6,204)
204 FORMAT(1X,'PERMUT.:(5 4 1 2 3)',1X,'(4 2 1 5 3)',1X,'(4 2 1 5
3)'),
    *1X,'(4 2 5 3 1)',1X,'(4 5 1 3 2)',1X,'(5 2 1 3 4)')
    WRITE(6,206) (NTYP5(I),I=55,60)
206 FORMAT(1X,'OCUR.:',2X,I5,5(7X,I5))
    WRITE(*,*)' '
    WRITE(6,210)
210 FORMAT(1X,'PERMUT.:(2 1 3 4 5)',1X,'(2 1 3 5 4)',1X,'(2 1 5 4
3)'),
    *1X,'(2 5 3 4 1)',1X,'(5 1 3 4 2)',1X,'(2 1 4 3 5)')
    WRITE(6,212) (NTYP5(I),I=61,66)
212 FORMAT(1X,'OCUR.:',2X,I5,5(7X,I5))
    WRITE(*,*)' '
    WRITE(6,214)
214 FORMAT(1X,'PERMUT.:(2 1 4 5 3)',1X,'(2 1 5 3 4)',1X,'(2 1 5 4
3)'),
    *1X,'(5 1 4 3 2)',1X,'(2 4 3 1 5)',1X,'(2 4 3 5 1)')
    WRITE(6,216) (NTYP5(I),I=67,72)
216 FORMAT(1X,'OCUR.:',2X,I5,5(7X,I5))
    WRITE(*,*)' '
    WRITE(6,218)
218 FORMAT(1X,'PERMUT.:(2 4 5 1 3)',1X,'(2 5 3 1 4)',1X,'(5 4 3 1
2)'),
    *1X,'(4 1 3 2 5)',1X,'(4 1 3 5 2)',1X,'(4 1 5 2 3)')
    WRITE(6,220) (NTYP5(I),I=73,78)
220 FORMAT(1X,'OCUR.:',2X,I5,5(7X,I5))
    WRITE(*,*)' '
    WRITE(6,222)
222 FORMAT(1X,'PERMUT.:(4 5 3 2 1)',1X,'(5 1 3 2 4)',1X,'(2 3 1 4
5)'),
    *1X,'(2 3 1 5 4)',1X,'(2 3 5 4 1)',1X,'(2 5 1 4 3)')
    WRITE(6,224) (NTYP5(I),I=79,84)
224 FORMAT(1X,'OCUR.:',2X,I5,5(7X,I5))
    WRITE(*,*)' '
    WRITE(6,226)
226 FORMAT(1X,'PERMUT.:(5 3 1 4 2)',1X,'(2 3 4 1 5)',1X,'(2 3 4 5
1)'),
    *1X,'(2 3 5 1 4)',1X,'(2 5 4 1 3)',1X,'(5 3 4 1 2)')
    WRITE(6,228) (NTYP5(I),I=85,90)
228 FORMAT(1X,'OCUR.:',2X,I5,5(7X,I5))
    WRITE(*,*)' '
    WRITE(6,230)
230 FORMAT(1X,'PERMUT.:(2 4 1 3 5)',1X,'(2 4 5 3 1)',1X,'(2 4 5 3
1)'),
    *1X,'(2 5 1 3 4)',1X,'(5 4 1 3 2)',1X,'(4 3 1 2 5)')
    WRITE(6,232) (NTYP5(I),I=91,96)
232 FORMAT(1X,'OCUR.:',2X,I5,5(7X,I5))

```

```

        WRITE(*,*)' '
        WRITE(6,234)
234  FORMAT(1X,'PERMUT.:(4 3 1 5 2)',1X,'(4 3 1 2 3)',1X,'(4 5 1 2
3)',
        *1X,'(5 3 1 2 4)',1X,'(3 1 2 4 5)',1X,'(3 1 2 5 4)')
        WRITE(6,236) (NTYP5(I),I=97,102)
236  FORMAT(1X,'OCUR.:',2X,I5,5(7X,I5))
        WRITE(*,*)' '
        WRITE(6,238)
238  FORMAT(1X,'PERMUT.:(3 1 5 4 2)',1X,'(3 5 2 4 1)',1X,'(5 1 2 4
3)',
        *1X,'(3 1 4 2 5)',1X,'(3 1 4 5 2)',1X,'(3 1 5 2 4)')
        WRITE(6,240) (NTYP5(I),I=103,108)
240  FORMAT(1X,'OCUR.:',2X,I5,5(7X,I5))
        WRITE(*,*)' '
        WRITE(6,242)
242  FORMAT(1X,'PERMUT.:(3 5 4 2 1)',1X,'(5 1 4 2 3)',1X,'(3 4 2 1
5)',
        *1X,'(3 4 2 5 1)',1X,'(3 4 5 1 2)',1X,'(3 5 2 1 4)')
        WRITE(6,244) (NTYP5(I),I=109,114)
244  FORMAT(1X,'OCUR.:',2X,I5,5(7X,I5))
        WRITE(*,*)' '
        WRITE(6,246)
246  FORMAT(1X,'PERMUT.:(5 4 2 1 3)',1X,'(4 1 2 3 5)',1X,'(4 1 2 5
3)',
        *1X,'(4 1 5 3 2)',1X,'(4 5 2 3 1)',1X,'(5 1 2 3 4)')
        WRITE(6,248) (NTYP5(I),I=115,120)
248  FORMAT(1X,'OCUR.:',2X,I5,5(7X,I5))
        WRITE(*,*)' '
        WRITE(6,250) XX(4,ICNT,3)
250  FORMAT(' ',4X,'CHI-CUADRADO CON 119 GRADOS DE LIBERTAD ES:
',F9.3)
        WRITE(6,252)
252  FORMAT(' ',4X,'PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-
CU
        *ADRADO')
        WRITE(6,254)
254  FORMAT(' ',4X,'CON 119 GRADOS DE LIBERTAD SEA <= X:')
        WRITE(*,*)' '
        WRITE(6,256)
256  FORMAT(' ',1X,'P|',2X,'.0001',2X,'.001',3X,'.01',3X,'.05',3X,
        *'.30',3X,'.50',3X,'.70',3X,'.95',3X,'.99',3X,'.999',3X,'.9999')
        WRITE(6,258)
258  FORMAT(' ',1X,'-----
--
        *-----')
        WRITE(6,260)
260  FORMAT(' ',1X,'X|',2X,'70.0',2X,'77.0',3X,'86.1',2X,'94.8',1X,
        * '110.0',1X,'118.3',1X,'126.6',1X,'145.5',1X,'157.8',2X,'172.4',
        *3X,'185.1')
999  RETURN
      END

C
C      SUBROUTINE TST08
C*****
C TST08 APLICA EL TEST DEL POKER
C
C PROPOSITO:
C DADA UNA LISTA DE N NUMEROS REALES EN PUNTO FLOTANTE ENTRE 0 Y 1 QUE
SE

```



C SUPONENE ALEATORIOS, SE LOS CONVIERTE A NUMEROS ENTEROS  
1,2,3,4,5,6,7,  
C 8,9,10 DE LA SIGUIENTE FORMA:  
C REEMPLAZAR X POR 1 SI  $0.0 \leq X < 0.1$   
C REEMPLAZAR X POR 2 SI  $0.1 \leq X < 0.2$   
C REEMPLAZAR X POR 3 SI  $0.2 \leq X < 0.3$   
C REEMPLAZAR X POR 4 SI  $0.3 \leq X < 0.4$   
C REEMPLAZAR X POR 5 SI  $0.4 \leq X < 0.5$   
C REEMPLAZAR X POR 6 SI  $0.5 \leq X < 0.6$   
C REEMPLAZAR X POR 7 SI  $0.6 \leq X < 0.7$   
C REEMPLAZAR X POR 8 SI  $0.7 \leq X < 0.8$   
C REEMPLAZAR X POR 9 SI  $0.8 \leq X < 0.9$   
C REEMPLAZAR X POR 10 SI  $0.9 \leq X \leq 1.0$ .  
C LUEGO LOS NUMEROS ENTEROS SERAN ALEATORIOS ENTRE 1 Y 10. EL TEST DEL  
POKER  
C SEGUN KENDALL Y BABINGTON SMITH PROPONE :TOMAR SUCEIVOS CONJUNTOS  
DE 5  
C ENTEROS, Y PARA CADA UNO, DETERMINAR SI CONTIENEN EL MISMO NUMERO  
("AAAAA")  
C DONDE A ES CUALQUER ENTERO ENTRE 1 Y 10), O BIEN UN ENTERO REPETIDO 4  
VECES  
C Y UNO DISTINTO ("AAAAB"), 3 IGUALES Y DOS DE OTRO ("AAABB"), TRES  
C IGUALES Y DOS DISTINTOS ("AABBC") , 2 DE CADA UNO Y EL TERCERO  
DIFERENTE  
C ("AABBC"), DE LA FORMA ("AABCD") O ("ABCDE"). UNA DE ESTAS  
POSIBILIDADES  
C OCURRIRA Y CUANDO HAY ALEATORIEDAD, LA PROPORCION DE OCURENCIA DE  
CADA UNA  
C SERA IGUAL A LAS PROBABILIDADES:  
C P(AAAAA) = .0001 P(AAAAB) = .0045  
C P(AAABB) = .0090 P(AAABC) = .0720  
C P(AABBC) = .1080 P(AABCD) = .5040  
C P(ABCDE) = .3024.  
C LUEGO, SE REALIZA UN TEST CHI-CUADRADO PARA VER SI LOS NUMEROS  
OBSERVADOS  
C EN CADA CATEGORIA CONCUERDAN CON LAS PROBABILIDADES TEORICAS  
MENCIONADAS  
C MIENTRAS QUE EL TEST UTILIZA 7 "REPARTICIONES" DEL CONJUNTO DE 5  
VALORES  
C ENTEROS, PUEDEN TOMARSE OTRAS. EL PROGRAMA REALIZA TAMBIEN EL TEST  
DEL  
C POKER PARA LAS REPARTICIONES BASADAS EN LOS ENTEROS DIFERENTES QUE  
HAY EN EL  
C 5, PARA LAS CUALES, LAS PROBABILIDADES BAJO ALEATORIEDAD SON:  
C P(1 DIFERENTE) = .0001 P(2 DIFERENTE) = .0135  
C P(3 DIFERENTE) = .1800 P(4 DIFERENTE) = .5040  
C P(5 DIFERENTE) = .3024.  
C NO ES NECESARIO RESTRINGIRSE A MANOS DE 5 CARTAS. TAMBIEN SE REALIZA  
EL  
C POKER TEST PARA MANOS DE 4 CARTAS CON 5 REPARTICIONES, CON  
PROBABILIDADES:  
C P(AAAA) = .001 P(AAAB) = .036  
C P(AABB) = .027 P(AABC) = .432  
C P(ABCD) = .504  
C  
C USO:  
C LOS PARAMETROS DE LA RUTINA TST08 SON:  
C VARIABLE LONG. NUM./CAR.  
C TNAME 8 CAR  
C IPARM(1) 11 NUM  
C IPARM(2) 11 NUM

```

C
C DESCRIPCION DE PARAMETROS
C   IPARM(1)...INDICE PARA COMENZAR EN LA LISTA DE NUMEROS ALEATORIOS
C   IPARM(2)...INDICE PARA FINALIZAR EN LA LISTA DE NUMEROS
ALEATORIOS
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS:
C   PNTCHI
C
C REFERENCIAS:
C   1. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C     GENERATION AND TESTING," AMERICAN SCIENCES PRESS,COLUMBUS,OHIO,
1981
C   2. KENDALL,M.G. AND BABINGTON SMITH,B.:"RANDOMNESS AND RANDOM
SAMPLING
C     NUMBERS"JOURNAL OF THE ROYAL STATISTICAL SOCIETY,VOL.101(1938)
C*****
C
C   SUBROUTINE TST08
C
C     DIMENSION PROB4(5),PROB7(7),PROB5(5)
C     DIMENSION EXP4(5),EXP7(7),EXP5(5)
C     DIMENSION NTYPE4(5),NTYPE7(7),NTYPE5(5)
C     DIMENSION HAND(10)
C     DIMENSION XXDF3(11),XXDF5(11)
C
C     COMMON /MESS/MESSG(9),VARIAB(9)
C     COMMON /FOURTH/Y(10000)
C     COMMON /SIXTH/XX(7,1000,3),YY(10000,2,3)
C     COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C     COMMON /T01PRM/NCHI,NPRNT
C     COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C     COMMON /TSTPRM/IPARM(5)
C     COMMON /CODE/GCODE
C     COMMON /T01VAL/INF01(10,6),NDEGF(3)
C
C     INTEGER HAND
C
C     character*8 name,messg,variab,t01nm,tstnm,ckname,accept,
*reject,stmt
c     REAL*8 MESSG,VARIAB,CKNAME,NAME
C
C     DATA PROB4/0.0010,0.0360,0.0270,0.4320,0.5040/
C     DATA PROB7/0.0001,0.0045,0.0090,0.0720,0.1080,0.5040,0.3024/
C     DATA PROB5/0.0001,0.0135,0.1800,0.5040,0.3024/
C     DATA CKNAME/'TST01'/
C     DATA
XXDF3/.005,.024,.114,.352,1.42,2.37,3.66,7.81,11.3,16.3,21.1/
C     DATA
XXDF5/.082,.210,.554,1.15,3.00,4.35,6.06,11.1,15.1,20.5,25.7/
C
C
OPEN(UNIT=28,FILE='ARCHT08',ACCESS='SEQUENTIAL',FORM='FORMATTED')
  READ(28,2349)IPARM(1)
  READ(28,2349)IPARM(2)
2349  FORMAT(I11)
  CLOSE(28)
  ISTART=IPARM(1)
  IEND=IPARM(2)
C
  IF(ISTART.LE.0) ISTART=1

```

```

        IF(IEND.LE.NBATCH) GOTO 33
        WRITE(6,100) IEND,NBATCH
100  FORMAT(1X,'VALOR ',I6,' DEL PARAMETRO 2 EN TST08 EXCEDE LA
CANTIDA
        *D DE NUMEROS ALEATORIOS DISPONIBLES',/1X,'PARA EL TEST. EL VALOR
H
        *A SIDO CAMBIADO A: ',I6,'. LA EJECUCION CONTINUA.')
        IEND=NBATCH
        INF01(5,2)=NBATCH
C
33  N=IEND-ISTART+1
C
        VERIFICACION DE ERRORES
C
        IF(N.GT.5) GOTO 1
        WRITE(6,101) N
101  FORMAT(1X,'PARAMETRO 1,',I11,' , EN TST08 ES DEMASIADO CHICO
PARA
        *RESULTADOS SIGNIFICATIVOS.',1X,'TST08 ES IGNORADO.')
C
        SI LOS VALORES FUERON GENERADOS POR TST01,
C        CAMBIAR EL SWITCH QUE LLAMA A CS SOBRE CS PARA TST08
C
7  IF(NAME.EQ.CKNAME) INF01(5,6)=0
        GOTO 999
C
        INICIALIZACIONES PARA CADA NUEVA APLICACION DEL TEST DEL
POKER
C
1  NGPS4=N/4
        NGPS5=N/5
C
        DO 2 J=1,5
            NTYPE4(J)=0
2  NTYPE5(J)=0
C
        DO 3 J=1,7
            NTYPE7(J)=0
C
        INICIO PROCESO DE MANOS DE 4 CARTAS
C
        NCARDS=4
C
        DO 11 I=1,NGPS4
C
        INICIALIZACIONES PARA CADA NUEVA MANO
C
        DO 4 J=1,10
4  HAND(J)=0
C
        PREPARACION DE NUEVA MANO PARA ARREGLO DE NUMEROS ALETORIOS
C
        DO 5 J=1,NCARDS
            K=(I-1)*NCARDS+J+ISTART-1
            INTGR=Y(K)*10+1
            IF(INTGR.GT.10) INTGR=10
5  HAND(INTGR)=HAND(INTGR)+1
C
        IDENTIFICACION DE TIPO DE REPARTICION
C

```

```

MXSAME=0
NDIFF=0
DO 6 J=1,10
  IF(HAND(J).EQ.0) GOTO 6
  IF(HAND(J).GT.MXSAME) MXSAME=HAND(J)
  NDIFF=NDIFF+1
6 CONTINUE
C
C      PROCESAMIENTO DE UNA PARTIDA
C
C      IDENTIFICACION TIPO DE REPARTICION SOBRE LA BASE DE NUMEROS
C      DE CARTAS DISTINTAS , NDIFF, Y DEL NUMERO DE REPETICIONES
C      DE CADA UNA DE LAS DISTINTAS CARTAS
C
C      HAND      NCARDS      MXSAME      NTYPE
C      AAAA      1          *          1
C      AAAB      2          3          2
C      AABB      2          2          3
C      AABC      3          *          4
C      ABCD      4          *          5
C
  IF(NDIFF.EQ.1) NTYPE4(1)=NTYPE4(1)+1
  IF((NDIFF.EQ.2).AND.(MXSAME.EQ.3)) NTYPE4(2)=NTYPE4(2)+1
  IF((NDIFF.EQ.2).AND.(MXSAME.EQ.2)) NTYPE4(3)=NTYPE4(3)+1
  IF(NDIFF.EQ.3) NTYPE4(4)=NTYPE4(4)+1
  IF(NDIFF.EQ.4) NTYPE4(5)=NTYPE4(5)+1
C
11 CONTINUE
C
  NTYP4S=NTYPE4(1)
  NTYPE4(2)=NTYPE4(2)+NTYPE4(1)
C
C      FIN PROCESO DEL BUCLE PARA GRUPOS DE 4 CARTAS
C      INICIO PROCESO DE MANOS DE 5 CARTAS
C
  NCARDS=5
C
  DO 19 I=1,NGPS5
C
C      INICIALIZACIONES PARA CADA NUEVA MANO
C
  DO 12 J=1,10
12 HAND(J)=0
C
C      PREPARACION DE UN ARREGLO DE NUMEROS PARA NUEVA MANO
C
  DO 13 J=1,NCARDS
  K=(I-1)*NCARDS+J+ISTART-1
  INTGR=Y(K)*10+1
  IF(INTGR.GT.10) INTGR=10
13 HAND(INTGR)=HAND(INTGR)+1
C
C      IDENTIFICACION TIPO DE REPARTICION
C
  MXSAME=0
  NDIFF=0
  DO 14 J=1,10
  IF(HAND(J).EQ.0) GOTO 14
  IF(HAND(J).GT.MXSAME) MXSAME=HAND(J)
  NDIFF=NDIFF+1
14 CONTINUE
C

```

```

C          IDENTIFICACION TIPO DE REPARTICION SOBRE LA BASE DE NUMEROS
C          DE CARTAS DISTINTAS , NDIFF, Y DEL NUMERO DE REPETICIONES
C          DE CADA UNA DE LAS DISTINTAS CARTAS
C          NSAME5(J), J=1,2,...,NCARDS.
C
C          HAND      NDIFF      MAXSAM      NTYPE
C          AAAAA      1          *          1
C          AAAAB      2          4          2
C          AAABB      2          3          3
C          AAABC      3          3          4
C          AABBC      3          2          5
C          AABCD      4          *          6
C          ABCDE      5          *          7
C
C          IF(NDIFF.EQ.1) NTYPE7(1)=NTYPE7(1)+1
C          IF((NDIFF.EQ.2).AND.(MXSAME.EQ.4)) NTYPE7(2)=NTYPE7(2)+1
C          IF((NDIFF.EQ.2).AND.(MXSAME.EQ.3)) NTYPE7(3)=NTYPE7(3)+1
C          IF((NDIFF.EQ.3).AND.(MXSAME.EQ.3)) NTYPE7(4)=NTYPE7(4)+1
C          IF((NDIFF.EQ.3).AND.(MXSAME.EQ.2)) NTYPE7(5)=NTYPE7(5)+1
C          IF(NDIFF.EQ.4) NTYPE7(6)=NTYPE7(6)+1
C          IF(NDIFF.EQ.5) NTYPE7(7)=NTYPE7(7)+1
C
C          NTYPE5(NDIFF)=NTYPE5(NDIFF)+1
C
C          19 CONTINUE
C
C          NTYP5S=NTYPE5(1)
C          NTYP7S=NTYPE7(1)
C          NTYPE5(2)=NTYPE5(2)+NTYPE5(1)
C          NTYPE7(2)=NTYPE7(2)+NTYPE7(1)
C
C          FIN PROCESO BUCLE PARA GRUPOS DE 5 CARTAS
C          CALCULO DE CHI-CUADRADO PARA ESTA MUESTRA
C
C          XX(5,ICNT,1)=0.0
C          XX(5,ICNT,2)=0.0
C          XX(5,ICNT,3)=0.0
C
C          DO 20 I=1,5
C          EXP4(I)=NGPS4*PROB4(I)
C          20 EXP5(I)=NGPS5*PROB5(I)
C
C          EXP4(2)=EXP4(2)+EXP4(1)
C          EXP5(2)=EXP5(2)+EXP5(1)
C
C          DO 21 I=1,7
C          21 EXP7(I)=NGPS5*PROB7(I)
C
C          EXP7(2)=EXP7(2)+EXP7(1)
C
C          DO 22 I=2,5
C          XX(5,ICNT,1)=XX(5,ICNT,1)+((NTYPE4(I)-EXP4(I))**2)/EXP4(I)
C          22 XX(5,ICNT,3)=XX(5,ICNT,3)+((NTYPE5(I)-EXP5(I))**2)/EXP5(I)
C
C          DO 23 I=2,7
C          23 XX(5,ICNT,2)=XX(5,ICNT,2)+((NTYPE7(I)-EXP7(I))**2)/EXP7(I)
C
C          IMPRESION DE RESULTADOS.LA IMPRESION SE OMITE CUANDO
C          LOS VALORES FUERON GENERADOS POR TST01 Y EL NUMERO
C          DE MUESTRAS CONSIDERADAS EXCEDE EL NUMERO DE TABLAS
C          REQUERIDAS POR EL USUARIO.

```

```

C
      IF((CKNAME.EQ.NAME).AND.(ICNT.GT.NPRNT)) GOTO 999
C
      E N C A B E Z A M I E N T O S
C
      WRITE(*,*)' '
      WRITE(*,*)' '
      WRITE(*,*)' '

WRITE(*,*)'*****'
      *****'

WRITE(*,*)'*****'
      *****'
      WRITE(*,*)' '
      WRITE(*,1234)
1234  FORMAT(1X,'TST08: TEST DEL POKER')
      WRITE(*,*)' '
      WRITE(6,201)
201  FORMAT(' ',1X,'EL TEST DEL POKER HA SIDO APLICADO (RUTINA TST08)
Y
      * HA PRODUCIDO')
      WRITE(6,202)
202  FORMAT(' ',1X,'LOS SIGUIENTES RESULTADOS:')
      WRITE(*,*)' '
      WRITE(6,240) ICNT
240  FORMAT(' ',5X,'NUMERO DE MUESTRA: ',I4)
      WRITE(*,2345)
2345  FORMAT(' ',5X,'-----')
      WRITE(*,*)' '
      WRITE(6,204) NGPS4,NGPS5
204  FORMAT(' ',1X,'EL TEST DEL POKER FUE APLICADO A',I5,' GRUPOS DE
4
      *NUMEROS ALEATORIOS Y',I5)
      WRITE(6,200)
200  FORMAT(' ',1X,'GRUPOS DE 5 NUMEROS ALEATORIOS.')
      WRITE(6,205)
205  FORMAT(' ',1X,'PARA UN GRUPO DE NUMEROS ALEATORIOS, EL TEST
PRUEBA
      * QUE TIPO DE REPARTICION')
      WRITE(*,*)' OCURRE.'
      WRITE(*,*)' '

C
      RESULTADOS PARA MANOS DE 4
C
      WRITE(6,207)
207  FORMAT(' ',1X,'MANOS DE 4 CON 4 REPARTICIONES')
      WRITE(*,*)' '
      WRITE(6,208)
208  FORMAT(' ',4X,'TIPO',16X,'ESPERADAS',8X,'OBSERVADAS')
      WRITE(6,210) EXP4(1),NTYPE4(1)
210  FORMAT(' ',4X,'AAAA',15X,F9.4,12X,I5)
      WRITE(6,211) EXP4(2),NTYPE4(2)
211  FORMAT(' ',1X,'AAAA OR AAAB',7X,F12.4,12X,I5)
      WRITE(6,212) EXP4(3),NTYPE4(3)
212  FORMAT(' ',4X,'AABB',15X,F9.4,12X,I5)
      WRITE(6,213) EXP4(4),NTYPE4(4)
213  FORMAT(' ',4X,'AABC',15X,F9.4,12X,I5)
      WRITE(6,214) EXP4(5),NTYPE4(5)
214  FORMAT(' ',4X,'ABCD',15X,F9.4,12X,I5)
      WRITE(*,*)' '

```

```

        WRITE(6,25)XX(5,ICNT,1)
25  FORMAT(' ',4X,'CHI-CUADRADO PARA 3 GRADOS DE LIBERTAD ES:
',F8.2)
        WRITE(6,206)
206  FORMAT(' ',4X,'PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-
CU
*ADRADO')
        WRITE(6,209)
209  FORMAT(' ',4X,'CON 3 GRADOS DE LIBERTAD SEA <= X:')
        WRITE(*,*)' '
        WRITE(6,24)
24  FORMAT('
',1X,'P|',1X,'.0001',1X,'.001',3X,'.01',3X,'.05',3X,'.30'
*,3X,'.50',3X,'.70',3X,'.95',3X,'.99',3X,'.999',3X,'.9999')
        WRITE(6,199)
199  FORMAT(' ', '-----
--
*-----')
        WRITE(6,198)(XXDF3(I),I=1,11)
198  FORMAT('
',1X,'X|',2X,2(F4.3,2X),2(F4.3,2X),4(F4.2,2X),F4.1,2X,F4.
*1,4X,F4.1)
        WRITE(*,*)' '
        WRITE(6,215)
215  FORMAT(' ',1X,'-----
--
*-----')
        WRITE(*,*)' '
C
C      RESULTADOS PARA MANOS DE 5 CON 6 REPARTICIONES
C
        WRITE(6,216)
216  FORMAT(' ',1X,'MANOS DE 5 CON 6 REPARTICIONES')
        WRITE(*,*)' '
        WRITE(6,217)
217  FORMAT(' ',4X,'TIPO',17X,'ESPERADAS',8X,'OBSERVADAS')
        WRITE(6,219) EXP7(1),NTYPE7(1)
219  FORMAT(' ',4X,'AAAAA',15X,F9.4,12X,I5)
        WRITE(6,220) EXP7(2),NTYPE7(2)
220  FORMAT(' ',1X,'AAAAA OR AAAAB',8X,F10.4,12X,I5)
        WRITE(6,221) EXP7(3),NTYPE7(3)
221  FORMAT(' ',4X,'AAABB',15X,F9.4,12X,I5)
        WRITE(6,222) EXP7(4),NTYPE7(4)
222  FORMAT(' ',4X,'AAABC',15X,F9.4,12X,I5)
        WRITE(6,223) EXP7(5),NTYPE7(5)
223  FORMAT(' ',4X,'AABBC',15X,F9.4,12X,I5)
        WRITE(6,1994)EXP7(6),NTYPE7(6)
1994  FORMAT(' ',4X,'AABCD',15X,F9.4,12X,I5)
        WRITE(6,1995)EXP7(7),NTYPE7(7)
1995  FORMAT(' ',4X,'ABCDE',15X,F9.4,12X,I5)
        WRITE(*,*)' '
        WRITE(6,26)XX(5,ICNT,2)
26  FORMAT(' ',4X,'CHI-CUADRADO PARA 5 GRADOS DE LIBERTAD ES:
',F8.2)
        WRITE(6,197)
197  FORMAT(' ',4X,'PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-
CU
*ADRADO')
        WRITE(6,218)
218  FORMAT(' ',4X,'CON 5 GRADOS DE LIBERTAD SEA <= X:')
        WRITE(*,*)' '

```

```

WRITE(6,196)
196 FORMAT('
',1X,'P|',2X,'.0001',2X,'.001',3X,'.01',3X,'.05',3X,'.30'
*,3X,'.50',3X,'.70',3X,'.95',3X,'.99',3X,'.999',3X,'.9999')
WRITE(6,195)
195 FORMAT(' ','-----'
---
*-----')
WRITE(6,194)(XXDF5(I),I=1,11)
194 FORMAT('
',1X,'X|',2X,F4.3,3X,F4.3,2X,F4.3,2X,4(F4.2,2X),2(F4.1,2X
*),1X,F4.1,4X,F4.1)

WRITE(*,*)' '
WRITE(6,224)
224 FORMAT(' ','-----'
---
*-----')
C
C RESULTADOS PARA MANOS DE 5 CON 4 REPARTICIONES
C
WRITE(*,*)' '
WRITE(6,225)
225 FORMAT(' ',1X,'MANOS DE 5 CON 4 REPARTICIONES')
WRITE(*,*)' '
WRITE(6,226)
226 FORMAT(' ',4X,'TIPO',16X,'ESPERADAS',8X,'OBSERVADAS')
WRITE(6,228) EXP5(1),NTYPE5(1)
228 FORMAT(' ',1X,'1 DIFERENTE',10X,F9.4,13X,I5)
WRITE(6,229) EXP5(2),NTYPE5(2)
229 FORMAT(' ',1X,'1 O 2 DIFERENTES',7X,F8.4,13X,I5)
WRITE(6,230) EXP5(3),NTYPE5(3)
230 FORMAT(' ',1X,'3 DIFERENTES',10X,F9.4,13X,I5)
WRITE(6,231) EXP5(4),NTYPE5(4)
231 FORMAT(' ',1X,'4 DIFERENTES',10X,F9.4,13X,I5)
WRITE(6,232) EXP5(5),NTYPE5(5)
232 FORMAT(' ',1X,'5 DIFERENTES',10X,F9.4,13X,I5)
WRITE(*,*)' '
WRITE(6,27) XX(5,ICNT,3)
27 FORMAT(' ',1X,'CHI-CUADRADO PARA 3 GRADOS DE LIBERTAD ES:
',F8.2)
WRITE(6,233)
233 FORMAT(' ',1X,'PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-
CU
*ADRADO')
WRITE(6,227)
227 FORMAT(' ',1X,'CON 3 GRADOS DE LIBERTAD SEA <= X:')
WRITE(*,*)' '
WRITE(6,234)
234
FORMAT(1X,'P|',2X,'.0001',2X,'.001',3X,'.01',2X,'.05',2X,'.30',3X,
*.50',2X,'.70',4X,'.95',3X,'.99',3X,'.999',3X,'.9999')
WRITE(6,235)
235 FORMAT(' ','-----'
---
*-----')
WRITE(6,236)(XXDF3(I),I=1,11)
236
FORMAT(1X,'X|',2X,2(F4.3,2X),2(F4.3,2X),4(F4.2,2X),F4.1,3X,F4.1,4X
*,F4.1)

```



999 RETURN

END

```
C
C      SUBROUTINE TST09
C*****
C TST09 APLICA EL TEST DE LAS CORRIDAS (RUNS UP) PARA PROBAR
C ALEATORIEDAD
C
C PROPOSITO
C DADA UNA LISTA DE N NUMEROS REALES EN PUNTO FLOTANTE ENTRE 0 Y 1 QUE
C SE SUPONEN ALEATORIOS, SE ESPERA QUE EL TAMA.%.O DE "RUNS UP"
C OBSERVADOS SIGA
C UNA CIERTA DISTRIBUCION DE PROBABILIDAD. SI DETERMINAMOS EL NUMERO
C COUNT(R) DE RUNS DE TAMA.%.O R (1<=R<=5) Y EL NUMERO COUNT(6) DE
C RUNS DE
C TAMANIO >=6, PUEDE USARSE UN TEST CHI-CUADRADO PARA VER SI LOS
C NUMEROS SON
C ALEATORIOS. (NOTAR QUE SE TIENE UN TEST CHI-CUADRADO COMPLEJO YA QUE
C LOS
C TAMANIOS DE LOS RUNS ESTAN CORRELACIONADOS. ES DECIR, SE ESPERA QUE UN
C RUN
C LARGO SEA SEGUIDO POR UNO CORTO). EL TEST SE APLICA A 3 CONJUNTOS DE
C NUMEROS ALEATORIOS:
C LISTA BASICA DE LOS X - LISTA DE LOS FR(10X) - LISTA DE FR(100X)
C DONDE FR (Z) DENOTA LA PARTE FRACCIONARIA DE Z.
C
C USO:
C TST09 TIENE LOS SIGUIENTES PARAMETROS:
C   VARIABLE           LONG.           NUM. /CAR.
C   TNAME              8               CAR
C   IPARM(1)           11             NUM
C   IPARM(2)           11             NUM
C
C DESCRIPCION DE PARAMETROS
C   IPARM(1)...INDICE DEL ARREGLO DONDE SE LOCALIZA EL PRIMERO A USAR
C   IPARM(2)...INDICE DEL ARREGLO DONDE SE LOCALIZA EL ULTIMO A USAR
C
C OBSERVACIONES:
C ESTA RUTINA ASUME QUE LA PROBABILIDAD DE TENER X IGUALES EN LA LISTA
C DE N NUMEROS ES CERO. CUANDO ESTO NO ES VERDADERO, LOS RESULTADOS SON
C SOLO APROXIMADOS.
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS:
C   PNTCHI
C METODO:
C VER REFERENCIAS
C 1. DUDEWICZ, E.J. AND RALLEY, T.: "THE HANDBOOK OF RANDOM NUMBER
C GENERATION AND TESTING," AMERICAN SCIENCES PRESS, COLUMBUS, OHIO, 1981
C
C*****
C
C      SUBROUTINE TST09
C
C      DIMENSION C(6,12), B(6), LEN(6), LENFR(6), LENFR2(6)
C
C      COMMON /MESS/MESSG(9), VARIAB(9)
C      COMMON /FOURTH/Y(10000)
C           COMMON /SIXTH/XX(7,1000,3), YY(10000,2,3)
C      COMMON /URNPRM/NAME, NBATCH, TIMES, IPRINT, IOPT
```

```

COMMON /T01PRM/NCHI,NPRNT
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON /TSTPRM/IPARM(5)
COMMON /T01VAL/INF01(10,6),NDEGF(3)
COMMON /CODE/GCODE

C
character*8 name,messg,variab,t01nm,tstnm,cknm(10),ckname
c   REAL*8 MESSG,VARIAB,CKNAME,NAME,C,XN,TEMP,PIVOT,BIG,COEFF
c   REAL*8 C,XN,TEMP,PIVOT,BIG,COEFF

C
DATA CKNAME/'TST01'/

C
OPEN(UNIT=29,FILE='ARCHT09',ACCESS='SEQUENTIAL',FORM='FORMATTED')
  READ(29,2349)IPARM(1)
  READ(29,2349)IPARM(2)
2349 FORMAT(I11)
  CLOSE(29)
  ISTART=IPARM(1)
  IEND=IPARM(2)

C
  IF(ISTART.LE.0) ISTART=1
  IF(IEND.LE.NBATCH) GOTO 2
  WRITE(6,99) IEND,NBATCH
99 FORMAT(1X,'VALOR ',I6,' DEL PARAMETRO 2 EN TST09 EXCEDE LA
  *CANTIDAD DE NUMEROS ALEATORIOS DISPONIBLES ',/1X,'PARA EL TEST.
  * EL VALOR
  *HA SIDO CAMBIADO A: ',I6,'. LA EJECUCION CONTINUA.')
  IEND=NBATCH
  INF01(6,2)=NBATCH

C
  2 N=IEND-ISTART+1

C
      INICIALIZACION PARA CADA NUEVA APLICACION DE TST09
C
DO 10 I=1,6
  LEN(I)=0
  LENFR(I)=0
10 LENFR2(I)=0

C
  MAX=0
  MAXFR=0
  MAXFR2=0

C
      PONER ULTIMO NUMERO IGUAL A -1 COMO PARTE DE UNA
C VERIFICACION
C      DEL FINAL DEL AREEGLO
C
  SAVE=Y(IEND)
  Y(IEND)=-1

C
      INICIALIZACIONES PARA TST09 APLICADAS A LA SUCESION
C      DE NUMEROS ALEATORIOS GENERADOS
C
  I=ISTART
  J=I+1
  L=1

C
12 IF(Y(I).GT.Y(J)) GOTO 14
  I=I+1
  J=I+1

```

```

L=L+1
GOTO 12
C
C      FIN DE CORRIDA . INFORMACION REGISTRADA
C
14 IF(J.LT.IEND) GOTO 17
   IF(Y(I).GT.SAVE) GOTO 15
C
C      FINAL CORRIDA PROCESADA Y VERIFICACION DE Y(N-1)<Y(N)
C      INCREMENTO EN 1 DEL TAMA.¥.O DEL ULTIMO RUN
C
L=L+1
IF(L.GT.MAX) MAX=L
IF(L.GT.6) L=6
LEN(L)=LEN(L)+1
GOTO 20
C
C      FINAL RUN PROCESADO Y VERIFICACION DE Y(N-1)>Y(N)
C      REGISTRAR Y CORRIDA ADICIONAL DE TAMA.¥.O 1
C
15 LEN(1)=LEN(1)+1
   IF(L.GT.MAX) MAX=L
   IF(L.GT.6) L=6
   LEN(L)=LEN(L)+1
   GOTO 20
C
C      REGISTRO DEL ANGO DE LA ULTIMA CORRIDA CALCULADA
C
17 IF(L.GT.MAX) MAX=L
   IF(L.GT.6) L=6
   LEN(L)=LEN(L)+1
C
C      INICIALIZACION PARA LA PROXIMA CORRIDA
C
L=1
I=I+1
J=J+1
GOTO 12
C
C      INICIALIZACIONES DE TST09 APLICADAS A LA SUCESION FR(10*X)
C
20 I=ISTART
   J=I+1
   L=1
C
22 I1=10*Y(I)
   J1=10*Y(J)
C
FRI=10*Y(I)-I1
FRJ=10*Y(J)-J1
IF(FRI.GT.FRJ) GOTO 24
I=I+1
J=J+1
L=L+1
GOTO 22
C
C      FIN DE UNA CORRIDA. REGISTRO DE INFORMACION
C
24 IF(J.LT.IEND) GOTO 27
   J1=10*SAVE
   SAVEFR=10*SAVE-J1

```

```

IF(FRI.GT.SAVEFR) GOTO 25
C
C      FINAL RUN PROCESSED AND CHECK SHOWS THAT
C      FR(10*Y(N-1))<=FR(10*Y(N)) SO INCREASE THE LENGTH OF THE
C      RUN BY 1
C
L=L+1
IF(L.GT.MAXFR) MAXFR=L
IF(L.GT.6) L=6
LENFR(L)=LENFR(L)+1
GOTO 30
C
C      FINAL RUN PROCESADO Y VERIFICACION DE QUE
C      FR(10*Y(N-1))>FR(10*Y(N)) . SI ES ASI, HAY UNA CORRIDA
C      ADICIONAL DE ANCHO 1
C
25 LENFR(1)=LENFR(1)+1
IF(L.GT.MAXFR) MAXFR=L
IF(L.GT.6) L=6
LENFR(L)=LENFR(L)+1
GOTO 30
C
C      GRABAR TAMA.%.O DEL ULTIMO RUN
C
27 IF(L.GT.MAXFR) MAXFR=L
IF(L.GT.6) L=6
LENFR(L)=LENFR(L)+1
C
C      INICIALIZA EL PROXIMO RUN
C
L=1
I=I+1
J=I+1
GOTO 22
C
C      INICIALIZACIONES DE TST09 APLICADAS A LA SUCESION FR(100*X)
C
30 I=ISTART
J=I+1
L=1
C
32 I1=100*Y(I)
J1=100*Y(J)
C
FRI=100*Y(I)-I1
FRJ=100*Y(J)-J1
IF(FRI.GT.FRJ) GOTO 34
I=I+1
J=I+1
L=L+1
GOTO 32
C
C      FIN DE UNA CORRIDA. GRABAR INFORMACION
C
34 IF(J.LT.IEND) GOTO 37
J1=100*SAVE
SAVFR2=100*SAVE-J1
IF(FRI.GT.SAVFR2) GOTO 35
C
C      FINAL RUN PROCESADO Y VERIFICACION QUE
C      FR(100*Y(N-1))<=FR(100*Y(N)) .SI ES ASI INCREMENTAR EN 1

```

```

C           EL TAMA.¥.O DEL RUN
C
L=L+1
IF(L.GT.MAXFR2) MAXFR2=L
IF(L.GT.6) L=6
LENFR2(L)=LENFR2(L)+1
GOTO 40

C
C           FINAL RUN PROCESADO Y VERIFICACION QUE
C           FR(100*Y(N-1))>FR(100 *Y(N)) .SI ES ASI HAY UNA CORRIDA
C           ADICIONAL DE TAMA.¥.O 1
C
35 LENFR2(1)=LENFR2(1)+1
IF(L.GT.MAXFR2) MAXFR2=L
IF(L.GT.6) L=6
LENFR2(L)=LENFR2(L)+1
GOTO 40

C
C           RECORD THE LENGTH OF THE LAST COMPUTED RUN
C
37 IF(L.GT.MAXFR2) MAXFR2=L

IF(L.GT.6) L=6
LENFR2(L)=LENFR2(L)+1

C
C           INICIALIZACION PARA EL PROXIMO RUN
C
L=1
I=I+1
J=J+1
GOTO 32

C
C           FORMULA DE LA MATRIZ C
C           MATRIZ C SEGUN KNUTH, P.63, ECUACION 22
C
40 XN=N
C(1,1) = (23.D0*XN + 83.D0)/180.D0
C(2,1) = (-7.D0*XN - 58.D0)/360.D0
C(1,2) = C(2,1)
C(3,1) = -5.D0*XN/336.D0 - 11.D0/210.D0
C(1,3) = C(3,1)
C(4,1) = -433.D0*XN/60480.D0 - 41.D0/12096.D0
C(1,4) = C(4,1)
C(5,1) = -13.D0*XN/5670.D0 + 91.D0/25920.D0
C(1,5) = C(5,1)
C(6,1) = (-121.D0*XN + 410.D0)/181440.D0
C(1,6) = C(6,1)
C(2,2) = 2843.D0*XN/20160.D0 - 305.D0/4032.D0
C(3,2) = (-989.D0*XN + 319.D0)/20160.D0
C(2,3) = C(3,2)
C(4,2) = -7159.D0*XN/362880.D0 + 2557.D0/72576.D0
C(2,4) = C(4,2)
C(5,2) = -10019.D0*XN/1814400.D0 + 10177.D0/604800.D0
C(2,5) = C(5,2)
C(6,2) = -1303.D0*XN/907200.D0 + 413.D0/64800.D0
C(2,6) = C(6,2)
C(3,3) = (54563.D0*XN - 58747.D0)/907200.D0
C(4,3) = -21311.D0*XN/1814400.D0 + 19703.D0/604800.D0
C(3,4) = C(4,3)
C(5,3) = (-62369.D0*XN + 239471.D0)/19958400.D0
C(3,5) = C(5,3)

```

```

C(6,3) = (-7783.D0*XN + 39517.D0)/9979200.D0
C(3,6) = C(6,3)
C(4,4) = 886657.D0*XN /39916800.D0 - 220837.D0/4435200.D0
C(5,4) = (-257699.D0*XN + 1196401.D0)/239500800.D0
C(4,5) = C(5,4)
C(6,4) = (-62611.D0*XN + 360989.D0)/239500800.D0
C(4,6) = C(6,4)
C(5,5) = 29874811.D0*XN/5448643200.D0-139126639.D0/7264857600.D0
C(6,5) = -1407179.D0*XN/21794572800.D0+4577641.D0/10897286400.D0
C(5,6) = C(6,5)
C(6,6) = 2134697.D0*XN/1816214400.D0-122953057.D0/21794572800.D0
C
B(1)=XN/6.D0+2.D0/3.D0
B(2)=5.D0*XN/24.D0+1.D0/24.D0
B(3)=11.D0*XN/120.D0-7.D0/60.D0
B(4)=19.D0*XN/720.D0-47.D0/720.D0
B(5)=29.D0*XN/5040.D0-19.D0/840.D0
B(6)=XN/840.D0-29.D0/5040.D0
C
C      DETERMINAR INVERSA DE C
C      ESTABLECER QUE LA SUBMATRIZ FORMADA POR LAS 6 ULTIMAS
C      COLUMNAS DE C ES IGUAL A LA MATRIZ IDENTIDAD
C      LUEGO USANDO EL METODO DE ELIMINACION DE GAUSS-JORDAN
C      HALLAR LA INVERSA DE C POR REDUCCION DE FILAS EN LA
SUBMATRIZ
C      FORMADA POR LAS SEIS PRIMERAS COLUMNAS DE C
C
DO 110 I=1,6
DO 110 J=7,12
C(I,J)=0.0D0
IF (J.EQ.6+I)C(I,J)=1.0D0
110 CONTINUE
C
DO 121 I=1,6
BIG=0.0D0
INDX=0
C
DO 116 K=I,6
IF(DABS(C(K,I)).LE.BIG) GOTO 116
BIG=DABS(C(K,I))
INDX=K
116 CONTINUE
C
C      SI TODOS LOS ELEMENTOS DE LA COLUMNA I ABAJO DE LA FILA
C      I-1 SON 0, LA MATRIZ ES SINGULAR Y NO TIENE INVERSA
C
IF(BIG.LT.1.D-8) GOTO 117
C
C      EL ELEMENTO MAXIMO EN VAL.ABSOLUTO EN LA COLUMNA I ESTA
C      EN LA POSICION (K,I). INTERCAMBIO DE LAS FILAS I Y K
C      DE MANERA DE USAR ESE ELEMENTO COMO PIVOTE.
C
IF(INDX.EQ.I) GOTO 129
C
DO 118 J1=1,12
TEMP=C(INDX,J1)
C(INDX,J1)=C(I,J1)
118 C(I,J1)=TEMP
C
129 PIVOT=C(I,I)
C

```

```

DO 120 K=1,6
IF(K.EQ.I) GOTO 120
COEFF=C(K,I)/PIVOT
DO 119 L=I,12
119 C(K,L)=C(K,L)-COEFF*C(I,L)
120 CONTINUE
C
121 CONTINUE
C
DO 122 K=1,6
PIVOT=C(K,K)
DO 123 L=K,12
123 C(K,L)=C(K,L)/PIVOT
122 CONTINUE
C
C          TRANSFERIR LA INVERSA A LAS 6 PRIMERAS COLUMNAS DEL ARREGLO
C
DO 124 I=1,6
DO 124 J=1,6
IND=J+6

124 C(I,J) = C(I,IND)
GO TO 130
C
117 WRITE(6,125)
125 FORMAT(1X,'EL COEFICIENTE DE LA MATRIZ EN TST09 ES SINGULAR. LA
EJ          *ECUCION DE TST09 HA TERMINADO.',/1X, 'EL CONTROL RETORNA AL
PROGRA          *MA DE TESTS.')
GOTO 998
C
C          CALCULO DE ESTADISTICAS
C
130 XX(6,ICNT,1)=0.0
XX(6,ICNT,2)=0.0
XX(6,ICNT,3)=0.0
C
DO 131 I=1,6
DO 131 J=1,6
XX(6,ICNT,1)=XX(6,ICNT,1)+(LEN(I)-B(I))*(LEN(J)-B(J))*C(I,J)
XX(6,ICNT,2)=
*   XX(6,ICNT,2)+(LENFR(I)-B(I))*(LENFR(J)-B(J))*C(I,J)
131 XX(6,ICNT,3)=
*   XX(6,ICNT,3)+(LENFR2(I)-B(I))*(LENFR2(J)-B(J))*C(I,J)
C
C          IMPRESION DE RESULTADOS. LA IMPRESION ES OMITIDA CUANDO
C          LOS VALORES FUERON GENERADOS POR TST01 Y LA CANTIDAD DE
C          MUESTRAS CONSIDERADAS EXCEDE LA CANTIDAD DE TABLAS
C          REQUERIDAS POR EL USUARIO.
C
IF((CKNAME.EQ.NAME).AND.(ICNT.GT.NPRNT)) GOTO 998
WRITE(*,*)' '
WRITE(*,*)' '
WRITE(*,*)' '

WRITE(*,*)'*****'
*****

WRITE(*,*)'*****'
*****

```

```

WRITE(*,*)' '
WRITE(*,1234)
1234 FORMAT(1X,'TST09: TEST DE LAS CORRIDAS (RUNS UP)')
WRITE(*,*)' '
WRITE(6,200)
200 FORMAT(' ',1X,'EL TEST "RUNS UP" FUE APLICADO (RUTINA TST09) Y
HA
*PRODUCIDO')
WRITE(6,202)
202 FORMAT(' ',1X,'LOS SIGUIENTES RESULTADOS:')
WRITE(*,*)' '
WRITE(6,204) ICNT
204 FORMAT(' ',4X,'NUMERO DE MUESTRA: ',I4)
WRITE(*,2345)
2345 FORMAT(' ',4X,'-----')
WRITE(*,*)' '
WRITE(6,208) N
208 FORMAT(' ',1X,'CADA MUESTRA CONSISTE DE ',I5,' NUMEROS
ALEATORIOS'
*)
WRITE(*,*)' '
WRITE(6,212)
212 FORMAT(' ',1X,'UTILIZANDO EL NUMERO ALEATORIO X COMO GENERADO')
WRITE(*,*)' '
WRITE(6,214)
214 FORMAT(' ',7X,'TABLA DE "RUNS UP" ')
WRITE(*,*)' '
DO 140 I=1,5
WRITE(6,216) LEN(I),I
216 FORMAT(' ',3X,I5,' CORRIDAS DE LONGITUD ',I1)
140 CONTINUE
WRITE(6,218) LEN(6)
218 FORMAT(' ',I5,' CORRIDAS DE LONGITUD 6 O MAS')
WRITE(*,*)' '
WRITE(6,220) MAX
220 FORMAT(' ',1X,'LA CORRIDA MAXIMA ES DE LONGITUD: ',I2)
WRITE(6,222)XX(6,ICNT,1)
222 FORMAT(' ',1X,'VAR. ESTADISTICA, APROX.CHI-CUADRADO CON 6 GRADOS
D
*E LIBERTAD: ',F6.2)
WRITE(*,*)' '
WRITE(*,*)' '
WRITE(6,225)
225 FORMAT(' ',1X,'UTILIZANDO LA PARTE FRACCIONARIA DE 10*X:')
WRITE(*,*)' '
WRITE(6,226)
226 FORMAT(' ',7X,'TABLA DE "RUNS UP"')
WRITE(*,*)' '
DO 228 I=1,5
WRITE(6,227) LENFR(I),I
227 FORMAT(' ',3X,I5,' CORRIDAS DE LONGITUD ',I1)
228 CONTINUE
WRITE(6,229) LENFR(6)
229 FORMAT(' ',I5,' CORRIDAS DE LONGITUD 6 O MAS')
WRITE(*,*)' '
WRITE(6,230) MAXFR
230 FORMAT(' ',1X,'LA CORRIDA MAXIMA ES DE LONGITUD: ',I2)
WRITE(6,231)XX(6,ICNT,2)
231 FORMAT(' ',1X,'VAR. ESTADISTICA, APROX.CHI-CUADRADO CON 6 GRADOS
D
*E LIBERTAD: ',F6.2)

```



```

        WRITE(*,*)' '
        WRITE(6,233)
233  FORMAT(' ',1X,'UTILIZANDO LA PARTE FRACCIONARIA DE 10*X:')
        WRITE(*,*)' '
        WRITE(6,234)
234  FORMAT(' ',7X,'TABLA DE "RUNS UP"')
        WRITE(*,*)' '
        DO 235 I=1,5
        WRITE(6,236) LENFR2(I),I
236  FORMAT(' ',3X,I5,' CORRIDAS DE LONGITUD ',I1)
235  CONTINUE
        WRITE(6,237) LENFR2(6)
237  FORMAT(' ',I5,' CORRIDAS DE LONGITUD 6 O MAS')
        WRITE(*,*)' '
        WRITE(6,238) MAXFR2
238  FORMAT(' ',1X,'LA CORRIDA MAXIMA ES DE LONGITUD: ',I2)
        WRITE(6,239)XX(6,ICNT,3)
239  FORMAT(' ',1X,'VAR. ESTADISTICA, APROX.CHI-CUADRADO CON 6 GRADOS
D
        *E LIBERTAD: ',F6.2)
        WRITE(*,*)' '
        WRITE(6,300)
300  FORMAT(' ',1X,'PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-
CU
        *ADRADO')
        WRITE(6,302)
302  FORMAT(' ',1X,'CON 6 GRADOS DE LIBERTAD SEA <= X:')
        WRITE(*,*)' '
        WRITE(6,304)
304  FORMAT(' ',1X,'P
|.0001',2X,'.001',3X,'.01',3X,'.05',3X,'.30',3X,'
*.50',3X,'.70',3X,'.95',3X,'.99',3X,'.999',3X,'.9999')
        WRITE(6,306)
306  FORMAT(' ',1X,'-----
',
        *'-----')
        WRITE(6,308)
308  FORMAT(' ',1X,'X
|.172',3X,'.381',2X,'.872',2X,'1.63',2X,'3.83',2X
*,'.5.35',2X,'7.23',2X,'12.6',2X,'16.8',3X,'22.5',4X,'27.9')
C
C          RESET Y(IEND) TO ITS ORIGINAL VALUE
C
998  Y(IEND)=SAVE
999  RETURN
      END
C
C          SUBROUTINE TST10
C*****
C PROPOSITO:
C DADA UNA LISTA DE N NUMEROS REALES EN PUNTO FLOTANTE ENTRE 0 Y 1,
C QUE SE
C SUPONENEN ALEATORIOS, SEA UN ENTERO D>=2 , SE CONSIDERA EL
C REEMPLAZO DE
C DICHOS NUMEROS POR ENTEROS 1,2,...,D MEDIANTE 1+INT(D*X) DONDE
C INT(D*X)
C ES LA PARTE ENTERA DE D*X (PERO SI D*X=D,TOMAMOS INT(D*X)=D-1).
C LOS NUEVOS NUMEROS SERAN ALEATORIOS ENTRE 1 Y D.PARA ESTOS ENTEROS
C KENDALL Y BABINGTON SMITH PROPONEN EL TEST "SERIAL PAIRS":
C TOMAR SUCESIVOS PARES DE ENTEROS Y DETERMINAR EN CUAL DE LAS D**2
C CATEGORIAS POSIBLES

```

```

C (1,1), (1,2), (1,3),..., (1,D-1), (1,D),
C (2,1), (2,2), (2,3),..., (2,D-1), (2,D),
C .
C .
C .
C (D,1), (D,2), (D,3),..., (D,D-1), (D,D)
C CAE CADA UNO.CUANDO HAY ALEATORIEDAD,LAS CATEGORIAS SON
C EQUIPROBABLES CON
C PROB.1/(D**2).LUEGO DE OBSERVAR LAS OCURRENCIAS EN CADA CATEGORIA
C SE REALI-
C ZA UN TEST CHI-CUAD.PARA VER SI LOS NUMEROS OBSERVADOS EN CADA
C CATEGORIA
C COINCIDEN CON LAS PROBAB.TEORICAS.AQUI SE REALIZA EL TEST PARA D=3,
C D=10 Y
C D=20
C PARAMETROS:
C VARIABLE LONG. NUM./CAR.
C TNAME 8 CAR
C IPARM(1) 11 NUM
C IPARM(2) 11 NUM
C
C
C DESCRIPCION DE PARAMETROS
C IPARM(1)...INDICE INICIAL PARA EL ARREGLO DE NUMEROS ALEATORIOS
C IPARM(2)...INDICE FINAL PARA EL ARREGLO DE NUMEROS ALEATORIOS
C
C OBSERVACIONES:
C MUCHAS PUBLICACIONES DEL USO DEL TEST DE PARES SERIALES SON
INCORRECTAS
C COMO LO HA MOSTRADO I.J. GOOD (REF. 1 PARA SU DISCUSION Y OTRAS
REFERENCIAS
C HAY DISTINTAS FORMAS DE REALIZAR ESTE TEST).EL QUE SE USA AQUI ES
UNO DE
C LOS MAS SIMPLES Y SE CONSIDERA RECOMENDABLE CUANDO SE USEN 2-UPLAS
DE
C NUMEROS ALEATORIOS EN UNA RUTINA DE GENERACION DE VARIABLES
ESTOCASTICAS.
C (REF.2 P.167)
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS:
C PNTCHI
C
C METODO:
C REFERENCIAS
C 1. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C GENERATION AND TESTING," AMERICAN SCIENCES
PRESS,COLUMBUS,OHIO,1981
C 2. LEARMONTH,G.P. AND LEWIS,P.A.W.:"STATISTICAL TESTS OF SOME
WIDELY USED
C AND RECENTLY PROPOSED UNIFORM RANDOM NUMBER GENERATORS"
PROCEEDINGS OF
C THE COMPUTER SCIENCE AND STATISTICS SEVENTH ANNUAL SYMPOSIUM ON
THE
C INTERFACE (W.J. KENNEDY,EDITOR),STATISTICAL LABORATORY,IOWA
STATE UNI-
C VERSITY, AMES, IOWA, 1973, PP. 163-171.
C*****
C
C SUBROUTINE TST10
C
C COMMON /MESS/MESSG(9),VARIAB(9)

```

```

COMMON /FOURTH/Y(10000)
COMMON /SIXTH/ XX(7,1000,3),YY(10000,2,3)
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOP
COMMON /T01PRM/NCHI,NPRNT
COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
COMMON /TSTPRM/IPARM(5)
COMMON /T01VAL/INF01(10,6),NDEGF(3)
COMMON /CODE/ GCODE

C
      CHARACTER*8 messg,variab,ckname,name
C
      REAL*8 MESSG,VARIAB,CKNAME,NAME
      DATA CKNAME /'TST01'/

C
      DIMENSION NPRS3(3,3),NPRS10(10,10),NPRS20(20,20)

C

OPEN(UNIT=30,FILE='ARCHT10',ACCESS='SEQUENTIAL',FORM='FORMATTED')
      READ(30,2349)IPARM(1)
      READ(30,2349)IPARM(2)
2349  FORMAT(I11)
      CLOSE(30)
      ISTART=IPARM(1)

C
C
      VERIFICACION DE ERRORES
      IF(ISTART.LE.0) ISTART=1
      IF(IPARM(2).LE.NBATCH) GOTO 1
      WRITE(6,98) IPARM(2),NBATCH
98  FORMAT(1X,'VALOR',I6,' DEL PARAMETRO 2 EN TST10 EXCEDE LA
CANTIDAD
      * DE NUMEROS ALEATORIOS DISPONIBLES',/1X,'PARA EL TEST.EL VALOR
HA
      *SIDO CAMBIADO A: ',I6,'. LA EJECUCION CONTINUA.')
      IPARM(2)=NBATCH
      INF01(7,2)=NBATCH

C
1  N=(IPARM(2)-ISTART+1)/2
      IEND=ISTART+2*N-1

C
      EXP3=N/9.
      EXP10=N/100.
      EXP20=N/400.

C
      DO 10 I=1,3
      DO 10 J=1,3
10  NPRS3(I,J)=0

C
      DO 12 I=1,10
      DO 12 J=1,10
12  NPRS10(I,J)=0

C
      DO 14 I=1,20
      DO 14 J=1,20
14  NPRS20(I,J)=0

C
C
      PARA UN ARREGLO DE 3X3

C
      DO 20 I=ISTART,IEND,2
      ROW=Y(I)*3
      COL=Y(I+1)*3
      IROW=ROW+1
      ICOL=COL+1

```

```

      IF(IROW.GT.3) IROW=3
      IF(ICOL.GT.3) ICOL=3
      NPRS3(IROW,ICOL)=NPRS3(IROW,ICOL)+1
20 CONTINUE
C
C      PARA UN ARREGLO DE 10X10
C
      DO 30 I=ISTART,IEND,2
      ROW=Y(I)*10
      COL=Y(I+1)*10
      IROW=ROW+1
      ICOL=COL+1
      IF(IROW.GT.10) IROW=10
      IF(ICOL.GT.10) ICOL=10
      NPRS10(IROW,ICOL)=NPRS10(IROW,ICOL)+1
30 CONTINUE
C
C      PARA UN ARREGLO DE 20X20
C
      DO 40 I=ISTART,IEND,2
      ROW=Y(I)*20
      COL=Y(I+1)*20
      IROW=ROW+1
      ICOL=COL+1
      IF(IROW.GT.20) IROW=20
      IF(ICOL.GT.20) ICOL=20
      NPRS20(IROW,ICOL)=NPRS20(IROW,ICOL)+1
40 CONTINUE
C
      XX(7,ICNT,1)=0.0
      XX(7,ICNT,2)=0.0
      XX(7,ICNT,3)=0.0
C
      DO 50 I=1,3
      DO 50 J=1,3
50 XX(7,ICNT,1)=XX(7,ICNT,1)+((NPRS3(I,J)-EXP3)**2)/EXP3
C
      DO 60 I=1,10
      DO 60 J=1,10
60 XX(7,ICNT,2)=XX(7,ICNT,2)+((NPRS10(I,J)-EXP10)**2)/EXP10
C
      DO 70 I=1,20
      DO 70 J=1,20
70 XX(7,ICNT,3)=XX(7,ICNT,3)+((NPRS20(I,J)-EXP20)**2)/EXP20
C
      IMPRESION DE RESULTADOS. SE OMITE SI LOS VALORES FUERON
      GENERADOS POR TST01 Y LA CANTIDAD DE MUESTRAS CONSIDERADAS
      EXCEDE LA CANTIDAD DE TABLAS REQUERIDAS POR EL USUARIO
C
      IF((CKNAME.EQ.NAME).AND.(ICNT.GT.NPRNT)) GOTO 999
C
      FORMATO DE LAS SALIDAS
C
      WRITE(*,*)' '
      WRITE(*,*)' '
      WRITE(*,*)' '
C
      WRITE(*,*)'*****'
      *****'
C
      WRITE(*,*)'*****'

```

```

*****
WRITE(*,*)' '
WRITE(*,1234)
1234 FORMAT(1X,'TST10: PARES SERIALES')
WRITE(*,*)' '
WRITE(6,100)
100 FORMAT(' ',1X,'EL TEST DE PARES SERIALES FUE APLICADO (RUTINA
TST1
*0) Y SE HAN OBTENIDO')
WRITE(6,120)
120 FORMAT(' ',1X,'LOS SIGUIENTES RESULTADOS:')
WRITE(*,*)' '
WRITE(6,130) ICNT
130 FORMAT(' ',4X,'NUMERO DE MUESTRA: ',I4)
WRITE(*,2345)
2345 FORMAT(' ',4X,'-----')
WRITE(*,*)' '
WRITE(6,140) N
140 FORMAT(' ',1X,'LA CANTIDAD DE PARES DE NUMEROS ALEATORIOS
PROBADOS
* ES: 'I5)
WRITE(*,*)' '
WRITE(6,160)
160 FORMAT(' ',1X,'PARES DE NUMEROS(Q,R) CON Q Y R ENTRE 1 Y 3')
WRITE(*,*)' '
WRITE(6,165)
165 FORMAT(' ',10X,'NUMEROS EN CATEGORIA(Q,R)')
WRITE(6,170)
170 FORMAT(' ',6X,'-----')
WRITE(6,180)
180 FORMAT(' ',6X,'|      | R=1 | R=2 | R=3 |')
WRITE(6,170)
DO 200 I=1,3
WRITE(6,190) I,(NPRS3(I,J),J=1,3)
190 FORMAT(' ',6X,'| Q=',I1,' | ',3(I5,' | '))
WRITE(6,170)
200 CONTINUE
WRITE(*,*)' '
WRITE(6,210) EXP3
210 FORMAT(' ',4X,'EL NUMERO ESPERADO EN CADA CATEGORIA ES: ',F6.2)
WRITE(6,220) XX(7,ICNT,1)
220 FORMAT(' ',4X,'CHI-CUADRADO PARA 8 GRADOS DE LIBERTAD ES:
',F6.2)
WRITE(6,230)
230 FORMAT(' ',4X,'PROBABILIDAD P QUE UNA VAR. ALEAT. CHI-CUADRADO
CON
* 8 G. DE LIBERTAD')
WRITE(6,240)
240 FORMAT(' ',4X,'SEA <= X:')
WRITE(*,*)' '
WRITE(6,250)
250 FORMAT(' ',1X,'P | .0001 .001 .01 .05 .30 .50 .70
.95
* .99 .999 .9999')
WRITE(6,260)
260 FORMAT(' ',1X,'-----')
--
*-----')
WRITE(6,270)
270 FORMAT(' ',1X,'X | .464 .857 1.65 2.73 5.53 7.34 9.52
15.5

```

```

* 20.1 26.1 31.8')
WRITE(*,*)' '
WRITE(6,290)
290 FORMAT(' ',2X,'PARES DE NUMEROS(Q,R) CON Q Y R ENTRE 1 Y 10')
WRITE(*,*)' '
WRITE(6,165)
WRITE(*,*)' '
WRITE(6,300)
300 FORMAT(' ',1X,'-----')
--
*-----')
WRITE(6,310)
310 FORMAT(' ',1X,' | R=1 | R=2 | R=3 | R=4 | R=5 | R=6 | R=7
|
*R=8 | R=9 | R=10|')
WRITE(6,300)
DO 350 I=1,10
WRITE(6,330) I,(NPRS10(I,J),J=1,10)
330 FORMAT(' ',1X,' | Q=',I2,' | ',10(I4,' | '))
WRITE(6,300)
350 CONTINUE
WRITE(*,*)' '
WRITE(6,410) EXP10
410 FORMAT(' ',4X,'EL NUMERO ESPERADO EN CADA CATEGORIA ES: ',F6.2)
WRITE(6,420) XX(7,ICNT,2)

420 FORMAT(' ',4X,'CHI-CUADRADO PARA 99 GRADOS DE LIBERTAD ES:
',F6.2)
WRITE(6,430)
430 FORMAT(' ',4X,'PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-
CU
*ADRADO')
WRITE(6,440)
440 FORMAT(' ',4X,'CON 99 GRADOS DE LIBERTAD SEA <= X:')
WRITE(*,*)' '
WRITE(6,450)
450 FORMAT(' ',1X,'P| .0001 .001 .01 .05 .30 .50 .70
.95
* .99 .999 .999;')
WRITE(6,460)
460 FORMAT(' ',1X,'-----')
--
*-----')
WRITE(6,470)
470 FORMAT(' ',1X,'X| 55.0 61.1 69.2 77.0 91.5 98.3 105.9
123.
*2 134.6 148.2 160.1')
WRITE(*,*)' '
WRITE(6,590)
590 FORMAT(' ',1X,'PARES DE NUMEROS(Q,R) CON Q Y R ENTRE 1 Y 20')
WRITE(*,*)' '
WRITE(6,165)
WRITE(*,*)' '
WRITE(6,600)
600 FORMAT(' ', '-----')
--
*-----')
WRITE(6,610)
610 FORMAT(' ', | R=1 | R=2 | R=3 | R=4 | R=5 | R=6 | R=7 |
R=8
* | R=9 | R=10|')

```

```

        WRITE(6,600)
        DO 650 I=1,20
        WRITE(6,630) I,(NPRS20(I,J),J=1,10)
630  FORMAT(' ', 'Q=', I2, ' | ', 10(I4, ' | '))
        WRITE(6,600)
650  CONTINUE
        WRITE(*,*)' '
        WRITE(6,601)
601  FORMAT(' ', '-----
--
        *-----')
        WRITE(6,611)
611  FORMAT(' ', ' | R=11| R=12| R=13| R=14| R=15| R=16| R=17|
R=18
        *| R=19| R=20|')
        WRITE(6,601)
        DO 651 I=1,20
        WRITE(6,631) I,(NPRS20(I,J),J=11,20)
631  FORMAT(' ', 'Q=', I2, ' | ', 10(I4, ' | '))
        WRITE(6,601)
651  CONTINUE
        WRITE(*,*)' '
        WRITE(6,710) EXP20
710  FORMAT(' ', 4X, 'EL NUMERO ESPERADO EN CADA CATEGORIA ES: ', F6.2)
        WRITE(6,720) XX(7, ICNT, 3)
720  FORMAT(' ', 4X, 'CHI-CUADRADO PARA 399 GRADOS DE LIBERTAD ES:
', F6.2
        *)
        WRITE(6,730)
730  FORMAT(' ', 4X, 'PROBABILIDAD P DE QUE UNA VARIABLE ALEATORIA CHI-
CU
        *ADRADO')
        WRITE(6,770)
770  FORMAT(' ', 4X, 'CON 399 GRADOS DE LIBERTAD SEA <= X: ')
        WRITE(*,*)' '
        WRITE(6,750)
750  FORMAT(' ', 1X, 'P| .0001 .001 .01 .05 .30 .50 .70
        * .95 .99 .999 .9999')
        WRITE(6,760)
760  FORMAT(' ', 1X, '-----
--
        *-----')
        WRITE(6,800)
800  FORMAT(' ', 1X, 'X| 302.4 317.4 336.2 353.7 383.7 398.3
413.3
        * 446.6 467.6 492.0 512.7')
999  RETURN
        END
C
C
C*****
C  SUBROUTINE TST11
C*****
C
C PROPOSITO:
C DADA UNA LISTA DE N NUMEROS REALES EN PUNTO FLOTANTE ENTRE 0 Y 1,
C QUE SE CONSIDERAN ALEATORIOS, SE TOMAN CONJUNTOS SUCEIVOS DE T
C NUME-
C ROS Y SE BUSCA EL MAXIMO EN CADA UNO. ENTONCES, BAJO ALEATORIEDAD DE
C LOS NUMEROS ORIGINALES, LA FUNCION DE DISTRIBUCION DE LA SUCESION DE
C MAXIMOS ES CONOCIDA.

```

```

C EL TEST K-S EVALUA LA DESVIACION DE LA FUNCION EMPIRICA DE LA SUCE
C SION DE MAXIMOS DE LA FUNCION DE DISTRIBUCION MENCIONADA,USANDO LAS
C MEDIDAS:
C
C   K+ = SQRT(M)*MAX(EMPIRIC(X)-X**T)
C   K- = SQRT(M)*MAX(X**T-EMPIRIC(X))
C   K* = MAX(K+,K-),
C
C DONDE M ES LA CANTIDAD DE CONJUNTOS DE T NUMEROS USADOS,Y EL MAXIMO
C SE TOMA SOBRE TODOS LOS NUMEROS REALES 0.0<=X<=1.0 ,PARA CADA
C CONJUNTO DE NUMEROS ESPECIFICADO.
C TST11 CALCULA E IMPRIME K+, K-, K* PUNTOS PORCENTUALES DE SUS
C RESPECTIVAS DISTRIBUCIONES PARA USARLOS EN LA INTERPRETACION DE SUS
C VALORES
C
C PARAMETROS:
C   VARIABLE                LONG.          NUM./CAR.
C   TNAME                   8             CAR
C   IPARM(1)                 11            NUM
C   IPARM(2)                 11            NUM
C   IPARM(3)                 11            NUM
C   IPARM(4)                 11            NUM
C
C DESCRIPCION DE PARAMETROS
C   IPARM(1)...VALOR DE T A SER USADO (SE PERMITEN 2,3,4,...).
C   IPARM(2)...CANTIDAD DE MAXIMOS (DE T NUMEROS ALEAT. CADA UNO) A
C   SER USADOS EN CADA TEST DE K-S.
C   IPARM(3)...CANTIDAD DE CONJUNTOS DE IPARM(2) MAXIMOS CADA UNO QUE
C   SERAN VERIFICADOS. DEBE COMPROBARSE:
C   T*IPARM(2)*IPARM(3)<=NBATCH.
C   IPARM(4)...INDICE PARA COMENZAR EN EL ARREGLO DE NUMEROS
C   ALEATORIOS
C   DEBE SER IPARM(4)-1+T*IPARM(2)*IPARM(3) <= NBATCH.
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS:
C   KSTPL,CDFKS,XTRMS, SORT
C
C METODO:
C REFERENCIAS
C 1. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C   GENERATION AND TESTING",AMERICAN SCIENCES PRESS,COLUMBUS,OHIO,
C   1981
C*****
C
C   SUBROUTINE TST11
C   COMMON /MESS/MESSG(9),VARIAB(9)
C   COMMON /FOURTH/Y(10000)
C   COMMON /SIXTH/XX(7,1000,3),YY(10000,2,3)
C   COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C   COMMON /T01PRM/NCHI,NPRNT
C   COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C   COMMON /TSTPRM/IPARM(5)
C   COMMON /CODE/GCODE
C   COMMON /T01VAL/INF01(10,6),NDEGF(3)
C
C   CHARACTER*8 messg,variab,ckname,name
C   REAL*8 MESSG,VARIAB,NAME,CKNAME
C   INTEGER T
C
C   DATA CKNAME/'TST01'/
C

```



```

OPEN(UNIT=31,FILE='ARCHT11',ACCESS='SEQUENTIAL',FORM='FORMATTED')
  READ(31,2349)IPARM(1)
  READ(31,2349)IPARM(2)
  READ(31,2349)IPARM(3)
  READ(31,2349)IPARM(4)
2349 FORMAT(I11)
  CLOSE(31)
  T=IPARM(1)
  NMAX=IPARM(2)
  NREP=IPARM(3)
  ISTART=IPARM(4)

C
  N=T*NMAX
  NTOT=N*NREP

C
C   VERIFICACION DE ERRORES
C
  IF(ISTART.LE.0) ISTART=1
  IF(NTOT.LE.NBATCH) GOTO 2
  WRITE(6,90) NMAX,NREP
  90 FORMAT(1X,'EL PRODUCTO DEL SEGUNDO PARAMETRO (EL NUMERO DE
MAXIMOS A SER UTILIZADOS EN CADA',/1X,'TEST K-S)',I7,/1X,'POR EL
TERCER PARAMETRO',I7,'(EL NUMERO DE TESTS INDIVIDUALES A SER
REALIZADOS)',
  */1X,'EXCEDE LA CANTIDAD DE NUMEROS ALEATORIOS
DISPONIBLES.',/1X,'T
*EST11 ES IGNORADO.')

C
  GOTO 6
  2 IF(NTOT.GT.0) GOTO 4
  WRITE(6,92) NMAX,NREP
  92 FORMAT(1X,'EL SEGUNDO O TERCER PARAMETRO,',I7,' O ',I7,
  *', COMO ESTAN ESPECIFICADOS NO PROVEEN SUFICIENTES NUMEROS',/1X,
  *' PARA RESULTADOS SIGNIFICATIVOS',/1X,'TST11 ES IGNORADO.')

C
C   SI LOS VALORES FUERON GENERADOS POR TST01,
C   PONER OFF EL SWITCH QUE LLAMA A KS SOBRE KS PARA TST11
C
  6 IF(NAME.EQ.CKNAME) INF01(10,6)=0
  GOTO 999

C
  4 IF((CKNAME.EQ.NAME).AND.(ICNT.GT.NPRNT)) GOTO 8

C
C   ENCABEZAMIENTO DE LA PRIMERA PAGINA IMPRESA CON LOS VALORES DE
LOS
C   PARAMETROS
  WRITE(*,*)' '
  WRITE(*,*)' '
  WRITE(*,*)' '

WRITE(*,*)'*****'
*****

WRITE(*,*)'*****'
*****

  WRITE(*,*)' '
  WRITE(*,1234)
1234 FORMAT(1X,'TST11: MAX DE T')
  WRITE(*,*)' '
  WRITE(6,500)

```

```

500 FORMAT(' ',1X,'EL TEST K-S EN MAX DE T FUE APLICADO (RUTINA
TST11)
      * Y SE HAN OBTENIDO')
      WRITE(6,401)
401 FORMAT(' ',1X,'LOS SIGUIENTES RESULTADOS:')
      WRITE(*,*)' '
      WRITE(6,345)
345 FORMAT(' ', 'ESPECIFICACIONES DEL USUARIO:')
      WRITE(6,110) T
C      WRITE(6,342)NMAX
110 FORMAT(' ', 'EL TEST USA EL MAXIMO DE LOS CONJUNTOS DE ',I2,'
VALOR
      *ES')
342 FORMAT(' ', 'CADA TEST USA ',I6,' DE ESOS VALORES MAXIMOS')
      WRITE(6,343)NREP
343 FORMAT(' ', 'HAY ',I6,' REPETICIONES DEL TEST POR MUESTRA.')
      WRITE(*,*)' '
C
C      BUCLE DE REPETICIONES
C
C      8 DO 20 I=1,NREP
C
C      EL TEST MAX T EMPIEZA AQUI
C
C      INDX=ISTART+(I-1)*N
C
C      DO 11 J=1,NMAX
C      I1=INDX+((J-1)*T)-1
C      RMAX=-1
C
C      DO 10 J1=1,T
C      K1=I1+J1
C      TEMP=Y(K1)
C      IF(TEMP.GT.RMAX) RMAX=TEMP
10 CONTINUE
C
C      K1=INDX+J-1
C      Y(K1)=RMAX
C
C      11 CONTINUE
C
C      IF((CKNAME.EQ.NAME).AND.(ICNT.GT.NPRNT)) GOTO 15
C
C      LINEA INFERIOR EN EL PLOT
C
C      WRITE(*,7777)
7777 FORMAT('-----
--
      *-----')
      WRITE(6,120) I,NREP,ICNT
120 FORMAT(' ', 'SUBMUESTRA ',I4,' (DE',I4,') EN LA MUESTRA ',I4,' DE
N
      *UMEROS ALEATORIOS ')
C
C      LLAMADO AL PLOT
C ( OPCION INHABILITADA EN ESTA VERSION )
C      CALL KSTPL(Y(INDX),NMAX,T)
C      GOTO 17
C
C      15 CALL SORT(Y(INDX),NMAX)
C

```

```

C          CALCULO DE LOS VALORES DE K-S
C
17 DO 12 J=1,NMAX
    K1=INDX+J-1
12 Y(K1)=FLOAT(J)/FLOAT(NMAX)-Y(K1)**T
    LAST=INDX+NMAX-1
    CALL XTRMS(Y,INDX, LAST,AMIN,AMAX)
C
    VALPOS=SQRT(FLOAT(NMAX))*AMAX
    VALMIN=SQRT(FLOAT(NMAX))*(-AMIN+1.0/NMAX)
    VAL=AMAX1(VALPOS,VALMIN)
    VALCK=VAL
    IF(VALCK.GT.50.0) VALCK=50.0
C
    EXPPOS=2*(VALPOS**2)
    EXPMIN=2*(VALMIN**2)
    IF(EXPPOS.GT.20) EXPPOS=20
    IF(EXPMIN.GT.20) EXPMIN=20
C
    PROBP=1-EXP(-EXPPPOS)
    PROBM=1-EXP(-EXPMIN)
    PROB=CDFKS(VALCK)
C
    IF(PROBP.GT.1.0) PROBP=1.0000
    IF(PROBM.GT.1.0) PROBM=1.0000
    IF(PROB.GT.1.0) PROB=1.0000
    IF((CKNAME.EQ.NAME).AND.(ICNT.GT.NPRNT)) GOTO 18
C
    WRITE(6,112) T,NMAX,VALPOS,VALMIN,VAL
112 FORMAT('0','EL TEST K-S EN MAX DE T ( T = ',I2,' ) CON N/T = ',
    *I6,' PRODUCE:',/,4X,'K+ = ',F9.4,' ',/,14X,'K- = ',F9.4,' ',/,14X,
    *'K = ',F9.4,' .')
    IF(NMAX.LT.100) GOTO 16
C
    WRITE(6,114)
114 FORMAT(' ','AQUI, LA SIGNIFICANCIA PUEDE SER EVALUADA USANDO LOS
H
    *ECHOS (OBTENIDOS DE LAS',/1X,'DISTRIBUCIONES LIMITES DE K+, K-,
K)
    * DE QUE: ',/)
    WRITE(6,116) VALPOS,PROBP,VALMIN,PROBM,VALCK,PROB
116 FORMAT(' ',2X,'P(K+ <= ',F8.4,' )=',G10.4,/3X,'P(K- <= ',F8.4,'
)=
    ',G10.4,/3X,'P(K <= ',F8.4,' )=',G10.4)
    GOTO 18
C
16 WRITE(6,118)
118 FORMAT(' ','AQUI, LA SIGNIFICANCIA PUEDE EVALUARSE USANDO LA
TABLA
    *1 DE L.H.MILLER,"TABLE OF',/1X,'PERCENTAGE POINTS OF KOLMOGOROV
ST
    *ATISTICS"', '- JOURNAL OF THE AMERICAN
STATISTICAL',/1X,'ASSOCIATIO
    *N,VOL.51(1956),PAG.111-112.',/1X,'NOTAR QUE ALLI SE USA LA
NOTACIO
    *N D,D*. LA CORRESPONDENCIA ES:',/1X,' K+ = SQRT(N/T)*D, K =
SQR
    *T(N/T)*(D*).' )
C
C
18 IF(I.GT.10) GOTO 20

```

```

        ICNTMT=ICNTMT+1
        YY(ICNTMT,2,1)=VALPOS
        YY(ICNTMT,2,2)=VALMIN
        YY(ICNTMT,2,3)=VAL
C
    20 CONTINUE
    999 RETURN
        END
C*****
    SUBROUTINE TSTENTR
C
C    PRUEBAS DE ENTROPIA CONTINUA EMPLEANDO
C    LOS SIGUIENTES TESTS:  VASICEK - Hmn
C    Y DUDEWICZ-VAN DER MEULEN - H*ALFA(m,n)
C*****
C X: ARREGLO UNIDIMENSIONAL DE 'CANTELEM' NÚMEROS ALEATORIOS
C    EN DOBLE PRECISIÓN, PARA CADA SUCESSION GENERADA A TRAVES
C    DE 'NLOTES' LLAMADOS AL GENERADOR URNWN
C
C M: ARREGLO DE VALORES ENTEROS POSITIVOS
C    (M=1,2,...,10,15,20,30,40) PARA UTILIZAR EN EL CÁLCULO DE
C    Hmn Y H*ALFA(m,n) EN CADA SUCESIÓN.
C
C DESCRIPCION DE ARCHIVOS
C
C ARCHDAT: ARCHIVO CON FORMATO F11.10 CONTENIENDO
C    NUMLOTES*CANTELEM NUMEROS ALEATORIOS GENERADOS POR URNWN
C*****
    DOUBLE PRECISION X,RES,SUM,N,LL,TT,MULT,RR,P,XKK
    DIMENSION X(10000),M(14),LOSM(14)
    INTEGER CANTELEM
    OPEN(UNIT=90,RECL=11,FORM='FORMATTED',FILE='ARCHDAT')
C
    READ(65,63) NUMLOTES
    READ(65,63) CANTELEM
    63 FORMAT(I8)
C    WRITE(*,*) 'Ingrese NUMLOTES,CANTELEM '
C    read(*,*)numlotes,cantelem
C
    WRITE(*,*)' '
    WRITE(*,*)' '

WRITE(*,*)'*****
*****'

WRITE(*,*)'*****
*****'
    WRITE(*,*)' '
    WRITE(*,1234)
1234 FORMAT(1X,' TESTS DE ENTROPIA CONTINUA')
    WRITE(*,*)' '
C-----
    M(1)=40.0
    M(2)=30.0
    M(3)=20.0
    M(4)=15.0
    M(5)=10.0
    M(6)=9.0
    M(7)=8.0
    M(8)=7.0
    M(9)=6.0

```

```

M(10)=5.0
M(11)=4.0
M(12)=3.0
M(13)=2.0
M(14)=1.0
DO 23 I= 1,14
23  LOSM(I)=0
C
WRITE(*,*)'CALCULO DE Hmn PARA M=1,...,10,15,20,30,40 (ARCHIVO ARC
*HDAT) '
WRITE(*,*)' '
C  N=10000.D0
   N=DBLE(CANTELEM)
C  WRITE(*,*)' INGRESAR NUMLOTES '
C  NUMLOTES=1000
C  READ(*,*) NUMLOTES
DO 444 ITER = 1 ,NUMLOTES
C
DO 666 KAPA = 1, CANTELEM
READ(90,456)X(KAPA)
666 CONTINUE
456 FORMAT(F11.10)
C
CALL SORTENTR(X,CANTELEM)
C
DO 30 JO=1,14
SUM=0.D0
RES=0.D0
MULT=0.D0
TT=0.D0
LL=0.D0
DO 20 J=1,CANTELEM
IF (J-M(JO).LT.1) SUBI1=1
IF (J-M(JO).GE.1) SUBI1=J-M(JO)
IF (J+M(JO).GT.N) SUBI2=CANTELEM
IF (J+M(JO).LE.N) SUBI2=J+M(JO)
RES=X(SUBI2)-X(SUBI1)
IF (ABS(RES).LT.1D0-10)THEN
WRITE(*,*)'H=0 , H*=0'
WRITE(*,*)
ENDIF
TT=N/(2*M(JO))
MULT=TT*RES
IF (MULT.EQ.0.0)THEN
LL=0.
IM=M(JO)
CALL LOSEMES(LOSM,IM)
GOTO 19
ENDIF
LL=LOG(MULT)
19  SUM=SUM+LL
20  CONTINUE
   H = SUM/N
30  CONTINUE
444 CONTINUE
CLOSE(90)
DO 3000 I = 1 ,14
IF (LOSM(I).NE.0) THEN
WRITE(*,*)'USARA EUXXFD10, PUES SE OBTIENEN VALORES=INF'
WRITE(*,*)
CALL EUXXFD10 (NUMLOTES, CANTELEM)

```

```

        GOTO 8000
    ENDIF
3000 CONTINUE
    WRITE(*,*) 'USARA EUXFIND, PUES SE OBTIENEN VALORES FINITOS'
    WRITE(*,*)
    CALL EUXXFIND(NUMLOTES, CANTELEM)
8000 CALL EUXXFINE(NUMLOTES)
    RETURN
    END
C
C
C*****
    SUBROUTINE SORTENTR(R,NN)
    DOUBLE PRECISION R(10000)
    MM=NN
1    MM=MM/2
    IF (MM.EQ.0.0) GOTO 5
    J=1
2    I=J
4    IF (R(I).LE.R(I+MM)) GOTO 3
    SAV=R(I)
    R(I)=R(I+MM)
    R(I+MM)=SAV
    I=I-MM
    IF (I.GE.1) GOTO 4
3    J=J+1
    IF (J.GT.NN-MM) GOTO 1
    GOTO 2
5    RETURN
    END
C
C*****
    SUBROUTINE LOSEMES(LOSM,IM)
    DIMENSION X(10000),LOSM(14)
    GOTO ( 1,2,3,4,5,6,7,8,9,10,11,12,13,14),IM
1    LOSM(1)=LOSM(1)+1
    RETURN
2    LOSM(2)=LOSM(2)+1
    RETURN
3    LOSM(3)=LOSM(3)+1
    RETURN
4    LOSM(4)=LOSM(4)+1
    RETURN
5    LOSM(5)=LOSM(5)+1
    RETURN
6    LOSM(6)=LOSM(6)+1
    RETURN
7    LOSM(7)=LOSM(7)+1
    RETURN
8    LOSM(8)=LOSM(8)+1
    RETURN
9    LOSM(9)=LOSM(9)+1
    RETURN
10   LOSM(10)=LOSM(10)+1
    RETURN
11   LOSM(11)=LOSM(11)+1
    RETURN
12   LOSM(12)=LOSM(12)+1
    RETURN
13   LOSM(13)=LOSM(13)+1
    RETURN

```

```

14   LOSM(14)=LOSM(14)+1
      RETURN
      END

C
C
C*****
      SUBROUTINE EUXXFD10(NUMLOTES,CANTELEM)

C
C   PARA CASOS DONDE SE OBTIENEN VALORES INFINITOS
C
C
C*****
*
C PROPOSITO:
C ANALOGO AL PROGRAMA EUXXFIND, PARA SER UTILIZADO CUANDO EXISTAN
C ALGUNOS M PARA LOS CUALES SE OBTIENEN VALORES INFINITOS
C (V.GR. URN10, URN13, URN39)
C
C - CALCULAR EL ESTIMADOR DE ENTROPIA DE VASICEK: Hmn
C - DETERMINAR EL ESTIMADOR ASINTOTICO AL PERCENTIL 100xALFA
C - PROBAR EL TEST DE DUDEWICZ-VAN DER MEULEN:
C       HMN <= H*ALFA(m,n)
C   CALCULANDO PORCENTAJES DE VERIFICACIONES
C
C DESCRIPCION DE PARAMETROS
C   X: ARREGLO UNIDIMENSIONAL PARA CADA SUCESSION DE 'CANTELEM'
C   NUMEROS ALEATORIOS GENERADOS A TRAVES DE 'NUMLOTES'
C   LLAMADOS AL GENERADOR URNWN
C   M: ARREGLO DE VALORES ENTEROS POSITIVOS (M=1,2,...,10,15,20,
C       30,40) PARA UTILIZAR EN EL CÁLCULO DE Hmn Y H*ALFA(m,n)
C   EN CADA SUCESIÓN
C   RESU,H14,PROMH: ARREGLOS PARA ALMACENAR VALORES DE LOS
C   ESTADÍSTICOS
C
C DESCRIPCION DE ARCHIVOS
C
C ARCHDAT: ARCHIVO DE DATOS CON FORMATO F11.10 CONTENIENDO
C NUMLOTES*CANTELEM NÚMEROS ALEATORIOS GENERADOS POR URNWN
C HMNUXX.DAT:ARCHIVO DE RESULTADOS DE VERIFICACIONES DE UN TEST.
C PRHUXX.DAT:ARCHIVO DE PROMEDIOS, DE LONGITUD QUINCE.
C HF14UXX.DAT:ARCHIVO DE REGISTROS DE LONGITUD QUINCE, PARA LOS
C VALORES
C DE H*ALFA(m,n)
C*****
      DOUBLE PRECISION X,RES,SUM,N,LL,TT,MULT,RR,P,XKK
      DIMENSION X(10000),M(14),RESU(14),H14(14),PROMH(14)
      INTEGER CANTELEM
      OPEN(UNIT=90,RECL=11,FORM='FORMATTED',FILE='ARCHDAT')
      OPEN(UNIT=98,ACCESS='DIRECT',RECL=3,FORM='FORMATTED',FILE=
* 'HMNUXX.DAT')
      OPEN(UNIT=99,ACCESS='DIRECT',RECL=15,FORM='FORMATTED',FILE=
* 'PRHUXX.DAT')
      OPEN(UNIT=61,ACCESS='DIRECT',RECL=15,FORM='FORMATTED',FILE=
* 'HF14UXX.DAT')
      M(1)=40.0
      M(2)=30.0
      M(3)=20.0
      M(4)=15.0
      M(5)=10.0
      M(6)=9.0
      M(7)=8.0

```

```

M(8)=7.0
M(9)=6.0
M(10)=5.0
M(11)=4.0
M(12)=3.0
M(13)=2.0
M(14)=1.0
DO 23 I=1,14
PROMH(I)=0.
H14(I)=0.
RESU(I)=0.
23 CONTINUE
N=DBLE(CANTELEM)
C N=10000.D0
C NUMLOTES=1000
DO 444 ITER = 1 ,NUMLOTES
DO 666 KAPA = 1,CANTELEM
READ(90,456)X(KAPA)
666 CONTINUE
456 FORMAT(F11.10)
C
CALL SORTENTR(X,CANTELEM)
DO 30 JO=1,14
SUM=0.D0
RES=0.D0
MULT=0.D0
TT=0.D0
LL=0.D0
DO 20 J=1,CANTELEM
IF (J-M(JO).LT.1) SUBI1=1
IF (J-M(JO).GE.1) SUBI1=J-M(JO)
IF (J+M(JO).GT.N) SUBI2=CANTELEM
IF (J+M(JO).LE.N) SUBI2=J+M(JO)
RES=X(SUBI2)-X(SUBI1)
C IF (ABS(RES).LT.1D0-10)THEN
C WRITE(*,*) 'H=0 , H*=0'
C WRITE(*,*)
C PAUSE
C ENDDIF
TT=N/(2*M(JO))
MULT=TT*RES
IF (MULT.EQ.0.0)THEN
LL=0.
GOTO 19
ENDDIF
LL=LOG(MULT)
19 SUM=SUM+LL
20 CONTINUE
C H=N**(-1)*SUM
C WRITE(*,*) 'SUM H , SALIENDO DE DO 20 '
C WRITE(*,*)SUM,H
C H = SUM/N
C
C CASOS
C IF(MULT.EQ.0.0)THEN
C WRITE(*,*) 'H ES -INF, PARA M='
C WRITE(*,*)M(JO)
C ENDDIF
C RESULTADOS EN RESU
NN=CANTELEM
C NN= 10000.

```



```

MM= INT(M(JO))
XM = M(JO)
GAMA = .5772156649D0
RR = 0.0
C CALCULO DEL ESTADISTICO H*ALFA(M,N), ALFA=0.05
  PH11ALF= -1.645
  RAIZCU = SQRT (6.* XM*NN)
  TERM1 = PH11ALF / RAIZCU
  TERM2 = LOG (2.* XM)
  TERM3 = LOG((N+1)/N)
C CALCULO DE R (1,2M-1) =R(J,K*)=K** -1 + (K-1)** -1 +...+J** -1
  KK = 2*MM-1
  DO 987 ITE = 2, KK
  XKK= 1./ITE
  RR = RR + XKK
987 CONTINUE
  RR = RR+1.
C CALCULO DE LA FORM.8 - H*ALF(M,N)
  HALFNM = TERM1- TERM2 - GAMA +RR - TERM3
C
C CALCULO DE LA FORM. 10
  H14(JO) = RAIZCU*( H + TERM2 +GAMA - RR)
C WRITE(*,17)HALFMN
C17 FORMAT(1X, 'H*ALF(M,N)=' ,F20.8)
  AAH=ABS(H)
  BBH=ABS(HALFMN)
  IF(AAH.LE.BBH) THEN
    RESU(JO) = 1.
    PROMH(JO) = PROMH(JO)+H
  ELSE
    RESU(JO) = 0.
  ENDIF
30 CONTINUE
C CREA ARCHIVOS 98,99 Y 61
  DO 56 I = 1,14
    WRITE(98,457)RESU(I)
    WRITE(99,458)PROMH(I)
    WRITE(61,458)H14(I)
457 FORMAT(F3.0)
458 FORMAT(F15.8)
56 CONTINUE
444 CONTINUE
  CLOSE(90)
  CLOSE(98)
  CLOSE(99)
  CLOSE(61)
  RETURN
  END
C
C
C*****
  SUBROUTINE EUXXFIND (NUMLOTES,CANTELEM)
C
C PARA CASOS DONDE SE OBTIENEN VALORES FINITOS
C*****
C PROPOSITO:
C - CALCULAR EL ESTIMADOR DE ENTROPÍA DE VASICEK :Hmn
C - DETERMINAR EL ESTIMADOR ASINTOTICO AL PERCENTIL 100*ALFA MEDIANTE
C - APROXIMACIÓN H*ALFA (FORM. 11, cfr. 2.4.3)
C - PROBAR EL TEST DE DUDEWICZ-VAN DER MEULEN: Hmn<=H*alfa(m,n)
C CALCULANDO PORCENTAJES DE VERIFICACIONES

```

```

C
C DESCRIPCION DE PARAMETROS
C X:  ARREGLO UNIDIMENSIONAL PARA CADA SUCESSION DE 'NUMLOTES' NUMEROS
C     ALEATORIOS GENERADOS A TRAVÉS DE 1000 LLAMADOS AL GENERADOR
URNWN
C     M:  ARREGLO DE VALORES ENTEROS POSITIVOS
C         (M=1,2,...,10,15,20,30,40)
C     PARA UTILIZAR EN EL CALCULO DE Hmn Y H*ALFA(m,n) EN CADA SUCESIÓN
C     RESU,H14,PROMH:  ARREGLOS PARA ALMACENAR VALORES DE LOS
C     ESTADISTICOS
C
C DESCRIPCION DE ARCHIVOS
C ARCHDAT:ARCHIVO DE ACCESO DIRECTO CON FORMATO F11.10 CONTENIENDO
C NUMLOTES*CANTELEM NUMEROS ALEATORIOS GENERADOS POR URNWN
C HMNUXX.DAT: ARCHIVO DE RESULTADOS DE VERIFICACIONES DE UN TEST.
C PRHUXX.DAT: ARCHIVO DE PROMEDIOS, DE LONGITUD QUINCE.
C HF14UXX.DAT: ARCHIVO DE REGISTROS DE LONGITUD QUINCE, PARA LOS
C VALORES DEL ESTIMADOR H*ALFA(m,n)

DOUBLE PRECISION X,RES,SUM,N,LL,TT,MULT,RR,P,XKK
DIMENSION X(10000),M(14),RESU(14),H14(14),PROMH(14)
INTEGER CANTELEM
OPEN(UNIT=90,RECL=11,FORM='FORMATTED',FILE='ARCHDAT')
OPEN(UNIT=98,ACCESS='DIRECT',RECL=3,FORM='FORMATTED',FILE=
*'HMNUXX.DAT')
OPEN(UNIT=99,ACCESS='DIRECT',RECL=15,FORM='FORMATTED',FILE=
*'PRHUXX.DAT')
OPEN(UNIT=61,ACCESS='DIRECT',RECL=15,FORM='FORMATTED',FILE=
*'HF14UXX.DAT')
M(1)=40.0
M(2)=30.0
M(3)=20.0
M(4)=15.0
M(5)=10.0
M(6)=9.0
M(7)=8.0
M(8)=7.0
M(9)=6.0
M(10)=5.0
M(11)=4.0
M(12)=3.0
M(13)=2.0
M(14)=1.0
DO 23 I=1,14
PROMH(I)=0.
H14(I)=0.
RESU(I)=0.
23 CONTINUE
N=DBLE(CANTELEM)
C     N=10000.D0
C     WRITE(*,*)' INGRESAR NUMLOTES'
C     NUMLOTES=1000
C     READ(*,*) NUMLOTES
DO 444 ITER = 1 ,NUMLOTES
DO 666 KAPA = 1,CANTELEM
READ(90,456)X(KAPA)
666 CONTINUE
456 FORMAT(F11.10)
C
CALL SORTENTR(X,CANTELEM)

```

```

DO 30 JO=1,14
SUM=0.D0
RES=0.D0
MULT=0.D0
TT=0.D0
LL=0.D0
DO 20 J=1,CANTELEM
  IF (J-M(JO).LT.1) SUBI1=1
  IF (J-M(JO).GE.1) SUBI1=J-M(JO)
  IF (J+M(JO).GT.N) SUBI2=CANTELEM
  IF (J+M(JO).LE.N) SUBI2=J+M(JO)
  RES=X(SUBI2)-X(SUBI1)
  IF (ABS(RES).LT.1D0-10)THEN
    WRITE(*,*)'H=0 , H*=0'
    WRITE(*,*)
  ENDIF
  TT=N/(2*M(JO))
  MULT=TT*RES
  LL=LOG(MULT)
  SUM=SUM+LL
20 CONTINUE
C   H=N**(-1)*SUM
   H = SUM/N
   IF(MULT.EQ.0.0)THEN
     WRITE(*,*)'H ES -INF, PARA M='
     WRITE(*,*)M(JO)
     GOTO 30
   ENDIF
C   RESULTADOS EN RESU
   NN=CANTELEM
C   NN= 10000.
   MM= INT(M(JO))
   XM = M(JO)
   GAMA = .5772156649D0
   RR = 0.0
C   CALCULO DEL ESTADISTICO H*ALFA(M,N), ALFA=0.05
   PHIALF= -1.645
   RAIZCU = SQRT (6.* XM*NN)
   TERM1 = PHIALF / RAIZCU
   TERM2 = LOG (2.* XM)
   TERM3 = LOG((N+1)/N)
C   CALCULO DE R (1,2M-1) =R(J,K*)=K**(-1) + (K-1)**(-1) +...+J**(-1)
   KK = 2*MM-1
   DO 987 ITE = 2, KK
     XKK= 1./ITE
     RR = RR + XKK
987 CONTINUE
   RR = RR+1.
C   CALCULO DE LA FORM.11 H*ALF(M,N)
   HALFMN = TERM1- TERM2 - GAMA +RR - TERM3
C   CALCULO DE LA FORM.7 (VER 2.4.1)
   H14(JO) = RAIZCU*( H + TERM2 +GAMA - RR)
C   WRITE(*,17)HALFMN
C17  FORMAT(1X, 'H*ALF(M,N)=' ,F20.8)
     AAH=ABS(H)
     BBH=ABS(HALFMN)
     IF(AAH.LE.BBH) THEN
       RESU(JO) = 1.
       PROMH(JO) = PROMH(JO)+H
     ELSE
       RESU(JO) = 0.

```

```

        ENDIF
30    CONTINUE
C    CREA ARCHIVOS 98,99 Y 61
        DO 56 I = 1,14
            WRITE(98,457)RESU(I)
            WRITE(99,458)PROMH(I)
            WRITE(61,458)H14(I)
457    FORMAT(F3.0)
458    FORMAT(F15.8)
56    CONTINUE
444   CONTINUE
        CLOSE(90)
        CLOSE(98)
        CLOSE(99)
        CLOSE(61)
        RETURN
        END
C
        SUBROUTINE EUXXFINE(NUMLOTES)
C
C    RESULTADOS DE LOS TESTS DE ENTROPIA
C*****
C    PROPOSITO:
C
C    IMPRIME LOS RESULTADOS DE CALCULAR EL ESTIMADOR DE ENTROPIA DE
C    VASICEK; C DEL ESTIMADOR ASINTÓTICO AL PERCENTIL 100*ALFA ; Y DE LAS
C    PRUEBAS DEL C TEST DE DUDEWICZ-VAN DER MEULEN.
C
C    DESCRIPCION DE PARAMETROS
C        M:  ARREGLO DE VALORES ENTEROS POSITIVOS
C            (M=1,2,...,10,15,20,30,40)
C        SM, PROMSM, PM, PMH, F14 Y PF14:  ARREGLOS DE LONGITUD 14 PARA
C        ALMACENAR VALORES PROMEDIOS.
C
C    DESCRIPCION DE ARCHIVOS
C
C    HMNUXX.DAT:  ARCHIVO DE RESULTADOS DE VERIFICACIONES DE UN TEST.
C    PRHUXX.DAT :  ARCHIVO DE PROMEDIOS DE LONGITUD QUINCE.
C    HF14UXX.DAT:  ARCHIVO DE REGISTROS DE LONGITUD QUINCE PARA LOS
C                  VALORES DEL ESTIMADOR H*ALFA(m,n)
C    OBSERVACION: SU USO ES COMUN PARA PROCESAR EN SU ULTIMA ETAPA TODOS
C    LOS
C    RESULTADOS OBTENIDOS EN LOS PROGRAMAS EJECUTADOS PRECEDENTEMENTE:
C    EUXXMM10, EUXMDP10, EUXXFIND, EUXXFD10 Y EUXFINDP.
C
C*****
*
        DIMENSION M(14),SM(14),PROMSM(14),PM(14),PMH(14),F14(14),
        *PF14(14)
        OPEN(UNIT=98,ACCESS='DIRECT',RECL=3,FORM='FORMATTED',FILE=
        &'HMNUXX.DAT')
        OPEN(UNIT=99,ACCESS='DIRECT',RECL=15,FORM='FORMATTED',FILE=
        &'PRHUXX.DAT')
        OPEN(UNIT=61,ACCESS='DIRECT',RECL=15,FORM='FORMATTED',FILE=
        &'HF14UXX.DAT')
C
        WRITE(*,11)
11    FORMAT(1X,' RESULTADOS TEST ENTROPIA HMN - GENERADOR URNWN')
        WRITE(*,*)
        M(1)=40.0

```

```

M(2)=30.0
M(3)=20.0
M(4)=15.0
M(5)=10.0
M(6)=9.0
M(7)=8.0
M(8)=7.0
M(9)=6.0
M(10)=5.0
M(11)=4.0
M(12)=3.0
M(13)=2.0
M(14)=1.0
DO 10 I = 1,14
  PM(I)=0.0
  SM(I)=0.0
  F14(I)=0.0
10  ENDDO
DO 20 I = 1,NUMLOTES
  DO 30 J= 1,14
    READ(98,123)VAL
    SM(J)=SM(J)+VAL
30  CONTINUE
123  FORMAT(F3.0)
20  CONTINUE
DO 25 I = 1,NUMLOTES
  DO 35 J=1,14
    READ(99,124)VALH
124  FORMAT(F15.8)
    PM(J)=PM(J)+VALH
35  CONTINUE
25  CONTINUE
DO 65 I = 1,NUMLOTES
  DO 75 J=1,14
    READ(61,124)VALF
    F14(J)=F14(J)+VALF
75  CONTINUE
65  CONTINUE
CLOSE(98)
CLOSE(99)
CLOSE(61)
OPEN(UNIT=80,RECL=15,FORM='FORMATTED',FILE=
*'VAL_F1.TXT')
WRITE(*,*) ' M                      Hm(m,PROMEDIO) <= H*(m,n)
*VALOR F EN 100*FI(F)'
DO 60 I = 1,14
  PROMSM(I) = SM(I)/10
  PMH(I) = PM(I)/NUMLOTES
  PF14(I) = F14(I)/NUMLOTES
125  WRITE(*,125)M(I),PMH(I),PF14(I)
  FORMAT(1X,I3,20X,F15.8,10X,F15.8)
  WRITE(80,126)PF14(I)
126  FORMAT(F15.8)
60  CONTINUE
CLOSE(90)
WRITE(*,12)
12  FORMAT(/,1X,'EL PTO.PERCENTIL 100*FI(F),SE OBTIENE POR METODOS
*ESTADISTICOS')
WRITE(*,*) ' CFR. MANUAL DEL USUARIO'
RETURN
END

```

```

C
C*****
C   SUBROUTINE REPORT(ISUB)
C*****
C PROPOSITO:
C REPORT ES LLAMADA CUANDO SE ENCUENTRA UNA ARCHIVO DE ENTRADA DE
C TST01.
C ESTA SUBROUTINA LEE Y ALMACENA LOS PARAMETROS DE TST01 Y CUENTA EL
C NUMERO
C TOTAL DE BITS 1 EN "CSCODE" Y "KSCODE".
C SE LLAMA LUEGO AL GENERADOR DE NUMEROS ALEAT.
C ESPECIFICADO, OBTENIENDO
C MUESTRAS DE "NBATCH" NUMEROS CADA UNA. LOS TEST INDIVIDUALES
C REQUERIDOS SE
C APLICAN A CADA NUEVA MUESTRA Y LOS RESULTADOS DE LAS ESTADISTICAS
C CHI-CUAD.
C Y K-S CALCULADAS POR LOS TESTS SON ALMACENADOS PARA SER PROCESADOS
C DESPUES POR TST01.
C LUEGO DE VERIFICAR "NCHI" MUESTRAS, REPORT IMPRIME UNA TABLA DE
C ESTADISTICAS
C GENERADAS POR EL TEST INDIVIDUAL Y, SI SE REQUIERE, UNA TABLA DE
C TIEMPOS
C PROMEDIO DE CPU NECESARIOS PARA GENERAR UN NUMERO ALEAT. EN CADA
C MUESTRA.
C FINALMENTE, REPORT LLAMA A TST01 PARA REALIZAR EL TEST CHI-CUADRADO
C O K-S
C CON LAS ESTADISTICAS DE LOS RESULTADOS INDIVIDUALES OBTENIDOS.
C DESCRIPCION DE PARAMETROS:
C ISUB...INDICA QUE TIPO DE GENERADOR DEBE USARSE
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS:
C   TST01, TST02, TST04, TST05, TST06, TST07, TST08, TST09, TST10,
C   TST11, RRN01, RRN02, RRN03, RRN04, RRN05, RRN06, RRN07, RRN08,
C   RRN09, RRN10, RRN11, RRN12, RRN13, RRN14, RRN15, RRN16, RRN17,
C   RRN18, RRN19, RRN20, RRN21, RRN22, RRN23, RRN24, RRN25, RRN26,
C   RRN27, RRN28, RRN29, RRN30, RRN31, RRN32, RRN33, RRN34, RRN35,
C   RRN36, RRN37, RRN38, RRN39, RRN40, RRNWN
C
C
C*****
C
C   SUBROUTINE REPORT(ISUB)
C
C   CHARACTER*11 TCODE, T2CODE, CSNM, KSNM
C   CHARACTER*8  NAME, MESSG, VARIAB, TSTNM
C   CHARACTER*8  T01NM(2,10), CKNM(10)
C   CHARACTER*3  TIMES, YES
C
C   REAL*8
C   NAME, MESSG, VARIAB, T01NM, C2, ZZ2, TSTNM, CKNM, *TCODE, *CSNM, *KSNM
C   REAL*8 C2, ZZ2
C   COMMON /MESS/MESSG(9), VARIAB(9)
C   COMMON /FOURTH/Y(10000)
C   COMMON /FIFTH/X(10000)
C   COMMON /SIXTH/XX(7,1000,3), YY(10000,2,3)
C   COMMON /URNPRM/NAME, NBATCH, TIMES, IPRINT, IOPT
C   COMMON /T01PRM/NCHI, NPRNT
C   COMMON /OWNPRM/INTIN(200), REALIN(200), IA, IR
C   COMMON /COUNTS/ICNT, ICNTKS, ICNTMT, NEWGEN
C   COMMON /TSTTIM/KTIME, CTIME(1000), TIMLOW, LOWSAM
C   COMMON /CODE/GCODE
C   COMMON /RPRMS/C, P, Q, ZZ

```

```

COMMON /RPMDBL/C2,ZZ2
COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKK,L,ML,MK,MMM
COMMON /TSTPRM/IPARM(5)
COMMON /GENARR/ MFSR(98),NN(128),ITABLE(128)
COMMON /T01VAL/INF01(10,6),NDEGF(3)
  DIMENSION CSCODE(8),KSCODE(8)
C   DIMENSION CSCODE(8),KSCODE(8),T01NM(2,10),CKNM(10)
  DATA T01NM/'UDISTB','TST02',
*           'CUPON ','TST04',
*           'GAP   ','TST05',
*           'PERMUT','TST07',
*           'POKER ','TST08',
*           'CORRID','TST09',
*           'PARSER','TST10',
*           '*****','*****',
*           ' K-S ','TST06',
*           ' MAX T','TST11'/
  DATA CKNM/'01T02','01T04','01T05','01T07','01T08','01T09',
*           '01T10','*****','01T06','01T11'/

C
  DATA CSNM/'      CSCS'/
  DATA KSNM/'      KSKS'/
  DATA YES/'YES'/

C
  TIMES='NO'
  NCARDS=0
  ICNT=1
  ICNTKS=0
  ICNMT=0

C
  OPEN(UNIT=21,FILE='ARCH1',ACCESS='SEQUENTIAL',FORM='FORMATTED')
  READ(21,4000)NCHI
  READ(21,4000)NPRNT
  READ(21,5010)T2CODE
4000  FORMAT(I11)
5010  FORMAT(A11)
  TCODE=T2CODE
  DO 3600 I=1,8
    READ(21,5020)CSCODE(I)
3600  END DO
5020  FORMAT(F1.0)
  DO 3700 I = 1,8
    READ(21,5021)KSCODE(I)
5021  FORMAT(I1)
3700  END DO
  CLOSE(21)
  WRITE(*,*)' '

C
  IF(NCHI.GT.1000) NCHI=1000
  IF(NPRNT.LE.0) NPRNT=1
  IF(NPRNT.LE.NCHI) GOTO 7
  WRITE(6,101) NPRNT,NCHI
  101 FORMAT(1X,'LA CANTIDAD DE REPORTE DE TESTS INDIVIDUALES
LLAMADOS
*EN TST01, ',I6,/1X,'EXCEDE LA CANTIDAD DE VECES QUE CADA TEST
SERA
* EJECUTADO, ',I6)
  NPRNT=MIN(10,NCHI)
  WRITE(6,104) NPRNT
  104 FORMAT(' ', 'EL VALOR DE NIMPR HA SIDO CAMBIADO A ',I3)
C   REPETIR ENTRADA VALORES

```

```

    7 WRITE(*,*)' '
      WRITE(6,214) NCHI
214 FORMAT(' ','LA CANTIDAD DE MUESTRAS DE NROS. ALEAT. A SER
GENERADO
      *S',/1X,'PARA SU VERIFICACION ES:',I7)
      WRITE(*,*)' '
      WRITE(6,105)
105 FORMAT(' ','LA CANTIDAD DE ESTOS, DESDE LOS CUALES SE REPORTARAN
L
      *OS RESULTADOS ')
      WRITE(6,888)NPRNT
888 FORMAT(' ','DE LOS TESTS INDIVIDUALES ES ',I7)
C
      IF(TCODE.EQ.KSNM) GOTO 9
      WRITE(*,*)' '
      WRITE(6,216)
216 FORMAT(' ','TESTS QUE SUMINISTRAN VALORES CHI-CUADRADO A
TST01:')
C
      DO 8 I=1,8
      IF(CSCODE(I).EQ.0) GOTO 8
      NCARDS=NCARDS+1
      WRITE(6,218) T01NM(1,I)
218 FORMAT(5X,A8)
      8 CONTINUE
C
      IF(TCODE.EQ.CSNM) GOTO 13
      9 WRITE(6,220)
220 FORMAT(' ','TESTS QUE SUMINISTRAN VALORES KOLMOGOROV-SMIRNOV A
TST
      *01:')
C
      DO 12 I=1,2
      IF(KSCODE(I).EQ.0) GOTO 12
      NCARDS=NCARDS+1
      J=I+8
      WRITE(6,222) T01NM(1,J)
222 FORMAT(5X,A8)
      12 CONTINUE
C
      13 IF(NCARDS.NE.0) GOTO 14
      WRITE(6,224)
224 FORMAT(' ',' ERROR EN EL VALOR CSCODE O KSCODE DE TST01. TESTS
NO
      *RECONOCIBLES. TST01 SE IGNORA.')
      RETURN
C
      INICIALIZACION PARA INGRESAR INFORMACION A TST01
C
14 DO 16 I=1,10
      DO 16 J=1,6
      16 INF01(I,J)=0
C
      DO 20 I=1,NCARDS
      READ(20,226) TSTNM,(IPARM(J),J=1,5)
226 FORMAT(A8,5I11)
C
      DO 18 J=1,10
      IF(TSTNM.NE.CKNM(J)) GOTO 18
      INF01(J,6)=1
      DO 17 K=1,5

```



```

17   INF01(J,K)=IPARM(K)
      GOTO 20
18   CONTINUE
20   CONTINUE
C
C       APLICACION DE LOS TESTS ESPECIFICADOS PARA GENERAR DATOS
PARA
C       TST01. SE ASUME QUE HA SIDO GENERADA LA PRIMERA MUESTRA DE
C       NUMEROS ALEATORIOS.
C
22   DO 48 I=1,10
      IF(INF01(I,6).EQ.0) GOTO 48
      DO 24 J=1,5
24   IPARM(J)=INF01(I,J)
      GOTO(26,28,30,32,34,36,38,40,42,44),I
26   CALL TST02
      GOTO 48
28   CALL TST04
      GOTO 48
30   CALL TST05
      GOTO 48
32   CALL TST07
      GOTO 48
34   CALL TST08
      GOTO 48
36   CALL TST09
      GOTO 48
38   CALL TST10
      GOTO 48
40   GOTO 48
42   IF(ICNTKS.GT.10000) GOTO 48
      CALL TST06
      DO 43 J=1,10000
43   Y(J)=X(J)
      GOTO 48
44   IF(ICNTMT.GT.10000) GOTO 48
      CALL TST11
48   CONTINUE
C
C       VERIFICAR SI TODOS LOS TESTS HAN SIDO RECHAZADOS A CAUSA DE
C       PARAMETROS INVALIDOS
C
      IF(ICNT.GT.1) GOTO 49
      DO 47 I=1,10
      IF(INF01(I,6).NE.0) GOTO 49
47   CONTINUE
      WRITE(6,99)
99   FORMAT(' ',' LOS TESTS ESPECIFICADOS EN TST01 TIENEN PARAMETROS
IN     *VALIDOS.',/, ' TST01 FINALIZA. LA EJECUCION CONTINUA.')
      GOTO 999
C
C       INCREMENTO DEL CONTADOR DE MUESTRAS Y
C       GENERAR UNA NUEVA MUESTRA SI ES NECESARIO
C
49   ICNT=ICNT+1
      IF(ICNT.GT.NCHI) GOTO 79
C
C       LLAMADA DEL GENERADOR APROPIADO
C
      GOTO(50,52,54,56,58,60,62,64,66,68,70,72,74,260,261,262,263,264,

```

```

* 265,266,2021,2022,2023,2024,2025,2026,2027,2028,2029,2030,
*2031,2032,2033,2034,2035,2036,2037,2038,2039,2040,2041), ISUB
50 CALL RRN01
   GOTO 78
52 CALL RRN02(NN,ML,MM,MK,L,MMM,KKK)
   GOTO 78
54 CALL RRN03
   GOTO 78
56 CONTINUE
c   CALL RRN04(ITABLE)
   GOTO 78
58 CALL RRN05(MFSR,IP,IQ,INTSIZ,JJ)
   GOTO 78
60 CALL RRN06
   GOTO 78
62 CONTINUE
c   CALL RRN07
   GOTO 78
64 CALL RRN08
   GOTO 78
66 CALL RRN09
   GOTO 78
68 CALL RRN10
   GOTO 78
70 CALL RRN11
   GOTO 78

C
C   LOS CALCULOS EN DOBLE PRECISION VERSION DE URN11 (Y DE
C   URN12) NECESITAN PARAMETROS EN DOBLE PRECISION
C   QUE POR LO TANTO SON PASADOS.
C

72 CALL RRN12(IX,L,C2,ZZ2)
   GOTO 78
74 CALL RRN13(X,NBATCH,TIMES)
   GOTO 78
260 CALL RRN14
   GOTO 78
261 CONTINUE
C CALL RRN15
   GOTO 78
262 CONTINUE
C CALL RRN16
   GOTO 78
263 CONTINUE
C CALL RRN17
   GOTO 78
264 CONTINUE
C CALL RRN18
   GOTO 78
265 CONTINUE
C CALL RRN19
   GOTO 78
266 CALL RRN20(X,NBATCH,TIMES)
   GOTO 78
2021 CALL RRN21(NBATCH)
   GO TO 78
2022 CALL RRN22(NBATCH)
   GO TO 78
2023 CONTINUE
2024 CONTINUE
2025 CONTINUE

```

```

2026 CONTINUE
2027 CONTINUE
2028 CALL RRN28(X,NBATCH,TIMES)
      GO TO 78
2029 CONTINUE
2030 CALL RRN30(X,NBATCH,TIMES)
      GO TO 78
2031 CALL RRN31(X,NBATCH,TIMES)
      GO TO 78
2032 CALL RRN32(X,NBATCH,TIMES)
      GO TO 78
2033 CALL RRN33(X,NBATCH,TIMES)
      GO TO 78
2034 CALL RRN34(X,NBATCH,TIMES)
      GO TO 78
2035 CALL RRN35(X,NBATCH,TIMES)
      GO TO 78
2036 CALL RRN36(X,NBATCH,TIMES)
      GO TO 78
2037 CALL RRN37(X,NBATCH,TIMES)
      GO TO 78
2038 CALL RRN38(X,NBATCH,TIMES)
      GO TO 78
2039 CALL RRN39(X,NBATCH,TIMES)
      GO TO 78
2040 CONTINUE
      CALL RRN40(X,NBATCH,TIMES)
      GO TO 78
2041 CALL RRNWN
C
C   REGISTROS DE TIEMPOS SI FUE REQUERIDO
C
      78 IF(TIMES.NE.YES) GOTO 21
          CTIME(ICNT)=KTIME/FLOAT(NBATCH)
          IF(CTIME(ICNT).LT.0) CTIME(ICNT)=99999
          IF(CTIME(ICNT).LT.TIMLOW) LOWSAM=ICNT
          IF(CTIME(ICNT).LT.TIMLOW) TIMLOW=CTIME(ICNT)
C
C           ALMACENAR NUMEROS ALEATORIOS PARA TEST
C
21   DO 23 J1=1,NBATCH
      Y(J1)=X(J1)
23   CONTINUE
C
      GOTO 22
C
C           FIN DE GENERACION DE NUMEROS Y BUCLE DE VERIFICACION
C           REPORTE DE TIEMPOS Y ESTADISTICAS DE LOS TESTS INDIVIDUALES
C
C           ENCABEZAMIENTOS PARA LOS TIEMPOS
C
      79 IF(TIMES.NE.YES) GOTO 85
          WRITE(6,230)
230  FORMAT(' ',7X,'TIEMPOS')
          WRITE(6,234) NBATCH
234  FORMAT(' ',1X,'TIEMPO = TIEMPO EN MICROSEGUNDOS POR CADA UNO DE
LO   *S',I7,' NUMEROS ALEATORIOS MAS ')
          WRITE(6,233)
233  FORMAT(' ',1X,'UN TIEMPO DE INTERRUPCION VARIABLE NO
CONTROLADO)')

```

```

        WRITE(*,*)' '
        WRITE(6,236)
236  FORMAT('
', 'MUEST.', 4X, 'TIEMPO', 4(4X, '|', 2X, 'MUEST.', 4X, 'TIEMPO'))
C
        NPAGES=NCHI/250
        NREM=MOD(NCHI,250)
        IF(NREM.NE.0) NPAGES=NPAGES+1
        ISMALL=-249
C
        DO 81 K2=1, NPAGES
        IF(K2.NE.1) WRITE(6,237)
237  FORMAT('
', 'MUEST.', 4X, 'TIEMPO', 4(4X, '|', 2X, 'MUEST.', 4X, 'TIEMPO'))
C
        ISMALL=ISMALL+250
        ILARGE=ISMALL+49
C
        DO 81 I1=ISMALL, ILARGE
        I2=I1+50
        I3=I1+100
        I4=I1+150
        I5=I1+200
        IF((K2.EQ.NPAGES).AND.(NREM.NE.0)) GOTO 80
        WRITE(6,238)
I1,CTIME(I1), I2,CTIME(I2), I3,CTIME(I3), I4,CTIME(I4),
    *I5,CTIME(I5)
238  FORMAT(' ', 1X, I4, 4X, F6.2, 4(4X, '|', 3X, I4, 4X, F6.2))
        GOTO 81
C
C
C          FORMATO DE LA ULTIMA PAGINA
C
        80 IF(I1.LE.NCHI) WRITE(6,240) I1,CTIME(I1)
240  FORMAT(' ', 1X, I4, 6X, F6.2, 6X, '|')
        IF(I2.LE.NCHI) WRITE(6,242) I2,CTIME(I2)
242  FORMAT(' ', 7X, I4, 6X, F6.2, 6X, '|')
        IF(I3.LE.NCHI) WRITE(6,244) I3,CTIME(I3)
244  FORMAT(' ', 3X, I4, 6X, F6.2, 6X, '|')
        IF(I4.LE.NCHI) WRITE(6,246) I4,CTIME(I4)
246  FORMAT(' ', 2X, I4, 6X, F6.2, 6X, '|')
        IF(I5.LE.NCHI) WRITE(6,248) I5,CTIME(I5)
248  FORMAT(' ', 5X, I4, 6X, F6.2)
C
        81 CONTINUE
C
        WRITE(6,250) TIMLOW, LOWSAM
250  FORMAT(2X, 'EL MENOR DE LOS TIEMPOS ES', F7.1, ' MICROSEGUNDOS, Y
OCU
*RRIO EN LA MUESTRA ', I7)
C
C          SI SE TRATA SOLO DEL TEST KS-KS, BIFURCAR A ESE REPORTE
C
        IF(TCODE.EQ.KSNM) GOTO 700
C
C          REPORTE DE VALORES CHI-CUADRADOS DE CADA MUESTRA
C          PARA CADA SUBTEST
C
        85 DO 180 I=1,7
        IF(INF01(I,6).EQ.0) GOTO 180
        GOTO(82,84,86,88,90,92,94), I
C

```

```

C          REPORTE DE VALORES DE TST02
C
      82
WRITE(*,*)'*****'
      *****'
      7777
FORMAT('*****'
      *****')
      WRITE(6,300)
      300 FORMAT(1X,'VALORES CHI-2 PARA CADA MUESTRA DE NUMEROS
ALEATORIOS')
      WRITE(6,301)NCHI,NBATCH
      301 FORMAT(1X,'DEL TEST DISTRIB.UNIF.(TST02) USANDO ',I6,' MUESTRAS
DE
      * ',I6,' NUMEROS')
      WRITE(*,*)' ALEATORIOS CADA UNA'
      WRITE(*,*)' '
C      WRITE(*,*)'ESTOS SON LOS VALORES DE LAS PRIMEROS 200 MUESTRAS'
C      WRITE(*,*)' '
      WRITE(6,302)
      302 FORMAT(' ','MUESTRA',2X,' CHI-2 ',4('|','MUESTRA',2X,' CHI-2
,
      *))
C
C      NPAGES=1
C      NPAGES=NCHI/250
C      NREM=MOD(NCHI,250)
C      NREM=MOD(200,200)
C      IF(NREM.NE.0) NPAGES=NPAGES+1
C      ISMALL=1
C
C      DO 103 J=1,NPAGES
C      IF(J.EQ.1) GOTO 98
C      ENCABEZAMIENTO PARA PAGINAS DE SALIDA DISTINTAS A LA PRIMEA
      WRITE(*,*)' '
      WRITE(6,304)
      304 FORMAT(' ','MUESTRA',2X,' CHI-2 ',4('|','MUESTRA',2X,' CHI-2
,
      *))
C
C      FORMATO PAGINA REGULAR COMPLETA
C
      98 ILARGE=ISMALL+49
C
C      DO 100 I1=ISMALL,ILARGE
C      I2=I1+50
C      I3=I1+100
C      I4=I1+150
C      I5=I1+200
C
C      VERIFICACION QUE UNA PAGINA NO ESTA COMPLETA
C
C      IF((J.EQ.NPAGES).AND.(NREM.NE.0)) GOTO 102
C
C      WRITE(6,306) I1,XX(1,I1,1),I2,XX(1,I2,1),I3,XX(1,I3,1),I4,
* XX(1,I4,1),I5,XX(1,I5,1)
      306 FORMAT(' ',1X,I4,4X,F6.2,4(3X,'|',1X,I4,4X,F6.2))
C
C      GOTO 100
C
C      FORMATO DE LA ULTIMA PAGINA CUANDO NO ES COMPLETA

```

```

C
102 IF(I1.LE.NCHI) WRITE(6,308) I1,XX(1,I1,1)
308 FORMAT(' ',1X,I4,4X,F6.2,3X,'|')
      IF(I2.LE.NCHI) WRITE(6,310) I2,XX(1,I2,1)
310 FORMAT(' ',18X,I4,4X,F6.2,3X,'|')
      IF(I3.LE.NCHI) WRITE(6,312) I3,XX(1,I3,1)
312 FORMAT(' ',36X,I4,4X,F6.2,3X,'|')
      IF(I4.LE.NCHI) WRITE(6,314) I4,XX(1,I4,1)
314 FORMAT(' ',54X,I4,4X,F6.2,3X,'|')
      IF(I5.LE.NCHI)WRITE(6,316)I5,XX(1,I5,1)
316 FORMAT(' ',101X,I4,6X,F6.2)

C
100 CONTINUE
      ISMALL=ISMALL+250
103 CONTINUE
      GOTO 180

C
C          FIN DEL REPORTE DE TST02
C*****
C          VALORES DEL REPORTE TST04
C*****

      84
WRITE(*,*)'*****'
      *****'
      WRITE(6,350)
350 FORMAT(' ',1X,'VALORES CHI-2 PARA CADA MUESTRA DE NUMEROS
ALEATORI
      *OS')
      WRITE(6,351)NCHI,NBATCH
351 FORMAT(' ',1X,'DEL TEST DE CUPONES (TST04) USANDO ',I6,'
MUESTRAS
      *DE ',I6,' NUMEROS')
      WRITE(*,*)' ALEATORIOS CADA UNA '
      WRITE(*,*)' '
      WRITE(6,352)
352 FORMAT(' ', 'MUESTRA',5X,'CHI-2',9X,'CHI-2',9X,'CHI-2',5X,'|','
MUE
      *STRA',7X,'CHI-2',9X,'CHI-2',9X,'CHI-2 ' )
      WRITE(6,354)
354 FORMAT(10X,'ENTEROS 1-5',' ENTEROS 1-5',' ENTEROS 1-
10',2X,'|',
      *11X,' ENTEROS 1-5',' ENTEROS 1-5',' ENTEROS 1-10')
      WRITE(6,356)
356 FORMAT(24X,'FRAC(100*X)',16X,'|',26X,'FRAC(100*X)')
      WRITE(6,358)
358 FORMAT(51X,'|')

C
      NPAGES=NCHI/100
      NREM=MOD(NCHI,100)
      IF(NREM.NE.0) NPAGES=NPAGES+1
      ISMALL=1

C
DO 122 J=1,NPAGES
      IF(J.EQ.1) GOTO 114

C
C          ENCABEZAMIENTO PARA PAGINA DE SALIDAS DISTINTA A LA PRIMERA
C
      WRITE(*,*)' '
      WRITE(6,360)

```

```

360 FORMAT(1X,'MUESTRA',5X,'CHI-2',9X,'CHI-2',9X,'CHI-
2',5X,'|',1X,'MU
      *ESTRA',7X,'CHI-2',9X,'CHI-2',9X,'CHI-2')
      WRITE(6,362)
362 FORMAT(10X,'ENTEROS 1-5','  ENTEROS 1-5','  ENTEROS 1-
10',2X,'|',
      *11X,' ENTEROS 1-5','  ENTEROS 1-5','  ENTEROS 1-10')
      WRITE(6,364)
364 FORMAT(24X,'FRAC(100*X)',16X,'|',26X,'FRAC(100*X)')

      WRITE(6,366)
366 FORMAT(51X,'|')
C
114 ILARGE=ISMALL+49
      DO 120 I1=ISMALL,ILARGE
          I2=I1+50
          IF((J.EQ.NPAGES).AND.(NREM.NE.0)) GOTO 118
          WRITE(6,368) I1,XX(2,I1,1),XX(2,I1,2),XX(2,I1,3),I2,XX(2,I2,1),
          *XX(2,I2,2),XX(2,I2,3)
368 FORMAT(' ',1X,I4,6X,2(F6.2,8X),F6.2,5X,'|',3X,I4,7X,3(F6.2,8X))
          GOTO 120
C
C          FORMATO PARA UNA PAGINA INCOMPLETA
C
118 IF(I1.LE.NCHI) WRITE(6,370) I1,XX(2,I1,1),XX(2,I1,2),XX(2,I1,3)
370 FORMAT(' ',1X,I4,6X,2(F6.2,8X),F6.2,5X,'|')
          IF(I2.LE.NCHI) WRITE(6,372) I2,XX(2,I2,1),XX(2,I2,2),XX(2,I2,3)
372 FORMAT(1X,57X,I4,7X,3(F6.2,8X))
C
120 CONTINUE
C
      ISMALL=ISMALL+100
C
122 CONTINUE
C
      GOTO 180
C
C*****
C          REPORTE DE VALORES DE TST05
C*****
86 A=INF01(3,1)/1000.
      B=INF01(3,2)/1000.
C
WRITE(*,*)'*****
*****'
      WRITE(6,400) NCHI,NBATCH
400 FORMAT(1X,' VALORES CHI-2 PARA CADA MUESTRA DE NUMEROS
ALEATORIOS
      *DEL TEST GAP (TST05)',/1X,' USANDO',I6,' MUESTRAS DE ',I6,'
NUMERO
      *S ALEATORIOS CADA UNA.',//)
      WRITE(*,*)' '
      WRITE(6,402)
402 FORMAT(1X,'MUESTRA',6X,'CHI-2',10X,'CHI-2',10X,'CHI-
2',5X,'|',1X,
      *'MUESTRA',7X,'CHI-2',10X,'CHI-2',10X,'CHI-2')
      WRITE(6,404)
404 FORMAT(6X,3(6X,'INTERVALO'),3X,'|',7X,3(6X,'INTERVALO'))
      WRITE(6,406) A,B,A,B
406 FORMAT(13X,'(0,0.5)',8X,'(0.5,1)',6X,'(,F4.3,',',F4.3,')',2X,

```

```

        *'|',14X,'(0,0.5)',8X,'(0.5,1)',6X,'(,F4.3,',',F4.3,')')
        WRITE(6,408)
408 FORMAT(' ',53X,'|')
C
        NPAGES=NCHI/100
        NREM=MOD(NCHI,100)
        IF(NREM.NE.0) NPAGES=NPAGES+1
        ISMALL=1
C
        DO 132 J=1, NPAGES
        IF(J.EQ.1) GOTO 126
C
        ENCABEZAMIENTO PARA PAGINA DE SALIDAS DISTINTA A LA PRIMERA
C
        WRITE(*,*)' '
        WRITE(6,410)
410 FORMAT(1X,'MUESTRA',6X,'CHI-2',10X,'CHI-2',10X,'CHI-
2',5X,'|',1X,
        *'MUESTRA',7X,'CHI-2',10X,'CHI-2',10X,'CHI-2')
        WRITE(6,412)
412 FORMAT(6X,3(6X,'INTERVALO'),3X,'|',7X,3(6X,'INTERVALO'))
        WRITE(6,414) A,B,A,B
414 FORMAT(13X,'(0,0.5)',8X,'(0.5,1)',6X,'(,F4.3,',',F4.3,')',2X,
        *'|',14X,'(0,0.5)',8X,'(0.5,1)',6X,'(,F4.3,',',F4.3,')')
        WRITE(6,418)
418 FORMAT(' ',53X,'|')
C
126 ILARGE=ISMALL+49
        DO 130 I1=ISMALL,ILARGE
        I2=I1+50
        IF((J.EQ.NPAGES).AND.(NREM.NE.0)) GOTO 128
        WRITE(6,420) I1,XX(3,I1,1),XX(3,I1,2),XX(3,I1,3),I2,XX(3,I2,1),
        *XX(3,I2,2),XX(3,I2,3)
420 FORMAT(' ',1X,I4,7X,2(F6.2,9X),F6.2,5X,'|',3X,I4,7X,3(F6.2,9X))
        GOTO 130
C
        FORMATO DE UNA PAGINA INCOMPLETA
128 IF(I1.LE.NCHI) WRITE(6,422) I1,XX(3,I1,1),XX(3,I1,2),XX(3,I1,3)
422 FORMAT(' ',1X,I4,7X,2(F6.2,9X),F6.2,5X,'|')
        IF(I2.LE.NCHI) WRITE(6,424) I2,XX(3,I2,1),XX(3,I2,2),XX(3,I2,3)
424 FORMAT('+',57X,I4,7X,3(F6.2,9X))
C
130 CONTINUE
C
        ISMALL=ISMALL+100
C
132 CONTINUE
C
        GOTO 180
C*****
C
        REPORTE PARA VALORES DE TST07
C*****
88
WRITE(*,*)'*****'
        *****
        WRITE(6,450) NCHI,NBATCH
450 FORMAT(1X,' VALORES CHI-2 PARA CADA MUESTRA DE NUMEROS
ALEATORIOS
        *DEL TEST DE ',/1X,' PERMUTACION (TST07) UTILIZANDO ',I6,'
MUESTRAS
        * DE ',I6,' NUMEROS',/1X,' ALEATORIOS CADA UNA.',/)
        WRITE(*,*)' '

```



```

WRITE(6,452)
452 FORMAT(1X,'MUESTRA',3X,' CHI-2 ',2(5X,' CHI-2 '),3X,'|',
*2X,'MUESTRA',4X,' CHI-2 ',2(5X,' CHI-2 '))
WRITE(6,454)
454 FORMAT(' ',11X,'3 TUPLAS',7X,'4 TUPLAS',7X,'5 TUPLAS',4X,'|',
* ' 13X,'3 TUPLAS',7X,'4 TUPLAS',7X,'5 TUPLAS')
WRITE(6,456)
456 FORMAT(' ',53X,'|')
C
NPAGES=NCHI/100
NREM=MOD(NCHI,100)
IF(NREM.NE.0) NPAGES=NPAGES+1
ISMALL=1
C
DO 146 J=1, NPAGES
IF(J.EQ.1) GOTO 140
C
ENCABEZAMIENTO PARA PAGINA DE SALIDAS DISTINTA A LA PRIMERA
C
WRITE(*,*)' '
WRITE(6,458)
458 FORMAT(1X,'MUESTRA',3X,' CHI-2 ',2(5X,' CHI-2 '),3X,'|',
*2X,'MUESTRA',4X,' CHI-2 ',2(5X,' CHI-2 '))
WRITE(6,460)
460 FORMAT(' ',11X,'3 TUPLAS',7X,'4 TUPLAS',7X,'5 TUPLAS',4X,'|',
* ' 13X,'3 TUPLAS',7X,'4 TUPLAS',7X,'5 TUPLAS')
WRITE(6,462)
462 FORMAT(' ',53X,'|')
C
140 ILARGE=ISMALL+49
DO 144 I1=ISMALL, ILARGE
I2=I1+50
IF((J.EQ.NPAGES).AND.(NREM.NE.0)) GOTO 142
WRITE(6,464) I1,XX(4,I1,1),XX(4,I1,2),XX(4,I1,3),I2,XX(4,I2,1),
* XX(4,I2,2),XX(4,I2,3)
464 FORMAT(' ',1X,I4,7X,2(F6.2,9X),F6.2,5X,'|',3X,I4,7X,3(F6.2,9X))
GOTO 144
C
FORMATO PARA PAGINAS INCOMPLETAS
C
142 IF(I1.LE.NCHI) WRITE(6,466) I1,XX(4,I1,1),XX(4,I1,2),XX(4,I1,3)
466 FORMAT(' ',1X,I4,7X,2(F6.2,9X),F6.2,5X,'|')
IF(I2.LE.NCHI) WRITE(6,468) I2,XX(4,I2,1),XX(4,I2,2),XX(4,I2,3)
468 FORMAT('+',57X,I4,7X,3(F6.2,9X))
C
144 CONTINUE
C
ISMALL=ISMALL+100
C
146 CONTINUE
C
GOTO 180
C*****
C REPORTE PARA VALORES DE TST08
C*****
90
WRITE(*,*)'*****'
*****
WRITE(6,500) NCHI,NBATCH
500 FORMAT(1X,' VALORES CHI-2 PARA CADA MUESTRA DE NUMEROS
ALEATORIOS

```

```

*DEL TEST DEL',/1X,' POKER (TST08) UTILIZANDO ',I6,' MUESTRAS DE
',
*I6,/1X,' NUMEROS ALEATORIOS CADA UNA.',//)
WRITE(*,*)' '
WRITE(6,502)
502 FORMAT(1X,'MUESTRA',4X,' CHI-2 ',2(5X,' CHI-2 '),4X,'|',
*2X,'MUESTRA',4X,' CHI-2 ',2(5X,' CHI-2 '))
WRITE(6,504)
504 FORMAT(' ',10X,'MANOS DE 4',2(5X,'MANOS DE 5'),3X,'|',
*
12X,'MANOS DE 4',2(5X,'MANOS DE 5'))
WRITE(6,506)
506 FORMAT(' ',10X,'4 REPARTIC. ',3X,'6 REPARTIC. ',3X,'4 REPARTIC.
',
*1X,'|',12X,'4 REPARTIC. ',3X,'6 REPARTIC. ',3X,'4 REPARTIC. ')
WRITE(6,508)
508 FORMAT(' ',53X,'|')
C
NPAGES=NCHI/100
NREM=MOD(NCHI,100)
IF(NREM.NE.0) NPAGES=NPAGES+1
ISMAILL=1
C
DO 156 J=1,NPAGES
IF(J.EQ.1) GOTO 150
C
C ENCABEZAMIENTO PARA PAGINA DE SALIDAS DISTINTA A LA PRIMERA
C
WRITE(*,*)' '
WRITE(6,510)
510 FORMAT(1X,'MUESTRA',4X,' CHI-2 ',2(5X,' CHI-2 '),4X,'|',
*2X,'MUESTRA',4X,' CHI-2 ',2(5X,' CHI-2 '))
WRITE(6,512)
512 FORMAT(' ',10X,'MANOS DE 4',2(5X,'MANOS DE 5'),3X,'|',
*12X,'MANOS DE 4',2(5X,'MANOS DE 5'))
WRITE(6,514)
514 FORMAT(' ',10X,'4 REPARTIC. ',3X,'6 REPARTIC. ',3X,'4 REPARTIC.
',
*1X,'|',12X,'4 REPARTIC. ',3X,'6 REPARTIC. ',3X,'4 REPARTIC. ')
WRITE(6,516)
516 FORMAT(' ',53X,'|')
C
150 ILARGE=ISMAILL+49
DO 154 I1=ISMAILL,ILARGE
I2=I1+50
IF((J.EQ.NPAGES).AND.(NREM.NE.0)) GOTO 152
WRITE(6,518) I1,XX(5,I1,1),XX(5,I1,2),XX(5,I1,3),I2,XX(5,I2,1),
* XX(5,I2,2),XX(5,I2,3)
518 FORMAT(' ',1X,I4,7X,2(F6.2,9X),F6.2,5X,'|',3X,I4,7X,3(F6.2,9X))
GOTO 154
C
FORMAT FOR UNFILLED PAGE
152 IF(I1.LE.NCHI) WRITE(6,520) I1,XX(5,I1,1),XX(5,I1,2),XX(5,I1,3)
520 FORMAT(' ',1X,I4,7X,2(F6.2,9X),F6.2,5X,'|')
IF(I2.LE.NCHI) WRITE(6,522) I2,XX(5,I2,1),XX(5,I2,2),XX(5,I2,3)
522 FORMAT('+',57X,I4,7X,3(F6.2,9X))
C
154 CONTINUE
C
ISMAILL=ISMAILL+100
C
156 CONTINUE
C

```

```

          GOTO 180
C*****
C          REPORTE PARA VALORES DE TST09
C*****
          92
WRITE(*,*)'*****
*****'
          WRITE(6,550) NCHI,NBATCH
          550 FORMAT(1X,' VALORES CHI-2 PARA CADA MUESTRA DE NUMEROS
ALEATORIOS D
          *EL TEST RUNS UP',/1X,' (TST09) UTILIZANDO ',I6,' MUESTRAS
DE',I6,
          *' NUMEROS ALEATORIOS CADA UNA.',//)
          WRITE(*,*)' '
          WRITE(6,552)
          552 FORMAT(1X,'MUESTRA',6X,'CHI-2',10X,'CHI-2',10X,'CHI-2',5X,'|',
          *2X,'MUESTRA',6X,'CHI-2',10X,'CHI-2',10X,'CHI-2')
          WRITE(6,554)
          554 FORMAT(17X,'X',9X,'FRAC(10*X)',4X,'FRAC(100*X)',2X,'|',
          *18X,'X',9X,'FRAC(10*X)',4X,'FRAC(100*X)')
          WRITE(6,556)
          556 FORMAT(' ',53X,'|')
C
          NPAGES=NCHI/100
          NREM=MOD(NCHI,100)
          IF(NREM.NE.0) NPAGES=NPAGES+1
          ISMALL=1
C
          DO 166 J=1,NPAGES
          IF(J.EQ.1) GOTO 160
C
C          ENCABEZAMIENTO PARA PAGINA DE SALIDAS DISTINTA A LA PRIMERA
C
          WRITE(*,*)' '
          WRITE(6,558)
          558 FORMAT(1X,'MUESTRA',6X,'CHI-2',10X,'CHI-2',10X,'CHI-2',5X,'|',
          *2X,'MUESTRA',6X,'CHI-2',10X,'CHI-2',10X,'CHI-2')
          WRITE(6,560)
          560 FORMAT(17X,'X',9X,'FRAC(10*X)',4X,'FRAC(100*X)',2X,'|',
          *18X,'X',9X,'FRAC(10*X)',4X,'FRAC(100*X)')
          WRITE(6,562)
          562 FORMAT(' ',53X,'|')
C
          160 ILARGE=ISMALL+49
          DO 164 I1=ISMALL,ILARGE
          I2=I1+50
          IF((J.EQ.NPAGES).AND.(NREM.NE.0)) GOTO 162
          WRITE(6,564) I1,XX(6,I1,1),XX(6,I1,2),XX(6,I1,3),I2,XX(6,I2,1),
          * XX(6,I2,2),XX(6,I2,3)
          564 FORMAT(' ',1X,I4,5X,2(F8.2,7X),F8.2,5X,'|',3X,I4,5X,3(F8.2,7X))
          GOTO 164
C
C          FORMATO PARA PAGINAS INCOMPLETAS
C
          162 IF(I1.LE.NCHI) WRITE(6,566) I1,XX(6,I1,1),XX(6,I1,2),XX(6,I1,3)
          566 FORMAT(' ',1X,I4,7X,2(F6.2,9X),F6.2,5X,'|')
          IF(I2.LE.NCHI) WRITE(6,568) I2,XX(6,I2,1),XX(6,I2,2),XX(6,I2,3)
          568 FORMAT('+',57X,I4,7X,3(F6.2,9X))
C
          164 CONTINUE
C

```

```

        ISMALL=ISMALL+100
C
    166 CONTINUE
C
        GOTO 180
C*****
C        REPORTE PARA VALORES DE TST10
C*****
    94
WRITE(*,*)'*****'
        *****
        WRITE(6,600) NCHI,NBATCH
    600 FORMAT(1X,' VALORES CHI-2 PARA CADA MUESTRA DE NUMEROS
ALEATORIOS
        *DEL TEST DE PARES',/1X,' SERIALES (TST10) USANDO ',I6,' MUESTRAS
D
        *E ',I6,' NUM. ALEATORIOS CADA UNA.',/)
        WRITE(*,*)' '
        WRITE(6,602)
    602 FORMAT(1X,'MUESTRA',6X,'CHI-2',10X,'CHI-2',10X,'CHI-2',5X,'|',
        *2X,'MUESTRA',6X,'CHI-2',10X,'CHI-2',10X,'CHI-2')
        WRITE(6,604)
    604 FORMAT(11X,'ENTEROS 1-3',3X,'ENTEROS 1-10',3X,'ENTEROS 1-20',2X,
        *'|',12X,'ENTEROS 1-3',3X,'ENTEROS 1-10',3X,'ENTEROS 1-20')
        WRITE(6,606)
    606 FORMAT(' ',53X,'|')
C
        NPAGES=NCHI/100
        NREM=MOD(NCHI,100)
        IF(NREM.NE.0) NPAGES=NPAGES+1
        ISMALL=1
C
        DO 176 J=1,NPAGES
        IF(J.EQ.1) GOTO 170
C
        ENCABEZAMIENTO PARA PAGINA DE SALIDAS DISTINTA A LA PRIMERA
C
        WRITE(*,*)' '
        WRITE(6,608)
    608 FORMAT(1X,'MUESTRA',6X,'CHI-2',10X,'CHI-2',10X,'CHI-2',5X,'|',
        *2X,'MUESTRA',6X,'CHI-2',10X,'CHI-2',10X,'CHI-2')
        WRITE(6,610)
    610 FORMAT(11X,'ENTEROS 1-3',3X,'ENTEROS 1-10',3X,'ENTEROS 1-20',2X,
        *'|',12X,'ENTEROS 1-3',3X,'ENTEROS 1-10',3X,'ENTEROS 1-20')
        WRITE(6,612)
    612 FORMAT(' ',53X,'|')
C
    170 ILARGE=ISMALL+49
        DO 174 I1=ISMALL,ILARGE
        I2=I1+50
        IF((J.EQ.NPAGES).AND.(NREM.NE.0)) GOTO 172
        WRITE(6,614) I1,XX(7,I1,1),XX(7,I1,2),XX(7,I1,3),I2,XX(7,I2,1),
        * XX(7,I2,2),XX(7,I2,3)
    614 FORMAT(' ',1X,I4,7X,2(F6.2,9X),F6.2,5X,'|',3X,I4,7X,3(F6.2,9X))
        GOTO 174
C
C        FORMATO PARA PAGINAS INCOMPLETAS
C
    172 IF(I1.LE.NCHI) WRITE(6,616) I1,XX(7,I1,1),XX(7,I1,2),XX(7,I1,3)
    616 FORMAT(' ',1X,I4,7X,2(F6.2,9X),F6.2,5X,'|')
        IF(I2.LE.NCHI) WRITE(6,618) I2,XX(7,I2,1),XX(7,I2,2),XX(7,I2,3)

```

```

618 FORMAT( '+', 57X, I4, 7X, 3(F6.2, 9X) )
C
174 CONTINUE
C
      ISMALL=ISMALL+100
C
176 CONTINUE
C*****

C          FIN DEL REPORTE DE VALORES DE CHI-CUADRADO
C
180 CONTINUE
C
      COMPLETADO DEL REPORTE DE VALORES CHI-CUADRADO
C          SOLO SI ES CHI2-CHI2, BIFURCAR AL LLAMADO DE TST01
C
      IF(TCODE.EQ.CSNM) GOTO 900
C
C*****
C          REPORTE DE VALORES KS DE CADA MUESTRA PARA CADA SUBTEST
C
700 IF(INF01(9,6).EQ.0) GOTO 800
C
      NREP=INF01(9,2)

WRITE(*,*) '*****'
      *****
      WRITE(6,702) NCHI,NREP,NBATCH
702 FORMAT(3X,'VALORES KOLMOGOROV-SMIRNOV DE CADA MUESTRA DE NUMEROS
A
      *LEATORIOS',/3X,'DEL TEST DE BONDAD DE AJUSTE K-S (TST06)
USANDO',
      *I5,' MUESTRAS CON ',/3X,I4,' SUBMUESTRAS DENTRO DE CADA MUESTRA
DE
      * ',I5,' NUMEROS ALEATORIOS.')
      WRITE(*,*) ' '
      WRITE(6,704)
704 FORMAT(1X,'MUESTRA',4X,'SUBMUESTRA',5X,'K+',10X,'K-',10X,'K',8X,
      *'|',2X,'MUESTRA',4X,'SUBMUESTRA',6X,'K+',10X,'K-',10X,'K')
      WRITE(6,706)
706 FORMAT(' ',58X,'|')
C
      NPAGES=ICNTKS/100
      NREM=MOD(ICNTKS,100)
      IF(NREM.NE.0) NPAGES=NPAGES+1
      ISMALL=1
C
      DO 190 J=1,NPAGES
      IF(J.EQ.1) GOTO 194
C
      ENCABEZAMIENTO PARA PAGINA DE SALIDAS DISTINTA A LA PRIMERA
C
      WRITE(*,*) ' '
      WRITE(6,708)
708 FORMAT(1X,'MUESTRA',4X,'SUBMUESTRA',5X,'K+',10X,'K-',10X,'K',8X,
      *'|',2X,'MUESTRA',4X,'SUBMUESTRA',6X,'K+',10X,'K-',10X,'K')
      WRITE(6,710)
710 FORMAT(' ',58X,'|')
C
194 ILARGE=ISMALL+49
      DO 196 I1=ISMALL,ILARGE

```

```

      I2=I1+50
      NSUBS1=MOD(I1,NREP)
      NSUBS2=MOD(I2,NREP)
      NSAMP1=I1/NREP+1
      NSAMP2=I2/NREP+1
      IF(NSUBS1.NE.0) GOTO 195
      NSUBS1=NREP
      NSAMP1=NSAMP1-1
195  IF(NSUBS2.NE.0) GOTO 197
      NSUBS2=NREP
      NSAMP2=NSAMP2-1
197  IF((J.EQ.NPAGES).AND.(NREM.NE.0)) GOTO 198
      WRITE(6,712) NSAMP1,NSUBS1,YY(I1,1,1),YY(I1,1,2),YY(I1,1,3),
      *          NSAMP2,NSUBS2,YY(I2,1,1),YY(I2,1,2),YY(I2,1,3)
712  FORMAT(' ',1X,I4,8X,I3,6X,3(F8.3,4X),'|',3X,I4,8X,I3,6X,
      * 3(F8.3,4X))
      GOTO 196
C      FORMATO PARA LA ULTIMA PAGINA INCOMPLETA
198  IF(I1.LE.ICNTKS)WRITE(6,714)
NSAMP1,NSUBS1,YY(I1,1,1),YY(I1,1,2),
      *YY(I1,1,3)
714  FORMAT(' ',1X,I4,8X,I3,6X,3(F8.3,4X))
      IF(I2.LE.ICNTKS)WRITE(6,716)
NSAMP2,NSUBS2,YY(I2,1,1),YY(I2,1,2),
      *YY(I2,1,3)
716  FORMAT('+',53X,'|',3X,I4,8X,I3,6X,3(F8.3,4X))
C
196  CONTINUE
C
      ISMALL=ISMALL+100
C
190  CONTINUE
C
      FIN DEL REPORTE DE TST06 REPORT
C*****
C      REPORTE DE LOS VALORES KS DE CADA MUESTRA PARA CADA SUBTEST
C
800  IF(INF01(10,6).EQ.0) GOTO 999
C
      NREP=INF01(10,3)
      WRITE(*,*)' '
      WRITE(*,7777)
      WRITE(6,802) NCHI,NREP,NBATCH
802  FORMAT(3X,'VALORES KOLMOGOROV-SMIRNOV DE CADA MUESTRA DE NUMEROS
A      *LEATORIOS ',/3X,'DEL TEST K-S MAXIMO-DE-T (TST11) USANDO ',I5,
      *' MUESTRAS CON ',I4,/3X,'SUBMUESTRAS DENTRO DE CADA MUESTRA DE
      ',
      *I5,' NUMEROS ALEATORIOS.')
      WRITE(*,*)' '
      WRITE(6,804)
804  FORMAT(1X,'MUESTRA',3X,'SUBMUESTRA',6X,'K+',10X,'K-',10X,'K',6X,
      *'|',2X,'MUESTRA',4X,'SUBMUESTRA',6X,'K+',10X,'K-',10X,'K')
      WRITE(6,806)
806  FORMAT(' ',57X,'|')
C
      NPAGES=ICNTMT/100
      NREM=MOD(ICNTMT,100)
      IF(NREM.NE.0) NPAGES=NPAGES+1
      ISMALL=1
C

```

```

DO 201 J=1,NPAGES
IF(J.EQ.1) GOTO 203
C
C           ENCABEZAMIENTO PARA PAGINA DE SALIDAS DISTINTA A LA PRIMERA
C
WRITE(*,*)' '
WRITE(6,808)
808 FORMAT(1X,'MUESTRA',3X,'SUBMUESTRA',6X,'K+',10X,'K-',10X,'K',6X,
*'|',2X,'MUESTRA',4X,'SUBMUESTRA',6X,'K+',10X,'K-',10X,'K')
WRITE(6,810)
810 FORMAT(' ',57X,'|')
C
203 ILARGE=ISMAIL+49
DO 207 I1=ISMAIL,ILARGE
I2=I1+50
NSUBS1=MOD(I1,NREP)
NSUBS2=MOD(I2,NREP)
NSAMP1=I1/NREP+1
NSAMP2=I2/NREP+1
IF(NSUBS1.NE.0) GOTO 205
NSUBS1=NREP
NSAMP1=NSAMP1-1
205 IF(NSUBS2.NE.0) GOTO 208
NSUBS2=NREP
NSAMP2=NSAMP2-1
208 IF((J.EQ.NPAGES).AND.(NREM.NE.0)) GOTO 209
WRITE(6,812) NSAMP1,NSUBS1,YY(I1,2,1),YY(I1,2,2),YY(I1,2,3),
*
* NSAMP2,NSUBS2,YY(I2,2,1),YY(I2,2,2),YY(I2,2,3)
812 FORMAT(' ',1X,I4,8X,I3,6X,3(F8.3,4X),'|',3X,I4,8X,I3,6X,
* 3(F8.3,4X))
GOTO 207
C
C           FORMATO DE LA ULTIMA PAGINA INCOMPLETA
C
209 IF(I1.LE.ICNIMT)WRITE(6,814)
NSAMP1,NSUBS1,YY(I1,2,1),YY(I1,2,2),
*YY(I1,2,3)
814 FORMAT(' ',1X,I4,8X,I3,6X,3(F8.3,4X))
IF(I2.LE.ICNIMT)WRITE(6,816)
NSAMP2,NSUBS2,YY(I2,2,1),YY(I2,2,2),
*YY(I2,2,3)
816 FORMAT(' ',53X,'|',3X,I4,8X,I3,6X,3(F8.3,4X))
C
207 CONTINUE
C
ISMAIL=ISMAIL+100
C
201 CONTINUE
C
C           FIN DEL REPORTE DE TST11
C*****
C           LLAMADO A TST01 PARA REALIZAR
C           CHI2 SOBRE CHI2 , KS SOBRE KS
C*****
900 DO 199 K1=1,10
IF(INF01(K1,6).EQ.0) GOTO 199
C
CALL TST01(K1)
C *****FIN SUBROUTINA
REPORT*****
199 CONTINUE

```

```

999 RETURN
END

C
C*****
C SUBROUTINE DBLPRE (IDBL)
C*****
C ESTA SUBRUTINA CONSTRUYE NUMEROS DE DOBLE PRECISION A PARTIR DE
C CUATRO
C NUMEROS SUCESIVOS DE SIMPLE PRECISION. LOS PRIMEROS DOS BYTES DEL
C NUMERO
C DE DOBLE PRECISION (SIGNO,EXPONENTE,Y EL PRIMER DIGITO HEXADECIMAL
C DE LA
C MANTISA)SON LA MITAD IZQUIERDA DEL PRIMER NUMERO DE SIMPLE
C PRECISION.
C LOS DIGITOS HEXADECIMALES RESTANTES DEL NUMERO DE DOBLE PRECISION SE
C OBTIENEN
C CONVIRTIENDO LOS BITS DE MENOR ORDEN DE LA MANTISA DE LOS TRES
C NUMEROS SUBSIGUIENTES DE SIMPLE PRECISION, EN ENTEROS DE MEDIA
C PALABRA.
C LUEGO LOS AGREGA A LA PORCION INICIAL.
C*****
C
C SUBROUTINE DBLPRE (IDBL)
COMMON /SECOND/XDBL(2500)
COMMON /FOURTH/Y(10000)
REAL*8 XDBL,DBL
EQUIVALENCE (DWORD(1),A,DBL),(DWORD(2),I1),(DWORD(3),I2),
*(DWORD(4),I3)
INTEGER*2 I1,DWORD(4),I3,I2

C
K=-3

C
DO 10 I=1,IDBL
DBL=0
K=K+4
A=Y(K)
I1=Y(K+1)*65536
I2=Y(K+2)*65536
I3=Y(K+3)*65536
XDBL(I)=DBL
10 CONTINUE
RETURN
END

C
C*****
C SUBROUTINE EXTPRE (IEXT)
C*****
CESTA SUBRUTINA CONSTRUYE UN NUMERO ALEATORIO DE CUADRUPLE PRECISION A
C PARTIR DE OCHO NUMEROS ALEATORIOS SUCESIVOS DE SIMPLE PRECISION.
C LOS PRIMEROS DOS BYTES DEL NUMERO ALEAT. DE CUADRUPLE PRECISION
C (SIGNO,
C EXPONENTE Y PRIMER DIGITO HEXADECIMAL DE LA MANTISA) SON LA MITAD
C IZQUIERDA
C DEL PRIMER NUMERO DE SIMPLE PRECISION. LOS DIGITOS HEXADECIMALES
C RESTANTES
C DEL NUMERO DE PRECISION EXTENDIDA SE OBTIENEN CONVIRTIENDO LOS BITS
C DE
C MENOR ORDEN DE LA MANTISA, CORRESPONDIENTES A LOS SIETE NUMEROS DE
C SIMPLE
C PRECISION RESTANTES,A ENTEROS DE MEDIA PALABRA. LUEGO SE AGREGAN A
C LA

```



```

C PORCION INICIAL.
C*****
C
      SUBROUTINE EXTPRE(IEXT)
      COMMON /THIRD/XEXT(1250)
      COMMON /FOURTH/Y(10000)
      EQUIVALENCE (DWORD(1),A,EXPRE),(DWORD(2),I1),(DWORD(3),I2),
      *(DWORD(4),I3),(DWORD(5),I4),(DWORD(6),I5),(DWORD(7),I6),
      *(DWORD(8),I7)
      INTEGER*2 DWORD(8),I1,I2,I3,I4,I5,I6,I7
      REAL*8 EXPRE,XEXT
C
      K=-7
C
      DO 10 I=1,IEXT
      EXPRE=0
      K=K+8
      A=Y(K)
      I1=Y(K+1)*65536
      I2=Y(K+2)*65536
      I3=Y(K+3)*65536
      I4=Y(K+4)*65536
      I5=Y(K+5)*65536
      I6=Y(K+6)*65536
      I7=Y(K+7)*65536
      XEXT(I)=EXPRE
10  CONTINUE
      RETURN
      END
C
C*****
C      SUBROUTINE KSTPL(R,N,T)
C*****
C PROPOSITO
C GRAFICAR LA FUNCION DE DISTRIBUCION EMPIRICA DE N NUMEROS (CADA UNO
C ESTA ENTRE 0.0 AND 1.0, INCLUSIVE), R(1),...,R(N), JUNTO CON LA
C FUNCION DE DISTRIBUCION ESTIMADA X**T (0.0<=X<=1.0).
C
C USO
C "CALL KSTPL(R,N,T)"
C
C DESCRIPCION DE PARAMETROS
C R...NOMBRE DEL ARREGLO UNIDIMENSIONAL EN EL CUAL LA FUNCION
C DE DISTRIBUCION EMPIRICA SE BASA.
C N...CANTIDAD DE ELEMENTOS DEL ARREGLO R
C T...ES LA POTENCIA DE X A SER USADA EN LA FUNCION DE
DISTRIBUCION
C ESTIMADA
C FUNCION X**T. (T DEBE SER UN ENTERO POSITIVO)
C
C OBSERVACIONES
C SUBROUTINA NO HABILITADA EN ESTA VERSION
C SI R(J) ES EL MAXIMO DE LOS NUMEROS ALEATORIOS ESTIMADOS
C K PARA J=1,...,N,
C ENTONCES ES CONVENIENTE ELEGIR T=K.
C
C SUBROUTINES Y FUNCIONES REQUIRIDAS
C SORT
C
C METODO

```

```

C   VECTORES DE PESOS DE CADA CURVA EN CADA PUNTO X=0.01(0.01)1.00
C   SON
C   CALCULADOS Y USADOS PARA CONTROLAR LOS SIMBOLOS IMPRESOS EN CADA
C   PESO(PONDERAL) .
C*****
C
C
C*****
C   SUBROUTINE XTRMS(C,KF,KL,RMIN,RMAX)
C*****
C PROPOSITO:
C
C   ENCONTRAR EL MENOR Y EL MAYOR NUMERO DE UN ARREGLO DADO.
C
C USO:
C   "CALL XTRMS(C,KF,KL,RMIN,RMAX) "
C DESCRIPCION DE PARAMETROS:
C   C...NOMBRE DEL ARREGLO UNIDIMENSIONAL A UTILIZAR.
C   KF...PRIMER INDICE A INCLUIR EN LA BUSQUEDA.
C   KL...ULTIMO INDICE A INCLUIR EN LA BUSQUEDA.KF Y KL DEBEN SER
C   ENTEROS POSITIVOS CON KF<=KL
C   RMIN...SE USA PARA RETORNAR EL MENOR VALOR DE C(KF),C(KF+1),
C   C(KF+2),...,C(KL).
C   RMAX...SE USA PARA RETORNAR EL MAYOR VALOR DE C(KF),C(KF+1),
C   C(KF+2),...,C(KL).
C
C OBSERVACIONES
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C NINGUNA
C
C METODO
C   COMIENZA CON RMIN Y RMAX IGUALES A C(KF), LUEGO RECORRE C(I),
C   I=KF,...,KL, ACTUALIZANDO VALORES DEL MENOR Y EL MAYOR REQUERIDOS.
C
C*****
C
C   SUBROUTINE XTRMS(C,KF,KL,RMIN,RMAX)
C   DIMENSION C(1)
C   RMAX=C(KF)
C   RMIN=C(KF)
C   DO 2 I=KF,KL
C   IF(C(I).LE.RMAX)GO TO 1
C   RMAX=C(I)
C   GO TO 2
C 1 IF(C(I).GE.RMIN)GO TO 2
C   RMIN=C(I)
C 2 CONTINUE
C   RETURN
C   END
C
C   SUBROUTINE SORT(R,N)
C*****
C PROPOSITO
C   ORDENAR UN ARREGLO DADO
C
C USO:
C   "CALL SORT(R,N) "
C
C DESCRIPCION DE PARAMETROS
C
C   R...NOMBRE DEL ARREGLO UNIDIMENSIONAL A ORDENAR

```

```

C   N...DIMENSION DEL ARREGLO A ORDENAR
C OBSERVACIONES:
C   ESTE ALGORITMO ES MAS EFICIENTE QUE EL METODO DE LA BURBUJA O EL
C   METODO POR INTERCAMBIO. AUNQUE NO SEA OPTIMO, PROVEE UNA TECNICA
C   SIMPLE Y RELATIVAMENTE EFICIENTE DE ORDENAR Y FACIL DE PROGRAMAR.
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C METODO:
C   EL METODO USADO ES "SHELL SORT" FUE INTRODUCIDO POR D.L.SHELL
C   EN LA REF.1 .OTROS MAS RECIENTES DESARROLLOS EN LA REFERENCIA
C   2 Y SU BIBLIOGRAFIA.
C
C REFERENCIAS:
C   1. SHELL,D.L.: "A HIGH-SPEED SORTING PROCEDURE," COMMUNICATIONS
OF
C   THE ASSOCIATION FOR COMPUTING MACHINERY, VOL.2, NO.7(JULY,
C   1959), PAG.30-32.
C   2. GHOSHDASTIDAR,D. AND ROY,M.K.: "A STUDY ON THE EVALUATION OF
C   SHELL'S SORTING TECHNIQUE," THE COMPUTER JOURNAL, VOL.18(1975),
C   PAG.234-235.
C
C*****
C
C   SUBROUTINE SORT(R,N)
C   DIMENSION R(1)
C   M=N
C   1 M=M/2
C   IF(M.EQ.0) GOTO 5
C   J=1
C   2 I=J
C   4 IF(R(I).LE.R(I+M)) GOTO 3
C   SAV=R(I)
C   R(I)=R(I+M)
C   R(I+M)=SAV
C   I=I-M
C   IF(I.GE.1) GOTO 4
C   3 J=J+1
C   IF(J.GT.N-M) GOTO 1
C   GOTO 2
C   5 RETURN
C   END
C
C*****
C   SUBROUTINE GCD(ISMALL,ILARGE,IGCD,INVERS)
C*****
C   GCD CALCULA EL MAXIMO COMUN DIVISOR DE DOS ENTEROS DADOS.
C
C PROPOSITO
C   PARA DETERMINAR SI DOS ENTEROS DADOS SON RELATIVAMENTE PRIMOS Y
C   PARA
C   CALCULAR LA INVERSA MULTIPLICATIVA DEL MENOR ENTERO SEGUN MODULO
C   DEL
C   MAYOR CUANDO AMBOS SON DADOS (ES DECIR, EL UNICO ENTERO X ENTRE
C   1,2,3
C   ,...,M-1, TAL QUE ISMALL*X=1 MODULO ILARGE)
C
C USO:
C   CALL GCD(ISMALL,ILARGE,IGCD,INV).GCD ES LLAMADO POR TSTM1 Y TSTM2
C   COMO UNA RUTINA UTILITARIA.

```

```

C
C DESCRIPCION DE PARAMETROS
C   ISMALL...MENOR DE LOS DOS ENTEROS DADOS
C   ILARGE...MAYOR DE LOS DOS ENTEROS DADOS
C   IGCD...MAXIMO COMUN DIVISOR DE ISMALL Y ILARGE
C   INVERS...INVERSA MULTIPLICATIVA DE ISMALL MODULO ILARGE
C           CUANDO IGCD=1 , 0 CUANDO IGCD>1.
C SUBROUTINAS Y FUNCIONES REQUERIDAS:
C   NINGUNA
C METODO:
C REFERENCIAS
C   1. KNUTH,D.E.: "THE ART OF COMPUTER PROGRAMMING, VOLUME 2/
C       SEMINUMERICAL ALGORITHMS", ADDISON-WESLEY PUBLISHING COMPANY,
C       INC., READING, MASSACHUSETTS, 1969, PAG.294-297.
C*****
C
C       SUBROUTINE GCD(ISMALL,ILARGE,IGCD,INVERS)
C
C       J1=IABS(ISMALL)
C       J2=IABS(ILARGE)
C
C       IP0=0
C       IQ0=1
C       IP1=1
C       IQ1=0
C       ISGN=1
30  IQUOT=J2/J1
C       IREM=MOD(J2,J1)
C       ISGN=-ISGN
C       IP=IQUOT*IP1+IP0
C       IQ=IQUOT*IQ1+IQ0
C       IF(IREM.EQ.0) GO TO 40
C       J2=J1
C       J1=IREM
C       IP0=IP1
C       IP1=IP
C       IQ0=IQ1
C       IQ1=IQ
C       GO TO 30
40  IGCD=J1
C       ICOEF1=IP1*(-ISGN)
C       ICOEF2=IQ1*(-ISGN)
C       INVERS=MOD(ICOEF1,IABS(ILARGE))
C       IF(IGCD.NE.1) INVERS=0
C       RETURN
C       END
C
C       SUBROUTINE PNTCHI(V)
C*****
C   ESTA SUBROUTINA IMPRIME LOS PUNTOS 1,5,25,50,75,95, Y 99 PORCIENTE DE
C   UNA VARIABLE ALEATORIA CHI-CUADRADO CON V GRADOS DE LIBERTAD. LA
C   SUBROUTINA POINTS ES USADA PARA CALCULAR ESTOS PUNTOS PORCENTUALES
C
C PROPOSITO
C   IMPRIME PUNTOS PORCENTUALES CHI-CUADRADO
C
C USO: "CALL PNTCHI(V)"
C
C DESCRIPCION DE PARAMETROS
C   V(CINCO)...GRADOS DE LIBERTAD PARA LOS PUNTOS PORCENTUALES
C

```

```

C SUBROUTINAS Y FUNCIONES REQUERIDAS:
C POINTS,DCDFNI,DCDFN
C
C METODO:
C REFERENCIAS CONSIGNADAS EN SUBROUTINA POINTS
C
C*****
*
C
SUBROUTINE PNTCHI(V)
DIMENSION STAT(7)
COMMON /FIRST/CHIVAL(100)
REAL*8 CHIVAL
INTEGER V
CALL POINTS(V)
STAT(1)=CHIVAL(1)
STAT(2)=CHIVAL(5)
STAT(3)=CHIVAL(25)
STAT(4)=CHIVAL(50)
STAT(5)=CHIVAL(75)
STAT(6)=CHIVAL(95)
STAT(7)=CHIVAL(99)
WRITE(6,7)
7 FORMAT(' ',38X,'PROBABILITY A CHI-SQUARE RANDOM VARIABLE',
*,38X,'WITH V DEGREES OF FREEDOM EXCEEDS THE')
WRITE(6,5)
5 FORMAT(20X,'DEGREES',11X,'VALUE GIVEN BELOW:',/,
*20X,'OF FREEDOM',8X,3H99%,6X,3H95%,6X,3H75%,6X,3H50%,
*6X,3H25%,6X,3H 5%,6X,3H 1%,/)
WRITE(6,6)V,STAT
6 FORMAT(20X,' V=' ,I3,9X,7(F7.3,2X))
RETURN
END
C
C SUBROUTINE POINTS(V)
C*****
C ESTA SUBROUTINA GENERA PUNTOS PORCENTUALES DE UNA VARIABLE ALEATORIA
C CHI-CUADRADO CON V GRADOS DE LIBERTAD.
C
C PROPOSITO:
C GENERA LOS PUNTOS PORCENTUALES 1,2,3,4,5,...,96,97,98,99 DE UNA
C VARIABLE ALEATORIA CHI-CUADRADO CON V GRADOS DE LIBERTAD.
C
C USO: "CALL POINTS(V)"
C
C DESCRIPCION DE PARAMETROS
C V(CINCO)...GRADOS DE LIBERTAD
C
C OBSERVACIONES:
C LOS PUNTOS PORCENTUALES RETORNAN EN EL ARREGLO CHIVAL, COMO
C CHIVAL(1),
C CHIVAL(2),...,CHIVAL(99). CHIVAL(100)=9999.9999.
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS:
C DCDFNI,CHIDF
C
C METODO
C PARA LOS NUMEROS CON GRADOS DE LIBERTAD V >=22, SE USA UNA
C APROXIMACION
C BASADA EN "CORNISH-FISHER EXPANSION"; PARA NUMEROS CON
C V=1,2,...,21, SE

```

```

C USA UN PROCESO "PINCH" CON LA FORMULA EXACTA DE LA FUNCION DE
DISTRIBUCION CHI-CUADRADO, QUE SE DA MEDIANTE LA FUNCION CHIDF(A,V);
C
C REFERENCIAS:
C 1. DUDEWICZ,E.J.:"IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM NUMBER
GENE-
C RATOR PACKAGE,"TECHNICAL REPORT,DEPARTMENT OF STATISTICS, THE
OHIO
C STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
C 2. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C GENERATION AND TESTING," AMERICAN SCIENCES
PRESS,COLUMBUS,OHIO,1981.
C
C*****
C
SUBROUTINE POINTS(V)
REAL*8 DCFNI,CHIDF,VV,P,Z,A1,A2,A3,A4,A5,A6,A7,CHIVAL
REAL*8 L,U,X,Y1,Y2,Y3,US,LU
INTEGER V
COMMON /FIRST/CHIVAL(100)
VV=V
IF(V.GE.22)GO TO 800
C
C THIS SECTION HANDLES CASES V=1,2,...,21
C
J=1
P=0.01D0*J
20 L=0.0
U=40.0
Y1=CHIDF(L,V)
Y3=CHIDF(U,V)
30 X=(L+U)/2.D0
Y2=CHIDF(X,V)
IF(Y2.GE.P) GO TO 40
L=X
Y1=Y2
GO TO 50
40 U=X
Y3=Y2
50 US=U-L
IF(US.GE.0.000001) GO TO 30
CHIVAL(J)=(L+U)/2.D0
LU=(L+U)/2.D0
Y2=CHIDF(LU,V)
IF(J.LT.99) GO TO 60
GO TO 80
60 J=J+1
P=0.01D0*J
GO TO 20
C
C THIS SECTION HANDLES CASES V=22,23,...
C
800 DO 820 II=2,100
P=.01D0*(II-1)
Z=DCFNI(P)
A1=DSQRT(VV)*(Z*DSQRT(2.D0))
A2=(2.D0/3.D0)*(Z**2-1.D0)
A3=(1.D0/DSQRT(VV))*((Z**3-7.D0*Z)/(9.D0*DSQRT(2.D0)))
A4=(6.D0*Z**4+(14.D0*Z**2)-32.D0)/405.D0

```

```

      A5=(9.D0*Z**5+256.D0*Z**3-433.D0*Z)/(4860.D0*DSQRT(2.D0))
      A6=(12.D0*Z**6-(243.D0*Z**4)-(923.D0*Z**2)+1472.D0)/25515.D0
      A7=((3753.D0*Z**7)+(4353.D0*Z**5)-(289517.D0*Z**3)-
(289717.D0*Z))/
      *(9185400.D0*DSQRT(2.D0))
      CHIVAL(II-1)=
      *      VV+A1+A2+A3-
(1.D0/VV)*A4+((1.D0/(VV*DSQRT(VV)))*A5)+(1.D0/VV**
      *2)*A6-(1.D0/((VV**2)*DSQRT(VV)))*A7
      820 CONTINUE
C
C          WE NOW SET CHIVAL(100) TO A LARGE NUMBER AND RETURN
C
      80 CHIVAL(100)=9999.99999
      RETURN
      END
C
C          SUBROUTINE BASE2(INTEGR,IBITS,LEADCF)
C*****
C PROPOSITO
C BASE2 CALCULA LA EXPANSION BINARIA DE UN ENTERO, ALMACENANDO LOS
BITS
C EN EL ARREGLO IBITS. LA EXPANSION BINARIA ES USADA PARA CALCULAR
C POTENCIAS CUANDO EL EXPONENTE ES MUY GRANDE
C
C USO:      "CALL BASE2(INTEGR,IBITS,LEADCF)"
C
C DESCRIPCION DE PARAMETROS
C INTEGR...ENTERO AL QUE SE LE CALCULARA EXPANSION BINARIA
C IBITS...ARREGLO DONDE RETORNAN LOS BITS DE LA EXPANSION BINARIA
C LEADCF...INDICES DE LA MAYOR POTENCIA DE 2 DADA EN LA EXPANSION
BINARIA
C
C OBSERVACIONES:
C BASE2 ES LAMADA PORLA SUBRUTINA ORED COMO UNA RUTINA UTILITARIA
C
C SUBRUTINAS Y FUNCIONES REQUERIDAS:
C      NINGUNA
C METODO:
C REFERENCIAS
C      1. KNUTH,D.E.: "THE ART OF COMPUTER PROGRAMMING, VOLUME 2/
C          SEMINUMERICAL ALGORITHMS," ADDISON-WESLEY PUBLISHING COMPANY,
C          INC., READING, MASSACHUSETTS, 1969, PP.280-282.
C*****
C          SUBROUTINE BASE2(INTEGR,IBITS,LEADCF)
C
C          DIMENSION IBITS(31)
C          IA=INTEGR
C
C          DO 5 I=1,31
      5  IBITS(I)=0
C
C          DO 10 I=1,31
          IBITS(I)=MOD(IA,2)
          IA=IA/2
          IF(IA.EQ.0) GOTO 20
      10 CONTINUE
C
      20 LEADCF=I
C

```

```

        RETURN
    END

C
C*****
C    SUBROUTINE NOGEN
C*****
C PROPOSITO:
C TERMINA LA EJECUCION CUANDO SE HACE UN LLAMADO A URNWN Y EL USUARIO
C NO HA PROVISTO DICHA RUTINA.
C*****
C
    SUBROUTINE NOGEN
    WRITE(6,100)
100 FORMAT(1X,'LA EJECUCION FUE TERMINADA PUESTO QUE URNWN HA SIDO
LLA
    *MADA PERO NO FUE PROVISTA POR EL USUARIO.')
    STOP
    RETURN
    END

C
C *****
C    SUBROUTINE ORDER(IELT,FACTOR,NFACTR,IORDER)
C*****
C PROPOSITO:
C PARA UN ENTERO ESPECIFICADO "IELT" ESTA SUBRUTINA CALCULA EL MENOR
ENTERO
C POSITIVO K (LLAMADO ORDEN DE IELT) PARA EL CUAL IELT**K=1 MODULO
M,DONDE
C M ES EL PRODUCTO DE LAS POTENCIAS PRIMAS ALMACENADAS EN EL ARREGLO
"FACTOR" .
C ESTA RUTINA SE USA PARA CALCULAR EL PERIODO DE UN GENERADOR
CONGRUENCIAL
C LINEAL
C
C USO:
C "CALL ORDER(IELT,FACTOR,NFACTR,IORDER)".
C
C DESCRIPCION DE PARAMETROS:
C IELT...ENTERO, RELATIVAMENTE PRIMO CON M, CUYO ORDEN EN EL GRUPO DE
C UNIDADES MODULO M SE CALCULARA
C FACTOR...ARREGLO DE 31x2 QUE DA LA FACTORIZACION DE POTENCIAS
PRIMAS DE
C MODULO M.FACTOR(I,1) ES EL I-ESIMO PRIMO EN LA
FACTORIZACION DE
C M. FACTOR(I,2) ES LA POTENCIA DEL FACTOR PRIMO(I,1) EN LA
FACTORIZACION DE M
C NFACTR...CANTIDAD DE PRIMOS DISTINTOS EN LA FACTORIZACION DE M
C IORDER...ORDEN DE IELT MODULO M COMO SE CALCULA EN LA RUTINA. ESTE
C VALOR ES RETORNADO A LA RUTINA QUE LLAMA.
C
C OBSERVACIONES:
C LOS CALCULOS INTERMEDIOS CON ENTEROS PUEDEN EXCEDER 2**31-1, MAXIMO
TAMA-
C NIO DE UN ENTERO POSITIVO.PARA SOLUCIONAR ESTE PROBLEMA,LA RUTINA
REQUIERE
C EL USO DE EXTENDED PRECISION.(OTROS COMPILADORES)
C LA RUTINA ORDER ES LLAMADA POR TSTM2.
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS:
C TSTM3, BASE2, IGCD
C

```



```

C METODO:
C SE USA TEORIA DE GRUPOS ABSTRACTOS PARA EVITAR GRANDES CALCULOS. EL
C GRUPO
C DE UNIDADES MODULO M ES DIVIDIDO (SPLIT) EN EL PRODUCTO DIRECTO DE
C UNIDA-
C DES MODULO LAS POTENCIAS PRIMAS DE M Y EL ORDEN DE LAS IMÁGENES DE
C IELT
C EN CADA UNA DE LAS COMPONENTES. PARA PRIMOS IMPARES, SE USA LA DES-
C COMPOSICION DEL GRUPO EN SUS SUBGRUPOS "SYLOW-P". SE TIENE EN
C CUENTA QUE SI
C IELT**K=1 Y K=K1*K2 ES UNA FACTORIZACION DE K EN FACTORES PRIMOS
C RELATIVOS
C PARA LOS CUALES IELT**K1=1 Y IELT**K2=1, ENTONCES K ES DEL ORDEN DE
C IELT.
C
C REFERENCIAS:
C 1. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C GENERATION AND TESTING," AMERICAN SCIENCES
C PRESS,COLUMBUS,OHIO,1981
C 2. ZASSENHAUS,H.J.: "THE THEORY OF GROUPS,"CHELSEA PUBLISHING CO.
C INC., NEW YORK, NEW YORK, 1949.
C*****
C
C SUBROUTINE ORDER(IELT,FACTOR,NFACTR,IORDER)
C INTEGER FACTOR(31,2)
C REAL*8 ELT,ELTPWR,FPRIME,FPRIPW,PRD,ONE,BASEP(31)
C DIMENSION IBITS(31),INTGR(31,2)
C
C INICIALIZACION PARA CADA LLAMADO A LA SUBRUTINA
C
C ONE=1.0D0
C LCM=1
C ISW=1
C IORDER=1
C
C VERIFICAR EL CASO TRIVIAL DONDE IELT=1
C
C IF(IELT.NE.1) GOTO 5
C IORDER=1
C RETURN
C
C BUCLE PARA CALCULAR EL ORDEN MODULO M DE CADA POTENCIA
C
C 5 DO 90 K=1,NFACTR
C
C INICIALIZACIONES PARA CADA NUEVA POTENCIA PRIMA
C
C ELT=IELT
C FPRIME=FACTOR(K,1)
C FPRIPW=FPRIME**FACTOR(K,2)
C
C MAXORD=FACTOR(K,1)-1
C
C VERIFICAR SI ELT = 1 MODULO PRIMO P (P=FACTOR(K,1))
C
C IF(ELT.NE.ONE) GOTO 10
C IORDER=1
C GOTO 40
C
C HALLAR LA FACTORIZACION PRIMA DE MAXIMO ORDEN

```

```

C
10 CALL TSTM3(MAXORD,INTGR,NDXORD,ISW)
C
C     TEST PARA HALLAR ORDEN DE ELT MODULO PRIMO P
C
C     EL SWITCH "NEW" INDICA SI ES EL PRIMER PASAJE
C     POR EL CICLO(EN TAL CASO LAS POTENCIAS
C     DE ELT SON CALCULADAS Y ALMACENADAS).
C
C     NEW=0
C     DO 35 I=1,NDXORD
15 IF(INTGR(I,2).EQ.0) GOTO 35
    IEXP=MAXORD/INTGR(I,1)
C
C     PROBAR SI ELT**IEXP=1 MOD P USANDO LA BASE 2
C     DESARROLLO DE IEXP.
C
C     ELTPWR=ONE
C     BASEP(1)=ELT
C     CALL BASE2(IEXP,IBITS,LEADCF)
C
C     DO 30 J=1,LEADCF
C     IF(IBITS(J).EQ.0) GOTO 20
C     ELTPWR=DMOD(ELTPWR*BASEP(J),FPRIME)
20 IF(NEW.NE.0) GOTO 30
    J1=J+1
    BASEP(J1)=DMOD(BASEP(J)*BASEP(J),FPRIME)
    IF(BASEP(J1).NE.ONE) GOTO 30
    IORDER=2**J
    GOTO 40
30 CONTINUE
C
C     PONER "OFF" EL SWITCH "NEW" DESPUES QUE LAS POTENCIAS DE
C     "ELT" HAN SIDO CALCULADAS Y ALMACENADAS
C
C     NEW=1
C     IF(ELTPWR.NE.ONE) GOTO 35
C
C     SI ELTPWR=1, EL ORDEN DEBE SER DIVISOR DE IEXP
C     (IEXP=MAXORD/P
C     DONDE P ES EL ACTUAL PRIMO EN CONSIDERACION) Y POR LO
C     TANTO,
C     UNA POTENCIA DE P PUEDE SER ELIMINADA DE LOS VALORES QUE
C     HAN
C     SIDO VERIFICADOS COMO ORDEN DE ELT.
C
C     MAXORD=IEXP
C     INTGR(I,2)=INTGR(I,2)-1
C     GOTO 15
C
C     35 CONTINUE
C
C     ESTO COMPLETA LA DETERMINACION DEL ORDEN DE "ELT"
C     MODULO P
C
C     IORDER=MAXORD
C
C     CALCULO DEL ORDEN DE "ELT" MODULO P**E, DONDE "E" ES LA
C     POTENCIA DE P PRODUCIDA EN LA FACTORIZACION DEL MODULO
C     ORIGINAL.
C

```

```

40 IF(FACTOR(K,2).EQ.1) GOTO 75
C
PRD=ELT
ELTPWR=ONE
C
CALL BASE2(IORDER,IBITS,LEADCF)
C
DO 50 J=1,LEADCF
IF(IBITS(J).EQ.0) GOTO 45
ELTPWR=DMOD(ELTPWR*PRD,FPRIPW)
45 PRD=DMOD(PRD*PRD,FPRIPW)
IF(PRD.NE.ONE) GOTO 50
IORDER=2**J
GOTO 75
50 CONTINUE
C
IF(ELTPWR.EQ.ONE) GOTO 75
C
C
C SI ELTPWR**IORDER = 1, EL ORDEN DE IELT DEBE SER
C EL PRODUCTO DEL ACTUAL VALOR DE IORDER POR ALGUNA
C POTENCIA P . VERIFICACION DE POTENCIAS P DE ELTPWR.
C
MXPOW=FACTOR(K,2)-1
CALL BASE2(FACTOR(K,1),IBITS,LEADCF)
C
DO 70 J=1,MXPOW
PRD=ELTPWR
ELTPWR=ONE
DO 65 L=1,LEADCF
IF(IBITS(L).EQ.0) GOTO 60
ELTPWR=DMOD(ELTPWR*PRD,FPRIPW)
60 PRD=DMOD(PRD*PRD,FPRIPW)
65 CONTINUE
C
IORDER=IORDER*FACTOR(K,1)
IF(ELTPWR.EQ.ONE) GOTO 75
70 CONTINUE
C
C ESTO COMPLETA EL CALCULO DEL ORDEN DE
C "ELT" MODULO P**E. EL ORDEN MODULO M ES EL
C MINIMO COMUN MULTIPLO DE LOS ORDENES MODULO
C DE LOS FACTORES PRIMOS DE M.
C
75 CALL GCD(LCM,IORDER,IGCD,INVR)
LCM=LCM*(IORDER/IGCD)
C
90 CONTINUE
C
C ESTO COMPLETA EL BUCLE DE TODOS LOS FACTORES
C PRIMOS DEL MODULO M.
C
IORDER=LCM
C
RETURN
END
C
C----- G E N E R A D O R E S : -----
C
C Orden de las Subrutinas : RRN**, URN**I, URN***
C-----

```

```

C*****
C      SUBROUTINE RRNWN
C*****
C
C      ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C      GENERADOR URNWN.
C      EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C      A URNWN1.
C      URNWN CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO
REQUIE-
C      RE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIAR URNWN
DESPUES
C      DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
C      SUBROUTINE RRNWN
C
C      REAL*8 NAME
character*8 name,messg,variab,t0lnm,tstnm,cknm(10)
CHARACTER C1,RESP
COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
COMMON /FIFTH/X(10000)
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
DATA YES/'YES '/
C
C      IF(NEWGEN.NE.0) GOTO 12
C
C      PARAMETROS IA,IR SERAN DEFINIDOS EN SUBROUTINA URNWN1
CALL URNWN1
IF(IA.EQ.0) GOTO 10
WRITE(6,100)
100 FORMAT(' ',1X,'VALORES INICIALES ENTEROS PARA EL GENERADOR DEL
USU
      *ARIO:')
WRITE(6,9)(INTIN(K),K=1,IA)
9 FORMAT(' ',1X,5(I11,1X))
WRITE(*,*)' '
10 IF(IR.EQ.0) GOTO 12
WRITE(6,104)
104 FORMAT('0',5X,'VALORES INICIALES REALES PARA EL GENERADOR DEL
USUA
      *ARIO:',//)
WRITE(6,106)(REALIN(K),K=1,IR)
106 FORMAT(' ',5F11.8)
WRITE(*,*)' '
C
C      12 IF(TIMES.EQ.YES)GO TO 14
C
C      GENERA NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
CALL URNWN(X,NBATCH)
GOTO 16
C
C      GENERA NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
14 KTIME=0
C      CALL TIMER(ITIME)
CALL URNWN(X,NBATCH)

```

```

C      CALL TIMER(ITIME)
      KTIME=KTIME+ITIME
C
C      16 IF(NEWGEN.NE.0) GOTO 999
C
      WRITE(*,*) ' '
      WRITE(6,200) NBATCH
      200 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO
(SIG
      *UIENTE A LOS',I6)
      WRITE(6,201)
      201 FORMAT(1X,'YA GENERADOS) ES:')
      IF(IA.EQ.0) GOTO 18
      WRITE(6,202) (INTIN(K),K=1,IA)
      202 FORMAT(' ',10(I11,1X))
      18 IF(IR.EQ.0) GOTO 999
      WRITE(6,206) (REALIN(K),K=1,IR)
      206 FORMAT(' ',10(F11.8,1X))
      999 RETURN
      END
C
C*****
C      SUBROUTINE URNWN
C*****
C
C      ESTA SUBROUTINA INICIALIZA A URNWN. EL GENERADOR DEL USUARIO
C      TIENE LAS SIGUIENTES OPCIONES:
C
C      IOPT=1  INICIALIZACION DE VALORES POR LECTURA
C      IOPT=2  INICIALIZACION DE VALORES EN LA MISMA URNWN
C
C*****
C
C      SUBROUTINE URNWN
      character*8 name,messg,variab,t01nm,tstnm,cknm(10)
c      REAL*8 NAME
      COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
C
      COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C
      IA=1
      IR=0
      INTIN(1)=32767
      IF((IA.EQ.0).AND.(IR.EQ.0)) GOTO 40
      IF(IOPT.EQ.0) GO TO 20
      IF(IOPT.EQ.2)GO TO 999
      IF(IA.EQ.0) GOTO 12
C      VALORES POR TECLADO
      WRITE(6,*)'DEFINIR VALORES INTIN(K), K=1,...,IA, EN LA
SUBROUTINA:
      *URNWNI'
      12 IF(IR.EQ.0) GOTO 999
C
      WRITE(6,*)'DEFINIR VALORES REALIN(K), K=1,...,IR, EN LA
SUBROUTINA:
      *URNWNI'
C
      40 WRITE(6,200)
      200 FORMAT(1X,'LOS PARAMETROS IA=IR=0 INDICAN QUE NO SE HAN
UTILIZADO
      *ESTOS VALORES',

```

```

*/1X,'PARTICULARES DE INICIALIZACION PARA EL GENERADOR URNWN.',
*//1X, ' EJECUCION TERMINADA. ')
GOTO 998
20 WRITE(6,210)
210 FORMAT(1X,'LA OPCION DE INICIALIZACION (0) PARA EL GENERADOR
URNWN
* ES UNA OPCION EQUIVOCADA'
*,/1X,'PARA UN GENERADOR DEL USUARIO'
*//1X, ' EJECUCION TERMINADA. ')
998 STOP
999 RETURN
END

C
C*****
C SUBROUTINE URNWN(X,NBATCH)
C*****
C
C ESTA SUBROUTINA SE PROVEE COMO UN PROTOTIPO DE LA FORMA EN LA QUE
C DEBE PROCESARSE EN GENTSTPC UN GENERADOR PROVISTO POR EL USUARIO. SI
C EL USUARIO PROVEE OTRA SUBROUTINA, DEBE SOBRESERIBIR ESTA. SI EL
C USUARIO INTENTA USAR URNWN SIN DAR SU PROPIO CODIGO, APARECERA
C UN MENSAJE DE ERROR Y TERMINARA LA EJECUCION. ESTE EJEMPLO ACTUA
C COMO URN09.
C EN LA DESCRIPCION DE LA SUBROUTINA URN09 SE VE "PROPOSITO","USO",
C "DESCRIPCION DE PARAMETROS", "OBSERVACIONES", "SUBROUTINAS Y
C FUNCIONES REQUERIDAS", Y "METODO".
C UNA SECUENCIA TIPICA DE LLAMADO EN EL PROGRAMA PRINCIPAL DEL
C USUARIO PUEDE SER (CON INICIALIZACION PROVISTA POR EL USUARIO) :
C COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
C IA=1
C IR=0
C INTIN(1)=524287
C CALL TESTER
C DE OTRO MODO, SI EN VEZ DE INICIALIZACION POR EL USUARIO
C FUERAN DATOS EN ARCHIVOS DE ENTRADA, SE TENDRIA:
C COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
C IA=1
C IR=0
C CALL TESTER
C*****
C
SUBROUTINE URNWN(X,NBATCH)
CHARACTER C1,RESP
COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
DIMENSION X(NBATCH)
OPEN(UNIT=60,ACCESS='DIRECT',RECL=1,FORM='FORMATTED',FILE=
*'OPCION')
READ(60,12)C1
12 FORMAT(1A)
IF(C1.EQ.'S')THEN
READ(60,12)RESP
GOTO 88
ENDIF
OPEN(UNIT=63,ACCESS='DIRECT',RECL=1,FORM='FORMATTED',FILE=
*'GENOARCH')
READ(63,12)RESP
CLOSE(63)
CLOSE(60)
OPEN(UNIT=60,ACCESS='DIRECT',RECL=1,FORM='FORMATTED',FILE=
*'OPCION')
WRITE(60,12)'S'

```

```

      WRITE(60,12)RESP
88  IF(Resp.EQ.'1') THEN
C
C  GENERACION DEL GENERADOR PROPIO "URNWN"
C
      J=INTIN(1)
      DO 10 I = 1,NBATCH
      J=J*1220703125
      IF(J.LT.0)J=J+2147483647+1
      X(I)=FLOAT(J)*0.4656613E-9
10  CONTINUE
      INTIN(1)=J
C
C  LECTURA DE "ARCHDAT"
C
      ELSE IF(Resp.EQ.'2') THEN
      DO 20 II = 1 ,NBATCH
      READ(90,900)XXX
      X(II)=XXX
900  FORMAT(F11.10)
20  CONTINUE
      ELSE
      WRITE(*,*)'HA INGRESADO UN VALOR INCORRECTO'
      STOP
      ENDIF
      CLOSE(60)
      RETURN
      END
C
C
C*****
C  SUBROUTINE RRN01
C*****
C  ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C  GENERADOR URN01.
C  EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C  A URN01I E IMPRIME LOS VALORES INICIALES.
C  URN01 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C  QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN01
C  DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
      SUBROUTINE RRN01
      character*8 name,messg,variab,t01nm,tstnm,cknm(10)
c  REAL*8 NAME
      INTEGER DBLPRT,EXTPRT
      COMMON /FOURTH/Y(10000)
      COMMON /FIFTH/X(10000)
      COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
      COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
      COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
      COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKK,L,ML,MK,MMM
      COMMON /CODE/GCODE
      DATA YES/'YES '/
      IF(NEWGEN.NE.0) GOTO 10
      CALL URN01I
      WRITE(6,990)IX
990  FORMAT(' ',8X,'IX=',I11)
      10 IF(TIMES.EQ.YES)GO TO 20
C

```

```

C          GENERACION DE NUMEROS ALEATORIOS SIN CALCULO TIEMPOS
C
C          CALL URN01(X,IX,NBATCH)
C
C          GOTO 30
C
C          GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C          20 KTIME=0
C             CALL TIMER(ITIME)
C             CALL URN01(X,IX,NBATCH)
C             CALL TIMER(ITIME)
C             KTIME=KTIME+ITIME
C
C          30 IF(NEWGEN.NE.0) GOTO 999
C             WRITE(6,400) NBATCH
C          400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
C             *' (SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES:')
C             WRITE(6,990) IX
C          999 RETURN
C             END
C
C*****
C          SUBROUTINE URN01I
C*****
C
C          ESTA SUBRUTINA INICIALIZA AQUELLOS GENERADORES QUE NECESITAN
C          UN SOLO VALOR INICIAL. LAS OPCIONES DISPONIBLES SON:
C
C          IOPT=0  INICIALIZACION POR DEFECTO
C          IOPT=1  INICIALIZACION MEDIANTE LECTURAS DE VALORES EN LA RUTINA
C          IOPT=2  INICIALIZACION DE VALORES PROVISTOS POR EL USUARIO
C
C          LOS GENERADORES INICIALIZADOS POR ESTA RUTINA SON LOS
C          SIGUIENTES:
C          URN01.
C          URN07.
C          URN08.
C          URN09.
C          URN13.(*)
C          URN15.
C          URN16.
C          URN17.
C          URN18.
C          URN19.
C          URN20.
C          (*)EN LA VERSION ASSEMBLER DEL PAQUETE TESTRAND
C*****
C
C          SUBROUTINE URN01I
C          ENTRY URN07I
C          ENTRY URN08I
C          ENTRY URN09I
C          ENTRY URN10I
C          ENTRY URN13I
C          ENTRY URN15I
C          ENTRY URN16I
C          ENTRY URN17I
C          ENTRY URN18I
C          ENTRY URN19I
C          ENTRY URN20I

```



```

      character*8 name,messg,variab,t01nm,tstnm,cknm(10),except
c     REAL*8 NAME,EXCEPT
      DATA EXCEPT/'URN10  '/
      COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
      COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
      IF(IOPT.EQ.1)GO TO 10
      IF(IOPT.EQ.2)GO TO 30
      IF(NAME.EQ.EXCEPT)IX=1
      IF(NAME.NE.EXCEPT)IX=524287
      GO TO 30
10    READ(5,600)IX
600   FORMAT(I11)
30    RETURN
      END
C
C*****
C     SUBROUTINE URN01(X,IX,NBATCH)
C*****
C
C     INICIALMENTE TESTRAND ADOPTO ESTA RUTINA DEL PAQUETE LLRANDOM EN
C     CODIGO ASSEMBLER ESPECIFICO AL SISTEMA IBM/370 (NAVAL
POSTGRADUATE
C     SCHOOL RANDOM NUMBER GENERATOR PACKAGE)
C     EN ESTA VERSION DE GENTSTPC SE DESARROLLA OTRO ALGORITMO.
C
C     PROPOSITO
C     GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE
SIMPLE
C     PRECISION EN (0.0,1.0)
C
C     USO
C     CALL URN01(X,IX,NBATCH)
C
C     DESCRIPCION DE PARAMETROS
C     IX...ES EL VALOR INICIAL DE UNA SUCESION Y PUEDE SE CUALQUIER
C     ENTERO ENTRE 1 Y 2147483647.(2**31 -1)
C     X... ES EL ARREGLO RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO
C     FLOTANTE DE SIMPLE PRECISION EN (0.0,1.0)
C     NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA
GENERAR
C     CON EL LLAMADO DE ESTA SUBRUTINA
C
C     OBSERVACIONES
C     IX NO DEBE SER ALTERADO POR EL USUARIO DURANTE LA EJECUCION
C     DEL PROGRAMA, SALVO QUE EL DECIDA REPETIR UNA SUCESION DE
C     NUMEROS ALEATORIOS.
C
C     SUBRUTINAS Y FUNCIONES REQUERIDAS
C     NINGUNA
C
C     METODO
C     REFERENCIAS:
C     1. ALGORITMO DE BRADLEY ,FOX Y SCHRAGE,"A GUIDE TO SIMULATION",
C     SPRINGER,2DA.ED. ,1987,P.212.
C     2. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C     GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C     COLUMBUS , OHIO,1981.
C*****
C
      SUBROUTINE URN01(X,IX,NBATCH)

```

```

REAL*8 X1(10000),X0,XM,XU
DIMENSION X(10000)
X0=IX
XM=2147483647D0
DO 11 I=1,NBATCH
X1(I)= DMOD(16807D0*X0,XM)
X0=X1(I)
XU=X1(I)/XM
X(I)=XU
11  END DO
IX=X0
RETURN
END

C
C*****
C      SUBROUTINE RRN02(NN,ML,MM,MK,L,MMM,KKK)
C*****
C
C      ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C      GENERADOR URN02.
C      EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C      A URN02I E IMPRIME LOS VALORES INICIALES.
C      URN02 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C      QUIERE) E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN02
C      DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C      SUBROUTINE RRN02(NN,ML,MM,MK,L,MMM,KKK)
C          character*8 name,messg,variab,t01nm,tstnm,cknm(10)
C      REAL*8 NAME
COMMON /FIFTH/X(10000)
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
DIMENSION NN(128)
DATA YES/'YES '/
IF(NEWGEN.NE.0) GOTO 10
CALL URN02I(NN,ML,MM,MK,L,MMM,KKK)
WRITE(6,990) ML,MM,MK,L,MMM,KKK
990 FORMAT(' ',8X,'ML=',I11,' MM=',I11,' MK=',I11,' L=',I11,
* ' MMM=',I11,' KKK=',I11)
WRITE(6,940)NN
940 FORMAT(9X,'NN= ',/,9(14(1X,Z8),/),1X,Z8,1X,Z8)
10 IF(TIMES.EQ.YES)GO TO 20

C
C      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO TIEMPOS
C
C      CALL URN02(X,NN,ML,MM,MK,L,MMM,KKK,NBATCH)
C      GOTO 30

C
C      GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20 KTIME=0
C      CALL TIMER(ITIME)
C      CALL URN02(X,NN,ML,MM,MK,L,MMM,KKK,NBATCH)
C      CALL TIMER(ITIME)
C      KTIME=KTIME+ITIME
C
30 IF(NEWGEN.NE.0) GOTO 999
WRITE(*,*)' '

```

```

WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO ',
*'(SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES:')
WRITE(6,990) ML,MM,MK,L,MMM,KKK
WRITE(6,940) NN
999 RETURN
END

C
C*****
C SUBROUTINE URN02I
C*****
C
C ESTA SUBROUTINA INICIALIZA URN02.LAS OPCIONES DISPONIBLES SON:
C
C IOPT=0 INICIALIZACION POR DEFECTO
C IOPT=1 INICIALIZACION MEDIANTE LECTURAS DE VALORES EN LA RUTINA
C IOPT=2 INICIALIZACION DE VALORES PROVISTOS POR EL USUARIO
C IOPT=3 TODAS LAS INICIALIZACIONES EXCEPTO NN PROVISTA POR EL
C USUARIO
C IOPT=4 TODAS LAS INICIALIZACIONES EXCEPTO NN LEIDA EN LA RUTINA
C
C*****
C
C SUBROUTINE URN02I(NN,ML,MM,MK,L,MMM,KKK)
DIMENSION N(128),NN(128)
C INICIALIZACION N(1),...,N(128) (MISMOS VALORES QUE PAQUETE
LLRND)
DATA N /880431333,845941495,233211304,1989617657,465185814,
*280672918,294923811,969688974,798989604,379880543,130022074,
*1958997525,1074191695,680854387,751282651,1208899767,695831691,
*1667008051,1682546364,1984522335,287570376,1137852001,1597983496,
*2015817872,1479672206,1468443024,1657203843,326324124,680973716,
*1451006002,1251441372,241092947,1815086916,1807193097,770906592,
*725422944,1822111098,470585328,939566271,1084841188,1988336409,
*229735215,1763201387,2072973152,1143606610,548108569,544252510,
*1980873641,1195919839,2089487851,1406149582,1839198022,2106705200,
*189238196,1170370207,1304402631,1936129483,810953177,706509560,
*476957499,1307077413,824336639,1487297852,1591453718,107382480,
*155452792,265840413,1440038626,785915439,1152058296,1726999383,
*1389732859,1838014251,1751063044,102451305,212848938,1046489181,
*976388856,1797117421,461971124,259337424,492056652,1152625277,
*1087711027,344810019,1477716555,809152324,1766452264,1687482934,
*1077592551,1906112218,328744821,1380339247,339750038,1993648985,
*1054008271,2006727977,1618648061,1300903972,168650429,1734500183,
*906733794,614096451,1092917209,1180334545,577024776,1406305431,
*648073629,973807028,883884075,1562357277,1705648154,1377603620,
*1845151798,220566094,768813055,571717967,218994012,212872559,
*1824677815,1573937649,450149130,284847256,2062965934,47834840,
*1766553923,1580332201,182920702/
C
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
character*8 name,messg,variab,t01nm,tstnm,cknm(10)
c REAL*8 NAME
C
IF (IOPT.EQ.1)GO TO 50
IF (IOPT.EQ.1)GO TO 50
IF (IOPT.EQ.2)GO TO 30
IF (IOPT.EQ.3.OR.IOPT.EQ.4)GO TO 60

```

```

C
    ML=65539
    MM=33554433
    MK=362436069
    L=089347405
    MMM=301467177
    KKK=240420681
C
    60 DO 20 I=1,128
        NN(I)=N(I)
    20 CONTINUE
C
    IF ( IOPT.EQ.4)READ(5,100)ML,MM,MK,L,MMM,KKK
100 FORMAT(6I11)
    GO TO 30
C
    50 READ(5,140)NN
140 FORMAT(15(8(Z8,1X),/),8(Z8,1X))
    READ(5,100)ML,MM,MK,L,MMM,KKK
C
    30 RETURN
    END
C*****
C    SUBROUTINE URN02(X,NN,ML,MM,MK,L,MMM,KKK,NBATCH)
C*****
C
C    SUBROUTINA DADA POR MARSAGLIA Y BRAY, PARA GENERAR NUMEROS ALEATO-
C    RIOS EN PUNTO FLOTANTE. EN EL PAQUETE IRCCRAND LA RUTINA "RN3"
C    UTILIZA EL MISMO METODO PARA GENERAR NUMEROS ALEATORIOS EN SIMPLE
C    PRECISION (REFERENCIA 2)
C
C    PROPOSITO
C    GENERACION DE VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE
C    SIMPLE PRECISION SOBRE (0.0,1.0)
C
C    USO
C    "CALL URN02I(ML,MM,MK,L,MMM,KKK,NN,)"
C    DESDE EL PROGRAMA PRINCIPAL.
C    LUEGO "CALL URN02(X,NN,ML,MM,MK,L,MMM,KKK,NBATCH)"
C
C    DESCRIPCION DE PARAMETROS
C    ML...CONSTANTE ENTERA (MULTIPLICADOR) DEL GENERADOR
C        DE ENTEROS ALEATORIOS
C    MM...CONSTANTE ENTERA (MULTIPLICADOR) DEL GENERADOR
C        DE ENTEROS ALEATORIOS
C    MK...CONSTANTE ENTERA (MULTIPLICADOR) DEL GENERADOR
C        DE ENTEROS ALEATORIOS
C    L.... SEMILLA PARA UNA SUCESSION ALEATORIA
C    MMM...SEMILLA PARA UNA SUCESSION ALEATORIA
C    KKK...SEMILLA PARA UNA SUCESSION ALEATORIA
C    NN... VECTOR VARIABLE CON DIMENSION ESPECIFICA 128 DONDE
C        SE ALMACENAN LOS ENTEROS ALEATORIOS USADOS
C    X... ARREGLO RESULTANTE EN PUNTO FLOTANTE DE SIMPLE PRECISION
C        PARA LOS NUMEROS ALEATORIOS EN (0.0,1.0)
C    NBATCH...CANTIDAD DE NUMEROS ALEATORIOS A SER GENERADOS POR EL
C        LLAMADO DE ESTA SUBROUTINA
C
C    OBSERVACIONES
C    LA SUBROUTINA URN02I INICIALIZA LAS VARIABLES
NN,ML,MM,MK,L,MMM,KKK;

```

```

C      SI EL USUARIO OPTA POR SUS PROPIOS VALORES NO DEBE INVOCAR EL
LLA-
C      MADO "CALL URN02I(ML,MM,MK,L,MMM,KKK,NN)", PERO EN CAMBIO DEBE
C      INICIALIZAR DICHAS VARIABLES.
C
C      ESTA SUBROUTINA ES ESPECIFICA PARA SISTEMAS IBM/360/370. EL METODO
C      FUE DADO EN REFERENCIA 1.
C
C      SUBROUTINAS Y FUNCIONES REQUERIDAS
C      NINGUNA
C
C      METODO
C      REFERENCIAS:
C      1. MARSAGLIA, G. AND BRAY, T.A.: "ONE-LINE RANDOM NUMBER
C      GENERATORS AND THEIR USE IN COMBINATIONS",COMMUNICATIONS OF TH
C      ACM, VOL.11(1968), PAG. 757-759.
C      2. DUDEWICZ,E.J.: "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM
C      NUMBER GENERATOR PACKAGE",TECHNICAL REPORT, DEPARTMENT OF
C      STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
C      3. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C      GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C      COLUMBUS, OHIO, 1981.
C      PARA SISTEMAS DISTINTOS A LAS MAQUINAS 360/370:
C      4. GROSENBAUGH,L.R., "MORE ON FORTRAN RANDOM NUMBER GENERATORS"
C      COMMUNICATIONS DE ACM, VOL.12(1969), P.639.
C
C*****
C
      SUBROUTINE URN02(X,NN,ML,MM,MK,L,MMM,KKK,NBATCH)
      DIMENSION NN(128)
      DIMENSION X(NBATCH)
      DO 12 I=1,NBATCH
      L=L*ML
      MMM=MMM*MM
      J=1+IABS(L)/16777216
      X(I)=.5+FLOAT(NN(J)+L+MMM)*.23283064E-9
      KKK=KKK*MK
      NN(J)=KKK
12  CONTINUE
      RETURN
      END
C
C
C*****
C      SUBROUTINE RRN03
C*****
C
C      ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C      GENERADOR URN03.
C      EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C      A URN03I E IMPRIME LOS VALORES INICIALES.
C      URN03 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C      QUIERE) E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN03
C      DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
      SUBROUTINE RRN03
      character*8 name,messg,variab,t01nm,tstnm,cknm(10)
c      REAL*8 NAME

```

```

COMMON /FIFTH/X(10000)
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKK,L,ML,MK,MMM
DATA YES/'YES '/

IF(NEWGEN.NE.0) GOTO 10
CALL URN03I
WRITE(6,990)IX,JX
10 IF(TIMES.EQ.YES) GOTO 20
C
C      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C      CALL URN03(X,IX,NBATCH)
C      GOTO 30
C
C      GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C      20 KTIME=0
C      CALL TIMER(ITIME)
C      CALL URN03(X,IX,NBATCH)
C      CALL TIMER(ITIME)
C      KTIME=KTIME+ITIME
C
C      30 IF(NEWGEN.NE.0) GOTO 999
C      WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
* ' SIGUIENTE A LOS ',I11/1X,'YA GENERADOS) ES:')
C      WRITE(6,990) IX,JX
990 FORMAT(1X,'IX= ',I11,5X,'JX= ',I11)
999 RETURN
END

C
C*****
C      SUBROUTINE URN03I
C*****
C
C      ESTA SUBRUTINA INICIALIZA AQUELLOS GENERADORES QUE NECESITAN
C      DOS VALORES INICIALES.LAS OPCIONES DISPONIBLES SON:
C
C      IOPT=0 INICIALIZACION POR DEFECTO
C      IOPT=1 INICIALIZACION MEDIANTE LECTURAS DE VALORES EN LA RUTINA
C      IOPT=2 INICIALIZACION DE VALORES PROVISTOS POR EL USUARIO
C
C      LOS GENERADORES INICIALIZADOS POR ESTA RUTINA SON LOS
SIGUIENTES:
C      URN03.
C      URN06.
C
C*****
C
C      SUBROUTINE URN03I
C      ENTRY URN06I
C      character*8 name,messg,variab,t01nm,tstnm,cknm(10)
c      REAL*8 NAME
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKK,L,ML,MK,MMM
IX=1
RETURN
END

```

```

C*****
C      SUBROUTINE URN03(X,IX,NBATCH)
C*****
C
C      INICIALMENTE TESTRAND ADOPTO ESTA RUTINA DEL PAQUETE SUPER-DUPER
C      EN CODIGO ASSEMBLER ESPECIFICO AL SISTEMA IBM/370 (MCGILL UNIV.
C      SCHOOL OF COMPUTER SCIENCE RANDOM NUMBER GENERATOR PACKAGE)
C      EN ESTA VERSION DE GENTSTPC SE DESARROLLA OTRO ALGORITMO.
C
C PROPOSITO
C  GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE
C  PRECISION EN (0.0,1.0)
C
C USO
C  CALL URN03(X,IX,NBATCH)
C
C DESCRIPCION DE PARAMETROS
C  IX...ES EL VALOR INICIAL DE UNA SUCESION Y PUEDE SE CUALQUIER
ENTE-
C      RO ENTRE 1 Y 2147483647.(2**31 -1)
C  X... ES EL ARREGLO RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO FLO-
C      TANTE DE SIMPLE PRECISION EN (0.0,1.0)
C  NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA
GENERAR
C      CON EL LLAMADO DE ESTA SUBRUTINA
C
C OBSERVACIONES
C  IX NO DEBE SER ALTERADO POR EL USUARIO DURANTE LA EJECUCION
C  DEL PROGRAMA, SALVO QUE DECIDA REPETIR UNA SUCESION DE
C  NUMEROS ALEATORIOS.
C
C SUBRUTINAS Y FUNCIONES REQUERIDAS
C  NINGUNA
C
C METODO
C  REFERENCIAS:
C  1. ALGORITMO DE BRADLEY ,FOX Y SCHRAGE,"A GUIDE TO SIMULATION",
C    SPRINGER, 2DA.ED.,1987,P.211
C  2. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C    GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C    COLUMBUS,OHIO,1981.
C*****
C
      SUBROUTINE URN03(X,IX,NBATCH)
      DIMENSION X(10000)
      INTEGER*4 XX(10000),X0,XM
      X0=IX
      XM=2147483648
      XM1=4.656612873e-10
      DO 11 I=1 ,NBATCH
      XX(I)= X0*65539
      IF(XX(I).LT.0) THEN
      XX(I)=1+(XX(I)+2147483647)
      END IF
      X0=XX(I)
      X(I)=XX(I)*XM1
11  END DO
      IX=X0
      RETURN
      END

```

```

C
C
C*****
C   SUBROUTINE RRN04(ITABLE)
C*****
C
C   ESTA SUBRUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN04.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN04I E IMPRIME LOS VALORES INICIALES.
C   URN04 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN04
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
C   SUBROUTINE RRN04(ITABLE)
C     character*8 name,messg,variab,t0lnm,tstnm,cknm(10)
c     REAL*8 NAME
C     COMMON /FIFTH/X(10000)
C     COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C     COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C     COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C     COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
C     DIMENSION ITABLE(128)
C     DATA YES/'YES '/
C     IF(NEWGEN.NE.0) GOTO 10
C
C     CALL URN04I
C     WRITE(6,990)IX
990  FORMAT(' ',8X,'IX=',I11)
C     WRITE(6,940) ITABLE
940  FORMAT(9X,'ITABLE=',/,9(14(1X,Z8),/),1X,Z8,1X,Z8)
C     CALL LLRND
C
C     10 IF(TIMES.EQ.YES)GO TO 20
C
C     GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C     CALL URN04(IX,X,NBATCH,ITABLE)
C     GOTO 30
C
C     GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C     20 KTIME=0
C     CALL TIMER(ITIME)
C     CALL URN04(IX,X,NBATCH,ITABLE)
C     CALL TIMER(ITIME)
C     KTIME=KTIME+ITIME
C
C     30 IF(NEWGEN.NE.0) GOTO 999
C     WRITE(6,400) NBATCH
400  FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
C     *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES:')
C     WRITE(6,990) IX
C     WRITE(6,940) ITABLE
999  RETURN
C     END
C
C*****
C

```



```

C      SUBROUTINE URN04 = URN01: ASSEMBLER
C*****
C*****
C      SUBROUTINE RRN05(MFSR,P,Q,INTSIZ,JJ)
C*****
C
C      ESTA SUBRUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C      GENERADOR URN05.
C      EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C      A URN05I E IMPRIME LOS VALORES INICIALES.
C      URN05 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE=
C      QUIERE)
C      E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN05 DESPUES
C      DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
C      SUBROUTINE RRN05(MFSR,P,Q,INTSIZ,JJ)
C      character*8 name,messg,variab,t01nm,tstnm,cknm(10)
C      REAL*8 NAME
C      COMMON /FIFTH/X(10000)
C      COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C      COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C      DATA YES/'YES '/
C      DIMENSION MFSR(98)
C      INTEGER P,Q
C
C      IF(NEWGEN.NE.0) GOTO 10
C
C      IFLAG=0
C      CALL URN05I(MFSR,P,Q,INTSIZ,JJ)
C      WRITE(6,940) P,Q,INTSIZ,JJ
C      940 FORMAT(' ',8X,'P=',I11,' Q=',I11,' INTSIZ=',I11,'
C      JJ=',I11)
C      WRITE(6,990)MFSR
C      990 FORMAT(9X,'MFSR=',/,9(10(1X,I11),/),8(1X,I11))
C      10 IF(TIMES.EQ.YES)GO TO 20
C
C      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C      CALL URN05(MFSR,P,Q,INTSIZ,JJ,IFLAG)
C      GOTO 30
C
C      GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C      20 KTIME=0
C      CALL TIMER(ITIME)
C      CALL URN05(MFSR,P,Q,INTSIZ,JJ,IFLAG)
C      CALL TIMER(ITIME)
C      KTIME=KTIME+ITIME
C
C      30 IF(NEWGEN.NE.0) GOTO 999
C      WRITE(6,400) NBATCH
C      400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
C      *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C      WRITE(6,940) P,Q,INTSIZ,JJ
C      WRITE(6,990) MFSR
C      999 RETURN
C      END

```

```

C
C*****
C   SUBROUTINE URN05I(MFSR,P,Q,INTSIZ,JJ)
C*****
C
C   ESTA SUBRUTINA INICIALIZA URN05.  LAS OPCIONES DISPONIBLES SON:
C
C   IOPT=0  INICIALIZACION POR DEFECTO
C   IOPT=1  INICIALIZACION MEDIANTE LECTURAS DE VALORES EN LA RUTINA
C   IOPT=2  INICIALIZACION DE VALORES PROVISTOS POR EL USUARIO
C   IOPT=3  TODAS LAS INICIALIZACIONES EXCEPTO MFSR PROVISTO POR EL
C           USUARIO
C   IOPT=4  TODAS LAS INICIALIZACIONES EXCEPTO MFSR LEIDO EN LA RUTINA
C
C*****
C
C   SUBROUTINE URN05I(MFSR,P,Q,INTSIZ,JJ)
C   DIMENSION MMFSR(98),MFSR(98)
C   CHARACTER*8 NAME
C   REAL*8 NAME
C   COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C   DATA MMFSR/346256726,591599773,1943131421,1173234223,1776849374,
C   *1119416586,172236044,985756773,1554281477,1503137291,650397619,
C   *1618395655,639939067,1448259547,1046853128,659170036,1034934222,
C   *279813371,326930100,367002640,648480182,1909733845,618563844,
C   *845531267,292262469,299413367,2139821356,1005803337,390139420,
C   *1161028423,2034360736,334070487,565633315,124796253,2104169336,
C   *2009751844,1999687407,83223028,1591328966,646701838,1935362333,
C   *795013136,680356918,1771711842,1324935502,1869840308,356745634,
C   *1061920662,614951490,261876461,703987800,797463948,178239686,
C
C   *1641708282,1539695556,1334926802,940547749,1957646566,1878491364,
C
C   *2033904942,1711106005,2138438575,647734238,1555990485,1210108489,
C   *1793192836,1819829578,751843064,345621400,575445974,1640918761,
C
C   *1379191461,1617832156,542966103,1305854952,1476721677,1466811698,
C   *1842260101,1666639833,217007402,685228354,902087789,32432242,
C   *789712994,702791444,1081111755,1572116899,321512624,644413114,
C   *863989644,1348681739,84379947,1955819746,941474606,984690559,
C   *1794209263,1704575856,1253913135/
C
C   INTEGER DELAY,P,Q,ONE
C
C   IF(IOPT.EQ.0)GO TO 90
C   IF(IOPT.EQ.1)GO TO 70
C   IF(IOPT.EQ.2)GO TO 80
C   IF(IOPT.EQ.4)READ(5,92)P,Q,INTSIZ,DELAY
92  FORMAT(4I11)
C   JJ=0
C   IFLAG=1
C   ONE=2**(INTSIZ-1)
C
C   DO 1 I=1,P
1   MFSR(I)=ONE
C
C   DO 4 K=1,INTSIZ
C   DO 2 J=1,DELAY
2   CALL URN05(MFSR,P,Q,INTSIZ,JJ,IFLAG)
C   KOUNT=0
C   DO 3 I=1,P

```

```

ITEMP=ONE/2**(K-1)
ITEMP=(MFSR(I)-MFSR(I)/ONE*ONE)/ITEMP
IF(ITEMP.EQ.1)KOUNT=KOUNT+1
IF(K.EQ.INTSIZ)GO TO 3
MFSR(I)=MFSR(I)/2+ONE
3 CONTINUE
4 CONTINUE
C
DO 6 I=1,5000
DO 5 J=1,P
5 CALL URN05(MFSR,P,Q,INTSIZ,JJ,IFLAG)
6 CONTINUE
GO TO 80
90 P=98
Q=27
INTSIZ=31
JJ=0
60 DO 20 I=1,98
MFSR(I)=MMFSR(I)
20 CONTINUE
GO TO 80
70 READ(5,75)MFSR,P,Q,INTSIZ,JJ
75 FORMAT(14(7I11,/),4I11)
80 RETURN
END
C
C*****
C SUBROUTINE URN05(MFSR,P,Q,INTSIZ,JJ,IFLAG)
C*****
C
C METODO GENERALIZADO FEEDBACK SHIFT REGISTER (GFSR) DADO POR T.G.
C LEWIS Y W.H. PAYNE, "GENERALIZED FEEDBACK SHIFT REGISTER PSEUDO-
C RANDOM NUMBER ALGORITHM",JOURNAL OF THE ASSOCIATION FOR COMPUTING
C MACHINERY, VOL.20(1973), PAG.456-468,PARA LA GENERACION DE NUMEROS
C ALEATORIOS EN PUNTO FLOTANTE
C
C PROPOSITO
C GENERACION DE NUMEROS ALEATORIOS EN (0.0,1.0)
C
C USO
C "CALL URN05I(MFSR,P,JJ,Q,INTSIZ)" SOLO UNA VEZ EN EL PROGRAMA PRIN-
C CIPAL. DESPUES "CALL URN05(MFSR,P,Q,INTSIZ,JJ,IFLAG)"
C
C DESCRIPCION DE PARAMETROS
C MFSR...ES UNA TABLA DE P NUMEROS ALEATORIOS INICIALES
C P...ES UN EXPONENTE DEL POLINOMIO PRIMITIVO  $X**P+X**Q+1$ 
C Q...ES UN EXPONENTE DEL POLINOMIO PRIMITIVO  $X**P+X**Q+1$ 
C DELAY...ES LA DIFERENCIA RELATIVA, EN BITS, ENTRE COLUMNAS DE
C MFSR(.)
C INTSIZ...ES EL TAMA.¥.O INTERO (EN BITS) DEL COMPUTADOR QUE HA
C SIDO
C UTILIZADO; POR EJEMPLO, 31 PARA LOS SISTEMAS
IBM/360/370
C 48 PARA EL COMPUTADOR CDC 6000, 35 PARA EL COMPUTADOR
SRU
C 1100 Y 15 PARA EL COMPUTADOR HP 2100
C X...ES EL ARREGLO RESULTANTE EN PUNTO FLOTANTE DE SIMPLE
PRECISION
C PARA LOS NUMEROS ALEATORIOS EN (0.0,1.0)
C JJ...ES UN PARAMETRO USADO CUANDO RECOMIENZA LA SUCESSION
C IFLAG...ES UN PARAMETRO QUE ES PASADO DEL PROGRAMA PRINCIPAL

```

```

C          Y USADO POR LA SUBROUTINA , QUE NO DEBE SER CAMBIADO POR
EL
C          USUARIO.
C
C OBSERVACIONES
C      URN05I INICIALIZA LAS VARIABLES MFSR,P,DELAY,Q,INTSIZ PARA
C      EL SISTEMA IBM /370 Y EL POLINOMIO PRIMITIVO X**98+X**27+1;
C      SI EL USUARIO DESEA EMPLEAR SUS PROPIOS VALORES NO DEBERA
INVOCAR
C      "CALL URN05I(MFSR,P,DELAY,Q,INTSIZ)", Y EN TAL CASO INICIALIZAR
C      ESAS VARIABLES.
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C      NINGUNA
C
C METODO
C      REFERENCIAS:
C      1. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C      GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C      COLUMBUS, OHIO, 1981.
C
C*****
C
C      SUBROUTINE URN05(MFSR,P,Q,INTSIZ,JJ,IFLAG)
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /FIFTH/X(10000)
INTEGER*4 AA,BB,LCOMPJ,LCOMPK,AUX1,AUX2
INTEGER A,B,P,Q,MFSR(1)
CHARACTER*8 NAME
C      REAL*8 NAME
EQUIVALENCE (AA,A),(BB,B),(MCOMPJ,LCOMPJ),(MCOMPK,LCOMPK)
C
NDIM=NBATCH
IF(IFLAG.EQ.1)NDIM=1
N=(2**(INTSIZ-1)-1)*2+1
C
DO 10 I=1,NDIM
JJ=JJ+1
IF(JJ.GT.P)JJ=1
K=JJ+Q
IF(K.GT.P)K=K-P
MCOMPJ=N-MFSR(JJ)
MCOMPK=N-MFSR(K)
A=MFSR(K)
B=MFSR(JJ)
AUX1=IAND(LCOMPJ,AA)
AUX2=IAND(LCOMPK,BB)
BB=IOR(AUX1,AUX2)
MFSR(JJ)=B
X(I)=FLOAT(MFSR(JJ))/FLOAT(N)
10 CONTINUE
RETURN
END
C
C
C*****
C
C      SUBROUTINE RRN06
C*****
C
C      ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL

```

```

C GENERADOR URN06.
C EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C A URN06I E IMPRIME LOS VALORES INICIALES.
C URN01 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C QUIERE) E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN01
C DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
C SUBROUTINE RRN06
C CHARACTER*8 NAME
C REAL*8 NAME
C COMMON /FIFTH/X(10000)
C COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
C DATA YES/'YES '/
C IF(NEWGEN.NE.0) GOTO 10
C CALL URN06I
C WRITE(6,990)IX,JX
C
C 10 IF(TIMES.EQ.YES)GO TO 20
C
C GENERA NUMEROS ALEATORIOS SIN CALCULOS DE TIEMPOS
C
C CALL URN06(IX,JX,X,NBATCH)
C GOTO 30
C
C GENERA NUMEROS ALEATORIOS CON CALCULOS DE TIEMPOS
C
C 20 KTIME=0
C CALL TIMER(ITIME)
C CALL URN06(IX,JX,X,NBATCH)
C CALL TIMER(ITIME)
C KTIME=KTIME+ITIME
C
C 30 IF(NEWGEN.NE.0) GOTO 999
C WRITE(6,400) NBATCH
C 400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
C *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C WRITE(6,990) IX,JX
C 990 FORMAT(' ',8X,'IX=',I11,' JX=',I11)
C 999 RETURN
C END
C*****
C SUBROUTINE URN06I = SUBROUTINE URN03I(EN URN03I, VER ENTRY URN06I)
C*****
C
C*****
C SUBROUTINE URN06(IX,JX,X,NBATCH)
C*****
C
C BASADA EN UNA FUNCION DADA POR D. LURIE AND R.L. MASON:
C "EMPIRICAL INVESTIGATION OF SEVERAL TECHNIQUES FOR COMPUTER
C GENERATION OF ORDER STATISTICS",COMMUNICATIONS IN STATISTICS,
C VOL.2(1973), PAG.363-371, PARA GENERAR NUMEROS ALEATORIOS EN PUNTO
C FLOTANTE EN EL PAQUETE IRCRAND (REFERENCIA 1 ), LA FUNCION USADA
C POR ESTE METODO FUE LLAMADA "RN1".
C
C PROPOSITO

```

```

C   GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE
C   PRECISION EN (0.0,1.0)
C
C   USO
C   CALL URN06(IX,JX,X,NBATCH)
C
C   DESCRIPCION DE PARAMETROS
C   IX...ES EL VALOR INICIAL DE UNA SUCESION Y PUEDE SE CUALQUIER
C   ENTERO ENTRE 1 Y 2147483647.(2**31 -1)
C   JX...ES EL VALOR INICIAL DE UNA SUCESION Y PUEDE SER CUALQUIER
C   ENTERO ENTRE 1 Y 2147483647.JX NO DEBER SER IGUAL A IX.
C   X... ES EL ARREGLO RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO
FLOTAN-
C   TE DE SIMPLE PRECISION EN (0.0,1.0)
C   NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA
GENERAR
C   CON EL LLAMADO DE ESTA SUBROUTINA
C
C   OBSERVACIONES
C   IX Y JX NO DEBEN SER ALTERADOS POR EL USUARIO DURANTE LA
EJECUCION
C   DEL PROGRAMA, SALVO QUE EL USUARIO DECIDA REPETIR UNA SUCESION DE
C   NUMEROS ALEATORIOS.
C
C   SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C   METODOD
C   REFERENCIAS :
C   1. DUDEWICZ,E.J.: "IRCRAND-THE OHIO STATE UNIVERSITY RANDOM
C   NUMBER GENERATOR PACKAGE",TECHNICAL REPORT, DEPARTMENT OF
C   STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
C   2. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C   GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C   COLUMBUS, OHIO, 1981.
C
C*****
C
SUBROUTINE URN06(IX,JX,X,NBATCH)
DIMENSION X(NBATCH)
DO 10 I=1,NBATCH
IX=IX*65539
JX=JX*262147
X(I)=0.4656613E-9*FLOAT(IABS(JX+IX))
10 CONTINUE
RETURN
END

C
C
C*****
C   SUBROUTINE RRN07
C
C*****
C
C   ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN07.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN07I E IMPRIME LOS VALORES INICIALES.
C   URN07 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN07
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.

```

```

C
  SUBROUTINE RRN07
    CHARACTER*8 NAME
C    REAL*8 NAME
    COMMON /FIFTH/X(10000)
    COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
    COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
    COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
    COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
    DATA YES/'YES '/
    IF(NEWGEN.NE.0) GOTO 10
    CALL URN07I
    WRITE(6,990)IX
990 FORMAT(' ',8X,'IX=',I11)
    10 IF(TIMES.EQ.YES)GO TO 20

C
C      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C    CALL URN07(IX,X,NBATCH)
C    GOTO 30

C
C      GENERACION DE NUEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C    20 KTIME=0
C    CALL TIMER(ITIME)
C    CALL URN07(IX,X,NBATCH)
C    CALL TIMER(ITIME)
C    KTIME=KTIME+ITIME
C
C    30 IF(NEWGEN.NE.0) GOTO 999
    WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
  * ' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
    WRITE(6,990) IX
999 RETURN
    END

C
C*****
C    SUBROUTINE URN07I = SUBROUTINE URN01I(EN URN01I, ENTRY URN07I)
C    SUBROUTINE URN07 = ASSEMBLER
C*****
C
C
C*****
C    SUBROUTINE RRN08
C*****
C
C    ESTA SUBRUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C    GENERADOR URN08.
C    EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C    A URN08I E IMPRIME LOS VALORES INICIALES.
C    URN08 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C    QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN08
C    DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C
C    SUBROUTINE RRN08
C    CHARACTER*8 NAME
C    REAL*8 NAME
C    COMMON /FIFTH/X(10000)
C    COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT

```

```

COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
DATA YES/'YES '/
IF(NEWGEN.NE.0) GOTO 10
CALL URN08I
IX=524287
WRITE(6,990)IX
990 FORMAT(' ',8X,'IX=',I11)
10 IF(TIMES.EQ.YES)GO TO 20
C
C          GENERACION DE NUMEROS ALEATORIOS SIN CALCULOS DE TIEMPOS
C
C          CALL URN08(IX,X,NBATCH)
C          GOTO 30
C
C          GENERACION DE NUMEROS ALEATORIOS CON CALCULOS DE TIEMPOS
C
20 KTIME=0
C          CALL TIMER(ITIME)
C          CALL URN08(IX,X,NBATCH)
C          CALL TIMER(ITIME)
C          KTIME=KTIME+ITIME
C
30 IF(NEWGEN.NE.0) GOTO 999
WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
*' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
WRITE(6,990) IX
999 RETURN
END
C
C*****
C          SUBROUTINE URN08I = URN01I (ENTRY URN08I EN URN01I)
C*****
C
C*****
C          SUBROUTINE URN08(IX,X,NBATCH)
C*****
C
C          ESTA SUBRUTINA SE BASA EN LA SUBRUTINA DE SSP DE IBM PARA LA
C          GENERACION DE NUMEROS ALEATORIOS ENTEROS Y EN PUNTO FLOTANTE.
C          EN EL PAQUETE IRCCRAND (REFERENCIA 2),LA SUBRUTINA EMPLEADA PARA
C          GENERAR NUMEROS ALEATORIOS EN SIMPLE PRECISION FUE LLAMADA "RN2"
C
C          PROPOSITO
C          GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE
C          PRECISION EN (0.0,1.0), Y NUMEROS ALEATORIOS ENTEROS K , SIENDO
C          0<K<2**31.
C
C          USO
C          CALL URN08(IX,X,NBATCH)
C
C          DESCRIPCION DE PARAMETROS
C          IX...PARA EL PRIMER VALOR, QUE PUEDE SER CUALQUIER NUMERO ENTERO
C          IMPAR ENTRE 1 Y 999999999.
C          LOS VALORES SUCEIVOS DE IX DEBERAN SER DEL CONJUNTO DE
C          VALORES PREVIOS DE IY CALCULADOS POR ESTA RUTINA.
C          X... ES EL VECTOR RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO
C          FLOTANTE DE SIMPLE PRECISION EN (0.0,1.0)
C          NBATCH...ES LA CANTIDAD DE NUMEROS ALETORIOS QUE SE DESEA GENERAR

```



```

C          CON EL LLAMADO DE ESTA SUBROUTINA.
C
C OBSERVACIONES
C  ESTA SUBROUTINA ES ESPECIFICA DE LOS  COMPUTADORES IBM SISTEMAS/360
C    /370.CORRESPONDE A LA SUBROUTINA  RANDU DE LA REFERENCIA 1  .
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C METODO
C  REFERENCIAS:
C   1. INTERNATIONAL BUSINESS MACHINES CORPORATION, "SYSTEM/360
C     SCIENTIFIC SUBROUTINE PACKAGE, VERSION III, PROGRAMMER'S MANUAL
C     PROGRAM NUMBER 360A-CM-03X",MANUAL GH20-0205-4 (FIFTH EDITION)
C     IBM CORPORATION, WHITE PLAINS, NEW YORK, AUGUST 1970, P.77;
C   2. DUDEWICZ,E.J.: "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM
C     NUMBER GENERATOR PACKAGE",TECHNICAL REPORT, DEPARTMENT OF
C     STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
C   3. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C     GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C     COLUMBUS, OHIO, 1981.

```

```

C*****

```

```

C
C      SUBROUTINE URN08(IX,X,NBATCH)
C      DIMENSION X(NBATCH)
C      DO 10 I=1,NBATCH
C      IY=IX*65539
C      IF(IY)5,6,6
5     IY=IY+2147483647+1
6     X(I)=IY
C      X(I)=X(I)*.4656613E-9
C      IX=IY
10    CONTINUE
C      RETURN
C      END

```

```

C
C

```

```

C*****

```

```

C      SUBROUTINE RRN09

```

```

C*****

```

```

C
C  ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C  GENERADOR URN09.
C  EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C  A URN09I E IMPRIME LOS VALORES INICIALES.
C  URN09 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C  QUIERE) E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN09
C  DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.

```

```

C

```

```

C*****

```

```

C
C      SUBROUTINE RRN09
C      CHARACTER*8 NAME
C      REAL*8 NAME
C      COMMON /FIFTH/X(10000)
C      COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C      COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C      COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
C      DATA YES/'YES '/

```

```

        IF(NEWGEN.NE.0) GOTO 10
        CALL URN09I
        WRITE(6,990)IX
990  FORMAT(' ',8X,'IX=',I11)
        10 IF(TIMES.EQ.YES)GO TO 20
C
C          GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C          CALL URN09(IX,X,NBATCH)
C          GOTO 30
C
C          GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C          20 KTIME=0
C             CALL TIMER(ITIME)
C             CALL URN09(IX,X,NBATCH)
C             CALL TIMER(ITIME)
C             KTIME=KTIME+ITIME
C
C          30 IF(NEWGEN.NE.0) GOTO 999
C             WRITE(6,400) NBATCH
400  FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
          *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C             WRITE(6,990) IX
999  RETURN
C             END
C
C*****
C          SUBROUTINE URN09I = SUBROUTINE URN01I (ENTRY URN09I EN URN01I)
C*****
C
C*****
C          SUBROUTINE URN09(IX,X,NBATCH)
C*****
C
C          ESTA SUBRUTINA ES LA SUBRUTINA SSP DE IBM PARA GENERAR NUMEROS
C          ALEATORIOS ENTEROS Y EN PUNTO FLOTANTE.EL CODIGO USADO DIFIERE DEL
C          CODIGO DE SSP(DONDE SE EMPLEA COMO RN2) .EL PRESENTE CODIGO SE
C          ENCUENTRA EN EL LISTADO FUENTE DE N. NIE, D.H. BENT, AND C.H.
C          HULL, "SPSS-STATISTICAL PACKAGE FOR THE SOCIAL SCIENCES", MCGRAW-
C          HILL, INC., NEW YORK, 1970. EN EL PAQUETE IRCCRAND (REF.1), LA
C          SUBRUTINA USADA PARA GENERAR NUMEROS ALEATORIOS EN SIMPLE PRECISION
C          SE DENOMINA "RN4".
C
C          PROPOSITO
C          GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE
C          SIMPLE
C          PRECISION EN (0.0,1.0)
C
C          USO
C          CALL URN09(IX,X,NBATCH)
C
C          DESCRIPCION DE PARAMETROS
C          IX...ES EL VALOR INICIAL Y PUEDE SER CUALQUIER ENTERO IMPAR ENTRE
C          1 AND 999999999.
C          X... ES EL VECTOR RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO
C          FLOTAN-
C          TE DE SIMPLE PRECISION EN (0.0, 1.0)
C          NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA
C          GENERAR
C          CON EL LLAMADO DE ESTA SUBRUTINA.

```

```

C
C OBSERVACIONES
C   IX NO DEBE SER ALTERADO POR EL USUARIO DURANTE LA EJECUCION DEL
C   PROGRAMA, SALVO QUE EL DECIDA REPETIR UNA SUCESION DE NUMEROS
C   ALEATORIOS.
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C METODO
C   REFERENCIAS:
C   1. DUDEWICZ,E.J.: "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM
C   NUMBER GENERATOR PACKAGE",TECHNICAL REPORT, DEPARTMENT OF
C   STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
C   2. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C   GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C   COLUMBUS, OHIO, 1981.
C
C*****
C
C   SUBROUTINE URN09(IX,X,NBATCH)
C   DIMENSION X(NBATCH)
C   DO 10 I=1,NBATCH
C   IY=IX*65539
C   IF(IY.GE.0) GO TO 6
C   IY=IY+2147483647
C   IY=IY+1
C   6   X(I)=FLOAT(IY)*.4656613E-9
C   IX=IY
C   10  CONTINUE
C   RETURN
C   END
C
C
C*****
C   SUBROUTINE RRN10
C*****
C
C   ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN10.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN10I E IMPRIME LOS VALORES INICIALES.
C   URN10 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN10
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C   SUBROUTINE RRN10
C   CHARACTER*8 NAME
C   REAL*8 NAME
C   COMMON /FIFTH/X(10000)
C   COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C   COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C   COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
C   COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
C   DATA YES/'YES '/
C   IF(NEWGEN.NE.0) GOTO 10
C   CALL URN10I
C   WRITE(6,990)IX
C   990 FORMAT(' ',8X,'IX=',I11)
C   10 IF(TIMES.EQ.YES)GO TO 20
C
C

```

```

C          GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C          CALL URN10(IX,X,NBATCH)
C          GOTO 30
C
C          GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C          20 KTIME=0
C          CALL TIMER(ITIME)
C          CALL URN10(IX,X,NBATCH)
C          CALL TIMER(ITIME)
C          KTIME=KTIME+ITIME
C
C          30 IF(NEWGEN.NE.0) GOTO 999
C          WRITE(6,400) NBATCH
C          400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
C          *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C          WRITE(6,990) IX
C          999 RETURN
C          END
C*****
C          SUBROUTINE URN10I = SUBROUTINE URN01I(ENTRY URN10I EN URN01I)
C*****
C
C*****
C          SUBROUTINE URN10(IX,X,NBATCH)
C*****
C
C          SUBROUTINA DESTACADA POR SU PORTABILIDAD SUGERIDA POR BY J.B.
C          KRUSKAL, Y USADA POR OMNITAB 2, PARA GENERAR NUMEROS ALEATORIOS
C          EN PUNTO FLOTANTE. EN EL PAQUETE IRCRAND (REFERENCIA 2 ), LA
C          SUBROUTINA QUE USA ESTE METODO PARA GENERAR NUMEROS ALEATORIOS EN
C          SIMPLE PRECISION SE LLAMA "RN5".
C
C          PROPOSITO
C          GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE
C          SIMPLE
C          PRECISION EN (0.0,1.0)
C
C          USO
C          CALL URN10(IX,X,NBATCH)
C
C          DESCRIPCION DE PARAMETROS
C          IX...PARA EL PRIMER VALOR, QUE DEBE SER PUESTO IGUAL A 1.LOS
C          VALO-
C          RES SUCESIVOS DE IX DEBERAN SER DEL CONJUNTO DE VALORES
C          PREVIOS DE IY CALCULADOS POR ESTA RUTINA.
C          X... ES EL VECTOR RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO
C          FLOTAN-
C          TE DE SIMPLE PRECISION EN (0.0,1.0)
C          NBATCH...ES LA CANTIDAD DE NUMEROS ALETORIOS QUE SE DESEA GENERAR
C          CON EL LLAMADO DE ESTA SUBROUTINA.
C
C          OBSERVACIONES
C          ESTA SUBROUTINA ES GENERAL...(VER REFERENCIA 1)
C
C          SUBROUTINAS Y FUNCIONES REQUERIDAS
C          NINGUNA
C
C          METODO
C          REFERENCIAS:

```

```

C 1. KRUSKAL, J.B. , "EXTREMELY PORTABLE RANDOM NUMBER GENERATOR",
C COMMUNICATIONS OF THE ACM, VOL.12(1969), PAG. 93-94.
C 2. DUDEWICZ, E.J.: "IRCCRAND-THE OHIO STATE UNIVGRSITY RANDOM
C NUMBER GENERATOR PACKAGE", TECHNICAL REPORT, DEPARTMENT OF
C STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
C 3. DUDEWICZ, E.J. AND RALLEY, T.: "THE HANDBOOK OF RANDOM NUMBER
C GENERATION AND TESTING", AMERICAN SCIENCES PRESS,
C COLUMBUS, OHIO, 1981.
C*****
C
C SUBROUTINE URN10(IX,X,NBATCH)
C DATA M,FLM/8192,8192./
C DIMENSION X(NBATCH)
C DO 110 J=1,NBATCH
C K=IX
C DO 10 I=1,3
10 K=MOD(5*K,M)
C X(J)=FLOAT(K)/FLM
C IY=K
C IX=IY
110 CONTINUE
C RETURN
C END
C
C
C*****
C SUBROUTINE RRN11
C*****
C
C ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C GENERADOR URN11.
C EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C A URN11I E IMPRIME LOS VALORES INICIALES.
C URN11 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C QUIERE) E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN11
C DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C SUBROUTINE RRN11
C CHARACTER*8 NAME
C CHARACTER*3 TIMES,YES
C REAL*8 NAME
C COMMON /FIFTH/X(1000)
C COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOP
C COMMON /TSTPIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C COMMON /RPRMS/C,P,Q,ZZ
C COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
C DATA YES/'YES'/
C IF(NEWGEN.NE.0) GOTO 10
C CALL URN11I
C WRITE(6,990)IX,L,C,ZZ
990 FORMAT(' ',8X,'IX=' ,I11,' L=' ,I11,' C=' ,F15.1,'
ZZ=' ,F15.1)
10 IF(TIMES.EQ.YES)GO TO 20
C
C GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C CALL URN11(IX,X,L,C,ZZ,NBATCH)
C GOTO 30
C
C GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS

```

```

C
20 KTIME=0
C   CALL TIMER(ITIME)
   CALL URN11 (IX,X,L,C,ZZ,NBATCH)
C   CALL TIMER(ITIME)
   KTIME=KTIME+ITIME
C
30 IF(NEWGEN.NE.0) GOTO 999
   WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
* ' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
   WRITE(6,990) IX,L,C,ZZ
999 RETURN
   END
C
C*****
C   SUBROUTINE URN11I
C*****
C
C   ESTA SUBRUTINA INICIALIZA URN11. LAS OPCIONES DISPONIBLES SON:
C
C   IOPT=0  INICIALIZACION POR DEFECTO
C   IOPT=1  TODOS LOS VALORES LEIDOS EN LA RUTINA
C   IOPT=2  VALORES DE INICIALIZACION PROVISTOS POR EL USUARIO
C
C*****
C
C   SUBROUTINE URN11I
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /RPRMS/C,P,Q,ZZ
COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKK,L,ML,MK,MMM
CHARACTER*8 NAME
C   REAL*8 NAME
IF(IOPT.EQ.1)GO TO 10
IF(IOPT.EQ.2)GO TO 30
IX=1
L=452807053
C=0.0
ZZ=2.0**31
GO TO 30
10 READ(5,100)IX,L,C2,ZZ2
100 FORMAT(2I11,2F16.0)
30 RETURN
   END
C
C*****
C   SUBROUTINE URN11 (IX,X,L,C,ZZ,NBATCH)
C*****
C
C   ESTA SUBRUTINA GENERA NUMEROS ALEATORIOS EN PUNTO FLOTANTE
C   USANDO UN GENERADOR CONGRUENCIAL MIXTO QUE ES ESPECIFICADO POR
C   EL USUARIO. EN EL PAQUETE IRCCRAND (REFERENCIA 2),UNA SUBRUTINA
C   QUE USA ESTE METODO PARA GENERAR NUMEROS ALEATORIOS EN SIMPLE
C   PRECISION SE DENOMINA "RNCG".
C
C PROPOSITO
C   GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE
SIMPLE
C   PRECISION EN (0.0,1.0)
C
C USO

```

```

C   CALL URN11(IX,X,L,C,ZZ,NBATCH)
C
C DESCRIPCION DE PARAMETROS
C   IX...PARA EL PRIMER VALOR DEBE SER PUESTO A 1.(UN USUARIO QUE
C       ENTIENDE LA OPERACION DE ESTE PROGRAMA PUEDE SEGURAMENTE
C       USAR OTROS VALORES INICIALES)
C   X...ES EL ARREGLO RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO FLO-
C       TANTE DE SIMPLE PRECISION EN (0.0,1.0).
C   L...ES EL ENTERO MULTIPLICATIVO DEL GENERADOR.
C   C...ES LA CONSTANTE ADITIVA DEL GENERADOR. (EL GENERADOR ES
C       DENOMINADO "CONGRUENCIAL MIXTO" SI C ES DISTINTO DE CERO
C       Y "CONGRUENCIAL MULTIPLICATIVO" SI C=0.)
C   ZZ...ES EL MODULO POR EL CUAL LOS NUMEROS SON REDUCIDOS
C   NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA
GENERAR
C
C       CON EL LLAMADO DE ESTA SUBROUTINA.
C
C OBSERVACIONES
C   PARA VER ALGUNAS SELECCIONES HABITUALES DE L,C,ZZ ,REFERENCIA 1
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C METODO
C   REFERENCIAS:
C   1. COVEYOU, R.R. AND MACPHERSON, R.D.: "FOURIER ANALYSIS OF
C       UNIFORM RANDOM NUMBER GENERATORS",JOURNAL OF THE ACM, VOL.14
C       (1967), PAG.100-119.
C   2. DUDEWICZ,E.J.: "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM
C       NUMBER GENERATOR PACKAGE",TECHNICAL REPORT, DEPARTMENT OF
C       STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
C   3. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C       GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C       COLUMBUS, OHIO, 1981.
C
C*****
C
SUBROUTINE URN11(IX,X,L,C,ZZ,NBATCH)
DIMENSION X(NBATCH)
DO 10 I=1,NBATCH
K=IX
Z=L*K+C
Z=AMOD(Z,ZZ)
Z=ABS(Z)
K=Z
X(I)=Z/ZZ
IY=K
IX=IY
10 CONTINUE
RETURN
END
C
C
C*****
C   SUBROUTINE RRN12(IX,L,C2,ZZ2)
C*****
C
C   ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN12.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN12I E IMPRIME LOS VALORES INICIALES.

```

```

C   URN12 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE) E IMPRIME LOS VALORES NECESARIOS PARA REINICIAR URN12
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
SUBROUTINE RRN12(IX,L,C2,ZZ2)
CHARACTER*8 NAME
CHARACTER*3 TIMES,YES
REAL*8 C2,ZZ2
COMMON /FIFTH/X(10000)
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
DATA YES/'YES'/
IF(NEWGEN.NE.0) GOTO 10
CALL URN12I(IX,L,C2,ZZ2)
WRITE(6,990) IX,L,C2,ZZ2
990 FORMAT(9X,'IX=',I11,' L=',I11,' C2=',F15.1,' ZZ2=',F15.1)
10 IF(TIMES.EQ.YES)GO TO 20
C
C           GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
CALL URN12(IX,X,L,C2,ZZ2,NBATCH)
GOTO 30
C
C           GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C
20 KTIME=0
CALL TIMER(ITIME)
CALL URN12(IX,X,L,C2,ZZ2,NBATCH)
CALL TIMER(ITIME)
KTIME=KTIME+ITIME
30 IF(NEWGEN.NE.0) GOTO 999
WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
* ' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
WRITE(6,990) IX,L,C2,ZZ2
999 RETURN
END
C
C*****
SUBROUTINE URN12I(IX,L,C2,ZZ2)
C*****
C
C   ESTA SUBROUTINA INICIALIZA URN12. EL GENERADOR ES UNA
C   VERSION DOBLE PRECISION DE URN11 Y REQUIERE VARIABLES
C   DOBLE PRECISION. LAS OPCIONES DISPONIBLES SON:
C
C   IOPT=0 INICIALIZACION POR DEFECTO
C   IOPT=1 TODOS LOS VALORES LEIDOS EN LA RUTINA
C   IOPT=2 VALORES DE INICIALIZACION PROVISTOS POR EL USUARIO
C
C*****
SUBROUTINE URN12I(IX,L,C2,ZZ2)
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
CHARACTER*8 NAME
REAL*8 ZZ2,C2
IF(IOPT.EQ.1)GO TO 10
IF(IOPT.EQ.2)GO TO 30

```



```

IX=1
L=452807053
C2=0.D0
ZZ2=2147483648.D0
GO TO 30
10 READ(5,600)IX,L,C2,ZZ2
600 FORMAT(2I11,2D16.0)
30 RETURN
END
C
C*****
C   SUBROUTINE URN12(IX,X,L,C2,ZZ2,NBATCH)
C*****
C
C   ESTA SUBROUTINA GENERA NUMEROS ALEATORIOS EN PUNTO FLOTANTE
C   USANDO UN GENERADOR CONGRUENCIAL MIXTO QUE ES ESPECIFICADO POR
C   EL USUARIO. EN EL PAQUETE IRCRAND (REFERENCIA 2),UNA SUBROUTINA
C   QUE USA ESTE METODO PARA GENERAR NUMEROS ALEATORIOS EN SIMPLE
C   PRECISION SE DENOMINA "RNCG".
C
C PROPOSITO
C   GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE
SIMPLE
C   PRECISION EN (0.0,1.0)
C
C USO
C   CALL URN12(IX,X,L,C2,ZZ2,NBATCH)
C
C DESCRIPCION DE PARAMETROS
C   IX...PARA EL PRIMER VALOR DEBE SER PUESTO A 1.(UN USUARIO QUE
C   ENTIENDE LA OPERACION DE ESTE PROGRAMA PUEDE SEGURAMENTE
C   USAR OTROS VALORES INICIALES)
C   LOS VALORES SUCEIVOS DE IX DEBEN TOMARSE DEL CONJUNTO
C   DE LOS VALORES PREVIOS IY CALCULADOS POR ESTA RUTINA
C   X...ES EL ARREGLO RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO FLO-
C   TANTE DE SIMPLE PRECISION EN (0.0,1.0).
C   L...ES EL ENTERO MULTIPLICATIVO DEL GENERADOR.
C   C2...ES LA CONSTANTE ADITIVA DEL GENERADOR. (EL GENERADOR ES
C   DENOMINADO "CONGRUENCIAL MIXTO" SI C2 ES DISTINTO DE CERO
C   Y "CONGRUENCIAL MULTIPLICATIVO" SI C2=0.)
C   ZZ2...ES EL MODULO POR EL CUAL LOS NUMEROS SON REDUCIDOS
C   NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA
GENERAR
C   CON EL LLAMADO DE ESTA SUBROUTINA.
C
C OBSERVACIONES
C   ESTA SUBROUTINA ES UNA VERSION EN DOBLE PRECISION DE URN11 (LOS
C   CALCULOS INTERMEDIOS SON EN DOBLE PRECISION PARA INCREMENTAR LA
C   PRECISION).RETORNA NUMEROS ALEATORIOS EN SIMPLE PRECISON.
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C METODO
C   REFERENCIAS:
C   1. COVEYOU, R.R. AND MACPHERSON, R.D.: "FOURIER ANALYSIS OF
C   UNIFORM RANDOM NUMBER GENERATORS",JOURNAL OF THE ACM, VOL.14
C   (1967), PAG.100-119.
C   2. DUDEWICZ,E.J.: "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM
C   NUMBER GENERATOR PACKAGE",TECHNICAL REPORT, DEPARTMENT OF
C   STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.

```

```

C      3. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C          GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C          COLUMBUS, OHIO, 1981.
C
C*****
C
C      SUBROUTINE URN12(IX,X,L,C2,ZZ2,NBATCH)
C          REAL*8 ZZ2,C2,Z
C          DIMENSION X(NBATCH)
C          DO 10 I=1,NBATCH
C              K=IX
C              Z=L*K+C2
C              Z=DMOD(Z,ZZ2)
C              IF(Z.GT.0.D0)GO TO 12
C              Z=Z+ZZ2
C          12 K=Z
C              X(I)=Z/ZZ2
C              IY=K
C              IX=IY
C          10 CONTINUE
C          RETURN
C          END
C
C*****
C      SUBROUTINE RRN13(X,NBATCH,TIMES)
C          ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C          GENERADOR URN13.
C          EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C          A URN33I E IMPRIME LOS VALORES INICIALES.
C          URN33 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO
C          REQUIERE)
C          E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN13 DESPUES
C          DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C
C          SUBROUTINE RRN13(X,NBATCH,TIMES)
C          INTEGER JEN,LCB2,RSUM,RB1C1,B1,B2,C1,C2
C          DIMENSION X(10000)
C          CHARACTER*3 TIMES,YES
C          COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
C          COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C          COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C          DATA YES/'YES'/
C          IF(NEWGEN.NE.0) GOTO 10
C
C          CALL URN13I
C          WRITE(6,990)INTIN(1),INTIN(2),INTIN(3),INTIN(4),INTIN(5)
C
C          10 IF(TIMES.EQ.YES)GO TO 20
C
C          GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C          CALL URN13(X,NBATCH)
C          GOTO 30
C
C          GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C          20 KTIME=0
C          CALL TIMER(ITIME)
C          CALL URN13 (X,NBATCH)
C          CALL TIMER(ITIME)

```

```

C      KTIME=KTIME+ITIME
C
C      30 IF(NEWGEN.NE.0) GOTO 999
C      WRITE(6,400) NBATCH
C      400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
C      * ' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C      WRITE(6,990) INTIN(1),INTIN(2),INTIN(3),INTIN(4),INTIN(5)
C      990 FORMAT(1X,'INTIN(1)=' ,I15,5X,'INTIN(2)=' ,I15,
C      /1X,'INTIN(3)=' ,I15,
C      *5X,'INTIN(4)=' ,I15,/1X,'INTIN(5)=' ,I15)
C      999 RETURN
C      END
C
C*****
C      SUBROUTINE URN13I
C*****
C      EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION
C      DEFAULT.(IOPT=0)
C      NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
C
C      SUBROUTINE URN13I
C      INTEGER JEN,LCB2,RSUM,RB1C1,B1,B2,C1,C2
C      DIMENSION X(10000)
C      CHARACTER*3 TIMES,YES
C      COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
C      COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C      DATA YES/'YES'/
C
C      IR=0
C      IA=5
C      MULT= 663608941
C      INTIN(1)=524287
C      INTIN(2)=INTIN(1)/32768
C      J = INTIN(1)*65536
C      IF (J.LT.0) J=J+ 2147483647 +1
C      INTIN(3)= J/65536
C      INTIN(4)= MULT/32768
C      J= MULT*65536
C      IF (J.LT.0) J= J+ 2147483647 +1
C      INTIN(5) = J/65536
C      RETURN
C      END
C
C*****
C      SUBROUTINE URN13(X,NBATCH)
C*****
C
C      PROPOSITO
C      GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C      PRECISION EN (0.0,1.0)
C
C      USO
C      CALL URN13(X,NBATCH)
C
C      DESCRIPCION DE PARAMETROS
C      IA.....DIMENSION VECTOR DE NUMEROS ENTEROS
C      IR.....DIMENSION VECTOR DE NUMEROS REALES
C      NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA
C      GENERAR

```

```

C           MEDIANTE ESTA SUBROUTINA
C     INTIN(1),INTIN(2),INTIN(3),INTIN(4),INTIN(5) VALORES ENTEROS
C
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C     NINGUNA
C
C METODO
C     UN GENERADOR CONGRUENCIAL MULTIPLICATIVO
C      $X_{i+1} = 663608941 * X_i \pmod{2^{32}}$ 
C
C REFERENCIAS:
C     AHRENS Y DIETER (1974)
C
C*****
C
C     SUBROUTINE URN13(X,NBATCH)
C     INTEGER JEN,LCB2,RSUM,RB1C1,B1,B2,C1,C2
C     DIMENSION X(10000)
C     CHARACTER*3 TIMES,YES
C
C     COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
C     COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C     COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C     DATA YES/'YES'/
C
C     C1=INTIN(4)
C     C2=INTIN(5)
C     B1=INTIN(2)
C     B2=INTIN(3)
C
C     DO 10 I =1 ,NBATCH
C     J2 = (B2*C1 + B1*C2)*16384
C     IF (J2.LT.0) J2=J2+ 2147483647 + 1
C     RSUM=J2/16384
C     J3= B1*C1*536870912
C     IF (J3.LT.0) J3=J3+ 2147483647 + 1
C     RB1C1 = J3/536870912
C     LCB2 = (C2*B2)/32768
C     J4= ((LCB2 + RSUM + RB1C1 * 32768)* 16384)
C     IF (J4.LT.0) J4=J4+ 2147483647 + 1
C     B1 = J4/16384
C     JEN = B1/4
C     J5= C2*B2*65536
C     IF (J5.LT.0) J5=J5+ 2147483647 + 1
C     B2 = J5/65536
C     X(I) = FLOAT(JEN)/ 32768.0
C 10 CONTINUE
C     INTIN(2)=B1
C     INTIN(3)=B2
C     RETURN
C     END
C
C*****
C     SUBROUTINE RRN14
C*****
C
C     ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C     GENERADOR URN14.
C     EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C     A URN14I E IMPRIME LOS VALORES INICIALES.

```

```

C     URN14 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C     QUIERE) E IMPRIME LOS VALORES NECESARIOS PARA REINICIAR URN14
C     DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
      SUBROUTINE RRN14
        CHARACTER*8 NAME
        CHARACTER*3 TIMES,YES
C     REAL*8 NAME
        INTEGER TABLE(64)
        COMMON /FIFTH/X(10000)
        COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
        COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
        COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
        COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
        DATA YES/'YES' /
        IF(NEWGEN.NE.0) GOTO 10
        CALL URN14I(TABLE)
        WRITE(6,990)IX,JX
        IF(IOPT.EQ.0) GOTO 10
        WRITE(6,992)
992  FORMAT(1X,'NOTA: EL VALOR INICIAL DE IX PROVISTO FUE USADO PARA
        *FORMAR UNA TABLA DE 64 NUMEROS,/1X,ALEATORIOS PARA USO INTERNO.
        *,/1X, EL VALOR IMPRESO DE IX ES EL 65-AVO EN LA SUCESION.')
10   IF(TIMES.EQ.YES)GOTO 20
C
C     GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C     CALL URN14(IX,JX,X,NBATCH, TABLE)
C     GOTO 30
C
C     GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C     20 KTIME=0
C     CALL TIMER(ITIME)
C     CALL URN14(IX,JX,X,NBATCH, TABLE)
C     CALL TIMER(ITIME)
C     KTIME=KTIME+ITIME
C
C     30 IF(NEWGEN.NE.0) GOTO 999
C     WRITE(6,400) NBATCH
C     400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
C     * ' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C     WRITE(6,990) IX,JX
C     990 FORMAT(' ',8X,'IX=',I11,' JX=',I11)
C     999 RETURN
C     END
C
C*****
C     SUBROUTINE URN14I(TABLE)
C*****
C
C     ESTA SUBROUTINA INICIALIZA URN14. LAS OPCIONES DISPONIBLES SON:
C
C     IOPT=0   INICIALIZACION POR DEFECTO
C     IOPT=1   INICIALIZACION DE VALORES, EXCEPTO PARA EL ARREGLO
C     TABLA
C             LEIDA EN LA RUTINA
C     IOPT=2   INICIALIZACION DE VALORES PROVISTOS POR EL USUARIO
C
C*****
C

```

```

SUBROUTINE URN14I(TABLE)
  CHARACTER*8 NAME
C   REAL*8 NAME
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKK,L,ML,MK,MMM
INTEGER TABLE(64)
DIMENSION ITABLE(64)
DATA ITABLE/ 453816693, 859273406, 1973808427, 2059993932,
* 509510129, 583551786, 2052114759, 810618840, 1560248493,
* 1724450006,1084172707, 2001645988, 557869481, 1854755266,
* 1609787967,1517139120, 222050533, 73145838, 1785734939,
* 1310401788,1741957729, 240883034, 1905540151, 126963336,
* 267106845, 247276550, 1192674707, 19119444, 1373938713,
* 254375410,1443045167, 2009808224, 458941525, 1668052766,
* 806675723,1844566700, 1226880721, 1868649354, 726922023,
* 2139021624, 192583565, 1545500470, 737174915, 321051908,
* 479937161,1751223842, 1985910815, 887331344, 1163899845,
* 1077642318, 632497915, 635970652, 1816864577, 133989818,
* 574549527,1837418472, 717428989, 106954342, 879430003,
* 2005340340, 613106937, 435722834, 265606415, 197249728/

C   IF(IOPT.EQ.0) GOTO 10
C   IF(IOPT.EQ.1) GOTO 20
C   IF(IOPT.EQ.2) GOTO 30

C
10 IX=197249728
   JX=0
   DO 12 K=1,64
12 TABLE(K)=ITABLE(K)
   GOTO 30

C
20 READ(5,100) IX,JX
100 FORMAT(2I11)
   DO 15 I=1,64
   TABLE(I)=IX
   IX=266245*IX+453816693
   IF(IX.LT.0) IX=IX+2147483647+1
15 CONTINUE
   GOTO 30

30 RETURN
   END

C
C*****
C   SUBROUTINE URN14(IX,JX,X,NBATCH, TABLE)
C*****
C
C   ESTA SUBRUTINA ES UNA EXPRESION EN FORTRAN DE UNA PARTE DEL
C   PAQUETE IRAND DE NUMEROS ALEATORIOS DESARROLLADO POR EL DR.
C   RAYMOND Y L. ZARLING EN 1971 EN EL MARIETTA COLLEGE, MARIETTA,
C   OHIO.
C
C   PROPOSITO
C   GENERA VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE
C   SIMPLE PRECISION EN (0.0,1.0)
C
C   USO
C   CALL URN14(IX,JX,X,NBATCH, TABLE)
C
C   DESCRIPCION DE PARAMETROS
C   IX...ES EL VALOR INICIAL DE UNA PORCION DEL GENERADOR
C   CONGRUENCIAL

```

```

C   JX...ES EL INDICE PARA EL PROCESO DE BARAJAR EN LA TABLA
C   X...ES EL ARRAY RESULTADO DE NUMEROS ALEATORIOS EN PUNTO
C   FLOTANTE DE SIMPE PRECISION EN (0.0,1.0)
C   NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA
GENERAR
C           EN LA LLAMADA A URN14
C   TABLE...ES UNA TABLA DE 64 INTEROS ALEATORIOS USADOS INTERNAMENTE
C           POR EL GENERADOR.
C
C OBSERVACIONES
C   EL VALOR DE JX ES ASIGNADO A LA MITAD DE LA PALABRA ENTERA
C   EL VALOR ES DIVIDIDO EN 512 PARA CONSTRUIR UN ENTERO ENTRE 0 Y 63
C   Y LUEGO EMPLEADO COMO INDICE EN UNA TABLA DE 64 ENTEROS
ALEATORIOS.
C   EL VALOR JX ES CONVERTIDO EN UN NUMERO EN PUNTO FLOTANTE Y
C   GUARDADO EN EL ARREGLO X DE NUMEROS ALEATORIOS A SER RETORNADO.
C   EL ENTERO IX ES USADO PARA REEMPLAZAR EL ELEMENTO EN LA TABLA
C   USADO PARA PRODUCIR EL NUMERO ALEATORIO.
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C METODO
C   REFERENCIAS:
C   1. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C       GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C       COLUMBUS, OHIO, 1981.
C
C*****
C
SUBROUTINE URN14(IX,JX,X,NBATCH,TABLE)
DIMENSION X(NBATCH)
INTEGER TABLE(64)
INTEGER*2 IY
DATA TWO31 /0.4656613E-9/
DO 10 I=1,NBATCH
IX=266245*IX+453816693
IF(IX.LT.0) IX=IX+2147483647+1
101 JX=10924*JX+6925
JX=JX-32769*(JX/32769)
IF(JX.EQ.32768) GOTO 101
IY=JX/512+1
X(I)=TWO31*FLOAT(TABLE(IY))
TABLE(IY)=IX
10 CONTINUE
RETURN
END
C
C
C*****
C   SUBROUTINE RRN15
C*****
C
C   ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN15.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN15I E IMPRIME LOS VALORES INICIALES.
C   URN15 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN15
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C

```

```

C*****
C
SUBROUTINE RRN15
CHARACTER*8 NAME
CHARACTER*3 TIMES,YES
C REAL*8 NAME
COMMON /FIFTH/X(1000)
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
DATA YES/'YES'/
IF(NEWGEN.NE.0) GOTO 10
C
CALL URN15I
WRITE(6,990)IX
990 FORMAT(' ',8X,'IX=',I11)
C
10 IF(TIMES.EQ.YES)GO TO 20
C
GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
CALL URN15(IX,X,NBATCH)
GOTO 30
C
GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20 KTIME=0
C CALL TIMER(ITIME)
C CALL URN15(IX,X,NBATCH)
C CALL TIMER(ITIME)
KTIME=KTIME+ITIME
30 IF(NEWGEN.NE.0) GOTO 999
WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
*' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
WRITE(6,990) IX
999 RETURN
END
C *****
C SUBROUTINE URN15I= SUBROUTINE URN01I(EN URN01I, ENTRY URN15I)
C
C*****
C SUBROUTINE RRN16
C*****
C
ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C GENERADOR URN16.
C EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C A URN16I E IMPRIME LOS VALORES INICIALES.
C URN16 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C QUIERE)
C E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN16 DESPUES
C DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
SUBROUTINE RRN16
CHARACTER*8 NAME
CHARACTER*3 TIMES,YES
C REAL*8 NAME

```



```

COMMON /FIFTH/X(10000)
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
DATA YES/'YES'/
IF(NEWGEN.NE.0) GOTO 10
CALL URN16I
WRITE(6,990)IX
990 FORMAT(' ',8X,'IX=',I11)
10 IF(TIMES.EQ.YES)GO TO 20
C
C          GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C          CALL URN16(IX,X,NBATCH)
C          GOTO 30
C
C          GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C          20 KTIME=0
C          CALL TIMER(ITIME)
C          CALL URN16(IX,X,NBATCH)
C          CALL TIMER(ITIME)
C          KTIME=KTIME+ITIME
C
C          30 IF(NEWGEN.NE.0) GOTO 999
C          WRITE(6,400) NBATCH
C          400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
C          * ' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C          WRITE(6,990) IX
C          999 RETURN
C          END
C
C*****
C          SUBROUTINE URN16I= SUBROUTINE URN01I(EN URN01I, ENTRY URN16I)
C*****
C
C
C*****
C          SUBROUTINE RRN17
C*****
C
C          ESTA SUBRUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C          GENERADOR URN17.
C          EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C          A URN17I E IMPRIME LOS VALORES INICIALES.
C          URN17 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C          QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN17
C          DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
C          SUBROUTINE RRN17
C          CHARACTER*8 NAME
C          CHARACTER*3 TIMES,YES
C          REAL*8 NAME
C
C          COMMON /FIFTH/X(10000)
C          COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C          COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C          COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C          COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM

```

```

DATA YES/'YES'/
C
IF(NEWGEN.NE.0) GOTO 10
CALL URN17I
WRITE(6,990)IX
990 FORMAT(' ',8X,'IX=',I11)
10 IF(TIMES.EQ.YES)GO TO 20
C
C GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C CALL URN17(IX,X,NBATCH)
C GOTO 30
C
C GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20 KTIME=0
C CALL TIMER(ITIME)
C CALL URN16(IX,X,NBATCH)
C CALL TIMER(ITIME)
C KTIME=KTIME+ITIME
C
30 IF(NEWGEN.NE.0) GOTO 999
WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
*' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
WRITE(6,990) IX
999 RETURN
END
C
C*****
C SUBROUTINE URN17I= SUBROUTINE URN01I(EN URN01I, ENTRY URN17I)
C*****
C
C
C*****
C SUBROUTINE RRN18
C*****
C
C ESTA SUBRUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C GENERADOR URN18.
C EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C A URN18I E IMPRIME LOS VALORES INICIALES.
C URN18 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C QUIERE) E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN18
C DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
C SUBROUTINE RRN18
C CHARACTER*8 NAME
C CHARACTER*3 TIMES,YES
C REAL*8 NAME
C COMMON /FIFTH/X(10000)
C COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
C COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
DATA YES/'YES'/
IF(NEWGEN.NE.0) GOTO 10
CALL URN18I

```

```

WRITE(6,990)IX

990 FORMAT(' ',8X,'IX=',I11)
C
10 IF(TIMES.EQ.YES)GO TO 20
C
C GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C CALL URN18(IX,X,NBATCH)
C GOTO 30
C
C GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20 KTIME=0
C CALL TIMER(ITIME)
C CALL URN18(IX,X,NBATCH)
C CALL TIMER(ITIME)
C KTIME=KTIME+ITIME
C
30 IF(NEWGEN.NE.0) GOTO 999
WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
*' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
WRITE(6,990) IX
999 RETURN
END
C
C*****
C SUBROUTINE URN18I= SUBROUTINE URN01I(EN URN01I, ENTRY URN18I)
C*****
C
C
C*****
C SUBROUTINE RRN19
C*****
C
C ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C GENERADOR URN19.
C EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C A URN19I E IMPRIME LOS VALORES INICIALES.
C URN19 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN19
C DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
C SUBROUTINE RRN19
C CHARACTER*8 NAME
C CHARACTER*3 TIMES,YES
C REAL*8 NAME
C COMMON /FIFTH/X(1000)
C COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOP
C COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
C DATA YES/'YES'/
C IF(NEWGEN.NE.0) GOTO 10
C CALL URN19I
C WRITE(6,990)IX
990 FORMAT(' ',8X,'IX=',I11)
10 IF(TIMES.EQ.YES)GO TO 20

```

```

C
C      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C      CALL URN19(IX,X,NBATCH)
C      GOTO 30
C
C      GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C      20 KTIME=0
C      CALL TIMER(ITIME)
C      CALL URN19(IX,X,NBATCH)
C      CALL TIMER(ITIME)
C      KTIME=KTIME+ITIME
C
C      30 IF(NEWGEN.NE.0) GOTO 999
C      WRITE(6,400) NBATCH
C      400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
C      *' SIGUIENTE A LOS',111/1X,'YA GENERADOS) ES')
C      WRITE(6,990) IX
C      999 RETURN
C      END
C
C*****
C      SUBROUTINE URN19= SUBROUTINE URN01I(EN URN01I, ENTRY URN19)
C*****
C
C
C*****
C      SUBROUTINE RRN20(X,NBATCH,TIMES)
C*****
C
C      ESTA SUBRUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C      GENERADOR URN20.
C      EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C      A URN20I E IMPRIME LOS VALORES INICIALES.
C      URN20 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C      QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN20
C      DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C      SUBROUTINE RRN20(X,NBATCH,TIMES)
C      DIMENSION X(10000)
C      CHARACTER*3 TIMES,YES
C      REAL*8 XR,X0,XM,XU
C      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C      COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOP
C      COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKK,L,ML,MK,MMM
C      DATA YES/'YES'/
C
C      IF(NEWGEN.NE.0) GOTO 10
C
C      CALL URN20I
C
C      10 IF(TIMES.EQ.YES)GO TO 20
C
C      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C      CALL URN20(X,NBATCH)
C      GOTO 30
C
C      GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS

```

```

C
20 KTIME=0
C   CALL TIMER(ITIME)
   CALL URN20 (X,NBATCH)
   WRITE(6,990)IX
C   CALL TIMER(ITIME)
C   KTIME=KTIME+ITIME
C
30 IF(NEWGEN.NE.0) GOTO 999
   WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
* ' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
   WRITE(6,990)IX
990 FORMAT(1X,'IX=',I20)
999 RETURN
   END
C*****
C   SUBROUTINE URN20I= SUBROUTINE URN01I(EN URN01I, ENTRY URN20I)
C*****
C
C*****
C   SUBROUTINE URN20(X,NBATCH)
C*****
C
C   PROPOSITO
C   GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C   PRECISION EN (0.0,1.0)
C
C   USO
C   CALL URN20(X,NBATCH)
C
C   DESCRIPCION DE PARAMETROS
C   XI...SEMILLA NUMERO ENTERO
C   NBATCH....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA
GENE-
C           RAR MEDIANTE ESTA SUBRUTINA
C
C
C   SUBRUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C   METODO
C
C    $X_{i+1} = 2027812808 \pmod{2^{*31} - 1}$ 
C
C   REFERENCIAS:
C   D.C. HOAGLIN (1976)
C
C*****
C
C   SUBROUTINE URN20(X,NBATCH)
C   DIMENSION X(10000)
C   CHARACTER*3 TIMES,YES
C   REAL*8 XR,X0,XM,XU
C   COMMON/ TSPTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C   COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
C   COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKK,L,ML,MK,MMM
C   COMMON /CODE/GCODE
C   DATA YES/'YES'/
C
C   X0=DFLOAT(IX)

```

```

        XM=2147483647D0
        DO 11 I=1,NBATCH
        XR= MOD(2027812808D0*X0,XM)
        X0=XR
C       X(I)=FLOAT(XR/XM)
        X(I)=XR/XM
11      END DO
        IX=INT(X0)
        RETURN
        END

C
C
C*****
C      SUBROUTINE RRN21(NBATCH)
C*****
C
C      ESTA SUBRUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C      GENERADOR URN21.
C      EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C      A URN21I E IMPRIME LOS VALORES INICIALES.
C      URN21 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C      QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN21
C      DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
        SUBROUTINE RRN21(NBATCH)
        CHARACTER*3 TIMES,YES
        COMMON /FIFTH/X(10000)
        COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
        COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
        COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
        COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKK,L,ML,MK,MMM
        DATA YES/'YES'/
        IF(NEWGEN.NE.0) GOTO 10
        CALL URN21I(M1,M2,M3,NBATCH)
10      IF(TIMES.EQ.YES)GOTO 20
C      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
        CALL URN21(M1,M2,M3,X,NBATCH)
        GOTO 30
C      GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
20      KTIME=0
C      CALL TIMER(ITIME)
        CALL URN21(M1,M2,M3,X,NBATCH)
C      CALL TIMER(ITIME)
C      KTIME=KTIME+ITIME
30      IF(NEWGEN.NE.0) GOTO 999
        WRITE(6,400) NBATCH
400     FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
        *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES:')
        WRITE(6,990) M1,M2,M3
990     FORMAT(' ',8X,' M1 =',I11,2X,' ', M2 =',I11,2X,' ', M3 =',I11)
999     RETURN
        END

C
C*****
C      SUBROUTINE URN21I(M1,M2,M3,NBATCH)
C*****
C
C      ESTA SUBRUTINA INICIALIZA AQUELLOS GENERADORES QUE UTILIZAN
C      SOLO UN VALOR INICIAL. LAS OPCIONES DISPONIBLES SON:
C
C      IOPT=0 INICIALIZACION POR DEFECTO

```

```

C      IOPT=1  INICIALIZACION DE VALORES POR LECTURA EN ESTA RUTINA
C      IOPT=2  INICIALIZACION DE VALORES PROVISTOS POR EL USUARIO
C
C      LOS GENERADORES INICIALIZADOS POR ESTA RUTINA SON LOS SIGUIENTES:
C      URN21
C
C      EN ESTA SUBROUTINA SE UTILIZA LA INICIALIZACION DEFAULT:(IOPT=0)
C      NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
C
      SUBROUTINE URN21I(M1,M2,M3,NBATCH)
      M1=32007779
      M2=23717810
      M3=52636370
      RETURN
      END
C*****
C      SUBROUTINE URN21(M1,M2,M3,NBATCH)
C*****
C
C      PROPOSITO
C      GENERAR VECTORES DE NUMEROS ALEATORIOS DE SIMPLE PRECISION
C      EN PUNTO FLOTANTE EN (0.0, 1.0)
C
C      USO
C      CALL URN21(M1,M2,M3,NBATCH)
C
C      DESCRIPTION DE PARAMETROS
C      M1,M2,M3...SEMILLAS DE NUMEROS ENTEROS EN EL INTERVALO (0..99999999)
C      NBATCH...LA CANTIDAD DE NUMEROS ALEATORIOS A SER GENERADOS.
C
C      SUBROUTINAS Y FUNCIONES REQUERIDAS
C      NINGUNA
C
C      METODO
C      EL METODO DE GENERACION DE NUMEROS ALEATORIOS, FUE DADO POR
C      C. GARDNER SWAIN Y MARGUERITE S. SWAIN, "A UNIFORM RANDOM
C      NUMBER GENERATOR THAT IS REPRODUCIBLE, HARDWARE-INDEPENDENT,
C      AND FAST", JOURNAL OF CHEMICAL INFORMATION AND COMPUTER
C      SCIENCES, VOL 20 (FEB. 1980) PAG.56-58.
C
C*****
C
      SUBROUTINE URN21(M1,M2,M3,X,NBATCH)
      DIMENSION X(NBATCH)
      DO 10 I=1,NBATCH
      M4=M1+M2+M3
      IF (M2.LT.50000000) THEN
      M4=M4+1357
      ENDIF
      IF (M4.GE.100000000) THEN
      M4=M4-100000000
      ENDIF
      IF (M4.GE.100000000) THEN
      M4=M4-100000000
      ENDIF
      M1=M2
      M2=M3
      M3=M4
      X(I)=(1E-8)*M3
10    CONTINUE

```

```

RETURN
END
C
C
C
*****
C      SUBROUTINE RRN22(NBATCH)
C*****
C      ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C      GENERADOR URN22.
C      EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C      A URN22I E IMPRIME LOS VALORES INICIALES.
C      URN22 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C      QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN22
C      DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
      SUBROUTINE RRN22(NBATCH)
      COMMON /FIFTH/X(10000)
      COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
      COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
      COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKK,L,ML,MK,MMM
      DOUBLE PRECISION M1
      CHARACTER*3 TIMES,YES
      DATA YES/'YES'/
      IF(NEWGEN.NE.0) GOTO 10
      CALL URN22I(M1,NBATCH)
      WRITE(*,990)M1
10 IF(TIMES.EQ.YES)GOTO 20
C      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
      CALL URN22(M1,X,NBATCH)
      GOTO 30
C      GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
20 KTIME=0
C      CALL TIMER(ITIME)
      CALL URN22(M1,X,NBATCH)
C      CALL TIMER(ITIME)
C      KTIME=KTIME+ITIME
C
30 IF(NEWGEN.NE.0) GOTO 999
      WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
      *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
      WRITE(6,990) M1
990 FORMAT(' ',8X,'M1=',d15.0)
999 RETURN
      END
C*****
C      SUBROUTINE URN22I(M1,NBATCH)
C*****
C
C      EN ESTA SUBROUTINA SE UTILIZA LA INICIALIZACION DEFAULT:(IOPT=0)
C      NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C
      SUBROUTINE URN22I(M1,NBATCH)
      DOUBLE PRECISION M1
      M1= 32007779.0D0
      RETURN
      END
C*****
C      SUBROUTINE URN22(M1,X,NBATCH)

```



```

C*****
C
C   ESTA SUBROUTINA GENERA NUMEROS PSEUDO ALEATORIOS CON EL METODO
C   USADO POR LA BIBLIOTECA MT$RANDOM DE VAX
C
C   PROPOSITO:
C   GENERAR VECTORES DE NUMEROS ALEATORIOS DE SIMPLE PRECISION
C   EN PUNTO FLOTANTE EN (0.0, 1.0)
C
C   USO
C   CALL URN22(M1,NBATCH)
C
C   DESCRIPCION DE PARAMETROS
C   M1...SEMILLA DE NUMERO ENTERO EN EL INTERVALO (0..99999999)
C   NBATCH...LA CANTIDAD DE NUMEROS ALEATORIOS A SER GENERADOS.
C
C   SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C   METODO
C*****
C
C   SUBROUTINE URN22(M1,X,NBATCH)
C   DOUBLE PRECISION XX(10000),S1,SK,M1
C   DIMENSION X(10000)
C   SK=2.D0**32
C   DO 10 I=1,NBATCH
C   S1=69069.0D0*M1+1.0D0
C   S1=MOD(S1,SK)
C   XX(I)=S1*0.23283064D-09
C   M1=S1
C   X(I)=XX(I)
10  CONTINUE
C   RETURN
C   END
C
C
C*****
C   SUBROUTINE RRN28(X,NBATCH,TIMES)
C*****
C
C   ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN28.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN28I.
C   RRN28 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN28
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C
C   SUBROUTINE RRN28(X,NBATCH,TIMES)
C   CHARACTER*8 NAME
C   CHARACTER*3 TIMES,YES
C   REAL*8 NAME
C   DIMENSION X(10000)
C   COMMON /TSTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C   COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
C   COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
C   DATA YES/'YES'/
C   IF(NEWGEN.NE.0) GOTO 10
C
C

```

```

        CALL URN28I(IX,NBATCH)
        WRITE(6,990)IX
C
10 IF(TIMES.EQ.YES)GO TO 20
C
C         GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
        CALL URN28(IX,X,NBATCH)
        GOTO 30
C
C         GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20 KTIME=0
C     CALL TIMER(ITIME)
C     CALL URN28(IX,X,NBATCH)
C     CALL TIMER(ITIME)
C     KTIME=KTIME+ITIME
C
30 IF(NEWGEN.NE.0) GOTO 999
    WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
    *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
    WRITE(6,990)IX
990 FORMAT(1X,'IX=',I20)
999 RETURN
    END
C
C     SUBROUTINE URN28I(IX,NBATCH)
C*****
C
C     EN ESTA SUBROUTINA SE UTILIZA LA INICIALIZACION DEFAULT.(IOPT=0)
C     NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
C
C     SUBROUTINE URN28I(IX,NBATCH)
C     IX=1
C     RETURN
C     END
C
C*****
C     SUBROUTINE URN28(IX,X,NBATCH)
C*****
C
C PROPOSITO
C     GENERAR NUMEROS PSEUDOALEATORIOS EN (0,1) EN PUNTO FLOTANTE CON
C     SIMPLE PRECISION
C
C USO
C     CALL URN28(IX,X,NBATCH)
C
C DESCRIPCION DE PARAMETROS
C     IX.....VALOR INICIAL DEL GENERADOR, PUEDE SER: 1,..., 2**32-2
C     X.....ES EL VECTOR DE LOS NUMEROS ALEATORIOS EN (0.0, 1.0)
C     NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR
C             MEDIANTE ESTA SUBROUTINA
C
C OBSERVACIONES
C     IX NO DEBE SER ALTERADO POR EL USUARIO DURANTE LA EJECUCION DEL
C     PROGRAMA SALVO QUE SE DESEE REPETIR UNA SUCESION DE NUMEROS
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS

```

```

C   NINGUNA
C
C METODO
C   UN GENERADOR CONGRUENCIAL MULTIPLICATIVO
C   CON CONSTANTE MULTIPLICATIVA 16807 Y MODULO 2**31 -1
C   REFERENCIAS:
C SCHRAGE, LINUS. "A MORE PROBABLE FORTRAN RANDOM NUMBER GENERATOR."
C ACM TRANSACTIONS ON MATHEMATICAL SOFTWARE, JUNIO 1979, PAG.132-138.
C*****
C
SUBROUTINE URN28(IX,X,NBATCH)
DIMENSION X(10000)
INTEGER TWO15,TWO16,P
MULT = 16807
TWO15 = 32768
TWO16 = 65536
P = 2147483647
IIX=IX
DO 10 I=1,NBATCH
KXHI=( IIX- MOD(IIX,TWO16) )/TWO16
KXALO=( IIX-KXHI*TWO16 ) *MULT
KLEFTLO=(KXALO-MOD(KXALO,TWO16) )/TWO16
KFHI=KXHI*MULT+KLEFTLO
K=(KFHI-MOD(KFHI,TWO15) )/TWO15
IIX=(( (KXALO-KLEFTLO*TWO16) -P) +(KFHI-K*TWO15)*TWO16 )+K
IF ( IIX.LT.0 ) THEN
IIX=IIX+P
END IF
X(I)=IIX*4.656612875E-10
10 CONTINUE
IX=IIX
RETURN
END

C
C
C*****
C SUBROUTINE RRN30(X,NBATCH,TIMES)
C*****
C
C ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C GENERADOR URN30.
C EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C A URN30I E IMPRIME LOS VALORES INICIALES.
C URN30 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIAR URN30
C DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
SUBROUTINE RRN30(X,NBATCH,TIMES)
DIMENSION X(10000)
CHARACTER*3 TIMES,YES
COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
COMMON/COMUN30/ISTRM,IZ
DATA YES/'YES'/
IF(NEWGEN.NE.0) GOTO 10

CALL URN30I
WRITE(*,990)ISTRM,IZ

10 IF(TIMES.EQ.YES)GO TO 20
C

```

```

C          GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C          CALL URN30(X,NBATCH)
C          GOTO 30
C
C          GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C          20 KTIME=0
C          CALL TIMER(ITIME)
C          CALL URN30 (X,NBATCH)
C          CALL TIMER(ITIME)
C          KTIME=KTIME+ITIME
C
C          30 IF(NEWGEN.NE.0) GOTO 999
C          WRITE(6,400) NBATCH
C          400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
C          *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C          WRITE(6,990)ISTRM,IZ
C          990 FORMAT(1X,'ISTRM=',I20,5X,'IZ=',I20)
C          999 RETURN
C          END
C
C*****
C          SUBROUTINE URN30I
C*****
C
C          EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION
C          DEFAULT.(IOPT=0)
C          NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
C          SUBROUTINE URN30I
C          DIMENSION X(10000)
C          CHARACTER*3 TIMES,YES
C          COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C          COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C
C          COMMON/COMUN30/ISTRM,IZ
C          DATA YES/'YES'/
C          ISTRM =0
C          IZ=0
C          RETURN
C          END
C
C*****
C          SUBROUTINE URN30(X,NBATCH)
C*****
C
C          PROPOSITO
C          GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C          PRECISION EN (0.0,1.0)
C
C          USO
C          CALL URN30(X,NBATCH)
C           $X_{i+1} = 630360016 X_i \pmod{2^{*31} - 1}$ 
C
C          DESCRIPCION DE PARAMETROS
C          ISTRM...SEMILLA NUMERO ENTERO ENTRE 1 Y 21474
C          IZ...VALOR ACTUAL DEL ARREGLO ZRNG(ISTRM) DE 100 POSICIONES
C          NBATCH....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA
C          GENE-
C          RAR MEDIANTE ESTA SUBROUTINA

```

```

C
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C METODO
C
C   REFERENCIAS:
C   MARSE Y ROBERTS (1983)
C
C
C   SUBROUTINE URN30(X,NBATCH)
C   DIMENSION X(10000)
C   INTEGER B2E15,B2E16,HI15,HI31,OVFLOW,ZI
C   DATA MULT1,MULT2 /24112,26143/
C   DATA B2E15,B2E16,MODLUS /32768,65536,2147483647/
C   COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C   COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C   COMMON/COMUN30/ISTRM,IZ
C
C   ISTRM2=ISTRM + 1
C   ZI=NXSEED(ISTRM2)
C   DO 20 I = 1 ,NBATCH
C   HI15=ZI / B2E16
C   LOWPRD=(ZI-HI15*B2E16)*MULT1
C   LOW15=LOWPRD/B2E16
C   HI31= HI15*MULT1 + LOW15
C   OVFLOW = HI31 / B2E15
C   ZI = (((LOWPRD-LOW15*B2E16)-MODLUS)+
C   *(HI31-OVFLOW*B2E15)*B2E16)+OVFLOW
C   IF (ZI.LT.0)ZI=ZI+MODLUS
C   HI15= ZI/B2E16
C   LOWPRD= (ZI-HI15*B2E16)*MULT2
C   LOW15=LOWPRD/B2E16
C   HI31=HI15*MULT2+LOW15
C   OVFLOW= HI31/B2E15
C   ZI = (((LOWPRD-LOW15*B2E16)-MODLUS)+
C   *(HI31-OVFLOW*B2E15)*B2E16)+OVFLOW
C   IF (ZI.LT.0)ZI=ZI+MODLUS
C   XR = (2*(ZI/256)+1)/16777216.0
C   X(I)=XR
20  CONTINUE
C   IZ=ZI
C   ISTRM=ISTRM2
C   RETURN
C   END
C
C   INTEGER FUNCTION NXSEED(ISTRM2)
C   INTEGER SEED
C   DOUBLE PRECISION Z
C   Z=1973272912
C   DO 10 I = 1,ISTRM2
C   Z = DMOD(715.D0*Z, 2147483647.D0)
C   Z = DMOD(1058.D0*Z, 2147483647.D0)
C   Z = DMOD(1385.D0*Z, 2147483647.D0)
10  CONTINUE
C   NXSEED=IDINT(Z)
C   RETURN
C   END
C

```

```

C
C*****
C      SUBROUTINE RRN31(X,NBATCH,TIMES)
C*****
C
C      ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C      GENERADOR URN31.
C      EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C      A URN31I E IMPRIME LOS VALORES INICIALES.
C      URN31 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C      QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN31
C      DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
      SUBROUTINE RRN31(X,NBATCH,TIMES)
      DIMENSION X(10000)
      CHARACTER*3 TIMES,YES
      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
      COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
      COMMON /COMUN3132/IXX
      DATA YES/'YES'/
      IF(NEWGEN.NE.0) GOTO 10

C
      CALL URN31I
      WRITE(*,990)IXX

C
      10 IF(TIMES.EQ.YES)GO TO 20

C
      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS

C
      CALL URN31(X,NBATCH)
      GOTO 31

C
      GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS

C
      20 KTIME=0
      CALL TIMER(ITIME)
      CALL URN31 (X,NBATCH)
      CALL TIMER(ITIME)
      KTIME=KTIME+ITIME

C
      31 IF(NEWGEN.NE.0) GOTO 999
      WRITE(6,400) NBATCH
      400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
      *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
      WRITE(6,990)IXX
      990 FORMAT(1X,'IXX=',I20)
      999 RETURN
      END

C

C      SUBROUTINE URN31I
C*****
C
C      EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION
C      DEFAULT.(IOPT=0)
C      NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
      SUBROUTINE URN31I
      DIMENSION X(10000)
      CHARACTER*3 TIMES,YES
      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM

```

```

COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON /COMUN3132/IXX
DATA YES/'YES'/
IXX =1234567
RETURN
END

C
C*****
C   SUBROUTINE URN31(X,NBATCH)
C*****
C
C   PROPOSITO
C   GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C   PRECISION EN (0.0,1.0)
C
C   USO
C   CALL URN31(X,NBATCH)
C
C   DESCRIPCION DE PARAMETROS
C   IX...SEMILLA NUMERO ENTERO
C   NBATCH....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA
GENE-
C           RAR MEDIANTE ESTA SUBRUTINA
C
C
C   SUBRUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C   METODO
C
C    $X_{i+1} = 2456949 X_i \pmod{2^{24}}$ 
C
C   REFERENCIAS:
C   H. MIYAZAKI (1987)
C
C*****
C
C   SUBROUTINE URN31(X,NBATCH)
C   DIMENSION X(1000)
C   CHARACTER*3 TIMES,YES
C   COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C   COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C   COMMON /COMUN3132/IXX
C   DATA YES/'YES'/
C
C   MM=2**24
C   DO 10 I=1,NBATCH
C   IY= 2456949*IXX
C   IY=MOD(IY,MM)
C   IF(IY)5,6,6
5   IY=(IY+2147483647)+1
6   IY2=IY*128
C   IXX =ABS(IY2/128)
C   XX =FLOAT(IXX)/MM
C   X(I) = XX
10  CONTINUE
C   RETURN
C   END
C
C
C*****

```

```

C      SUBROUTINE RRN32(X,NBATCH,TIMES)
C*****
C
C      ESTA SUBRUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C      GENERADOR URN32.
C      EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C      A URN32I E IMPRIME LOS VALORES INICIALES.
C      URN32 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C      QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN32
C      DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
      SUBROUTINE RRN32(X,NBATCH,TIMES)
      DIMENSION X(10000)
      CHARACTER*3 TIMES,YES
      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
      COMMON /COMUN3132/IXX
      DATA YES/'YES'/
      IF(NEWGEN.NE.0) GOTO 10

C
      CALL URN32I
      WRITE(*,990)IXX

C
      10 IF(TIMES.EQ.YES)GO TO 20

C
      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS

C
      CALL URN32(X,NBATCH)
      GOTO 32

C
      GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS

C
      20 KTIME=0
      CALL TIMER(ITIME)
      CALL URN32 (X,NBATCH)
      CALL TIMER(ITIME)
      KTIME=KTIME+ITIME

C
      32 IF(NEWGEN.NE.0) GOTO 999
      WRITE(6,400) NBATCH
      400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
      *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
      WRITE(6,990)IXX
      990 FORMAT(1X,'IXX=',I20)
      999 RETURN

      END

C
C*****
C      SUBROUTINE URN32I
C*****
C
C      EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION DEFAULT.(IOPT=0)
C      NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
      SUBROUTINE URN32I
      DIMENSION X(10000)
      CHARACTER*3 TIMES,YES
      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
      COMMON /COMUN3132/IXX

```



```

DATA YES/'YES'/
IXX =1234567
RETURN
END
C*****
C   SUBROUTINE URN32(X,NBATCH)
C*****
C
C   PROPOSITO
C   GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C   PRECISION EN (0.0,1.0)
C
C   USO
C   CALL URN32(X,NBATCH)
C
C   DESCRIPCION DE PARAMETROS
C   IX...SEMILLA NUMERO ENTERO
C   NBATCH.....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENE-
C           RAR MEDIANTE ESTA SUBROUTINA
C
C
C   SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C   METODO
C    $X_{i+1} = 3513383 X_i \pmod{2^{25}}$ 
C
C   REFERENCIAS:
C   H. MIYAZAKI (1987)
C
C*****
C
C   SUBROUTINE URN32(X,NBATCH)
C   DIMENSION X(10000)
C   CHARACTER*3 TIMES,YES
C   COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C   COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C   COMMON /COMUN3132/IXX
C   DATA YES/'YES'/
C
C   MM=2**25
C   DO 10 I=1,NBATCH
C   IY= 3513383*IXX
C   IY=MOD(IY,MM)
C   IF(IY)5,6,6
C 5   IY=(IY+2147483647)+1
C 6   IY2=IY*64
C   IXX =ABS(IY2/64)
C   XX =FLOAT(IXX)/MM
C   X(I) =XX
C 10  CONTINUE
C   RETURN
C   END
C
C
C*****
C   SUBROUTINE RRN33(X,NBATCH,TIMES)
C*****
C
C   ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN33.

```

```

C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN33I E IMPRIME LOS VALORES INICIALES.
C   URN33 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN33
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
      SUBROUTINE RRN33(X,NBATCH,TIMES)
      DIMENSION X(10000)
      CHARACTER*3 TIMES,YES
      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
      COMMON /COMUN33/XI
      DATA YES/'YES'/
      IF(NEWGEN.NE.0) GOTO 10

C
      CALL URN33I
      WRITE(*,990)XI

C
10  IF(TIMES.EQ.YES)GO TO 20

C
      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
      CALL URN33(X,NBATCH)
      GOTO 33

C
      GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20  KTIME=0
C     CALL TIMER(ITIME)
C     CALL URN33 (X,NBATCH)
C     CALL TIMER(ITIME)
C     KTIME=KTIME+ITIME
C
33  IF(NEWGEN.NE.0) GOTO 999
      WRITE(6,400) NBATCH
400  FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
      *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
      WRITE(6,990)XI
990  FORMAT(1X,'XI=',G15.10)
999  RETURN
      END

C
C*****
C     SUBROUTINE URN33I
C*****
C
C   EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION DEFAULT.(IOPT=0)
C   NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
      SUBROUTINE URN33I
      DIMENSION X(10000)
      CHARACTER*3 TIMES,YES
      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
      COMMON /COMUN33/XI
      DATA YES/'YES'/
      XI =12345678.0
      RETURN
      END

C

```

```

C      SUBROUTINE URN33(X,NBATCH)
C*****
C
C      PROPOSITO
C      GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C      PRECISION EN (0.0,1.0)
C
C      USO
C      CALL URN33(X,NBATCH)
C
C      DESCRIPCION DE PARAMETROS
C      XI...SEMILLA NUMERO ENTERO
C      NBATCH.....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENE-
C          RAR MEDIANTE ESTA SUBROUTINA
C
C
C      SUBROUTINAS Y FUNCIONES REQUERIDAS
C      NINGUNA
C
C      METODO
C      COMBINA DOS SUCESIONES PARA FORMAR :
C      Xi = Y 2i+2
C      SIENDO:  Y2i+1 = 11257 Y2i (MOD. 10**8 + 7)
C              Y2i+2 = 12553 Y2i+1 (MOD.10**8 +37)
C
C
C      REFERENCIAS:
C      H. MIYAZAKI (1987)
C
C*****
C
C      SUBROUTINE URN33(X,NBATCH)
C      DIMENSION X(10000)
C      CHARACTER*3 TIMES,YES
C      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C      COMMON /COMUN33/XI
C      DATA YES/'YES'/
C
C      XMD1=100000007.0
C      XMD2=100000037.0
C      DO 10 I=1,NBATCH
C      Y1=11257.0*XI
C      Y1=AMOD(Y1,XMD1)
C      IF(Y1)5,6,6
C      5  Y1=(Y1+2147483647)+1
C      6  Y2=12553.0*Y1
C      Y2=AMOD(Y2,XMD2)
C      IF(Y2)7,8,8
C      7  Y2=(Y2+2147483647)+1
C      8  XI=Y2
C      Y2=Y2/XMD2
C      X(I)=Y2
C      10 CONTINUE
C      END
C
C
C*****
C      SUBROUTINE RRN34(X,NBATCH,TIMES)
C*****
C

```

```

C   ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN34.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN34I E IMPRIME LOS VALORES INICIALES.
C   URN34 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN34
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
      SUBROUTINE RRN34(X,NBATCH,TIMES)
      DIMENSION X(10000)
      CHARACTER*3 TIMES,YES
      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN

      COMMON/COMUN3435/KW1,KY1,KZ1,XR1
      DATA YES/'YES'/
      IF(NEWGEN.NE.0) GOTO 10

C
      CALL URN34I
      WRITE(*,990)KW1,KY1,KZ1,XR1

C
      10 IF(TIMES.EQ.YES)GO TO 20
C
      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
      CALL URN34(X,NBATCH)
      GOTO 30

C
      GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
      20 KTIME=0
      CALL TIMER(ITIME)
      CALL URN34 (X,NBATCH)
      CALL TIMER(ITIME)
      KTIME=KTIME+ITIME
C
      30 IF(NEWGEN.NE.0) GOTO 999
      WRITE(6,400) NBATCH
      400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
      *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
      WRITE(6,990)KW1,KY1,KZ1,XR1
      990
      FORMAT(1X,'KW=' ,I15,3X,'KY=' ,I15,3X,'KZ=' ,I15,3X,/1X,'XR=' ,F12.10)
      999 RETURN
      END

C
C*****
C   SUBROUTINE URN34I
C*****
C   EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION DEFAULT.(IOPT=0)
C   NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
      SUBROUTINE URN34I
      DIMENSION X(10000)
      CHARACTER*3 TIMES,YES
      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
      COMMON /COMUN3435/KW1,KY1,KZ1,XR1
      DATA YES/'YES'/
      KW1 =1

```

```

        KY1 =1
        KZ1 =1
        XR1 = 0.453
        RETURN
        END
C
C*****
C      SUBROUTINE URN34(X,NBATCH)
C*****
C
C      PROPOSITO
C      GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C      PRECISION EN (0.0,1.0)
C
C      USO
C      CALL URN34(X,NBATCH)
C
C      DESCRIPCION DE PARAMETROS
C      KW1...SEMILLA NUMERO ENTERO PARA LA SUCESION {Wi}
C      KY1...SEMILLA DE LA SUCESION {Yi}
C      KZ1...SEMILLA DE LA SUCESION {Zi}
C      XR1...VALOR INICIAL REAL PARA EL CALCULO DE INDICES DE LA TABLA
C      NBATCH.....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA
GENE-
C          RAR MEDIANTE ESTA SUBROUTINA
C
C
C      SUBROUTINAS Y FUNCIONES REQUERIDAS
C      NINGUNA
C
C      METODO
C      UTILIZA TRES SUCESIONES :
C       $W_{i+1} = 44 W_i \pmod{2039}$ 
C       $Y_{i+1} = 45 Y_i \pmod{2037}$ 
C       $Z_{i+1} = 41 Z_i \pmod{2003}$ 
C      LA SUCESION {Xi} SE OBTIENE DESPUES DE PERMUTACIONES EN UNA
TABLA
C       $X_i = W_i/2039 + Y_i/2039*2037 + Z_i/2039*2037*2003$ 
C
C      REFERENCIAS:
C      F. SEZGIN (1987)
C
C*****
C
C      SUBROUTINE URN34(X,NBATCH)
C      DIMENSION X(10000)
C      CHARACTER*3 TIMES,YES
C      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C      COMMON /COMUN3435/KW1,KY1,KZ1,XR1
C      DATA YES/'YES'/
C
C      DIMENSION TW(10),TY(10),TZ(10)
C      KW=KW1
C      KY=KY1
C      KZ=KZ1
C      XR=XR1
C      DO 1 I=1,10
C      KW=MOD(44*KW,2039)
C      TW(I)=KW
C      KY=MOD(45*KY,2027)

```

```

        TY(I)=KY
        KZ=MOD(41*KZ,2003)
        TZ(I)=KZ
1  CONTINUE
    DO 2 J=1,NBATCH
        KW=MOD(44*KW,2039)
        KY=MOD(45*KY,2027)
        KZ=MOD(41*KZ,2003)
        IU=1000*XR
        I1=IU/100
        I=IU-I1*100
        I2=I/10
        I3=I-10*I2+1
        I1=I1+1
        I2=I2+1
        IW=TW(I1)
        IY=TY(I2)
        IZ=TZ(I3)
        TW(I1)=KW
        TY(I2)=KY
        TZ(I3)=KZ
        NMBR=MOD(IW,6)+1
        GO TO(10,20,30,40,50,60),NMBR
    10  XR=FLOAT(IW)/2038.0 - FLOAT(IY)/4128988.0 +
        FLOAT(IZ)/8266233976.0
        GO TO 100
    20  XR=FLOAT(IW)/2038.0 - FLOAT(IZ)/4080076.0 +
        FLOAT(IY)/8266233976.0
        GO TO 100
    30  XR=FLOAT(IY)/2026.0 - FLOAT(IZ)/4056052.0 +
        FLOAT(IW)/8266233976.0
        GO TO 100
    40  XR=FLOAT(IY)/2026.0 - FLOAT(IW)/4128988.0 +
        FLOAT(IZ)/8266233976.0
        GO TO 100
    50  XR=FLOAT(IZ)/2002.0 - FLOAT(IW)/4080076.0 +
        FLOAT(IY)/8266233976.0
        GO TO 100
    60  XR=FLOAT(IZ)/2002.0 - FLOAT(IY)/4056052.0 +
        FLOAT(IW)/8266233976.0
    100 X(J)=XR
    2  CONTINUE
        KW1=KW
        KY1=KY
        KZ1=KZ
        XR1=XR
        RETURN
    END
C
C
C*****
C      SUBROUTINE RRN35(X,NBATCH,TIMES)
C*****
C
C      ESTA SUBRUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C      GENERADOR URN35.
C      EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C      A URN35I E IMPRIME LOS VALORES INICIALES.
C      URN35 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C      QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN35
C      DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.

```

```

C*****
SUBROUTINE RRN35(X,NBATCH,TIMES)
DIMENSION X(10000)
CHARACTER*3 TIMES,YES
COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON /COMUN3435/KW1,KY1,KZ1,XR1
DATA YES/'YES'/
IF(NEWGEN.NE.0) GOTO 10

C
CALL URN35I
WRITE(*,990)KW1,KY1,KZ1,XR1

C
10 IF(TIMES.EQ.YES)GO TO 20

C
C GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
CALL URN35(X,NBATCH)
GOTO 30

C
C GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20 KTIME=0
CALL TIMER(ITIME)
CALL URN35 (X,NBATCH)
CALL TIMER(ITIME)
KTIME=KTIME+ITIME

C
30 IF(NEWGEN.NE.0) GOTO 999
WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
*' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
WRITE(6,990)KW1,KY1,KZ1,XR1
990
FORMAT(1X,'KW=',I15,3X,'KY=',I15,3X,'KZ=',I15,3X,/1X,'XR=',F12.10)
999 RETURN
END

C
C*****
C SUBROUTINE URN35I
C*****
C EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION DEFAULT.(IOPT=0)
C NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
SUBROUTINE URN35I
DIMENSION X(10000)
CHARACTER*3 YES
COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON /COMUN3435/KW1,KY1,KZ1,XR1
DATA YES/'YES'/
KW1 =1
KY1 =1
KZ1 =1
XR1=0.0
RETURN
END

C
C*****
C SUBROUTINE URN35(X,NBATCH)
C*****

```

```

C
C PROPOSITO
C GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C PRECISION EN (0.0,1.0)
C
C USO
C CALL URN35(X,NBATCH)
C
C DESCRIPCION DE PARAMETROS
C KW1...SEMILLA NUMERO ENTERO PARA LA SUCESION {Wi}
C KY1...SEMILLA DE LA SUCESION {Yi}
C KZ1...SEMILLA DE LA SUCESION {Zi}
C XR1...VALOR INICIAL REAL
C NBATCH.....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENE-
C RAR MEDIANTE ESTA SUBROUTINA
C
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C NINGUNA
C
C METODO
C UTILIZA CUATRO SUCESIONES :
C  $W_{i+1} = 157 W_i \pmod{32363}$ 
C  $Y_{i+1} = 146 Y_i \pmod{31727}$ 
C  $Z_{i+1} = 142 Z_i \pmod{31657}$ 
C  $X_i = (W_i + Y_i + Z_i - 3) \pmod{32362}$ 
C LUEGO:  $U_i = (X_i + 1) / 32363$ 
C
C REFERENCIAS:
C L' ECUYER P. (1987)
C
C*****
C
C SUBROUTINE URN35(X,NBATCH)
C DIMENSION X(10000)
C CHARACTER*3 TIMES,YES
C COMMON/ TSTTIM/KTIME ,CTIME(1000) ,TIMLOW,LOWSAM
C COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C COMMON /COMUN3435/KW1,KY1,KZ1,XR1
C DATA YES/'YES'/
C
C DIMENSION TW(10),TY(10),TZ(10)
C KW=KW1
C KY=KY1
C KZ=KZ1
C XR=XR1
C DO 2 J=1,NBATCH
C KW=MOD(157*KW,32363)
C KY=MOD(146*KY,31727)
C KZ=MOD(142*KZ,31657)
C XR=MOD(KW+KY+KZ-3,32362)
C XR=(XR+1)/32363
C X(J)=XR
2 CONTINUE
C KW1=KW
C KY1=KY
C KZ1=KZ
C XR1=XR
C RETURN
C END
C

```



```

C
C*****
C   SUBROUTINE RRN36
C*****
C
C   ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN36.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN36I.
C   RRN36 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN36
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C
C   SUBROUTINE RRN36(X,NBATCH,TIMES)
C   INTEGER INTIN,JEN,L16CB2,R20SUM,R5B1C1,B1,B2,C1,C2
C   DIMENSION X(10000)
C   CHARACTER*3 TIMES,YES
C   COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
C   COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C   COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C   DATA YES/'YES'/
C   IF(NEWGEN.NE.0) GOTO 10
C   CALL URN36I
C   WRITE(6,990)INTIN(1),INTIN(2),INTIN(3),INTIN(4),INTIN(5)
C
C   10 IF(TIMES.EQ.YES)GO TO 20
C
C   GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C   CALL URN36(X,NBATCH)
C   GOTO 30
C
C   GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C   20 KTIME=0
C   CALL TIMER(ITIME)
C   CALL URN36 (X,NBATCH)
C   CALL TIMER(ITIME)
C   KTIME=KTIME+ITIME
C
C   30 IF(NEWGEN.NE.0) GOTO 999
C   WRITE(6,400) NBATCH
C   400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
C   *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C   WRITE(6,990)INTIN(1),INTIN(2),INTIN(3),INTIN(4),INTIN(5)
C   990 FORMAT(1X,'INTIN(1)=' ,I15,5X,'INTIN(2)=' ,I15
C   /1X,'INTIN(3)=' ,I15,
C   *5X,'INTIN(4)=' ,I15,/1X,'INTIN(5)=' ,I15)
C   999 RETURN
C   END
C
C*****
C   SUBROUTINE URN36I
C*****
C
C   EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION
C   DEFAULT.(IOPT=0)
C   NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
C   SUBROUTINE URN36I

```

```

      INTEGER INTIN,JEN,L16CB2,R20SUM,R5B1C1,B1,B2,C1,C2
      DIMENSION X(10000)
      CHARACTER*3 TIMES,YES
      COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
      DATA YES/'YES'/
      IA=5
      IR=0
      MULT = 1220703125
      INTIN(1) = 32767
      INTIN(2) = INTIN(1) / 32768
      J= INTIN(1)*65536
      IF(J.LT.0) J = J+2147483647+1
      INTIN(3) = J/65536
      INTIN(4) = MULT/32768
      J = MULT*65536
      IF(J.LT.0) J = J+2147483647+1
      INTIN(5) = J/65536
      RETURN
      END
C
C*****
C      SUBROUTINE URN36
C*****
C
C      PROPOSITO
C      GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
PRECISION EN (0.0, 1.0)
C
C      USO
C      CALL URN36 (X,NBATCH)
C
C      DESCRIPCION DE PARAMETROS
C      IA.....DIMENSION VECTOR DE NUMEROS ENTEROS
C      IR.....DIMENSION VECTOR DE NUMEROS REALES
C      NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR
C      MEDIANTE ESTA SUBROUTINA
C      INTIN(1),INTIN(2),INTIN(3),INTIN(4),INTIN(5) VALORES ENTEROS
C
C
C      SUBROUTINAS Y FUNCIONES REQUERIDAS
C      NINGUNA
C
C      METODO
C      UN GENERADOR CONGRUENCIAL MULTIPLICATIVO
C      CON CONSTANTE MULTIPLICATIVA 5**13 Y MODULO 2**35
C      Xi+1=5**13*Xi (MOD 2**35)
C
C      REFERENCIAS:
C      DUDEWICZ E., KARIAN Z. Y MARSHALL R. , "RANDOM NUMBER GENERATION
C      ON MICROCOMPUTERS', THE SOCIETY FOR COMPUTER SIMULATION,1985,
C      PAG.9-14
C
C*****
      SUBROUTINE URN36 (X,NBATCH)
      INTEGER INTIN,JEN,L16CB2,R20SUM,R5B1C1,B1,B2,C1,C2
      DIMENSION X(10000)
      CHARACTER*3 TIMES,YES
      COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR

```

```

COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
DATA YES/'YES'/
C
C1=INTIN(4)
C2=INTIN(5)
B1=INTIN(2)
B2=INTIN(3)
C
DO 10 I =1 ,NBATCH
J2= (B2*C1 + B1*C2)*2048
IF (J2.LT.0) J2=J2+ 2147483647 + 1
R20SUM=J2/2048
J3= B1*C1*67108864
IF (J3.LT.0) J3=J3+ 2147483647 + 1
R5B1C1 = J3/67108864
L16CB2 = (C2*B2)/32768
J4= ((L16CB2 + R20SUM + R5B1C1 * 32768)* 2048)
IF (J4.LT.0) J4=J4+ 2147483647 + 1
B1 = J4/2048
JEN = B1/32
J5= C2*B2*65536
IF (J5.LT.0) J5=J5+ 2147483647 + 1
B2 = J5/65536
X(I) = FLOAT(JEN)/ 32768.0
10 CONTINUE
INTIN(2)=B1
INTIN(3)=B2
RETURN
END
C
C
C*****
C SUBROUTINE RRN37(X,NBATCH,TIMES)
C*****
C
C ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C GENERADOR URN37.
C EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C A URN37I E IMPRIME LOS VALORES INICIALES.
C URN37 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN37
C DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C SUBROUTINE RRN37(X,NBATCH,TIMES)
C INTEGER INTIN
C DIMENSION X(10000)
C CHARACTER*3 TIMES,YES
C COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
C COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C DATA YES/'YES'/
C IF(NEWGEN.NE.0) GOTO 10
C
C CALL URN37I
C WRITE(6,990)INTIN(1)
C
C 10 IF(TIMES.EQ.YES)GO TO 20
C
C GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C

```

```

CALL URN37(X,NBATCH)
GOTO 30
C
C          GENERACION DE NUMEROS ALEATORIOS  CON CALCULO DE TIEMPOS
C
20 KTIME=0
C      CALL TIMER(ITIME)
      CALL URN37 (X,NBATCH)
C      CALL TIMER(ITIME)
C      KTIME=KTIME+ITIME
C
30 IF(NEWGEN.NE.0) GOTO 999
      WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
* ' SIGUIENTE A LOS ',I11/1X,' YA GENERADOS) ES')
      WRITE(6,990)INTIN(1)
990 FORMAT(1X,'INTIN(1)=' ,I20)
999 RETURN
      END
C
C*****
C      SUBROUTINE URN37I
C*****
C
C      EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION
DEFAULT.(IOPT=0)
C      NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
      SUBROUTINE URN37I
      INTEGER INTIN
      DIMENSION X(10000)
      CHARACTER*3 TIMES,YES
      COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
      DATA YES/'YES'/
      IA=1
      IR=0
      INTIN(1) = 32767
      RETURN
      END
C
C*****
C      SUBROUTINE URN37
C*****
C
C      PROPOSITO
C      GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C      PRECISION EN (0.0,1.0)
C
C      USO
C      CALL URN37 (X,NBATCH)
C
C      DESCRIPCION DE PARAMETROS
C      IA.....DIMENSION VECTOR DE NUMEROS ENTEROS
C      IR.....DIMENSION VECTOR DE NUMEROS REALES
C      NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR
C      MEDIANTE ESTA SUBROUTINA
C      INTIN(1) VALOR ENTERO
C
C

```

```

C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C METODO
C   UN GENERADOR CONGRUENCIAL MULTIPLICATIVO
C   CON CONSTANTE MULTIPLICATIVA 5**13 Y MODULO 2**31
C       Xi+1=5**13*Xi (MOD 2**31)
C
C REFERENCIAS:
C   DUDEWICZ E., KARIAN Z. Y MARSHALL R. , "RANDOM NUMBER GENERATION
C   ON MICROCOMPUTERS', THE SOCIETY FOR COMPUTER SIMULATION,1985,
C   PAG.9-14
C
C*****
SUBROUTINE URN37 (X,NBATCH)
  INTEGER INTIN
  DIMENSION X(10000)
  COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
  COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
  COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
  DATA YES/'YES '/
  J=INTIN(1)
  DO 10 I =1 ,NBATCH
    J=J*1220703125
    IF (J.LT.0) J= J+ 2147483647 + 1
    X(I) = FLOAT(J)* 0.4656613E-9
10  CONTINUE
    INTIN(1)=J
  RETURN
END
C
C
C*****
SUBROUTINE RRN38(X,NBATCH,TIMES)
C*****
C   ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN38.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN38I E IMPRIME LOS VALORES INICIALES.
C   URN38 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE)
C   E IMPRIME LOS VALORES NECESARIOS PARA REINICIAR URN38 DESPUES
C   DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
SUBROUTINE RRN38(X,NBATCH,TIMES)
  INTEGER INTIN
  DIMENSION X(10000)
  CHARACTER*3 TIMES,YES
  COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
  COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
  COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
  DATA YES/'YES '/
  IF(NEWGEN.NE.0) GOTO 10
C
  CALL URN38I
  WRITE(6,990)INTIN(1)
C
10 IF(TIMES.EQ.YES)GO TO 20
C
C   GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS

```

```

C
CALL URN38(X,NBATCH)
GOTO 30

C
C          GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20 KTIME=0
C   CALL TIMER(ITIME)
C   CALL URN38 (X,NBATCH)
C   CALL TIMER(ITIME)
C   KTIME=KTIME+ITIME
C
30 IF(NEWGEN.NE.0) GOTO 999
WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
* ' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
WRITE(6,990) INTIN(1)
990 FORMAT(1X,'INTIN(1)=' ,I20)
999 RETURN
END

C
C*****
C   SUBROUTINE URN38I
C*****
C   EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION DEFAULT.(IOPT=0)
C   NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
SUBROUTINE URN38I
INTEGER INTIN
DIMENSION X(10000)
CHARACTER*3 TIMES,YES
COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
DATA YES/'YES'/
IA=1
IR=0
INTIN(1) = 32767
RETURN
END

C
C*****
C   SUBROUTINE URN38
C*****
C   PROPOSITO
C   GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C   PRECISION EN (0.0,1.0)
C
C USO
C CALL URN38 (X,NBATCH)
C
C DESCRIPCION DE PARAMETROS
C   IA.....DIMENSION VECTOR DE NUMEROS ENTEROS
C   IR.....DIMENSION VECTOR DE NUMEROS REALES
C   NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR
C           MEDIANTE ESTA SUBROUTINA
C   INTIN(1) VALOR ENTERO
C
C
C

```

```

C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C METODO
C   UN GENERADOR CONGRUENCIAL MULTIPLICATIVO
C   CON CONSTANTE MULTIPLICATIVA 5**13 Y MODULO 2**15
C       Xi+1 = 5**13*Xi (MOD 2**15)
C
C   REFERENCIAS:
C   DUDEWICZ E., KARIAN Z. Y MARSHALL R. , "RANDOM NUMBER GENERATION
C   ON MICROCOMPUTERS', THE SOCIETY FOR COMPUTER SIMULATION,1985,
C   PAG.9-14
C
C*****
SUBROUTINE URN38 (X,NBATCH)
INTEGER INTIN
DIMENSION X(10000)
CHARACTER*3 TIMES,YES
COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
DATA YES/'YES'/
J = INTIN(1)
DO 10 I =1 ,NBATCH
J=J*15625
J=J*65536
IF (J.LT.0) J=J+ 2147483647+1
J = J/65536
J=J*15625
J=J*65536
IF (J.LT.0) J=J+ 2147483647+1
J=J/65536
J=J*5
J=J*65536
IF (J.LT.0) J= J+ 2147483647+1
J=J/65536
X(I) = FLOAT(J)/ 32768.0
10 CONTINUE
INTIN(1)=J
RETURN
END
C
C
C*****
SUBROUTINE RRN39(X,NBATCH,TIMES)
C*****
C
C   ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN39.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN39I E IMPRIME LOS VALORES INICIALES.
C   URN39 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE)
C   E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN39 DESPUES
C   DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
SUBROUTINE RRN39(X,NBATCH,TIMES)
DIMENSION X(10000)
DOUBLE PRECISION SEED,PASS,TEMP,XR
CHARACTER*3 TIMES,YES
COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM

```

```

COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
COMMON/COMUN39/SEED,ITOTAL39
DATA YES/'YES' /
IF(NEWGEN.NE.0) GOTO 10
C
CALL URN39I
WRITE(6,990)SEED,ITOTAL39
C
10 IF(TIMES.EQ.YES)GO TO 20
C
C GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
CALL URN39(X,NBATCH)
GOTO 30
C
C GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20 KTIME=0
C CALL TIMER(ITIME)
CALL URN39 (X,NBATCH)
C CALL TIMER(ITIME)
C KTIME=KTIME+ITIME
C
30 IF(NEWGEN.NE.0) GOTO 999
WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
*' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
WRITE(6,990)SEED,ITOTAL39
990 FORMAT(1X,'SEED=',D15.10,/1X, 'ITOTAL39 =',I20)
999 RETURN
END
C
C*****
C SUBROUTINE URN39I
C*****
C
C EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION
DEFAULT.(IOPT=0)
C NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
SUBROUTINE URN39I
DIMENSION X(10000)
DOUBLE PRECISION SEED,PASS,TEMP,XR
CHARACTER*3 TIMES,YES
COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
COMMON/COMUN39/SEED,ITOTAL39
DATA YES/'YES' /
SEED = 2.0 /(1.0+SQRT(5.0))
ITOTAL39 = 0
RETURN
END
C
C*****
C SUBROUTINE URN39(X,NBATCH)
C*****
C
C PROPOSITO
C GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C PRECISION EN (0.0,1.0)
C

```



```

C USO
C CALL URN39(X,NBATCH)
C
C DESCRIPCION DE PARAMETROS
C SEED .....SEMILLA NUMERO REAL SIMPLE PRECISION
C IGUAL A 2/(1+SQRT(5))
C TOTAL39 ...VALOR INICIAL REAL SIMPLE PRECISION
C NBATCH.....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA
GENE-
C RAR MEDIANTE ESTA SUBROUTINA
C
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C NINGUNA
C
C METODO
C UTILIZA EL ALGORITMO
C  $X_i = [X_{i-1} + (i+X_{i-1})\text{SEN}(i)] \pmod{1}$ 
C
C REFERENCIAS:
C REY (1990)
C
C*****
C EN ESTA SUBROUTINA SE UTILIZA LA INICIALIZACION DEFAULT.(IOPT=0)
C NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
C
SUBROUTINE URN39(X,NBATCH)
DIMENSION X(10000)
DOUBLE PRECISION SEED,PASS,TEMP,XR
CHARACTER*3 TIMES,YES
COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON/COMUN39/SEED,ITOTAL39
DATA YES/'YES'/
DO 10 I= 1,NBATCH
PASS=(ITOTAL39+I)*1.0
TEMP=SEED+(PASS+SEED)*DSIN(PASS)
XR=TEMP-INT(TEMP)
IF (XR.LT.0.0) THEN
XR=XR+1.0
ENDIF
SEED=XR
X(I)=XR
10 CONTINUE
ITOTAL39=ITOTAL39+NBATCH
RETURN
END
C
C*****
C SUBROUTINE RRN40(X,NBATCH,TIMES)
C*****
C
C ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C GENERADOR URN40.
C EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C A URN40I E IMPRIME LOS VALORES INICIALES.
C RRN40 TAMBIEN IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR
C URN40 DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUM. ALEATORIOS.
C
C*****

```

```

SUBROUTINE RRN40(X,NBATCH,TIMES)
DIMENSION X(10000)
DOUBLE PRECISION PASS,SEED,TEMP,ABSTEMP,XLOGAR,XR
CHARACTER*3 TIMES,YES
COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON/COMUN40/SEED,ITOTAL40
DATA YES/'YES '/
IF(NEWGEN.NE.0) GOTO 10
C
CALL URN40I
C
10 IF(TIMES.EQ.YES)GO TO 20
C
GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
CALL URN40(X,NBATCH)
GOTO 30
C
GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20 KTIME=0
CALL TIMER(ITIME)
CALL URN40 (X,NBATCH)
CALL TIMER(ITIME)
KTIME=KTIME+ITIME
C
30 IF(NEWGEN.NE.0) GOTO 999
WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
*' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
WRITE(6,990)SEED,ITOTAL40
990 FORMAT(1X,'SEED=',D15.10,/1X,'ITOTAL40 =',I20)
999 RETURN
END
C
C*****
C SUBROUTINE URN40I
C*****
C
C EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION DEFAULT.(IOPT=0)
C NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
SUBROUTINE URN40I
DIMENSION X(10000)
DOUBLE PRECISION PASS,SEED,TEMP,ABSTEMP,XLOGAR,XR
COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON/COMUN40/SEED,ITOTAL40
DATA YES/'YES '/
SEED = 2.0 /(1.0+SQRT(5.0))
ITOTAL40 = 0
RETURN
END
C
C*****
C SUBROUTINE URN40(X,NBATCH)
C*****
C
C PROPOSITO
C GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE

```

```

C   PRECISION EN (0.0,1.0)
C
C   USO
C   CALL URN40(X,NBATCH)
C
C   DESCRIPCION DE PARAMETROS
C   SEED .....SEMILLA NUMERO REAL SIMPLE PRECISION
C   TOTAL40 ...VALOR INICIAL REAL SIMPLE PRECISION
C   NBATCH.....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA
GENE-
C           RAR MEDIANTE ESTA SUBROUTINA
C
C   SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C   METODO
C   UTILIZA TRES SUCESIONES :  $X_i = W_i + Y_i + Z_i \pmod{2^{32}}$ 
C   Y DESPLAZAMIENTOS DE BITS
C
C   REFERENCIAS:
C   MARSAGLIA Y ZAMAN
C
C*****
C
C   SUBROUTINE URN40(X,NBATCH)
C   DIMENSION X(10000)
C   DOUBLE PRECISION PASS,SEED,TEMP,ABSTEMP,XLOGAR,XR
C   COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C   COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C   COMMON/COMUN40/SEED,ITOTAL40
C   DATA YES/'YES '/
C
C   DO 10 I = 1,NBATCH
C   PASS = (ITOTAL40 + I) * 1.0
C   TEMP = SEED + (PASS + SEED)*SIN(PASS)
C   ABSTEMP = ABS(TEMP)
C   IF (ABSTEMP.GT.1.0) THEN
C   XLOGAR = DLOG10(ABSTEMP)
C   XLOGAR = INT(XLOGAR)+1.0
C   TEMP = TEMP / (10 ** XLOGAR)
C   ENDIF
C   XR = TEMP - INT(TEMP)
C   IF (XR.LT.0) THEN
C   XR = XR + 1.0
C   ENDIF
C   SEED = XR
C   X(I) = XR
10  CONTINUE
C   ITOTAL40 = ITOTAL40 + NBATCH
C   RETURN
C   END
C
C*****
C
C*****
C   SUBROUTINE PRINTSTS (GNOMBRE)
C*****
C
C   SUBROUTINA PARA IMPRIMIR RESULTADOS DEL TEST TST01 CON LA
C   OPCION CS2KS2
C   IMPRESION DE UNA TABLA CON RESULTADOS DE LOS 19 TESTS.

```

```

C
C*****
SUBROUTINE PRINTSTS(GNOMBRE)
COMMON/GAMCHI2/TT(19,2)
CHARACTER*5 GNOMBRE
C CALCULAR CANTIDAD DE GAMA APROX. ENTRE 0.50 ,095 Y 0.99
J50=0
J95=0
J99=0
DO 80 I = 1,19
IF (TT(I,2).GE.0.99) THEN
J99=J99+1
ELSE IF (TT(I,2).GE.0.95) THEN
J95=J95+1
ELSE IF (TT(I,2).GE.0.50) THEN
J50=J50+1
ENDIF
80 CONTINUE
J95= J99+J95
J50=J95+J50
C IMPRESION RESULTADOS 19 TESTS Y CANT. PROBAB.
WRITE(*,7890)
7890 FORMAT(/,/,/22X,30('*'))
WRITE(*,123) GNOMBRE
123 FORMAT(25X,'G E N E R A D O R: ',1X,5A)
WRITE(*,5678)
5678 FORMAT(22X,30('*'),/,/)
WRITE(*,101)
101 FORMAT(1X,'T E S T',T32,'T(I)',T38,'P[CHI-CUAD<=T(I)]')
WRITE(*,102)
102 FORMAT(1X,'-----
')
WRITE(*,11)TT(1,1),TT(1,2)
11 FORMAT(1X,'DISTR.UNIF.(TST02)',T30,F6.2,T40,F5.2)
WRITE(*,12)TT(2,1),TT(2,2)
12 FORMAT(1X,'CUPONES D=5(TST04)',T30,F6.2,T40,F5.2)
WRITE(*,13)TT(3,1),TT(3,2)
13 FORMAT(1X,'CUPONES FR(100R)',T30,F6.2,T40,F5.2)
WRITE(*,14)TT(4,1),TT(4,2)
14 FORMAT(1X,'CUPONES D=10',T30,F6.2,T40,F5.2)
WRITE(*,15)TT(5,1),TT(5,2)
15 FORMAT(1X,'GAP ARRIBA MEDIA(TST05)',T30,F6.2,T40,F5.2)
WRITE(*,16)TT(6,1),TT(6,2)
16 FORMAT(1X,'GAP BAJO MEDIA',T30,F6.2,T40,F5.2)
WRITE(*,17)TT(7,1),TT(7,2)
17 FORMAT(1X,'GAP (0.33,0.667)',T30,F6.2,T40,F5.2)
WRITE(*,18)TT(8,1),TT(8,2)
18 FORMAT(1X,'PERMUTACIONES 3S(TST07)',T30,F6.2,T40,F5.2)
WRITE(*,19)TT(9,1),TT(9,2)
19 FORMAT(1X,'PERMUTACIONES 4S',T30,F6.2,T40,F5.2)
WRITE(*,20)TT(10,1),TT(10,2)
20 FORMAT(1X,'PERMUTACIONES 5S',T30,F6.2,T40,F5.2)
WRITE(*,21)TT(11,1),TT(11,2)
21 FORMAT(1X,'POKER (4,4) (TST08)',T30,F6.2,T40,F5.2)
WRITE(*,22)TT(12,1),TT(12,2)
22 FORMAT(1X,'POKER (5,6)',T30,F6.2,T40,F5.2)
WRITE(*,23)TT(13,1),TT(13,2)
23 FORMAT(1X,'POKER (5,4)',T30,F6.2,T40,F5.2)
WRITE(*,24)TT(14,1),TT(14,2)
24 FORMAT(1X,'CORRIDAS R(TST09)',T30,F6.2,T40,F5.2)
WRITE(*,25)TT(15,1),TT(15,2)

```

```

25  FORMAT(1X,'CORRIDAS FR(10R)',T30,F6.2,T40,F5.2)
    WRITE(*,26)TT(16,1),TT(16,2)
26  FORMAT(1X,'CORRIDAS FR(100R)',T30,F6.2,T40,F5.2)
    WRITE(*,27)TT(17,1),TT(17,2)
27  FORMAT(1X,'PARES SERIALES 3X3(TST10)',T30,F6.2,T40,F5.2)
    WRITE(*,28)TT(18,1),TT(18,2)
28  FORMAT(1X,'PARES SERIALES 10X10',T30,F6.2,T40,F5.2)
    WRITE(*,29)TT(19,1),TT(19,2)
29  FORMAT(1X,'PARES SERIALES 20X20',T30,F6.2,T40,F5.2)
    WRITE(*,102)
    WRITE(*,300)J99
300 FORMAT(1X,'CANTIDAD DE G`= P[CHI-CUAD<= T(I)]>= 0.99 ',2X,I6)
    WRITE(*,301)J95
301 FORMAT(1X,'CANTIDAD DE G`= P[CHI-CUAD<= T(I)]>= 0.95 ',2X,I6)
    WRITE(*,302)J50
302 FORMAT(1X,'CANTIDAD DE G`= P[CHI-CUAD<= T(I)]>= 0.50 ',2X,I6)
    RETURN
    END

C
C*****
C          F U N C I O N E S
C          CDFKS - CHIDF - DCDFN - DCDFNI
C*****
C
C*****
C          FUNCTION CDFKS(A)
C*****
C ESTE SUBPROGRAMA CALCULA LAS PROBABILIDADES PARA UNA VARIABLE
C ALEATORIA DE KOLMOGOROV-SMIRNOV.
C PROPOSITO:
C   CALCULAR LA PROBABILIDAD DE QUE X SEA MENOR QUE A, DONDE X ES UNA
C VARIABLE ALEATORIA KOLMOGOROV-SMIRNOV, PARA USAR EN TST01,TST06 Y
C TST11
C USO:
C   " P = CDFKS(A) "
C DESCRIPCION Y PARAMETROS:
C   A...ES EL PUNTO PARA EL CUAL LA PROBABILIDAD P(X<A) ES
C   CALCULADA
C METODO:
C   LA PROBABILIDAD DESEADA ESTA DADA POR:
C    $P(X<A)=COEFF*(EXP(-1*C)+EXP(-9*C)+...+EXP(-(2K-1)**2)*C)$ 
C   DONDE
C    $COEFF=SQRT(2*PI)/A$ 
C    $C=(PI**2)/(8*(A**2))$ 
C    $PI=3.1415926$ 
C   Y LA SUMATORIA ES TERMINADA CUANDO EL SUMANDO ES MENOR QUE EXP(-
C 20).
C
C REFERENCIA:
C   1. FELLER,W.: "ON THE KOLMOGOROV-SMIRNOV THEOREMS",ANNALS OF
C   MATHEMATICAL STATISTICS, VOL.19(1948), PP.177-189.
C*****

C
C          FUNCTION CDFKS(VALUE)
C          PI=3.141592654
C          CONST=(PI**2)/(8*(VALUE**2))
C          COEFF=SQRT(2*PI)/VALUE
C          SUM=0

C
C   DO 1 I=1,100

```

```

        POWER=((2*I-1)**2)*CONST
        IF(POWER.GT.20) GOTO 2
    1 SUM=SUM+EXP(-POWER)
C
    2 CDFKS=COEFF*SUM
C
        RETURN
        END
C
C*****
C
C        FUNCTION CHIDF(A,V)
C
C*****
C ESTA FUNCION CALCULA PROBABILIDADES PARA VARIABLES ALEATORIAS CHI
C CUADRADO CON "V" GRADOS DE LIBERTAD, V=1,2,...,21.
C
C PROPOSITO
C GENERAR LA PROBABILIDAD DE QUE "X" SEA MENOR O IGUAL QUE "A",
C DONDE X ES UNA VARIABLE ALEATORIA CHI-CUADRADO
C CON V GRADOS DE LIBERTAD.
C
C USO
C " P = CHIDF(A,V) "
C
C DESCRIPCION DE PARAMETROS
C A...ES EL PUNTO PARA EL CUAL SE DESEA QUE LA PROBABILIDAD DE
C X SEA MENOR O IGUAL QUE A
C V...SON LOS GRADOS DE LIBERTAD PARA DICHA PROBABILIDAD
C
C OBSERVACIONES
C ES UNA FUNCION EN DOBLE PRECISION Y DEBE OPERAR CON UN ARGUMENTO
C DOBLE PRECISION.(CUANDO "A" NO LO ES PUEDEN PRODUCIRSE ERRORES
C EN LA RUTINA QUE LA INVOCA)
C LA EXACTITUD DE ESTE SUBPROGRAMA FUE VERIFICADO PARA V=1,2,...,21
C LO CUAL ES SUFICIENTE PARA EL USO QUE SE HACE EN ESTE PAQUETE.
C (ES DECIR QUE LA FUNCION ES LLAMADA POR LOS PUNTOS EN ESE RANGO
DE V
C Y LOS VALORES DE LA FUNCION DE DISTRIBUCION SON USADOS EN UN
PROCESO
C DENOMINADO "PINCH" PARA HALLAR LOS PUNTOS PORCENTUALES DE CHI
CUADRADO
C CADA 0.000001). LA EXACTITUD NO FUE EVALUADA PARA V>21.
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C DCDFN
C
C METODO
C FORMULAS DE P.941, REFERENCIA 2.
C
C REFERENCIAS:
C 1. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C GENERATION AND TESTING," AMERICAN SCIENCES PRESS,
C COLUMBUS, OHIO, 1981.
C 2. ZELEN, M. AND SEVERO, N.C.: "PROBABILITY FUNCTIONS,"
C HANDBOOK OF MATHEMATICAL FUNCTIONS WITH FORMULAS, GRAPHS, AND
C MATHEMATICAL TABLES (EDITED BY M. ABRAMOWITZ AND I.A. STEGUN),
C NATIONAL BUREAU OF STANDARDS, APPLIED MATHEMATICS SERIES, VOL.
C 55, JUNE 1964 (FOURTH PRINTING, DECEMBER 1965, WITH
C CORRECTIONS), CHAPTER 26, PP. 925-995.
C

```

```

C*****
C
      REAL*8 FUNCTION CHIDF(A,V)
      REAL*8  SUM,B,A,EE,DCDFN
      INTEGER V
      IF(MOD(V,2).EQ.0) GO TO 120
      IF(A.GT.0) GO TO 105
      CHIDF=0.D0
      RETURN
105  IF(V.NE.1) GO TO 107
      CHIDF=2.D0*DCDFN(DSQRT(A))-1.D0
      RETURN
107  SUM=0.D0
      LIM2=(V-1)/2
      B=1.D0/DSQRT(A)
      DO 110 I=1,LIM2
      B=B*(A/(2*I-1))
      SUM=SUM+B
110  CONTINUE
      EE=DSQRT(2.D0/3.1415926535897932D0)*DEXP(-A/2.D0)
      CHIDF=2.D0*DCDFN(DSQRT(A))-1.D0-EE*SUM
      RETURN
120  IF(V.NE.2) GO TO 122
      CHIDF=1.D0-DEXP(-A/2.D0)
      RETURN
122  SUM=0.D0
      LIM2=(V-2)/2
      B=1.D0
      DO 210 I=1,LIM2
      B=B*(A/(2*I))
      SUM=SUM+B
210  CONTINUE
      EE=DEXP(-A/2.D0)
      CHIDF=1.D0-EE*(1.D0+SUM)
      RETURN
      END
C
C*****
C      FUNCTION DCDFN(X)
C
C*****
C
C PROPOSITO
C SUBPROGRAMA PARA CALCULAR PROBABILIDADES DE VARIABLES NORMALES
C ESTANDAR.
C ESTA FUNCION GENERA LA PROBABILIDAD DE QUE Z SEA MENOR O IGUAL
C QUE X, SIENDO Z UNA VARIABLE NORMAL ESTANDAR.
C
C USO
C      " P = DCDFN(X) "
C
C DESCRIPCION DE PARAMETROS
C      X...ES EL PUNTO PARA EL CUAL SE DESEA QUE LA PROBABILIDAD DE
C          Z SEA MENOR O IGUAL QUE X
C
C OBSERVACIONES
C      ES UNA FUNCION DOBLE PRECISION QUE SE AJUSTA A UN ARGUMENTO
C      EN DOBLE PRECISION
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C      NINGUNA

```

```

C
C METODO
C   PARA EL ALGORITMO EMPLEADO Y SU EXACTITUD, REFERENCIA 2.
C   COMPARACIONES CON OTROS ALGORITMOS Y VELOCIDAD, REFERENCIA 1.
C
C REFERENCIAS
C 1. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C   GENERATION AND TESTING," AMERICAN SCIENCES PRESS,
C   COLUMBUS, OHIO, 1981.
C 2. MILTON,R.C. AND HOTCHKISS,R.: "COMPUTER EVALUATION OF THE
C   NORMAL AND INVERSE NORMAL DISTRIBUTION FUNCTIONS,"
C   TECHNOMETRICS, VOL.11(1969), PP. 817-822.
C
C*****

```

```

C
C   DOUBLE PRECISION FUNCTION DCDFN(X)
C   DIMENSION A(6),D(6),C(6,10)
C   REAL*8 A,C,D,X,Y,Z,SGNY
C   DATA A/.625D0,1.25D0,2.0D0,2.45D0,3.5D0,4.62D0/
C   DATA D/.3D0,.925D0,1.625D0,2.225D0,2.95D0,4.15D0/
C   DATA C/6.7982403291D-04,-1.2709753598D-03,6.7964525797D-04,
C   *-8.8500314297D-05,1.4791554152D-07,5.4879072878D-07,
C   *5.1028640888D-03,-1.4822649744D-03,-3.4589883732D-04,
C   *5.6826070991D-04,-6.1965013125D-05,-8.6656125337D-07,
C   *-6.0160862379D-03,8.7718965604D-03,-3.0883401043D-03,
C   *-4.5015531032D-04,2.2266710841D-04,6.6877769441D-07,
C   *-2.2725611373D-02,-2.0200415601D-03,7.2141082947D-03,
C   *-1.3872389725D-03,-4.4390018986D-04,-6.4084653798D-07,
C   *3.3555772127D-02,-3.4681674488D-02,-1.0677039440D-03,
C   *5.5422185916D-03,6.2978750848D-04,6.2471782478D-07,
C   *7.2703311825D-02,4.7642403207D-02,-2.4859750804D-02,
C   *-1.0221129286D-02,-6.6382630882D-04,-4.2105004094D-07,
C   *-1.4093895854D-01,5.6850293056D-02,5.7420315819D-02,
C   *1.1850195831D-02,5.1265678121D-04,2.0742133302D-07,
C   *-1.5468912970D-01,-2.2180615138D-01,-6.5383705170D-02,
C   *-8.8864030978D-03,-2.7657162532D-04,-7.6856040218D-08,
C   *5.1563045460D-01,2.3979043073D-01,4.0236129479D-02,
C   *3.9938885701D-03,9.3751413487D-05,1.8722020822D-08,
C   *1.6431337971D-01,4.0458836515D-01,4.8922186659D-01,
C   *4.9917416726D-01,4.9998489849D-01,4.9999999779D-01/
C   Y=X*0.70710678119D0
C   SGNY=1.D0
C   IF(Y)2,1,3
1  DCDFN=.5D0
C   RETURN
2  SGNY=-1.D0
C   Y=-Y
3  DO 4 I=1,6
C   IF(Y-A(I))5,5,4
4  CONTINUE
C   Z=.5D0
C   GO TO 7
5  Y=Y-D(I)
C   Z=C(I,1)
C   DO 6 J=2,10
6  Z=Z*Y+C(I,J)
7  DCDFN=.5D0+SGNY*Z
C   RETURN
C   END
C
C*****

```



```

C      FUNCTION DCDFNI(P)
C
C*****
C
C PROPOSITO
C   SUBPROGRAMA PARA CALCULAR PROBABILIDADES DE VARIABLES NORMALES
C   ESTANDAR
C   ESTA FUNCION GENERAL100 PUNTOS PORCENTUALES DE UNA VARIABLE NORMAL
C   ESTANDAR.
C
C USO
C   " X = DCDFNI(P) "
C
C DESCRIPCION OF PARAMETROS
C   P...ES LA PROPORCION PARA LA CUAL EL PUNTO PORCENTUAL 100P DE
C   UNA DISTRIBUCION NORMAL ESTANDAR ES DESEADO
C
C OBSERVACIONES
C   ES UNA FUNCION DOBLE PRECISION QUE SE AJUSTA A UN ARGUMENTO
C   EN DOBLE PRECISION
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   DCDFN
C
C METODO
C   PARA EL ALGORITMO EMPLEADO Y SU EXACTITUD, REFERENCIA 2.
C   COMPARACIONES CON OTROS ALGORITMOS Y VELOCIDAD, REFERENCIA 1.
C   1. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C   GENERATION AND TESTING," AMERICAN SCIENCES PRESS,
C   COLUMBUS, OHIO, 1981.
C   2. MILTON,R.C. AND HOTCHKISS,R.: "COMPUTER EVALUATION OF THE
C   NORMAL AND INVERSE NORMAL DISTRIBUTION FUNCTIONS,"
C   TECHNOMETRICS, VOL.11(1969), PP. 817-822.
C
C*****
C
C      DOUBLE PRECISION FUNCTION DCDFNI(P)
C      REAL*8 X,Q,T,R,P,DCDFN
C      R=P
C      NN=1
C      Q=1.D0-P
C      IF(R)1,3,4
C 1  WRITE(6,2)P
C 2  FORMAT('0','ILLEGAL ARGUMENT IN DCDFNI. P=',D20.10)
C      STOP
C 3  DCDFNI=-7.D0
C      RETURN
C 4  IF(1.D0-R)1,5,6
C 5  DCDFNI=7.D0
C      RETURN
C 6  IF(R-.5D0)9,7,8
C 7  DCDFNI=0.D0
C      RETURN
C 8  R=1.D0-R
C      NN=2
C      Q=P
C 9  IF(R-1.D-10)10,11,11
C10  X=6.41D0
C      GO TO 14
C11  T=DSQRT(DLOG(1.D0/(R*R)))

```

```

      X=T-
      ((.010328D0*T+.802853D0)*T+2.515517D0)/(((.001308D0*T+.189269D
      10)*T+1.432788D0)*T+1.D0)
      12 DO 13 I=1,2
      13 X=X-(DCDFN(X)-Q)/(.3989422804D0*DEXP(-.5D0*X*X))
      14 GO TO (15,16),NN
      15 DCDFNI=-X
      RETURN
      16 DCDFNI=X
      RETURN
      END
C*****FIN GENTSTPC1 *****
C*****12960 LINEAS CODIGO*****

```

```

PROGRAM GENTSTPC2
C   Gentstpc2 corre en PC con sistemas operativos WINDOWS 9*
C   COMPILADOR:Micro Soft Developer Studio-FORTRAN POWER STATION 4.0
C   Autor: ANGEL ADOLFO OLMOS CERVANTES - 2005

CHARACTER*8 NAME
CHARACTER*12 ARCHURN
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT,ARCHURN
write(*,*)'Ingresar(mayusculas)el nombre, URN**,del Generador de
*seado para obtener el archivo ARCHURN** de NumerosPseudoaleator
*ios'
C   AQUI SE GENERARAN NLOT SUCESIONES DE NUMALE(NBATCH) CADA UNA
READ(*,900)NAME
900  FORMAT(A5)
WRITE(*,9001)NAME
9001  FORMAT(1X,A5)
C
C*****
C
CALL TESTER
C
C*****
C
WRITE(*,5161)
5161  FORMAT(1X/)
WRITE(6,5162)ARCHURN
5162  FORMAT(1X,'LA EJECUCION HA FINALIZADO',1X//' EL ARCHIVO:',A12,
*'CONTIENE EL TOTAL DE NUMEROS PSEUDOALEATORIOS GENERADOS')
WRITE(*,*)
PAUSE
END
C
C*****
C   SUBROUTINA TESTER
C*****
C "TESTER" ES LA RUTINA USADA POR GENTSTPC PARA VERIFICAR LA CALIDAD
C ESTADISTICA DE LOS NUMEROS PSEUDOALEATORIOS GENERADOS.
C TESTER PUEDE SER APLICADA A UNO DE LOS GENERADORES PROVISTOS POR
C GENTSTPC (URN01-URN40) O AL PROPIO GENERADOR (SUBRUTINA "URNWN")
C LAS RUTINAS PUEDEN SER LLAMADAS EN FORMA INDIVIDUAL, POR LO QUE EL
C USUARIO PUEDE USARLAS DE DISTINTAS FORMAS NO ESTANDAR CON UN MINIMO
C DE REPROGRAMACION. PARA ESTE PROPOSITO, LOS BLOQUES "COMMON" DEBEN
C USARSE COMO SE ESPECIFICA.
C PROPOSITO:
C   GENERACION Y VERIFICACION DE NUMEROS ALEATORIOS
C USO:
C   PARA USAR LAS FACILIDADES PROVISTAS POR LA SUBRUTINA "TESTER", EL
C   PROGRAMA PRINCIPAL DEBE CONTENER LA INSTRUCCION "CALL TESTER".
C   IPRMS   TESTER, REPORT, RRN01, RRN03, RRN04, RRN06, RRN07,
C           RRN08, RRN09, RRN10, RRN11, RRN13, RRN14, RRN15,
C           RRN16, RRN17, RRN18, RRN19, RRN20, URN01I, URN03I,
C           URN04I
C SUBRUTINAS Y FUNCIONES REQUERIDAS:
C   RRN01, RRN02, RRN03,
C   RRN04, RRN05, RRN06, RRN07, RRN08, RRN09, RRN10, RRN11, RRN12,

```

```

C   RRN13, RRN14, RRN15, RRN16, RRN17, RRN18, RRN19, RRN20, RRNWN,
C *****
C

```

```

SUBROUTINE TESTER
  CHARACTER*8 NAME,MESSG,BL, LASTGN,URNM(41)
  CHARACTER*12 ARCHURN
  REAL*8 XEXT,XDBL,XDOUBLE(10000)
  REAL*8 C2,ZZ2,SEED
  INTEGER GCODE

COMMON /MESS/MESSG(9),VARIAB(9)
COMMON /SECOND/XDBL(2500)
COMMON /THIRD/XEXT(1250)
COMMON /FOURTH/Y(10000)
COMMON /FIFTH/X(10000)
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT,ARCHURN
COMMON /T01PRM/NCHI,NPRNT
COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /CODE/GCODE
COMMON /RPRMS/C,P,Q,ZZ
COMMON /RPMDBL/C2,ZZ2
COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKK,L,ML,MK,MMM
COMMON /TSTPRM/IPARM(5)
COMMON /GENARR/MFSR(98),NN(128),ITABLE(128)
COMMON /COMUN30/ISTRM,IZ
COMMON /COMUN3132/IXX
COMMON /COMUN33/XI
COMMON /COMUN3435/KW1,KY1,KZ1,XR1
COMMON /COMUN39/SEED,ITOTAL39

```

```

C
C
C
C   EL VECTOR URNM(41)CONTIENE CARACTERES ALFANUMERICOS QUE SON NOMBRES
C   DE GENERADORES. LOS GENERADORES SE DENOMINAN ASI:
C       URN01...GENERADOR BASICO LLRANDOM ("RANDOM" EN IRCCRAND)
C       URN02...GENERADOR MARSAGLIA-BRAY ("RN3" EN IRCCRAND)
C       URN03...GENERADOR SUPER-DUPER DE MCGILL ("UNI" EN IRCCRAND)
C       URN04...GENERADOR SHUFFLED LLRANDOM ("SRAND"EN IRCCRAND)
C       URN05...GENERADOR GFSR DE LEWIS-PAYNE (NO ESTA EN IRCCRAND)
C       URN06...GENERADOR LURIE-MASON ("RN1" EN IRCCRAND)
C       URN07...RANDU EN ASSEMBLER ("KERAND" EN IRCCRAND)
C       URN08...RANDU COMO EN SSP DE IBM ("RN2" EN IRCCRAND)
C       URN09...RANDU COMO EN SPSS ("RN4" EN IRCCRAND)
C       URN10...GENERADOR OMNITAB 2 DE KRUSKAL ("RN5" EN IRCCRAND)
C       URN11...GENERADOR CONGRUENCIAL MIXTO CON PARAMETROS DEL
C           USUARIO ("RNCG" EN IRCCRAND)
C       URN12...UNO MAS PRECISO, DOBLE PRECISION INTERNA, URN11
C           (NO ESTA EN IRCCRAND)
C       URN13...GENERADOR AHRENS-DIETER (NO ESTA EN IRCCRAND)
C       URN14...GENERADOR ZARLING
C       URN15...GENERADOR HOAGLIN
C       URN16...GENERADOR HOAGLIN
C       URN17...GENERADOR HOAGLIN
C       URN18...GENERADOR HOAGLIN
C       URN19...GENERADOR HOAGLIN
C       URN20...GENERADOR HOAGLIN
C       URN21...GENRADOR G. Y M. SWAIN (1980)
C       URN22...GENERADOR VAX-11/780
C       URN23...GENERADOR BRODY (1984)

```

```

C      URN24...GENERADOR THESEN (1985)
C      URN25...GENERADOR GAIT (1977)
C      URN26...GENERADOR MATHEWS (1986)
C      URN27...GENERADOR HENRIKSEN-CRANE (1983)
C      URN28...GENERADOR ANALOGO URN01
C      URN29...GENERADOR ANALOGO URN01
C      URN30...GENERADOR SIMSCRIPT ,WEST-JOHNSON (1984)
C      URN31...GENERADOR MIYAZAKI (1987)
C      URN32...GENERADOR MIYAZAKI (1987)
C      URN33...GENERADOR MIYAZAKI (1987)
C      URN34...GENERADOR SEZGIN (1987)
C      URN35...GENERADOR L' ECUYER (1987)
C      URN36...GENERADOR JENNERGREN (1984)
C      URN37...GENERADOR DUDEWICZ-KARIAN-MARSHALL (1985)
C      URN38...GENERADOR JENNERGREN
C      URN39...GENERADOR REY (1990)
C      URN40...GENERADOR MARSAGLIA Y ZAMAN
C      URNWN...UN NOMBRE RESERVADO PARA UN GENERADOR SUMINISTRADO
C      POR EL USUARIO
C
      DATA URNM/'URN01','URN02','URN03','URN04','URN05','URN06',
* 'URN07','URN08','URN09','URN10','URN11','URN12','URN13','URN14',
* 'URN15','URN16','URN17','URN18','URN19','URN20','URN21','URN22',
* 'URN23','URN24','URN25','URN26','URN27','URN28','URN29','URN30',
* 'URN31','URN32','URN33','URN34','URN35','URN36','URN37','URN38',
* 'URN39','URN40','URNWN'/
C
C      EL SIGUIENTE ENCABEZAMIENTO SIRVE PARA LA GENERACION
C      DE NUMEROS CADA VEZ QUE SE HACE UNA NUEVA
C      LLAMADA A TESTER
C
C      ABRIR ARCHIVO PARA GUARDAR LOS NUMEROS CREADOS
      ARCHURN='ARCH'//NAME//'.DAT'
C      WRITE(*,789)URNDAT
C789  FORMAT(1X,A12)
C
      OPEN(UNIT=50,FILE=ARCHURN,ACCESS='SEQUENTIAL',RECL=11)
C
C
      WRITE(6,70)
70  FORMAT(1X,
*****
***** )
      WRITE(6,73)
73  FORMAT(1X,'*',2X,'LAS SIGUIENTES SALIDAS HAN SIDO PRODUCIDASPOR
*EL SISTEMA: " GENTSTPC "' ,2X,'*')
      WRITE(6,75)
75  FORMAT(1X,'*',2X,'EL PAQUETE ESTA CONSTITUIDO POR RUTINAS PARA
LA
* GENERACION DE NUMEROS      ','*')
      WRITE(6,74)
74  FORMAT(1X,'*',2X,'PSEUDOALEATORIOS EN MICROCOMPUTADORES ',35x,'*
*')
      WRITE(6,87)
87
FORMAT(1X,*****
***** )
C      GENCODE
C      COMIENZO SUBROUTINA TESTER:
      DO 10 I=1,9
      MESSG(I)=BL

```

```

10 CONTINUE
   GCODE=0
C
C PREGUNTA SI EL NOMBRE DEL GENERADOR INGRESADO COINCIDE CON ALGUNO DE
C LOS INCLUIDOS EN EL ARREGLO URNM
   DO 20 KSUB=1,41
C     IF(NAME.EQ.URNM(KSUB))GO TO 800
     IF(NAME.EQ.URNM(KSUB))GO TO 8001
20 CONTINUE
   WRITE(*,*)'ERROR DE TECLEO EN EL NOMBRE DE URN**,RECOMENZAR'
   STOP
8001 LASTGN=NAME
   ISUB=KSUB

C     TITULO DE PAGINA PARA EL NUEVO GENERADOR
C
C     WRITE(6,300)
C 300 FORMAT('
', '*****')
C     *****')
C
   WRITE(*,*)
c 304 FORMAT(1X,'*',1X,'SE HA ENCONTRADO UN GENERADOR EN LOS DATOS DE
c *ENnTRADA.TODOS LOS TESTS      ', '*')
c     WRITE(6,305)
c 305 FORMAT(1X,'*',1X,'DISPONIBLES SERAN APLICADOS A ESTE GENERADOR
c *HASTA EL FIN DE LA EJECUCION.', '*')
c     WRITE(6,306)
c 306 FORMAT(' ', '*', 75X, '*')
c     WRITE(6,307)
c 307 FORMAT(' ', '*', 1X, 'SE ASIGNARA UN NUMERO DE CODIGO AL PRESENTE
c *GENERADOR', 21X, '*')
c     WRITE(6,310)GCODE
c 310 FORMAT(' ', '*', 1X, 'EL CODIGO ASIGNADO ES GENCODE:', I2, '
c *                               ', 5X, '*')
c     WRITE(6,360)
c 360
FORMAT(1X, '*****')
c     *****')
c
c     AQUI SE GENERARAN NLOT SUCESSIONES DE NUMALE(NBATCH) CADA UNA
   WRITE(*,*)'Ingrese la cantidad de Numeros Pseudoaleatorios que
   *desea generar en cada sucesion (<=10 mil)'
   READ(*,*)NBATCH
   WRITE(*,*)'Entre la cantidad de sucesiones (<= 1 mil)'
   READ(*,*)NLOT
C     INICIO DE GENERACION PSEUDOALEATORIOS PARA ARCHIVO
C
   DO 9888 LOTES = 1, NLOT
   GO TO (110,120,130,140,150,160,170,180,190,200,210,215,220,161,
   *171,181,191,201,211,212,2021,2022,2023,2024,2025,2026,2027,
   *2028,2029,2030,2031,2032,2033,2034,2035,2036,2037,2038,2039,
   *2040,2041), ISUB
110 CALL RRN01
   GO TO 630
120 CALL RRN02(NN,ML,MM,MK,L,MMM,KKK)
   GO TO 630
130 CALL RRN03
   GO TO 630
140 CALL RRN04(ITABLE)
   GO TO 6303

```

150 CALL RRN05(MFSR,IP,IQ,INTSIZ,JJ)  
GO TO 630  
160 CALL RRN06  
GO TO 630  
170 CALL RRN07  
GO TO 6303  
180 CALL RRN08  
GO TO 630  
190 CALL RRN09  
GO TO 630  
200 CALL RRN10  
GO TO 630  
210 CALL RRN11  
GO TO 630  
215 CALL RRN12(IX,L,C2,ZZ2)  
GO TO 630  
220 CALL RRN13(X,NBATCH,TIMES)  
GO TO 630  
161 CALL RRN14  
GO TO 630  
171 CALL RRN15  
GO TO 6303  
181 CALL RRN16  
GO TO 6303  
191 CALL RRN17  
GO TO 6303  
201 CALL RRN18  
GO TO 6303  
211 CALL RRN19  
GO TO 6303  
212 CALL RRN20(X,NBATCH,TIMES)  
GO TO 630  
2021 CALL RRN21(NBATCH)  
GO TO 630  
2022 CALL RRN22(NBATCH)  
GO TO 630  
2023 GOTO 6303  
2024 GOTO 6303  
2025 GOTO 6303  
2026 GOTO 6303  
2027 GOTO 6303  
2028 CALL RRN28(X,NBATCH,TIMES)  
GO TO 630  
2029 GOTO 6303  
2030 CALL RRN30(X,NBATCH,TIMES)  
GO TO 630  
2031 CALL RRN31(X,NBATCH,TIMES)  
GO TO 630  
2032 CALL RRN32(X,NBATCH,TIMES)  
GO TO 630  
2033 CALL RRN33(X,NBATCH,TIMES)  
GO TO 630  
2034 CALL RRN34(X,NBATCH,TIMES)  
GOTO 630  
2035 CALL RRN35(X,NBATCH,TIMES)  
GOTO 630  
2036 CALL RRN36(X,NBATCH,TIMES)  
GOTO 630  
2037 CALL RRN37(X,NBATCH,TIMES)  
GOTO 630  
2038 CALL RRN38(X,NBATCH,TIMES)

```

        GOTO 630
2039 CALL RRN39(X,NBATCH,TIMES)
        GOTO 630
2040 GOTO 6303
C      CALL RRN40(X,NBATCH,TIMES)
2041 GOTO 6303
C      CALL RRNWN
630   NEWGEN=1
C
C
C      A CONTINUACION SE IMPRIMEN LOS NUMEROS DE LA PRIMER MUESTRA SI FUE
C      REQUERIDO
C
C
C      GUARDA LOS NUMEROS      EN ARCHIVO UNIT=50
C
        940 DO 941 K1=1,NBATCH
            XDOBLE(K1)=X(K1)
C      941 Y(K1)=X(K1)
            WRITE(50,5050)XDOBLE(K1)
5050   FORMAT(F11.10)
        941 ENDDO
9888   CONTINUE
        RETURN
6303   WRITE(*,6304)NAME
6304   FORMAT(1X,'EL GENERADOR:',A8,' NO ESTA IMPLEMENTADO EN
GENTSTPC2')
        PAUSE
        STOP
        END
C631   CONTINUE
C
C      CICLO FINAL DE PROCESAMIENTO DE DATOS DE ENTRADA
C      626 RETURN

C      ***** FIN TESTER *****
C
C
C
C*****
C      SUBROUTINE SORT(R,N)
C*****
C      PROPOSITO
C      ORDENAR UN ARREGLO DADO
C
C      USO:
C      "CALL SORT(R,N)"
C
C      DESCRIPCION DE PARAMETROS
C
C      R...NOMBRE DEL ARREGLO UNIDIMENSIONAL A ORDENAR
C      N...DIMENSION DEL ARREGLO A ORDENAR
C      OBSERVACIONES:
C      ESTE ALGORITMO ES MAS EFICIENTE QUE EL METODO DE LA BURBUJA O EL
C      METODO POR INTERCAMBIO. AUNQUE NO SEA OPTIMO, PROVEE UNA TECNICA
C      SIMPLE Y RELATIVAMENTE EFICIENTE DE ORDENAR Y FACIL DE PROGRAMAR.
C
C      SUBROUTINAS Y FUNCIONES REQUERIDAS
C      NINGUNA

```



```

C
C METODO:
C EL METODO USADO ES "SHELL SORT" FUE INTRODUCIDO POR D.L.SHELL
C EN LA REF.1 .OTROS MAS RECIENTES DESARROLLOS EN LA REFERENCIA
C 2 Y SU BIBLIOGRAFIA.
C
C REFERENCIAS:
C
C 1. SHELL,D.L.:"A HIGH-SPEED SORTING PROCEDURE," COMMUNICATIONS OF
C THE ASSOCIATION FOR COMPUTING MACHINERY, VOL.2, NO.7(JULY,
C 1959), PAG.30-32.
C 2. GHOSHDASTIDAR,D. AND ROY,M.K.: "A STUDY ON THE EVALUATION OF
C SHELL'S SORTING TECHNIQUE," THE COMPUTER JOURNAL,VOL.18(1975),
C PAG.234-235.
C
C*****
C
C SUBROUTINE SORT(R,N)
C DIMENSION R(1)
C M=N
C 1 M=M/2
C IF(M.EQ.0) GOTO 5
C J=1
C 2 I=J
C 4 IF(R(I).LE.R(I+M)) GOTO 3
C SAV=R(I)
C R(I)=R(I+M)
C R(I+M)=SAV
C I=I-M
C IF(I.GE.1) GOTO 4
C 3 J=J+1
C IF(J.GT.N-M) GOTO 1
C GOTO 2
C 5 RETURN
C END
C
C
C
C
C----- G E N E R A D O R E S : -----
C
C Orden de las Subrutinas : RRN**, URN**I, URN***
C-----
C*****
C SUBROUTINE RRNWN
C*****
C
C ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C GENERADOR URNWN.
C EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C A URNWN I.
C URNWN CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO
C REQUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URNWN
C DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
C SUBROUTINE RRNWN
C
C REAL*8 NAME
C character*8 name,messg,variab,t01nm,tstnm,cknm(10)

```

```

CHARACTER C1,RESP
COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
COMMON /FIFTH/X(10000)
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
DATA YES/'YES '/

C
  IF(NEWGEN.NE.0) GOTO 12

C
C
  PARAMETROS IA,IR SERAN DEFINIDOS EN SUBROUTINA URNWN
  CALL URNWN
  IF(IA.EQ.0) GOTO 10
  WRITE(6,100)
100 FORMAT(' ',1X,'VALORES INICIALES ENTEROS PARA EL GENERADOR DEL U
  *SUARIO:')
  WRITE(6,9)(INTIN(K),K=1,IA)
  9 FORMAT(' ',1X,5(I11,1X))
  WRITE(*,*)' '
  10 IF(IR.EQ.0) GOTO 12
  WRITE(6,104)
104 FORMAT('0',5X,'VALORES INICIALES REALES PARA EL GENERADOR DEL
  *USUARIO:',//)
  WRITE(6,106)(REALIN(K),K=1,IR)
106 FORMAT(' ',5F11.8)
  WRITE(*,*)' '

C
  12 IF(TIMES.EQ.YES)GO TO 14

C
C
  GENERA NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
  CALL URNWN(X,NBATCH)
  GOTO 16

C
C
  GENERA NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
  14 KTIME=0
  CALL TIMER(ITIME)
  CALL URNWN(X,NBATCH)
  CALL TIMER(ITIME)
  KTIME=KTIME+ITIME

C
  16 IF(NEWGEN.NE.0) GOTO 999

C
  WRITE(*,*)' '
  WRITE(6,200) NBATCH
200 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO (
  *SIGUIENTE A LOS',I6)
  WRITE(6,201)
201 FORMAT(1X,'YA GENERADOS) ES:')
  IF(IA.EQ.0) GOTO 18
  WRITE(6,202)(INTIN(K),K=1,IA)
202 FORMAT(' ',10(I11,1X))
  18 IF(IR.EQ.0) GOTO 999
  WRITE(6,206)(REALIN(K),K=1,IR)
206 FORMAT(' ',10(F11.8,1X))
999 RETURN
  END

C
C*****
C
  SUBROUTINE URNWN

```

```

C*****
C
C   ESTA SUBROUTINA INICIALIZA A URNWN. EL GENERADOR DEL USUARIO
C   TIENE LAS SIGUIENTES OPCIONES:
C
C   IOPT=1  INICIALIZACION DE VALORES POR LECTURA
C   IOPT=2  INICIALIZACION DE VALORES EN LA MISMA URNWN
C
C*****
C
C   SUBROUTINE URNWNI
C       character*8 name,messg,variab,t01nm,tstnm,cknm(10)
C       REAL*8 NAME
C       COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
C       COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C
C       IA=1
C       IR=0
C       INTIN(1)=32767
C       IF((IA.EQ.0).AND.(IR.EQ.0)) GOTO 40
C       IF(IOPT.EQ.0) GO TO 20
C       IF(IOPT.EQ.2)GO TO 999
C       IF(IA.EQ.0) GOTO 12
C   VALORES POR TECLADO
C       WRITE(6,*)'DEFINIR VALORES INTIN(K), K=1,...,IA, EN LA SUBRUT
C       *INE:URNWNI '
C   12 IF(IR.EQ.0) GOTO 999
C
C       WRITE(6,*)'DEFINIR VALORES REALIN(K), K=1,...,IR, EN LA
C       SUBROUTINA:      *URNWNI '
C
C   40 WRITE(6,200)
C   200 FORMAT(1X,'LOS PARAMETROS IA=IR=0 INDICAN QUE NO SE HAN
C       *UTILIZADO ESTOS VALORES',
C       */1X,'PARTICULARES DE INICIALIZACION PARA EL GENERADOR URNWN.',
C       */1X, ' EJECUCION TERMINADA. ')
C       GOTO 998
C   20 WRITE(6,210)
C   210 FORMAT(1X,'LA OPCION DE INICIALIZACION (0) PARA EL GENERADOR URN
C       *WN ES UNA OPCION EQUIVOCADA'
C       *,/1X,'PARA UN GENERADOR DEL USUARIO'
C       *,//1X, ' EJECUCION TERMINADA. ')
C   998 STOP
C   999 RETURN
C       END
C
C*****
C   SUBROUTINE URNWN(X,NBATCH)
C*****
C
C   ESTA SUBROUTINA SE PROVEE COMO UN PROTOTIPO DE LA FORMA EN LA QUE
C   DEBE PROCESARSE EN GENTSTPC UN GENERADOR PROVISTO POR EL USUARIO. SI
C   EL USUARIO PROVEE OTRA SUBROUTINA, DEBE SOBRESERIBIR ESTA. SI EL
C   USUARIO INTENTA USAR URNWN SIN DAR SU PROPIO CODIGO, APARECERA
C   UN MENSAJE DE ERROR Y TERMINARA LA EJECUCION. ESTE EJEMPLO ACTUA
C   COMO URN09.
C   EN LA DESCRIPCION DE LA SUBROUTINA URN09 SE VE "PROPOSITO","USO",
C   "DESCRIPCION DE PARAMETROS", "OBSERVACIONES", "SUBROUTINAS Y
C   FUNCIONES REQUERIDAS", Y "METODO".
C   UNA SECUENCIA TIPICA DE LLAMADO EN EL PROGRAMA PRINCIPAL DEL
C   USUARIO PUEDE SER (CON INICIALIZACION PROVISTA POR EL USUARIO) :

```

```

C      COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
C      IA=1
C      IR=0
C      INTIN(1)=524287
C      CALL TESTER
C DE OTRO MODO, SI EN VEZ DE INICIALIZACION POR EL USUARIO
C FUERAN DATOS EN ARCHIVOS DE ENTRADA, SE TENDRIA:
C      COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
C      IA=1
C      IR=0
C      CALL TESTER
C*****
C
      SUBROUTINE URNWN(X,NBATCH)

      CHARACTER C1,RESP
      COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
      DIMENSION X(NBATCH)
      OPEN(UNIT=60,ACCESS='DIRECT',RECL=1,FORM='FORMATTED',FILE=
* 'OPCION')
      READ(60,12)C1
12  FORMAT(1A)
      IF(C1.EQ.'S')THEN
          READ(60,12)RESP
          GOTO 88
      ENDIF
      OPEN(UNIT=63,ACCESS='DIRECT',RECL=1,FORM='FORMATTED',FILE=
* 'GENOARCH')
      READ(63,12)RESP
      CLOSE(63)
      CLOSE(60)
      OPEN(UNIT=60,ACCESS='DIRECT',RECL=1,FORM='FORMATTED',FILE=
* 'OPCION')
      WRITE(60,12)'S'
      WRITE(60,12)RESP
88  IF(RESP.EQ.'1') THEN
C
C      GENERACION DEL GENERADOR PROPIO "URNWN"
C
      J=INTIN(1)
      DO 10 I = 1,NBATCH
      J=J*1220703125
      IF(J.LT.0)J=J+2147483647+1
      X(I)=FLOAT(J)*0.4656613E-9
10  CONTINUE
      INTIN(1)=J
C
C      LECTURA DE "ARCHDAT"
C
      ELSE IF(RESP.EQ.'2') THEN
      DO 20 II = 1 ,NBATCH
      READ(90,900)XXX
      X(II)=XXX
900  FORMAT(F11.10)
20  CONTINUE
      ELSE
      WRITE(*,*)'HA INGRESADO UN VALOR INCORRECTO'
      STOP
      ENDIF
      CLOSE(60)
      RETURN

```

```

END
C
C
C*****
C   SUBROUTINE RRN01
C*****
C   ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN01.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN01I E IMPRIME LOS VALORES INICIALES.
C   URN01 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN01
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
C   SUBROUTINE RRN01
C     character*8 name,messg,variab,t0lnm,tstnm,cknm(10)
C     REAL*8 NAME
C     INTEGER DBLPRT,EXTPRT
C     COMMON /FOURTH/Y(10000)
C     COMMON /FIFTH/X(10000)
C     COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C     COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C     COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C     COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKK,L,ML,MK,MMM
C     COMMON /CODE/GCODE
C     DATA YES/'YES '/
C     IF(NEWGEN.NE.0) GOTO 10
C     CALL URN01I
C     WRITE(6,990)IX
990 FORMAT(' ',8X,'IX=',I11)
10 IF(TIMES.EQ.YES)GO TO 20
C
C     GENERACION DE NUMEROS ALEATORIOS SIN CALCULO TIEMPOS
C
C     CALL URN01(X,IX,NBATCH)
C
C     GOTO 30
C
C     GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20 KTIME=0
C     CALL TIMER(ITIME)
C     CALL URN01(X,IX,NBATCH)
C     CALL TIMER(ITIME)
C     KTIME=KTIME+ITIME
C
30 IF(NEWGEN.NE.0) GOTO 999
C     WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
* '(SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES:')
C     WRITE(6,990) IX
999 RETURN
C     END
C
C*****
C   SUBROUTINE URN01I
C*****
C
C   ESTA SUBROUTINA INICIALIZA AQUELLOS GENERADORES QUE NECESITAN

```

```

C      UN SOLO VALOR INICIAL. LAS OPCIONES DISPONIBLES SON:
C
C      IOPT=0  INICIALIZACION POR DEFECTO
C      IOPT=1  INICIALIZACION MEDIANTE LECTURAS DE VALORES EN LA RUTINA
C      IOPT=2  INICIALIZACION DE VALORES PROVISTOS POR EL USUARIO
C
C      LOS GENERADORES INICIALIZADOS POR ESTA RUTINA SON LOS SIGUIENTES:
C      URN01.
C      URN07.
C      URN08.
C      URN09.
C      URN13.( *)
C      URN15.
C      URN16.
C      URN17.
C      URN18.
C      URN19.
C      URN20.
C (*)EN LA VERSION ASSEMBLER DEL PAQUETE TESTRAND
C*****
C
C      SUBROUTINE URN01I
C      ENTRY URN07I
C      ENTRY URN08I
C      ENTRY URN09I
C      ENTRY URN10I
C      ENTRY URN13I
C      ENTRY URN15I
C      ENTRY URN16I
C      ENTRY URN17I
C      ENTRY URN18I
C      ENTRY URN19I
C      ENTRY URN20I
C      character*8 name,messg,variab,t0lnm,tstnm,cknm(10),except
C      REAL*8 NAME,EXCEPT
C      DATA EXCEPT/'URN10  '/
C      COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C      COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
C      IF(IOPT.EQ.1)GO TO 10
C      IF(IOPT.EQ.2)GO TO 30
C      IF(NAME.EQ.EXCEPT)IX=1
C      IF(NAME.NE.EXCEPT)IX=524287
C      GO TO 30
C      10 READ(5,600)IX
C      600 FORMAT(I11)
C      30 RETURN
C      END
C
C*****
C      SUBROUTINE URN01(X,IX,NBATCH)
C*****
C
C      INICIALMENTE TESTRAND ADOPTO ESTA RUTINA DEL PAQUETE LLRANDOM EN
C      CODIGO ASSEMBLER ESPECIFICO AL SISTEMA IBM/370 (NAVAL POSTGRADUATE
C      SCHOOL RANDOM NUMBER GENERATOR PACKAGE)
C      EN ESTA VERSION DE GENTSTPC SE DESARROLLA OTRO ALGORITMO.
C
C      PROPOSITO
C      GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE
C      PRECISION EN (0.0,1.0)
C

```

```

C USO
C   CALL URN01(X,IX,NBATCH)
C
C DESCRIPCION DE PARAMETROS
C   IX...ES EL VALOR INICIAL DE UNA SUCESION Y PUEDE SE CUALQUIER
C       ENTERO ENTRE 1 Y 2147483647.(2**31 -1)
C   X... ES EL ARREGLO RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO
C       FLOTANTE DE SIMPLE PRECISION EN (0.0,1.0)
C   NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR
C       CON EL LLAMADO DE ESTA SUBRUTINA
C
C OBSERVACIONES
C   IX NO DEBE SER ALTERADO POR EL USUARIO DURANTE LA EJECUCION
C   DEL PROGRAMA, SALVO QUE EL DECIDA REPETIR UNA SUCESION DE
C   NUMEROS ALEATORIOS.
C
C SUBRUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C METODO
C   REFERENCIAS:
C   1. ALGORITMO DE BRADLEY ,FOX Y SCHRAGE,"A GUIDE TO SIMULATION",
C       SPRINGER,2DA.ED.,1987,P.212.
C   2. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C       GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C       COLUMBUS, OHIO,1981.
C
C*****
C
SUBROUTINE URN01(X,IX,NBATCH)
REAL*8 X1(10000),X0,XM,XU
DIMENSION X(10000)
X0=IX
XM=2147483647D0
DO 11 I=1,NBATCH
X1(I)= DMOD(16807D0*X0,XM)
X0=X1(I)
XU=X1(I)/XM
X(I)=XU
11  END DO
IX=X0
RETURN
END

C
C
C*****
C   SUBROUTINE RRN02(NN,ML,MM,MK,L,MMM,KKK)
C*****
C
C   ESTA SUBRUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN02.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN02I E IMPRIME LOS VALORES INICIALES.
C   URN02 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE) E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN02
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
SUBROUTINE RRN02(NN,ML,MM,MK,L,MMM,KKK)
character*8 name,messg,variab,t01nm,tstnm,cknm(10)

```

```

C      REAL*8 NAME
COMMON /FIFTH/X(10000)
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
DIMENSION NN(128)
DATA YES/'YES '/
IF(NEWGEN.NE.0) GOTO 10
CALL URN02I(NN,ML,MM,MK,L,MMM,KKK)
WRITE(6,990) ML,MM,MK,L,MMM,KKK
990 FORMAT(' ',8X,'ML=',I11,' MM=',I11,' MK=',I11,' L=',I11,
*' MMM=',I11,' KKK=',I11)
WRITE(6,940)NN
940 FORMAT(9X,'NN= ',/,9(14(1X,Z8),/),1X,Z8,1X,Z8)
10 IF(TIMES.EQ.YES)GO TO 20

C
C      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO TIEMPOS
C
CALL URN02(X,NN,ML,MM,MK,L,MMM,KKK,NBATCH)
GOTO 30

C
C      GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20 KTIME=0
CALL TIMER(ITIME)
CALL URN02(X,NN,ML,MM,MK,L,MMM,KKK,NBATCH)
CALL TIMER(ITIME)
KTIME=KTIME+ITIME

C
30 IF(NEWGEN.NE.0) GOTO 999
WRITE(*,*)' '
WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO ',
*' (SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES:')
WRITE(6,990) ML,MM,MK,L,MMM,KKK
WRITE(6,940) NN
999 RETURN
END

C
C*****
C      SUBROUTINE URN02I
C*****
C
C      ESTA SUBRUTINA INICIALIZA URN02.LAS OPCIONES DISPONIBLES SON:
C
C      IOPT=0 INICIALIZACION POR DEFECTO
C      IOPT=1 INICIALIZACION MEDIANTE LECTURAS DE VALORES EN LA RUTINA
C      IOPT=2 INICIALIZACION DE VALORES PROVISTOS POR EL USUARIO
C      IOPT=3 TODAS LAS INICIALIZACIONES EXCEPTO NN PROVISTA POR EL
C      USUARIO
C      IOPT=4 TODAS LAS INICIALIZACIONES EXCEPTO NN LEIDA EN LA RUTINA
C
C*****
C
SUBROUTINE URN02I(NN,ML,MM,MK,L,MMM,KKK)
DIMENSION N(128),NN(128)
C      INICIALIZACION N(1),...,N(128) (MISMOS VALORES QUE PAQUETE
LLRND)
DATA N /880431333,845941495,233211304,1989617657,465185814,
*280672918,294923811,969688974,798989604,379880543,130022074,
*1958997525,1074191695,680854387,751282651,1208899767,695831691,

```



```

*1667008051,1682546364,1984522335,287570376,1137852001,1597983496,
  *2015817872,1479672206,1468443024,1657203843,326324124,680973716,
  *1451006002,1251441372,241092947,1815086916,1807193097,770906592,
  *725422944,1822111098,470585328,939566271,1084841188,1988336409,
  *229735215,1763201387,2072973152,1143606610,548108569,544252510,

*1980873641,1195919839,2089487851,1406149582,1839198022,2106705200,
  *189238196,1170370207,1304402631,1936129483,810953177,706509560,
  *476957499,1307077413,824336639,1487297852,1591453718,107382480,
  *155452792,265840413,1440038626,785915439,1152058296,1726999383,
  *1389732859,1838014251,1751063044,102451305,212848938,1046489181,
  *976388856,1797117421,461971124,259337424,492056652,1152625277,
  *1087711027,344810019,1477716555,809152324,1766452264,1687482934,
  *1077592551,1906112218,328744821,1380339247,339750038,1993648985,

*1054008271,2006727977,1618648061,1300903972,168650429,1734500183,
  *906733794,614096451,1092917209,1180334545,577024776,1406305431,
  *648073629,973807028,883884075,1562357277,1705648154,1377603620,
  *1845151798,220566094,768813055,571717967,218994012,212872559,
  *1824677815,1573937649,450149130,284847256,2062965934,47834840,
  *1766553923,1580332201,182920702/

```

```

C
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
      character*8 name,messg,variab,t01nm,tstnm,cknm(10)
c
      REAL*8 NAME
C
      IF (IOPT.EQ.1)GO TO 50
      IF (IOPT.EQ.1)GO TO 50
      IF (IOPT.EQ.2)GO TO 30
      IF (IOPT.EQ.3.OR.IOPT.EQ.4)GO TO 60
C
      ML=65539
      MM=33554433
      MK=362436069
      L=089347405
      MMM=301467177
      KKK=240420681
C
      60 DO 20 I=1,128
          NN(I)=N(I)
      20 CONTINUE
C
      IF (IOPT.EQ.4)READ(5,100)ML,MM,MK,L,MMM,KKK
      100 FORMAT(6I11)
          GO TO 30
C
      50 READ(5,140)NN
      140 FORMAT(15(8(Z8,1X),/),8(Z8,1X))
          READ(5,100)ML,MM,MK,L,MMM,KKK
C
      30 RETURN
      END
C*****
C      SUBROUTINE URN02(X,NN,ML,MM,MK,L,MMM,KKK,NBATCH)
C*****
C
C      SUBROUTINA DADA POR MARSAGLIA Y BRAY, PARA GENERAR NUMEROS ALEATO-
C      RIOS EN PUNTO FLOTANTE. EN EL PAQUETE IRCCRAND LA RUTINA "RN3"
C      UTILIZA EL MISMO METODO PARA GENERAR NUMEROS ALEATORIOS EN SIMPLE
C      PRECISION (REFERENCIA 2)

```

```

C
C PROPOSITO
C   GENERACION DE VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE
C   SIMPLE PRECISION SOBRE (0.0,1.0)
C
C USO
C   "CALL URN02I(ML,MM,MK,L,MMM,KKK,NN,)"
C   DESDE EL PROGRAMA PRINCIPAL.
C   LUEGO "CALL URN02(X,NN,ML,MM,MK,L,MMM,KKK,NBATCH)"
C
C DESCRIPCION DE PARAMETROS
C   ML...CONSTANTE ENTERA (MULTIPLICADOR) DEL GENERADOR
C   DE ENTEROS ALEATORIOS
C   MM...CONSTANTE ENTERA (MULTIPLICADOR) DEL GENERADOR
C   DE ENTEROS ALEATORIOS
C   MK...CONSTANTE ENTERA (MULTIPLICADOR) DEL GENERADOR
C   DE ENTEROS ALEATORIOS
C   L... SEMILLA PARA UNA SUCESSION ALEATORIA
C   MMM...SEMILLA PARA UNA SUCESSION ALEATORIA
C   KKK...SEMILLA PARA UNA SUCESSION ALEATORIA
C   NN... VECTOR VARIABLE CON DIMENSION ESPECIFICA 128 DONDE
C   SE ALMACENAN LOS ENTEROS ALEATORIOS USADOS
C   X... ARREGLO RESULTANTE EN PUNTO FLOTANTE DE SIMPLE PRECISION
C   PARA LOS NUMEROS ALEATORIOS EN (0.0,1.0)
C   NBATCH...CANTIDAD DE NUMEROS ALEATORIOS A SER GENERADOS POR EL
C   LLAMADO DE ESTA SUBROUTINA
C
C OBSERVACIONES
C LA SUBROUTINA URN02I INICIALIZA LAS VARIABLES NN,ML,MM,MK,L,MMM,KKK;
C SI EL USUARIO OPTA POR SUS PROPIOS VALORES NO DEBE INVOCAR EL LLA-
C MADO "CALL URN02I(ML,MM,MK,L,MMM,KKK,NN)", PERO EN CAMBIO DEBE
C INICIALIZAR DICHAS VARIABLES.
C
C ESTA SUBROUTINA ES ESPECIFICA PARA SISTEMAS IBM/360/370. EL METODO
C FUE DADO EN REFERENCIA 1.
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C METODO
C   REFERENCIAS:
C   1. MARSAGLIA, G. AND BRAY, T.A.: "ONE-LINE RANDOM NUMBER
C   GENERATORS AND THEIR USE IN COMBINATIONS",COMMUNICATIONS OF TH
C   ACM, VOL.11(1968), PAG. 757-759.
C   2. DUDEWICZ,E.J.: "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM
C   NUMBER GENERATOR PACKAGE",TECHNICAL REPORT, DEPARTMENT OF
C   STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
C   3. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C   GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C   COLUMBUS, OHIO, 1981.
C   PARA SISTEMAS DISTINTOS A LAS MAQUINAS 360/370:
C   4. GROSENBAUGH,L.R., "MORE ON FORTRAN RANDOM NUMBER GENERATORS"
C   COMMUNICATIONS DE ACM, VOL.12(1969), P.639.
C
C*****
C
SUBROUTINE URN02(X,NN,ML,MM,MK,L,MMM,KKK,NBATCH)
DIMENSION NN(128)
DIMENSION X(NBATCH)
DO 12 I=1,NBATCH
L=L*ML

```

```

      MMM=MMM*MM
      J=1+IABS(L)/16777216
      X(I)=.5+FLOAT(NN(J)+L+MMM)*.23283064E-9
      KKK=KKK*MK
      NN(J)=KKK
12  CONTINUE
      RETURN
      END

C
C
C*****
C   SUBROUTINE RRN03
C*****
C
C   ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN03.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN03I E IMPRIME LOS VALORES INICIALES.
C   URN03 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE) E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN03
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
C   SUBROUTINE RRN03
      character*8 name,messg,variab,t0lnm,tstnm,cknm(10)
c   REAL*8 NAME
      COMMON /FIFTH/X(10000)
      COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
      COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
      COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
      COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKK,L,ML,MK,MMM
      DATA YES/'YES '/
      IF(NEWGEN.NE.0) GOTO 10
      CALL URN03I
      WRITE(6,990)IX,JX
10  IF(TIMES.EQ.YES) GOTO 20

C
C   GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
      CALL URN03(X,IX,NBATCH)
      GOTO 30

C
C   GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20  KTIME=0
C   CALL TIMER(ITIME)
      CALL URN03(X,IX,NBATCH)
C   CALL TIMER(ITIME)
      KTIME=KTIME+ITIME

C
30  IF(NEWGEN.NE.0) GOTO 999
      WRITE(6,400) NBATCH
400  FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
      *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES:')
      WRITE(6,990) IX,JX
990  FORMAT(1X,'IX= ',I11,5X,'JX= ',I11)
999  RETURN
      END

C
C*****

```

```

C      SUBROUTINE URN03I
C*****
C
C      ESTA SUBRUTINA INICIALIZA AQUELLOS GENERADORES QUE NECESITAN
C      DOS VALORES INICIALES.LAS OPCIONES DISPONIBLES SON:
C
C      IOPT=0  INICIALIZACION POR DEFECTO
C      IOPT=1  INICIALIZACION MEDIANTE LECTURAS DE VALORES EN LA RUTINA
C      IOPT=2  INICIALIZACION DE VALORES PROVISTOS POR EL USUARIO
C
C      LOS GENERADORES INICIALIZADOS POR ESTA RUTINA SON LOS SIGUIENTES:
C      URN03.
C      URN06.
C*****
C
C      SUBROUTINE URN03I
C      ENTRY URN06I
C      character*8 name,messg,variab,t01nm,tstnm,cknm(10)
C      REAL*8 NAME
C      COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C      COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKK,L,ML,MK,MMM
C      IX=1
C      RETURN
C      END
C*****
C      SUBROUTINE URN03(X,IX,NBATCH)
C*****
C
C      INICIALMENTE TESTRAND ADOPTO ESTA RUTINA DEL PAQUETE SUPER-DUPER
C      EN CODIGO ASSEMBLER ESPECIFICO AL SISTEMA IBM/370 (MCGILL UNIV.
C      SCHOOL OF COMPUTER SCIENCE RANDOM NUMBER GENERATOR PACKAGE)
C      EN ESTA VERSION DE GENTSTPC SE DESARROLLA OTRO ALGORITMO.
C
C      PROPOSITO
C      GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE
C      PRECISION EN (0.0,1.0)
C
C      USO
C      CALL URN03(X,IX,NBATCH)
C
C      DESCRIPCION DE PARAMETROS
C      IX...ES EL VALOR INICIAL DE UNA SUCESSION Y PUEDE SE CUALQUIER ENTE-
C      RO ENTRE 1 Y 2147483647.(2*31 -1)
C      X... ES EL ARREGLO RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO FLO-
C      TANTE DE SIMPLE PRECISION EN (0.0,1.0)
C      NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR
C      CON EL LLAMADO DE ESTA SUBRUTINA
C
C      OBSERVACIONES
C      IX NO DEBE SER ALTERADO POR EL USUARIO DURANTE LA EJECUCION
C      DEL PROGRAMA, SALVO QUE DECIDA REPETIR UNA SUCESSION DE
C      NUMEROS ALEATORIOS.
C
C      SUBRUTINAS Y FUNCIONES REQUERIDAS
C      NINGUNA
C
C      METODO
C      REFERENCIAS:
C      1. ALGORITMO DE BRADLEY ,FOX Y SCHRAGE,"A GUIDE TO SIMULATION",
C      SPRINGER, 2DA.ED.,1987,P.211

```

```

C      2. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C      GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C      COLUMBUS,OHIO,1981.
C
C*****
C
C      SUBROUTINE URN03(X,IX,NBATCH)
C      DIMENSION X(10000)
C      INTEGER*4 XX(10000),X0,XM
C      X0=IX
C      XM=2147483648
C      XM1=4.656612873e-10
C      DO 11 I=1 ,NBATCH
C      XX(I)= X0*65539
C      IF(XX(I).LT.0) THEN
C      XX(I)=1+(XX(I)+2147483647)
C      END IF
C      X0=XX(I)
C      X(I)=XX(I)*XM1
11    END DO
C      IX=X0
C      RETURN
C      END
C
C
C*****
C      SUBROUTINE RRN04(ITABLE)
C*****
C
C      ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C      GENERADOR URN04.
C      EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C      A URN04I E IMPRIME LOS VALORES INICIALES.
C      URN04 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C      QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN04
C      DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
C      SUBROUTINE RRN04(ITABLE)
C      character*8 name,messg,variab,t01nm,tstnm,cknm(10)
C      REAL*8 NAME
C      COMMON /FIFTH/X(10000)
C      COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C      COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C      COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
C      DIMENSION ITABLE(128)
C      DATA YES/'YES '/
C      IF(NEWGEN.NE.0) GOTO 10
C
C      CALL URN04I
C      WRITE(6,990)IX
C      990 FORMAT(' ',8X,'IX=',I11)
C      WRITE(6,940) ITABLE
C      940 FORMAT(9X,'ITABLE=',/,9(14(1X,Z8),/),1X,Z8,1X,Z8)
C      CALL LLRND
C
C      10 IF(TIMES.EQ.YES)GO TO 20
C
C      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS

```

```

C
C      CALL URN04(IX,X,NBATCH,ITABLE)
C      GOTO 30
C
C      GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C      20 KTIME=0
C      CALL TIMER(ITIME)
C      CALL URN04(IX,X,NBATCH,ITABLE)
C      CALL TIMER(ITIME)
C      KTIME=KTIME+ITIME
C
C      30 IF(NEWGEN.NE.0) GOTO 999
C      WRITE(6,400) NBATCH
C      400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
C      *' SIGUIENTE A LOS ',I11/1X,' YA GENERADOS) ES:')
C      WRITE(6,990) IX
C      WRITE(6,940) ITABLE
C      999 RETURN
C      END
C
C*****
C
C      SUBROUTINE URN04 = URN01: ASSEMBLER
C*****
C*****
C      SUBROUTINE RRN05(MFSR,P,Q,INTSIZ,JJ)
C*****
C
C      ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C      GENERADOR URN05.
C      EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C      A URN05I E IMPRIME LOS VALORES INICIALES.
C      URN05 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE=
C      QUIERE)
C      E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN05 DESPUES
C      DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
C      SUBROUTINE RRN05(MFSR,P,Q,INTSIZ,JJ)
C      character*8 name,messg,variab,t01nm,tstnm,cknm(10)
C      REAL*8 NAME
C      COMMON /FIFTH/X(10000)
C      COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,ILOPT
C      COMMON /TSTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C      COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
C      DATA YES/'YES '/
C      DIMENSION MFSR(98)
C      INTEGER P,Q
C
C
C      IF(NEWGEN.NE.0) GOTO 10
C
C      IFLAG=0
C      CALL URN05I(MFSR,P,Q,INTSIZ,JJ)
C      WRITE(6,940) P,Q,INTSIZ,JJ
C      940 FORMAT(' ',8X,'P=',I11,' Q=',I11,' INTSIZ=',I11,'JJ=',I11)
C      WRITE(6,990)MFSR
C      990 FORMAT(9X,'MFSR=',/,9(10(1X,I11),/),8(1X,I11))
C      10 IF(TIMES.EQ.YES)GO TO 20

```

```

C
C          GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C          CALL URN05(MFSR,P,Q,INTSIZ,JJ,IFLAG)
C          GOTO 30
C
C          GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C          20 KTIME=0
C             CALL TIMER(ITIME)
C             CALL URN05(MFSR,P,Q,INTSIZ,JJ,IFLAG)
C             CALL TIMER(ITIME)
C             KTIME=KTIME+ITIME
C
C          30 IF(NEWGEN.NE.0) GOTO 999
C             WRITE(6,400) NBATCH
C          400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
C             *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C             WRITE(6,940) P,Q,INTSIZ,JJ
C             WRITE(6,990) MFSR
C          999 RETURN
C             END
C
C*****
C          SUBROUTINE URN05I(MFSR,P,Q,INTSIZ,JJ)
C*****
C
C          ESTA SUBRUTINA INICIALIZA URN05.  LAS OPCIONES DISPONIBLES SON:
C
C          IOPT=0  INICIALIZACION POR DEFECTO
C          IOPT=1  INICIALIZACION MEDIANTE LECTURAS DE VALORES EN LA RUTINA
C          IOPT=2  INICIALIZACION DE VALORES PROVISTOS POR EL USUARIO
C          IOPT=3  TODAS LAS INICIALIZACIONES EXCEPTO MFSR PROVISTO POR EL
C                  USUARIO
C          IOPT=4  TODAS LAS INICIALIZACIONES EXCEPTO MFSR LEIDO EN LA RUTINA
C
C*****
C
C          SUBROUTINE URN05I(MFSR,P,Q,INTSIZ,JJ)
C          DIMENSION MMFSR(98),MFSR(98)
C          CHARACTER*8 NAME
C          REAL*8 NAME
C          COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C          DATA MMFSR/346256726,591599773,1943131421,1173234223,1776849374,
C          *1119416586,172236044,985756773,1554281477,1503137291,650397619,
C          *1618395655,639939067,1448259547,1046853128,659170036,1034934222,
C          *279813371,326930100,367002640,648480182,1909733845,618563844,
C          *845531267,292262469,299413367,2139821356,1005803337,390139420,
C          *1161028423,2034360736,334070487,565633315,124796253,2104169336,
C          *2009751844,1999687407,83223028,1591328966,646701838,1935362333,
C          *795013136,680356918,1771711842,1324935502,1869840308,356745634,
C          *1061920662,614951490,261876461,703987800,797463948,178239686,
C
C          *1641708282,1539695556,1334926802,940547749,1957646566,1878491364,
C
C          *2033904942,1711106005,2138438575,647734238,1555990485,1210108489,
C          *1793192836,1819829578,751843064,345621400,575445974,1640918761,
C
C          *1379191461,1617832156,542966103,1305854952,1476721677,1466811698,
C          *1842260101,1666639833,217007402,685228354,902087789,32432242,
C          *789712994,702791444,1081111755,1572116899,321512624,644413114,

```

```

*863989644,1348681739,84379947,1955819746,941474606,984690559,
*1794209263,1704575856,1253913135/
C
    INTEGER DELAY,P,Q,ONE
C
    IF(IOPT.EQ.0)GO TO 90
    IF(IOPT.EQ.1)GO TO 70
    IF(IOPT.EQ.2)GO TO 80
    IF(IOPT.EQ.4)READ(5,92)P,Q,INTSIZ,DELAY
92  FORMAT(4I11)
    JJ=0
    IFLAG=1
    ONE=2**(INTSIZ-1)
C
    DO 1 I=1,P
1   MFSR(I)=ONE
C
    DO 4 K=1,INTSIZ
    DO 2 J=1,DELAY
2   CALL URN05(MFSR,P,Q,INTSIZ,JJ,IFLAG)
    KOUNT=0
    DO 3 I=1,P
    ITEMP=ONE/2**(K-1)
    ITEMP=(MFSR(I)-MFSR(I)/ONE*ONE)/ITEMP
    IF(ITEMP.EQ.1)KOUNT=KOUNT+1
    IF(K.EQ.INTSIZ)GO TO 3
    MFSR(I)=MFSR(I)/2+ONE
3   CONTINUE
4   CONTINUE
C
    DO 6 I=1,5000
    DO 5 J=1,P
5   CALL URN05(MFSR,P,Q,INTSIZ,JJ,IFLAG)
6   CONTINUE
    GO TO 80
90  P=98
    Q=27
    INTSIZ=31
    JJ=0
60  DO 20 I=1,98
    MFSR(I)=MMFSR(I)
20  CONTINUE
    GO TO 80
70  READ(5,75)MFSR,P,Q,INTSIZ,JJ
75  FORMAT(14(7I11,/),4I11)
80  RETURN
    END
C
C*****
C   SUBROUTINE URN05(MFSR,P,Q,INTSIZ,JJ,IFLAG)
C*****
C
C   METODO GENERALIZADO FEEDBACK SHIFT REGISTER (GFSR) DADO POR T.G.
C   LEWIS Y W.H. PAYNE, "GENERALIZED FEEDBACK SHIFT REGISTER PSEUDO-
C   RANDOM NUMBER ALGORITHM",JOURNAL OF THE ASSOCIATION FOR COMPUTING
C   MACHINERY, VOL.20(1973), PAG.456-468,PARA LA GENERACION DE NUMEROS
C   ALEATORIOS EN PUNTO FLOTANTE
C
C PROPOSITO
C   GENERACION DE MUMEROS ALEATORIOS EN (0.0,1.0)
C

```



```

C USO
C "CALL URN05I(MFSR,P,JJ,Q,INTSIZ)" SOLO UNA VEZ EN EL PROGRAMA PRIN-
C CIPAL. DESPUES "CALL URN05(MFSR,P,Q,INTSIZ,JJ,IFLAG)"
C
C DESCRIPCION DE PARAMETROS
C MFSR...ES UNA TABLA DE P NUMEROS ALEATORIOS INICIALES
C P...ES UN EXPONENTE DEL POLINOMIO PRIMITIVO X**P+X**Q+1
C Q...ES UN EXPONENTE DEL POLINOMIO PRIMITIVO X**P+X**Q+1
C DELAY...ES LA DIFERENCIA RELATIVA, EN BITS, ENTRE COLUMNAS DE
C MFSR(.)
C INTSIZ...ES EL TAMA.¥.O INTERO (EN BITS) DEL COMPUTADOR QUE HA SIDO
C UTILIZADO; POR EJEMPLO, 31 PARA LOS SISTEMAS IBM/360/370
C 48 PARA EL COMPUTADOR CDC 6000, 35 PARA EL COMPUTADOR SRU
C 1100 Y 15 PARA EL COMPUTADOR HP 2100
C X...ES EL ARREGLO RESULTANTE EN PUNTO FLOTANTE DE SIMPLE PRECISION
C PARA LOS NUMEROS ALEATORIOS EN (0.0,1.0)
C JJ...ES UN PARAMETRO USADO CUANDO RECOMIENZA LA SUCESION
C IFLAG...ES UN PARAMETRO QUE ES PASADO DEL PROGRAMA PRINCIPAL
C Y USADO POR LA SUBROUTINA , QUE NO DEBE SER CAMBIADO POR EL
C USUARIO.
C
C OBSERVACIONES
C URN05I INICIALIZA LAS VARIABLES MFSR,P,DELAY,Q,INTSIZ PARA
C EL SISTEMA IBM /370 Y EL POLINOMIO PRIMITIVO X**98+X**27+1;
C SI EL USUARIO DESEA EMPLEAR SUS PROPIOS VALORES NO DEBERA INVOCAR
C "CALL URN05I(MFSR,P,DELAY,Q,INTSIZ)", Y EN TAL CASO INICIALIZAR
C ESAS VARIABLES.
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C NINGUNA
C
C METODO
C REFERENCIAS:
C 1. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C COLUMBUS, OHIO, 1981.
C
C*****
C
C SUBROUTINE URN05(MFSR,P,Q,INTSIZ,JJ,IFLAG)
C COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C COMMON /FIFTH/X(10000)
C INTEGER*4 AA,BB,LCOMPJ,LCOMPK,AUX1,AUX2
C INTEGER A,B,P,Q,MFSR(1)
C CHARACTER*8 NAME
C REAL*8 NAME
C EQUIVALENCE (AA,A),(BB,B),(MCOMPJ,LCOMPJ),(MCOMPK,LCOMPK)
C
C NDIM=NBATCH
C IF(IFLAG.EQ.1)NDIM=1
C N=(2**(INTSIZ-1)-1)*2+1
C
C DO 10 I=1,NDIM
C JJ=JJ+1
C IF(JJ.GT.P)JJ=1
C K=JJ+Q
C IF(K.GT.P)K=K-P
C MCOMPJ=N-MFSR(JJ)
C MCOMPK=N-MFSR(K)
C A=MFSR(K)
C B=MFSR(JJ)

```

```

    AUX1=IAND(LCOMPJ,AA)
    AUX2=IAND(LCOMPK,BB)
    BB=IOR(AUX1,AUX2)
    MFSR(JJ)=B
    X(I)=FLOAT(MFSR(JJ))/FLOAT(N)
10  CONTINUE
    RETURN
    END

C
C
C*****
C   SUBROUTINE RRN06
C*****
C
C   ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN06.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN06I E IMPRIME LOS VALORES INICIALES.
C   URN01 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE) E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN01
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
C   SUBROUTINE RRN06
C   CHARACTER*8 NAME
C   REAL*8 NAME
C   COMMON /FIFTH/X(10000)
C   COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C   COMMON /TSTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C   COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
C   COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
C   DATA YES/'YES '/
C   IF(NEWGEN.NE.0) GOTO 10
C   CALL URN06I
C   WRITE(6,990)IX,JX

C
10  IF(TIMES.EQ.YES)GO TO 20

C
C   GENERA NUMEROS ALEATORIOS SIN CALCULOS DE TIEMPOS
C
C   CALL URN06(IX,JX,X,NBATCH)
C   GOTO 30

C
C   GENERA NUMEROS ALEATORIOS CON CALCULOS DE TIEMPOS
C
20  KTIME=0
C   CALL TIMER(ITIME)
C   CALL URN06(IX,JX,X,NBATCH)
C   CALL TIMER(ITIME)
C   KTIME=KTIME+ITIME

C
30  IF(NEWGEN.NE.0) GOTO 999
    WRITE(6,400) NBATCH
400  FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
* ' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
    WRITE(6,990) IX,JX
990  FORMAT(' ',8X,'IX=',I11,' JX=',I11)
999  RETURN
    END
C*****

```

```

C SUBROUTINE URN06I = SUBROUTINE URN03I(EN URN03I, VER ENTRY URN06I)
C*****
C
C*****
C   SUBROUTINE URN06(IX,JX,X,NBATCH)
C*****
C
C   BASADA EN UNA FUNCION DADA POR D. LURIE AND R.L. MASON:
C   "EMPIRICAL INVESTIGATION OF SEVERAL TECHNIQUES FOR COMPUTER
C   GENERATION OF ORDER STATISTICS",COMMUNICATIONS IN STATISTICS,
C   VOL.2(1973), PAG.363-371, PARA GENERAR NUMEROS ALEATORIOS EN PUNTO
C   FLOTANTE EN EL PAQUETE IRCRAND (REFERENCIA 1 ), LA FUNCION USADA
C   POR ESTE METODO FUE LLAMADA "RN1".
C
C PROPOSITO
C   GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE
C   PRECISION EN (0.0,1.0)
C
C USO
C   CALL URN06(IX,JX,X,NBATCH)
C
C DESCRIPCION DE PARAMETROS
C   IX...ES EL VALOR INICIAL DE UNA SUCESION Y PUEDE SE CUALQUIER
C   ENTERO ENTRE 1 Y 2147483647.(2**31 -1)
C   JX...ES EL VALOR INICIAL DE UNA SUCESION Y PUEDE SER CUALQUIER
C   ENTERO ENTRE 1 Y 2147483647.JX NO DEBER SER IGUAL A IX.
C   X... ES EL ARREGLO RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO FLOTAN-
C   TE DE SIMPLE PRECISION EN (0.0,1.0)
C   NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR
C   CON EL LLAMADO DE ESTA SUBRUTINA
C
C OBSERVACIONES
C   IX Y JX NO DEBEN SER ALTERADOS POR EL USUARIO DURANTE LA EJECUCION
C   DEL PROGRAMA, SALVO QUE EL USUARIO DECIDA REPETIR UNA SUCESION DE
C   NUMEROS ALEATORIOS.
C
C SUBRUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C METODOD
C   REFERENCIAS:
C   1. DUDEWICZ,E.J.: "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM
C   NUMBER GENERATOR PACKAGE",TECHNICAL REPORT, DEPARTMENT OF
C   STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
C   2. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C   GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C   COLUMBUS, OHIO, 1981.
C
C*****
C
C   SUBROUTINE URN06(IX,JX,X,NBATCH)
C   DIMENSION X(NBATCH)
C   DO 10 I=1,NBATCH
C   IX=IX*65539
C   JX=JX*262147
C   X(I)=0.4656613E-9*FLOAT(IABS(JX+IX))
10 CONTINUE
C   RETURN
C   END
C
C

```

```

C*****
C      SUBROUTINE RRN07
C*****
C
C      ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C      GENERADOR URN07.
C      EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C      A URN07I E IMPRIME LOS VALORES INICIALES.
C      URN07 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C      QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIAR URN07
C      DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C
C      SUBROUTINE RRN07
C      CHARACTER*8 NAME
C      REAL*8 NAME
C      COMMON /FIFTH/X(1000)
C      COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C      COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C      COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
C      DATA YES/'YES '/
C      IF(NEWGEN.NE.0) GOTO 10
C      CALL URN07I
C      WRITE(6,990)IX
990 FORMAT(' ',8X,'IX=',I11)
10 IF(TIMES.EQ.YES)GO TO 20
C
C      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C      CALL URN07(IX,X,NBATCH)
C      GOTO 30
C
C      GENERACION DE NUEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20 KTIME=0
C      CALL TIMER(ITIME)
C      CALL URN07(IX,X,NBATCH)
C      CALL TIMER(ITIME)
C      KTIME=KTIME+ITIME
C
30 IF(NEWGEN.NE.0) GOTO 999
C      WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
* ' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C      WRITE(6,990) IX
999 RETURN
C      END
C
C*****
C      SUBROUTINE URN07I = SUBROUTINE URN01I(EN URN01I, ENTRY URN07I)
C      SUBROUTINE URN07 = ASSEMBLER
C*****
C
C*****
C      SUBROUTINE RRN08
C*****
C
C      ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C      GENERADOR URN08.

```

```

C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN08I E IMPRIME LOS VALORES INICIALES.
C   URN08 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE) E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN08
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C
C   SUBROUTINE RRN08
C   CHARACTER*8 NAME
C   REAL*8 NAME
C   COMMON /FIFTH/X(10000)
C   COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C   COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C   COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C   COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
C   DATA YES/'YES '/
C   IF(NEWGEN.NE.0) GOTO 10
C   CALL URN08I
C   IX=524287
C   WRITE(6,990)IX
990 FORMAT(' ',8X,'IX=',I11)
10 IF(TIMES.EQ.YES)GO TO 20
C
C   GENERACION DE NUMEROS ALEATORIOS SIN CALCULOS DE TIEMPOS
C
C   CALL URN08(IX,X,NBATCH)
C   GOTO 30
C
C   GENERACION DE NUMEROS ALEATORIOS CON CALCULOS DE TIEMPOS
C
20 KTIME=0
C   CALL TIMER(ITIME)
C   CALL URN08(IX,X,NBATCH)
C   CALL TIMER(ITIME)
C   KTIME=KTIME+ITIME
C
30 IF(NEWGEN.NE.0) GOTO 999
C   WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
* ' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C   WRITE(6,990) IX
999 RETURN
C   END
C
C*****
C   SUBROUTINE URN08I = URN01I (ENTRY URN08I EN URN01I)
C*****
C
C   SUBROUTINE URN08(IX,X,NBATCH)
C*****
C
C   ESTA SUBROUTINA SE BASA EN LA SUBROUTINA DE SSP DE IBM PARA LA
C   GENERACION DE NUMEROS ALEATORIOS ENTEROS Y EN PUNTO FLOTANTE.
C   EN EL PAQUETE IRCRAND (REFERENCIA 2),LA SUBROUTINA EMPLEADA PARA
C   GENERAR NUMEROS ALEATORIOS EN SIMPLE PRECISION FUE LLAMADA "RN2"
C
C PROPOSITO
C   GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE
C   PRECISION EN (0.0,1.0), Y NUMEROS ALEATORIOS ENTEROS K , SIENDO
C   0<K<2**31.

```

```

C
C USO
C   CALL URN08(IX,X,NBATCH)
C
C DESCRIPCION DE PARAMETROS
C   IX...PARA EL PRIMER VALOR, QUE PUEDE SER CUALQUIER NUMERO ENTERO
C       IMPAR ENTRE 1 Y 999999999.
C       LOS VALORES SUCEIVOS DE IX DEBERAN SER DEL CONJUNTO DE
C       VALORES PREVIOS DE IY CALCULADOS POR ESTA RUTINA.
C   X... ES EL VECTOR RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO
C       FLOTANTE DE SIMPLE PRECISION EN (0.0,1.0)
C   NBATCH...ES LA CANTIDAD DE NUMEROS ALETORIOS QUE SE DESEA GENERAR
C       CON EL LLAMADO DE ESTA SUBRUTINA.
C
C OBSERVACIONES
C   ESTA SUBRUTINA ES ESPECIFICA DE LOS COMPUTADORES IBM SISTEMAS/360
C       /370.CORRESPONDE A LA SUBRUTINA RANDU DE LA REFERENCIA 1 .
C
C SUBRUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C METODO
C   REFERENCIAS:
C   1. INTERNATIONAL BUSINESS MACHINES CORPORATION, "SYSTEM/360
C       SCIENTIFIC SUBROUTINE PACKAGE, VERSION III, PROGRAMMER'S MANUAL
C       PROGRAM NUMBER 360A-CM-03X",MANUAL GH20-0205-4 (FIFTH EDITION)
C       IBM CORPORATION, WHITE PLAINS, NEW YORK, AUGUST 1970, P.77;
C   2. DUDEWICZ,E.J.: "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM
C       NUMBER GENERATOR PACKAGE",TECHNICAL REPORT, DEPARTMENT OF
C       STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
C   3. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C       GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C       COLUMBUS, OHIO, 1981.
C
C*****
C
C   SUBROUTINE URN08(IX,X,NBATCH)
C   DIMENSION X(NBATCH)
C   DO 10 I=1,NBATCH
C     IY=IX*65539
C     IF(IY)5,6,6
C 5    IY=IY+2147483647+1
C 6    X(I)=IY
C     X(I)=X(I)*.4656613E-9
C     IX=IY
C 10  CONTINUE
C     RETURN
C     END
C
C
C*****
C   SUBROUTINE RRN09
C*****
C
C   ESTA SUBRUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN09.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN09I E IMPRIME LOS VALORES INICIALES.
C   URN09 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE) E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN09
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.

```

```

C
C*****
C
      SUBROUTINE RRN09
      CHARACTER*8 NAME
C     REAL*8 NAME
      COMMON /FIFTH/X(1000)
      COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
      COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
      COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
      DATA YES/'YES '/
      IF(NEWGEN.NE.0) GOTO 10
      CALL URN09I
      WRITE(6,990)IX
990  FORMAT(' ',8X,'IX=',I11)
      10 IF(TIMES.EQ.YES)GO TO 20
C
C     GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
      CALL URN09(IX,X,NBATCH)
      GOTO 30
C
C     GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
      20 KTIME=0
      CALL TIMER(ITIME)
      CALL URN09(IX,X,NBATCH)
      CALL TIMER(ITIME)
      KTIME=KTIME+ITIME
C
      30 IF(NEWGEN.NE.0) GOTO 999
      WRITE(6,400) NBATCH
400  FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
      *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
      WRITE(6,990) IX
999  RETURN
      END
C
C*****
C     SUBROUTINE URN09I = SUBROUTINE URN01I (ENTRY URN09I EN URN01I)
C*****
C
C*****
C     SUBROUTINE URN09(IX,X,NBATCH)
C*****
C
C     ESTA SUBRUTINA ES LA SUBRUTINA SSP DE IBM PARA GENERAR NUMEROS
C     ALEATORIOS ENTEROS Y EN PUNTO FLOTANTE.EL CODIGO USADO DIFIERE DEL
C     CODIGO DE SSP(DONDE SE EMPLEA COMO RN2) .EL PRESENTE CODIGO SE
C     ENCUENTRA EN EL LISTADO FUENTE DE N. NIE, D.H. BENT, AND C.H.
C     HULL, "SPSS-STATISTICAL PACKAGE FOR THE SOCIAL SCIENCES", MCGRAW-
C
C     HILL, INC., NEW YORK, 1970. EN EL PAQUETE IRCCRAND (REF.1), LA
C     SUBRUTINA USADA PARA GENERAR NUMEROS ALEATORIOS EN SIMPLE
PRECISION
C     SE DENOMINA "RN4" .
C
C PROPOSITO
C     GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE
SIMPLE

```

```

C   PRECISION EN (0.0,1.0)
C
C   USO
C   CALL URN09(IX,X,NBATCH)
C
C   DESCRIPCION DE PARAMETROS
C   IX...ES EL VALOR INICIAL Y PUEDE SER CUALQUIER ENTERO IMPAR ENTRE
C       1 AND 999999999.
C   X... ES EL VECTOR RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO FLOTAN-
C       TE DE SIMPLE PRECISION EN (0.0, 1.0)
C   NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR
C       CON EL LLAMADO DE ESTA SUBROUTINA.
C
C   OBSERVACIONES
C   IX NO DEBE SER ALTERADO POR EL USUARIO DURANTE LA EJECUCION DEL
C   PROGRAMA, SALVO QUE EL DECIDA REPETIR UNA SUCCESION DE NUMEROS
C   ALEATORIOS.
C
C   SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C   METODO
C   REFERENCIAS:
C   1. DUDEWICZ,E.J.: "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM
C       NUMBER GENERATOR PACKAGE",TECHNICAL REPORT, DEPARTMENT OF
C       STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
C   2. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C       GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C       COLUMBUS, OHIO, 1981.
C
C*****
C
C   SUBROUTINE URN09(IX,X,NBATCH)
C   DIMENSION X(NBATCH)
C   DO 10 I=1,NBATCH
C   IY=IX*65539
C   IF(IY.GE.0) GO TO 6
C   IY=IY+2147483647
C   IY=IY+1
C 6   X(I)=FLOAT(IY)*.4656613E-9
C   IX=IY
C 10  CONTINUE
C   RETURN
C   END
C
C
C*****
C
C   SUBROUTINE RRN10
C*****
C
C   ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN10.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN10I E IMPRIME LOS VALORES INICIALES.
C   URN10 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN10
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C   SUBROUTINE RRN10
C   CHARACTER*8 NAME

```



```

C      REAL*8 NAME
COMMON /FIFTH/X(10000)
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
DATA YES/'YES '/
IF(NEWGEN.NE.0) GOTO 10
CALL URN10I
WRITE(6,990)IX
990 FORMAT(' ',8X,'IX=',I11)
10 IF(TIMES.EQ.YES)GO TO 20

C
C      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C      CALL URN10(IX,X,NBATCH)
C      GOTO 30

C
C      GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C      20 KTIME=0
C      CALL TIMER(ITIME)
C      CALL URN10(IX,X,NBATCH)
C      CALL TIMER(ITIME)
C      KTIME=KTIME+ITIME

C
C      30 IF(NEWGEN.NE.0) GOTO 999
C      WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
* ' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C      WRITE(6,990) IX
999 RETURN
C      END

C*****
C      SUBROUTINE URN10I = SUBROUTINE URN01I(ENTRY URN10I EN URN01I)
C*****
C
C*****
C      SUBROUTINE URN10(IX,X,NBATCH)
C*****
C
C      SUBROUTINA DESTACADA POR SU PORTABILIDAD SUGERIDA POR BY J.B.
C      KRUSKAL, Y USADA POR OMNITAB 2, PARA GENERAR NUMEROS ALEATORIOS
C      EN PUNTO FLOTANTE. EN EL PAQUETE IRCCRAND (REFERENCIA 2 ), LA
C      SUBROUTINA QUE USA ESTE METODO PARA GENERAR NUMEROS ALEATORIOS EN
C      SIMPLE PRECISION SE LLAMA "RN5".
C
C      PROPOSITO
C      GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE
C      PRECISION EN (0.0,1.0)
C
C      USO
C      CALL URN10(IX,X,NBATCH)
C
C      DESCRIPCION DE PARAMETROS
C      IX...PARA EL PRIMER VALOR, QUE DEBE SER PUESTO IGUAL A 1.LOS VALO-
C      RES SUCEIVOS DE IX DEBERAN SER DEL CONJUNTO DE VALORES
C      PREVIOS DE IY CALCULADOS POR ESTA RUTINA.
C      X... ES EL VECTOR RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO FLOTAN-
C      TE DE SIMPLE PRECISION EN (0.0,1.0)
C      NBATCH...ES LA CANTIDAD DE NUMEROS ALETORIOS QUE SE DESEA GENERAR

```

```

C          CON EL LLAMADO DE ESTA SUBROUTINA.
C
C OBSERVACIONES
C   ESTA SUBROUTINA ES GENERAL...(VER REFERENCIA 1)
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C METODO
C   REFERENCIAS:
C   1. KRUSKAL, J.B. , "EXTREMELY PORTABLE RANDOM NUMBER GENERATOR",
C     COMMUNICATIONS OF THE ACM, VOL.12(1969), PAG. 93-94.
C   2. DUDEWICZ,E.J.: "IRCCRAND-THE OHIO STATE UNIVGRSITY RANDOM
C     NUMBER GENERATOR PACKAGE",TECHNICAL REPORT, DEPARTMENT OF
C     STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
C   3. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C     GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C     COLUMBUS, OHIO, 1981.
C*****
C
C   SUBROUTINE URN10(IX,X,NBATCH)
C     DATA M,FLM/8192,8192./
C     DIMENSION X(NBATCH)
C     DO 110 J=1,NBATCH
C       K=IX
C       DO 10 I=1,3
10    K=MOD(5*K,M)
C       X(J)=FLOAT(K)/FLM
C       IY=K
C       IX=IY
110  CONTINUE
C     RETURN
C     END
C
C
C*****
C   SUBROUTINE RRN11
C*****
C
C   ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN11.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN11I E IMPRIME LOS VALORES INICIALES.
C   URN11 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN11
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C   SUBROUTINE RRN11
C     CHARACTER*8 NAME
C
C     REAL*8 NAME
C     COMMON /FIFTH/X(1000)
C     COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C     COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C     COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
C     COMMON /RPRMS/C,P,Q,ZZ
C     COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
C
C     IF(NEWGEN.NE.0) GOTO 10
C     CALL URN11I
C     WRITE(6,990)IX,L,C,ZZ

```

```

990 FORMAT(' ',8X,'IX=',I11,' L=',I11,' C=',F15.1,'
ZZ=',F15.1)
10 IF(TIMES.EQ.YES)GO TO 20
C
C      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C      CALL URN11(IX,X,L,C,ZZ,NBATCH)
C      GOTO 30
C
C      GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C      20 KTIME=0
C      CALL TIMER(ITIME)
C      CALL URN11(IX,X,L,C,ZZ,NBATCH)
C      CALL TIMER(ITIME)
C      KTIME=KTIME+ITIME
C
C      30 IF(NEWGEN.NE.0) GOTO 999
C      WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
*' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C      WRITE(6,990) IX,L,C,ZZ
999 RETURN
C      END
C
C*****
C      SUBROUTINE URN11I
C*****
C
C      ESTA SUBRUTINA INICIALIZA URN11. LAS OPCIONES DISPONIBLES SON:
C
C      IOPT=0 INICIALIZACION POR DEFECTO
C      IOPT=1 TODOS LOS VALORES LEIDOS EN LA RUTINA
C      IOPT=2 VALORES DE INICIALIZACION PROVISTOS POR EL USUARIO
C
C*****
C
C      SUBROUTINE URN11I
C      COMMON /URNPRM/NAME ,NBATCH ,TIMES ,IPRINT ,IOPT
C      COMMON /RPRMS/C ,P ,Q ,ZZ
C      COMMON /IPRMS/IX ,INTSIZ ,JJ ,JX ,KKK ,L ,ML ,MK ,MMM
C      CHARACTER*8 NAME
C      REAL*8 NAME
C      IF(IOPT.EQ.1)GO TO 10
C      IF(IOPT.EQ.2)GO TO 30
C      IX=1
C      L=452807053
C      C=0.0
C      ZZ=2.0**31
C      GO TO 30
10 READ(5,100)IX,L,C2,ZZ2
100 FORMAT(2I11,2F16.0)
30 RETURN
C      END
C
C*****
C      SUBROUTINE URN11(IX,X,L,C,ZZ,NBATCH)
C*****
C
C      ESTA SUBRUTINA GENERA NUMEROS ALEATORIOS EN PUNTO FLOTANTE

```

```

C   USANDO UN GENERADOR CONGRUENCIAL MIXTO QUE ES ESPECIFICADO POR
C   EL USUARIO. EN EL PAQUETE IRCRAND (REFERENCIA 2),UNA SUBROUTINA
C   QUE USA ESTE METODO PARA GENERAR NUMEROS ALEATORIOS EN SIMPLE
C   PRECISION SE DENOMINA "RNCG".
C
C PROPOSITO
C   GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE
C   PRECISION EN (0.0,1.0)
C
C USO
C   CALL URN11(IX,X,L,C,ZZ,NBATCH)
C
C DESCRIPCION DE PARAMETROS
C   IX...PARA EL PRIMER VALOR DEBE SER PUESTO A 1.(UN USUARIO QUE
C   ENTIENDE LA OPERACION DE ESTE PROGRAMA PUEDE SEGURAMENTE
C   USAR OTROS VALORES INICIALES)
C   X...ES EL ARREGLO RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO FLO-
C   TANTE DE SIMPLE PRECISION EN (0.0,1.0).
C   L...ES EL ENTERO MULTIPLICATIVO DEL GENERADOR.
C   C...ES LA CONSTANTE ADITIVA DEL GENERADOR. (EL GENERADOR ES
C   DENOMINADO "CONGRUENCIAL MIXTO" SI C ES DISTINTO DE CERO
C   Y "CONGRUENCIAL MULTIPLICATIVO" SI C=0.)
C   ZZ...ES EL MODULO POR EL CUAL LOS NUMEROS SON REDUCIDOS
C   NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR
C   CON EL LLAMADO DE ESTA SUBROUTINA.
C
C OBSERVACIONES
C   PARA VER ALGUNAS SELECCIONES HABITUALES DE L,C,ZZ ,REFERENCIA 1
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C METODO
C   REFERENCIAS:
C   1. COVEYOU, R.R. AND MACPHERSON, R.D.: "FOURIER ANALYSIS OF
C   UNIFORM RANDOM NUMBER GENERATORS",JOURNAL OF THE ACM, VOL.14
C   (1967), PAG.100-119.
C   2. DUDEWICZ,E.J.: "IRCRAND-THE OHIO STATE UNIVERSITY RANDOM
C   NUMBER GENERATOR PACKAGE",TECHNICAL REPORT, DEPARTMENT OF
C   STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
C   3. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C   GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C   COLUMBUS, OHIO, 1981.
C
C*****
C
SUBROUTINE URN11(IX,X,L,C,ZZ,NBATCH)
DIMENSION X(NBATCH)
DO 10 I=1,NBATCH
K=IX
Z=L*K+C
Z=AMOD(Z,ZZ)
Z=ABS(Z)
K=Z
X(I)=Z/ZZ
IY=K
IX=IY
10 CONTINUE
RETURN
END
C

```

```

C
C*****
C   SUBROUTINE RRN12(IX,L,C2,ZZ2)
C*****
C
C   ESTA SUBRUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN12.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN12I E IMPRIME LOS VALORES INICIALES.
C   URN12 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE) E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN12
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C   SUBROUTINE RRN12(IX,L,C2,ZZ2)
C   CHARACTER*8 NAME
C
C   REAL*8 C2,ZZ2
C   COMMON /FIFTH/X(10000)
C   COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C   COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C   COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
C
C   IF(NEWGEN.NE.0) GOTO 10
C   CALL URN12I(IX,L,C2,ZZ2)
C   WRITE(6,990) IX,L,C2,ZZ2
990 FORMAT(9X,'IX=',I11,' L=',I11,' C2=',F15.1,' ZZ2=',F15.1)
10 IF(TIMES.EQ.YES)GO TO 20
C
C   GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C   CALL URN12(IX,X,L,C2,ZZ2,NBATCH)
C   GOTO 30
C
C   GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C
20 KTIME=0
C   CALL TIMER(ITIME)
C   CALL URN12(IX,X,L,C2,ZZ2,NBATCH)
C   CALL TIMER(ITIME)
C   KTIME=KTIME+ITIME
30 IF(NEWGEN.NE.0) GOTO 999
C   WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
* ' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C   WRITE(6,990) IX,L,C2,ZZ2
999 RETURN
C   END
C
C*****
C   SUBROUTINE URN12I(IX,L,C2,ZZ2)
C*****
C
C   ESTA SUBRUTINA INICIALIZA URN12. EL GENERADOR ES UNA
C   VERSION DOBLE PRECISION DE URN11 Y REQUIERE VARIABLES
C   DOBLE PRECISION. LAS OPCIONES DISPONIBLES SON:
C
C   IOPT=0 INICIALIZACION POR DEFECTO
C   IOPT=1 TODOS LOS VALORES LEIDOS EN LA RUTINA
C   IOPT=2 VALORES DE INICIALIZACION PROVISTOS POR EL USUARIO

```

```

C
C*****
C
      SUBROUTINE URN12I(IX,L,C2,ZZ2)
      COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
      CHARACTER*8 NAME
      REAL*8 ZZ2,C2
      IF(IOPT.EQ.1)GO TO 10
      IF(IOPT.EQ.2)GO TO 30
      IX=1
      L=452807053
      C2=0.D0
      ZZ2=2147483648.D0
      GO TO 30
10  READ(5,600)IX,L,C2,ZZ2
600  FORMAT(2I11,2D16.0)
30  RETURN
      END
C
C*****
C      SUBROUTINE URN12(IX,X,L,C2,ZZ2,NBATCH)
C*****
C
C      ESTA SUBRUTINA GENERA NUMEROS ALEATORIOS EN PUNTO FLOTANTE
C      USANDO UN GENERADOR CONGRUENCIAL MIXTO QUE ES ESPECIFICADO POR
C      EL USUARIO. EN EL PAQUETE IRCRAND (REFERENCIA 2),UNA SUBRUTINA
C      QUE USA ESTE METODO PARA GENERAR NUMEROS ALEATORIOS EN SIMPLE
C      PRECISION SE DENOMINA "RNCG".
C
C      PROPOSITO
C      GENERAR VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE SIMPLE
C      PRECISION EN (0.0,1.0)
C
C      USO
C      CALL URN12(IX,X,L,C2,ZZ2,NBATCH)
C
C      DESCRIPCION DE PARAMETROS
C      IX...PARA EL PRIMER VALOR DEBE SER PUESTO A 1.(UN USUARIO QUE
C      ENTIENDE LA OPERACION DE ESTE PROGRAMA PUEDE SEGURAMENTE
C      USAR OTROS VALORES INICIALES)
C      LOS VALORES SUCEIVOS DE IX DEBEN TOMARSE DEL CONJUNTO
C      DE LOS VALORES PREVIOS IY CALCULADOS POR ESTA RUTINA
C      X...ES EL ARREGLO RESULTANTE DE NUMEROS ALEATORIOS EN PUNTO FLO-
C      TANTE DE SIMPLE PRECISION EN (0.0,1.0).
C      L...ES EL ENTERO MULTIPLICATIVO DEL GENERADOR.
C      C2...ES LA CONSTANTE ADITIVA DEL GENERADOR. (EL GENERADOR ES
C      DENOMINADO "CONGRUENCIAL MIXTO" SI C2 ES DISTINTO DE CERO
C      Y "CONGRUENCIAL MULTIPLICATIVO" SI C2=0.)
C      ZZ2...ES EL MODULO POR EL CUAL LOS NUMEROS SON REDUCIDOS
C      NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR
C      CON EL LLAMADO DE ESTA SUBRUTINA.
C
C      OBSERVACIONES
C      ESTA SUBRUTINA ES UNA VERSION EN DOBLE PRECISION DE URN11 (LOS
C      CALCULOS INTERMEDIOS SON EN DOBLE PRECISION PARA INCREMENTAR LA
C      PRECISION).RETORNA NUMEROS ALEATORIOS EN SIMPLE PRECISION.
C
C      SUBRUTINAS Y FUNCIONES REQUERIDAS
C      NINGUNA
C
C      METODO

```

```

C   REFERENCIAS:
C   1. COVEYOU, R.R. AND MACPHERSON, R.D.: "FOURIER ANALYSIS OF
C       UNIFORM RANDOM NUMBER GENERATORS", JOURNAL OF THE ACM, VOL.14
C       (1967), PAG.100-119.
C   2. DUDEWICZ, E.J.: "IRCCRAND-THE OHIO STATE UNIVERSITY RANDOM
C       NUMBER GENERATOR PACKAGE", TECHNICAL REPORT, DEPARTMENT OF
C       STATISTICS, THE OHIO STATE UNIVERSITY, COLUMBUS, OHIO, 1974.
C   3. DUDEWICZ, E.J. AND RALLEY, T.: "THE HANDBOOK OF RANDOM NUMBER
C       GENERATION AND TESTING", AMERICAN SCIENCES PRESS,
C       COLUMBUS, OHIO, 1981.
C
C*****
C
C       SUBROUTINE URN12(IX,X,L,C2,ZZ2,NBATCH)
C       REAL*8 ZZ2,C2,Z
C       DIMENSION X(NBATCH)
C       DO 10 I=1,NBATCH
C         K=IX
C         Z=L*K+C2
C         Z=DMOD(Z,ZZ2)
C         IF(Z.GT.0.D0)GO TO 12
C         Z=Z+ZZ2
12      K=Z
C         X(I)=Z/ZZ2
C         IY=K
C         IX=IY
10     CONTINUE
C       RETURN
C       END
C
C*****
C       SUBROUTINE RRN13(X,NBATCH,TIMES)
C       ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C       GENERADOR URN13.
C       EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C       A URN33I E IMPRIME LOS VALORES INICIALES.
C       URN33 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO
C       REQUIERE)
C       E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN13 DESPUES
C       DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C
C       SUBROUTINE RRN13(X,NBATCH,TIMES)
C       INTEGER JEN,LCB2,RSUM,RB1C1,B1,B2,C1,C2
C       DIMENSION X(10000)
C
C       COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
C       COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C       COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C
C       IF(NEWGEN.NE.0) GOTO 10
C
C       CALL URN13I
C       WRITE(6,990)INTIN(1),INTIN(2),INTIN(3),INTIN(4),INTIN(5)
C
C   10 IF(TIMES.EQ.YES)GO TO 20
C
C       GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C       CALL URN13(X,NBATCH)
C       GOTO 30

```

```

C
C           GENERACION DE NUMEROS ALEATORIOS   CON CALCULO DE TIEMPOS
C
C   20 KTIME=0
C       CALL TIMER(ITIME)
C       CALL URN13 (X,NBATCH)
C       CALL TIMER(ITIME)
C       KTIME=KTIME+ITIME
C
C   30 IF(NEWGEN.NE.0) GOTO 999
C       WRITE(6,400) NBATCH
400  FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
* ' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C       WRITE(6,990)INTIN(1),INTIN(2),INTIN(3),INTIN(4),INTIN(5)
990  FORMAT(1X,'INTIN(1)=' ,I15,5X, 'INTIN(2)=' ,I15,
/1X,'INTIN(3)=' ,I15,
*5X,'INTIN(4)=' ,I15,/1X,'INTIN(5)=' ,I15)
C   999 RETURN
C       END
C
C*****
C       SUBROUTINE URN13I
C*****
C   EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION DEFAULT.(IOPT=0)
C   NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
*
C
C       SUBROUTINE URN13I
C       INTEGER JEN,LCB2,RSUM,RB1C1,B1,B2,C1,C2
C       DIMENSION X(10000)
C
C       COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
C       COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C       COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C
C
C   IR=0
C   IA=5
C   MULT= 663608941
C   INTIN(1)=524287
C   INTIN(2)=INTIN(1)/32768
C   J = INTIN(1)*65536
C   IF (J.LT.0) J=J+ 2147483647 +1
C   INTIN(3)= J/65536
C   INTIN(4)= MULT/32768
C   J= MULT*65536
C   IF (J.LT.0) J= J+ 2147483647 +1
C   INTIN(5) = J/65536
C   RETURN
C   END
C
C*****
C       SUBROUTINE URN13(X,NBATCH)
C*****
C
C   PROPOSITO
C   GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C   PRECISION EN (0.0,1.0)
C
C   USO

```



```

C CALL URN13(X,NBATCH)
C
C DESCRIPCION DE PARAMETROS
C   IA.....DIMENSION VECTOR DE NUMEROS ENTEROS
C   IR.....DIMENSION VECTOR DE NUMEROS REALES
C   NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA
GENERAR
C           MEDIANTE ESTA SUBROUTINA
C   INTIN(1),INTIN(2),INTIN(3),INTIN(4),INTIN(5) VALORES ENTEROS
C
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C METODO
C   UN GENERADOR CONGRUENCIAL MULTIPLICATIVO
C    $X_{i+1} = 663608941 * X_i \pmod{2^{**}32}$ 
C
C REFERENCIAS:
C   AHRENS Y DIETER (1974)
C
C*****
C
C   SUBROUTINE URN13(X,NBATCH)
C   INTEGER JEN,LCB2,RSUM,RB1C1,B1,B2,C1,C2
C   DIMENSION X(10000)
C
C   COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
C   COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C   COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
C
C
C   C1=INTIN(4)
C   C2=INTIN(5)
C   B1=INTIN(2)
C   B2=INTIN(3)
C
C
C   DO 10 I =1 ,NBATCH
C   J2 = (B2*C1 + B1*C2)*16384
C   IF (J2.LT.0) J2=J2+ 2147483647 + 1
C   RSUM=J2/16384
C   J3= B1*C1*536870912
C   IF (J3.LT.0) J3=J3+ 2147483647 + 1
C   RB1C1 = J3/536870912
C   LCB2 = (C2*B2)/32768
C   J4= ((LCB2 + RSUM + RB1C1 * 32768)* 16384)
C   IF (J4.LT.0) J4=J4+ 2147483647 + 1
C   B1 = J4/16384
C   JEN = B1/4
C   J5= C2*B2*65536
C   IF (J5.LT.0) J5=J5+ 2147483647 + 1
C   B2 = J5/65536
C   X(I) = FLOAT(JEN)/ 32768.0
C 10 CONTINUE
C   INTIN(2)=B1
C   INTIN(3)=B2
C   RETURN
C   END
C*****
C   SUBROUTINE RRN14
C*****

```

```

C
C   ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN14.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN14I E IMPRIME LOS VALORES INICIALES.
C   URN14 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN14
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C   SUBROUTINE RRN14
C     CHARACTER*8 NAME
C
C     REAL*8 NAME
C     INTEGER TABLE(64)
C     COMMON /FIFTH/X(10000)
C     COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C     COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C     COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
C     COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
C
C     IF(NEWGEN.NE.0) GOTO 10
C     CALL URN14I(TABLE)
C     WRITE(6,990)IX,JX
C     IF(IOPT.EQ.0) GOTO 10
C     WRITE(6,992)
992  FORMAT(1X,'NOTA: EL VALOR INICIAL DE IX PROVISTO FUE USADO PARA
      *FORMAR UNA TABLA DE 64 NUMEROS,/1X,ALEATORIOS PARA USO INTERNO.
      * ,/1X, EL VALOR IMPRESO DE IX ES EL 65-AVO EN LA SUCESION.')
10   IF(TIMES.EQ.YES)GOTO 20
C
C     GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C     CALL URN14(IX,JX,X,NBATCH,TABLE)
C     GOTO 30
C
C     GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C     20 KTIME=0
C     CALL TIMER(ITIME)
C     CALL URN14(IX,JX,X,NBATCH,TABLE)
C     CALL TIMER(ITIME)
C     KTIME=KTIME+ITIME
C
C     30 IF(NEWGEN.NE.0) GOTO 999
C     WRITE(6,400) NBATCH
400  FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
      *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
      WRITE(6,990) IX,JX
990  FORMAT(' ',8X,'IX=',I11,' JX=',I11)
999  RETURN
      END
C
C*****
C   SUBROUTINE URN14I(TABLE)
C*****
C
C   ESTA SUBROUTINA INICIALIZA URN14. LAS OPCIONES DISPONIBLES SON:
C
C   IOPT=0   INICIALIZACION POR DEFECTO
C   IOPT=1   INICIALIZACION DE VALORES, EXCEPTO PARA EL ARREGLO
C   TABLA

```

```

C          LEIDA EN LA RUTINA
C      IOPT=2  INICIALIZACION DE VALORES PROVISTOS POR EL USUARIO
C
C*****
C
C      SUBROUTINE URN14I(TABLE)
C          CHARACTER*8 NAME
C          REAL*8 NAME
C      COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C      COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKK,L,ML,MK,MMM
C      INTEGER TABLE(64)
C      DIMENSION ITABLE(64)
C      DATA ITABLE/ 453816693, 859273406, 1973808427, 2059993932,
C      * 509510129, 583551786, 2052114759, 810618840, 1560248493,
C      * 1724450006,1084172707, 2001645988, 557869481, 1854755266,
C      * 1609787967,1517139120, 222050533, 73145838, 1785734939,
C      * 1310401788,1741957729, 240883034, 1905540151, 126963336,
C      * 267106845, 247276550, 1192674707, 19119444, 1373938713,
C      * 254375410,1443045167, 2009808224, 458941525, 1668052766,
C      * 806675723,1844566700, 1226880721, 1868649354, 726922023,
C      * 2139021624, 192583565, 1545500470, 737174915, 321051908,
C      * 479937161,1751223842, 1985910815, 887331344, 1163899845,
C      * 1077642318, 632497915, 635970652, 1816864577, 133989818,
C      * 574549527,1837418472, 717428989, 106954342, 879430003,
C      * 2005340340, 613106937, 435722834, 265606415, 197249728/
C
C      IF(IOPT.EQ.0) GOTO 10
C      IF(IOPT.EQ.1) GOTO 20
C      IF(IOPT.EQ.2) GOTO 30
C
C
C      10 IX=197249728
C          JX=0
C          DO 12 K=1,64
C      12 TABLE(K)=ITABLE(K)
C          GOTO 30
C
C
C      20 READ(5,100) IX,JX
C      100 FORMAT(2I11)
C          DO 15 I=1,64
C          TABLE(I)=IX
C          IX=266245*IX+453816693
C          IF(IX.LT.0) IX=IX+2147483647+1
C      15 CONTINUE
C          GOTO 30
C
C      30 RETURN
C          END
C
C*****
C      SUBROUTINE URN14(IX,JX,X,NBATCH,TABLE)
C*****
C
C      ESTA SUBRUTINA ES UNA EXPRESION EN FORTRAN DE UNA PARTE DEL
C      PAQUETE IRAND DE NUMEROS ALEATORIOS DESARROLLADO POR EL DR.
C      RAYMOND Y L. ZARLING EN 1971 EN EL MARIETTA COLLEGE, MARIETTA,
C      OHIO.
C
C      C
C      PROPOSITO
C      GENERA VECTORES DE NUMEROS ALEATORIOS EN PUNTO FLOTANTE DE
C      SIMPLE PRECISION EN (0.0,1.0)
C
C
C      USO

```

```

C   CALL URN14(IX,JX,X,NBATCH,TABLE)
C
C DESCRIPCION DE PARAMETROS
C   IX...ES EL VALOR INICIAL DE UNA PORCION DEL GENERADOR
C       CONGRUENCIAL
C   JX...ES EL INDICE PARA EL PROCESO DE BARAJAR EN LA TABLA
C   X...ES EL ARRAY RESULTADO DE NUMEROS ALEATORIOS EN PUNTO
C       FLOTANTE DE SIMPE PRECISION EN (0.0,1.0)
C   NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA
GENERAR
C       EN LA LLAMADA A URN14
C   TABLE...ES UNA TABLA DE 64 INTEROS ALEATORIOS USADOS INTERNAMENTE
C       POR EL GENERADOR.
C
C OBSERVACIONES
C   EL VALOR DE JX ES ASIGNADO A LA MITAD DE LA PALABRA ENTERA
C   EL VALOR ES DIVIDIDO EN 512 PARA CONSTRUIR UN ENTERO ENTRE 0 Y 63
C   Y LUEGO EMPLEADO COMO INDICE EN UNA TABLA DE 64 ENTEROS
ALEATORIOS.
C   EL VALOR JX ES CONVERTIDO EN UN NUMERO EN PUNTO FLOTANTE Y
C   GUARDADO EN EL ARREGLO X DE NUMEROS ALEATORIOS A SER RETORNADO.
C   EL ENTERO IX ES USADO PARA REEMPLAZAR EL ELEMENTO EN LA TABLA
C   USADO PARA PRODUCIR EL NUMERO ALEATORIO.
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C METODO
C   REFERENCIAS:
C   1. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C       GENERATION AND TESTING",AMERICAN SCIENCES PRESS,
C       COLUMBUS, OHIO, 1981.
C
C*****
C
SUBROUTINE URN14(IX,JX,X,NBATCH,TABLE)
DIMENSION X(NBATCH)
INTEGER TABLE(64)
INTEGER*2 IY
DATA TWO31 /0.4656613E-9/
DO 10 I=1,NBATCH
IX=266245*IX+453816693
IF(IX.LT.0) IX=IX+2147483647+1
101 JX=10924*JX+6925
JX=JX-32769*(JX/32769)
IF(JX.EQ.32768) GOTO 101
IY=JX/512+1
X(I)=TWO31*FLOAT(TABLE(IY))
TABLE(IY)=IX
10 CONTINUE
RETURN
END
C
C
C*****
C   SUBROUTINE RRN15
C*****
C
C   ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
GENERADOR URN15.

```

```

C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN15I E IMPRIME LOS VALORES INICIALES.
C   URN15 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN15
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
C   SUBROUTINE RRN15
C   CHARACTER*8 NAME
C
C   REAL*8 NAME
COMMON /FIFTH/X(10000)
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM

C   IF(NEWGEN.NE.0) GOTO 10
C
C   CALL URN15I
C   WRITE(6,990)IX
990 FORMAT(' ',8X,'IX=',I11)
C
C   10 IF(TIMES.EQ.YES)GO TO 20
C
C   GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C   CALL URN15(IX,X,NBATCH)
C   GOTO 30
C
C   GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C   20 KTIME=0
C   CALL TIMER(ITIME)
C   CALL URN15(IX,X,NBATCH)
C   CALL TIMER(ITIME)
C   KTIME=KTIME+ITIME
C   30 IF(NEWGEN.NE.0) GOTO 999
C   WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
* ' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C   WRITE(6,990) IX
999 RETURN
C   END
C
C*****
C   SUBROUTINE URN15I= SUBROUTINE URN01I(EN URN01I, ENTRY URN15I)
C*****
C
C*****
C   SUBROUTINE RRN16
C*****
C
C   ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN16.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN16I E IMPRIME LOS VALORES INICIALES.
C   URN16 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE)
C   E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN16 DESPUES

```

```

C   DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
C   SUBROUTINE RRN16
C     CHARACTER*8 NAME
C
C   REAL*8 NAME
C   COMMON /FIFTH/X(10000)
C   COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C   COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C   COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C   COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM
C
C   IF(NEWGEN.NE.0) GOTO 10
C   CALL URN16I
C   WRITE(6,990)IX
990 FORMAT(' ',8X,'IX=',I11)
10 IF(TIMES.EQ.YES)GO TO 20
C
C   GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C   CALL URN16(IX,X,NBATCH)
C   GOTO 30
C
C   GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20 KTIME=0
C   CALL TIMER(ITIME)
C   CALL URN16(IX,X,NBATCH)
C   CALL TIMER(ITIME)
C   KTIME=KTIME+ITIME
C
30 IF(NEWGEN.NE.0) GOTO 999
C   WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
* ' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C   WRITE(6,990) IX
999 RETURN
C   END
C
C*****
C   SUBROUTINE URN16I= SUBROUTINE URN01I(EN URN01I, ENTRY URN16I)
C*****
C
C
C*****
C   SUBROUTINE RRN17
C*****
C
C   ESTA SUBRUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN17.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN17I E IMPRIME LOS VALORES INICIALES.
C   URN17 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN17
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
C   SUBROUTINE RRN17

```

```

CHARACTER*8 NAME

C   REAL*8 NAME
COMMON /FIFTH/X(10000)
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM

C
IF(NEWGEN.NE.0) GOTO 10
CALL URN17I
WRITE(6,990)IX
990 FORMAT(' ',8X,'IX=',I11)
10 IF(TIMES.EQ.YES)GO TO 20

C
C           GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C   CALL URN17(IX,X,NBATCH)
C   GOTO 30

C
C           GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C   20 KTIME=0
C   CALL TIMER(ITIME)
C   CALL URN16(IX,X,NBATCH)
C   CALL TIMER(ITIME)
C   KTIME=KTIME+ITIME

C
30 IF(NEWGEN.NE.0) GOTO 999
WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
* ' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
WRITE(6,990) IX
999 RETURN
END

C
C*****
C   SUBROUTINE URN17I= SUBROUTINE URN01I(EN URN01I, ENTRY URN17I)
C*****
C
C
C*****
C   SUBROUTINE RRN18
C*****
C
C   ESTA SUBRUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN18.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN18I E IMPRIME LOS VALORES INICIALES.
C   URN18 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE) E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN18
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
C   SUBROUTINE RRN18
C   CHARACTER*8 NAME

C   REAL*8 NAME
COMMON /FIFTH/X(10000)

```

```

COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM

IF(NEWGEN.NE.0) GOTO 10
CALL URN18I
WRITE(6,990)IX
990 FORMAT(' ',8X,'IX=',I11)
C
10 IF(TIMES.EQ.YES)GO TO 20
C
C GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C CALL URN18(IX,X,NBATCH)
C GOTO 30
C
C GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20 KTIME=0
C CALL TIMER(ITIME)
C CALL URN18(IX,X,NBATCH)
C CALL TIMER(ITIME)
C KTIME=KTIME+ITIME
C
30 IF(NEWGEN.NE.0) GOTO 999
WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
*' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
WRITE(6,990) IX
999 RETURN
END
C
C*****
C SUBROUTINE URN18I= SUBROUTINE URN01I(EN URN01I, ENTRY URN18I)
C*****
C
C
C*****
C SUBROUTINE RRN19
C*****
C
C ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C GENERADOR URN19.
C EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C A URN19I E IMPRIME LOS VALORES INICIALES.
C URN19 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN19
C DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C
C*****
C
C SUBROUTINE RRN19
C CHARACTER*8 NAME
C
C REAL*8 NAME
COMMON /FIFTH/X(1000)
COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM

```



```

        IF(NEWGEN.NE.0) GOTO 10
        CALL URN19I
        WRITE(6,990)IX
990  FORMAT(' ',8X,'IX=',I11)
        10 IF(TIMES.EQ.YES)GO TO 20
C
C          GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C          CALL URN19(IX,X,NBATCH)
C          GOTO 30
C
C          GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C          20 KTIME=0
C          CALL TIMER(ITIME)
C          CALL URN19(IX,X,NBATCH)
C          CALL TIMER(ITIME)
C          KTIME=KTIME+ITIME
C
C          30 IF(NEWGEN.NE.0) GOTO 999
C          WRITE(6,400) NBATCH
400  FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
        *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
        WRITE(6,990) IX
999  RETURN
        END
C
C*****
C          SUBROUTINE URN19= SUBROUTINE URN01I(EN URN01I, ENTRY URN19)
C*****
C
C
C*****
C          SUBROUTINE RRN20(X,NBATCH,TIMES)
C*****
C
C          ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C          GENERADOR URN20.
C          EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C          A URN20I E IMPRIME LOS VALORES INICIALES.
C          URN20 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C          QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN20
C          DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C          SUBROUTINE RRN20(X,NBATCH,TIMES)
C          DIMENSION X(10000)
C
C          REAL*8 XR,X0,XM,XU
C          COMMON/ TSPTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C          COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C          COMMON /URNPRM/NAME,NBATCH,TIMES,IPRINT,IOPT
C          COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKK,L,ML,MK,MMM
C
C          IF(NEWGEN.NE.0) GOTO 10
C
C          CALL URN20I
C
C          10 IF(TIMES.EQ.YES)GO TO 20
C

```

```

C          GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C          CALL URN20(X,NBATCH)
C          GOTO 30
C
C          GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C          20 KTIME=0
C          CALL TIMER(ITIME)
C          CALL URN20 (X,NBATCH)
C          WRITE(6,990)IX
C          CALL TIMER(ITIME)
C          KTIME=KTIME+ITIME
C
C          30 IF(NEWGEN.NE.0) GOTO 999
C          WRITE(6,400) NBATCH
C          400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
C          *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C          WRITE(6,990)IX
C          990 FORMAT(1X,'IX=',I20)
C          999 RETURN
C          END
C*****
C          SUBROUTINE URN20I= SUBROUTINE URN01I(EN URN01I, ENTRY URN20I)
C*****
C
C*****
C          SUBROUTINE URN20(X,NBATCH)
C*****
C
C          PROPOSITO
C          GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C          PRECISION EN (0.0,1.0)
C
C          USO
C          CALL URN20(X,NBATCH)
C
C          DESCRIPCION DE PARAMETROS
C          XI...SEMILLA NUMERO ENTERO
C          NBATCH.....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENE-
C          RAR MEDIANTE ESTA SUBROUTINA
C
C          SUBROUTINAS Y FUNCIONES REQUERIDAS
C          NINGUNA
C
C          METODO
C
C          Xi+1 = 2027812808 (MOD. 2**31 -1)
C
C          REFERENCIAS:
C          D.C. HOAGLIN (1976)
C*****
C          *
C
C          SUBROUTINE URN20(X,NBATCH)
C          DIMENSION X(10000)
C
C          REAL*8 XR,X0,XM,XU
C          COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM

```

```

COMMON /COUNTS/ICNT, ICNTKS, ICNTMT, NEWGEN
COMMON /IPRMS/IX, INTSIZ, JJ, JX, KKK, L, ML, MK, MMM
COMMON /CODE/GCODE

C
  X0=DFLOAT(IX)
  XM=2147483647D0
  DO 11 I=1, NBATCH
  XR= MOD(2027812808D0*X0, XM)
  X0=XR
C
  X(I)=FLOAT(XR/XM)
  X(I)=XR/XM
11  END DO
  IX=INT(X0)
  RETURN
  END

C
C
C*****
C  SUBROUTINE RRN21(NBATCH)
C*****
C
C  ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C  GENERADOR URN21.
C  EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C  A URN21I E IMPRIME LOS VALORES INICIALES.
C  URN21 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C  QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN21
C  DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C  SUBROUTINE RRN21(NBATCH)

      COMMON /FIFTH/X(10000)
      COMMON /OWNPRM/INTIN(200), REALIN(200), IA, IR
      COMMON /COUNTS/ICNT, ICNTKS, ICNTMT, NEWGEN
      COMMON /TSTTIM/KTIME, CTIME(1000), TIMLOW, LOWSAM
      COMMON /IPRMS/IX, INTSIZ, JJ, JX, KKK, L, ML, MK, MMM

      IF(NEWGEN.NE.0) GOTO 10
      CALL URN21I(M1, M2, M3, NBATCH)
10  IF(TIMES.EQ.YES)GOTO 20
C      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
      CALL URN21(M1, M2, M3, X, NBATCH)
      GOTO 30
C      GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
20  KTIME=0
C      CALL TIMER(ITIME)
      CALL URN21(M1, M2, M3, X, NBATCH)
C      CALL TIMER(ITIME)
C      KTIME=KTIME+ITIME
30  IF(NEWGEN.NE.0) GOTO 999
      WRITE(6, 400) NBATCH
400  FORMAT(1X, 'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
      * ' SIGUIENTE A LOS', I11/1X, 'YA GENERADOS) ES:')
      WRITE(6, 990) M1, M2, M3
990  FORMAT(' ', 8X, ' M1 =', I11, 2X, ' ', M2 =', I11, 2X, ' ', M3 =', I11)
999  RETURN
      END

C
C*****
C  SUBROUTINE URN21I(M1, M2, M3, NBATCH)

```

```

C*****
C   ESTA SUBROUTINA INICIALIZA AQUELLOS GENERADORES QUE UTILIZAN
C   SOLO UN VALOR INICIAL. LAS OPCIONES DISPONIBLES SON:
C
C   IOPT=0  INICIALIZACION POR DEFECTO
C   IOPT=1  INICIALIZACION DE VALORES POR LECTURA EN ESTA RUTINA
C   IOPT=2  INICIALIZACION DE VALORES PROVISTOS POR EL USUARIO
C
C   LOS GENERADORES INICIALIZADOS POR ESTA RUTINA SON LOS SIGUIENTES:
C       URN21
C
C   EN ESTA SUBROUTINA SE UTILIZA LA INICIALIZACION DEFAULT:(IOPT=0)
C   NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
C
C       SUBROUTINE URN21I(M1,M2,M3,NBATCH)
C           M1=32007779
C           M2=23717810
C           M3=52636370
C           RETURN
C       END
C*****
C       SUBROUTINE URN21(M1,M2,M3,NBATCH)
C*****
C   PROPOSITO
C   GENERAR VECTORES DE NUMEROS ALEATORIOS DE SIMPLE PRECISION
C   EN PUNTO FLOTANTE EN (0.0, 1.0)
C
C   USO
C       CALL URN21(M1,M2,M3,NBATCH)
C
C   DESCRIPTION DE PARAMETROS
C       M1,M2,M3...SEMILLAS DE NUMEROS ENTEROS EN EL INTERVALO
C       (0..99999999)
C       NBATCH...LA CANTIDAD DE NUMEROS ALEATORIOS A SER GENERADOS.
C
C   SUBROUTINAS Y FUNCIONES REQUERIDAS
C       NINGUNA
C
C   METODO
C       EL METODO DE GENERACION DE NUMEROS ALEATORIOS, FUE DADO POR
C       C. GARDNER SWAIN Y MARGUERITE S. SWAIN, "A UNIFORM RANDOM
C       NUMBER GENERATOR THAT IS REPRODUCIBLE, HARDWARE-INDEPENDENT,
C       AND FAST", JOURNAL OF CHEMICAL INFORMATION AND COMPUTER
C       SCIENCES, VOL 20 (FEB. 1980) PAG.56-58.
C
C*****
C
C       SUBROUTINE URN21(M1,M2,M3,X,NBATCH)
C           DIMENSION X(NBATCH)
C           DO 10 I=1,NBATCH
C               M4=M1+M2+M3
C               IF (M2.LT.50000000) THEN
C                   M4=M4+1357
C               ENDIF
C               IF (M4.GE.100000000) THEN
C                   M4=M4-100000000
C               ENDIF
C               IF (M4.GE.100000000) THEN
C                   M4=M4-100000000
C

```

```

        ENDIF
        M1=M2
        M2=M3
        M3=M4
        X(I)=(1E-8)*M3
10     CONTINUE
        RETURN
        END

C
C
C
*****
C     SUBROUTINE RRN22(NBATCH)
C*****
C     ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C     GENERADOR URN22.
C     EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C     A URN22I E IMPRIME LOS VALORES INICIALES.
C     URN22 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C     QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN22
C     DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
        SUBROUTINE RRN22(NBATCH)
        COMMON /FIFTH/X(10000)
        COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
        COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
        COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
        COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKK,L,ML,MK,MMM
        DOUBLE PRECISION M1

        IF(NEWGEN.NE.0) GOTO 10
        CALL URN22I(M1,NBATCH)
        WRITE(*,990)M1
10     IF(TIMES.EQ.YES)GOTO 20
C     GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
        CALL URN22(M1,X,NBATCH)
        GOTO 30
C     GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
20     KTIME=0
C     CALL TIMER(ITIME)
        CALL URN22(M1,X,NBATCH)
C     CALL TIMER(ITIME)
C     KTIME=KTIME+ITIME
C
30     IF(NEWGEN.NE.0) GOTO 999
        WRITE(6,400) NBATCH
400    FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
        *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
        WRITE(6,990) M1
990    FORMAT(' ',8X,'M1=',d15.0)
999    RETURN
        END
C*****
C     SUBROUTINE URN22I(M1,NBATCH)
C*****
C
C     EN ESTA SUBROUTINA SE UTILIZA LA INICIALIZACION DEFAULT:(IOPT=0)
C     NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C
        SUBROUTINE URN22I(M1,NBATCH)

```

```

DOUBLE PRECISION M1
M1= 32007779.0D0
RETURN
END
C*****
C   SUBROUTINE URN22(M1,X,NBATCH)
C*****
C
C   ESTA SUBROUTINA GENERA NUMEROS PSEUDO ALEATORIOS CON EL METODO
C   USADO POR LA BIBLIOTECA MT$RANDOM DE VAX
C
C   PROPOSITO:
C   GENERAR VECTORES DE NUMEROS ALEATORIOS DE SIMPLE PRECISION
C   EN PUNTO FLOTANTE EN (0.0, 1.0)
C
C   USO
C   CALL URN22(M1,NBATCH)
C
C   DESCRIPCION DE PARAMETROS
C   M1...SEMILLA DE NUMERO ENTERO EN EL INTERVALO (0..99999999)
C   NBATCH...LA CANTIDAD DE NUMEROS ALEATORIOS A SER GENERADOS.
C
C   SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C   METODO
C*****
C
C   SUBROUTINE URN22(M1,X,NBATCH)
C   DOUBLE PRECISION XX(10000),S1,SK,M1
C   DIMENSION X(10000)
C   SK=2.D0**32
C   DO 10 I=1,NBATCH
C   S1=69069.0D0*M1+1.0D0
C   S1=MOD(S1,SK)
C   XX(I)=S1*0.23283064D-09
C   M1=S1
C   X(I)=XX(I)
10  CONTINUE
RETURN
END
C
C
C*****
C   SUBROUTINE RRN28(X,NBATCH,TIMES)
C*****
C
C   ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN28.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN28I.
C   RRN28 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN28
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C
C   SUBROUTINE RRN28(X,NBATCH,TIMES)
C   CHARACTER*8 NAME
C
C   REAL*8 NAME
C   DIMENSION X(10000)

```

```

COMMON /TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON /IPRMS/IX,INTSIZ,JJ,JX,KKKK,L,ML,MK,MMM

IF(NEWGEN.NE.0) GOTO 10
C
CALL URN28I(IX,NBATCH)
WRITE(6,990)IX
C
10 IF(TIMES.EQ.YES)GO TO 20
C
GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
CALL URN28(IX,X,NBATCH)
GOTO 30
C
GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20 KTIME=0
CALL TIMER(ITIME)
CALL URN28(IX,X,NBATCH)
CALL TIMER(ITIME)
KTIME=KTIME+ITIME
C
30 IF(NEWGEN.NE.0) GOTO 999
WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
*' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
WRITE(6,990)IX
990 FORMAT(1X,'IX=',I20)
999 RETURN
END
C
C*****
C SUBROUTINE URN28I(IX,NBATCH)
C*****
C
EN ESTA SUBROUTINA SE UTILIZA LA INICIALIZACION DEFAULT.(IOPT=0)
C NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
C
SUBROUTINE URN28I(IX,NBATCH)
IX=1
RETURN
END
C
C*****
C SUBROUTINE URN28(IX,X,NBATCH)
C*****
C
PROPOSITO
C GENERAR NUMEROS PSEUDOALEATORIOS EN (0,1) EN PUNTO FLOTANTE CON
C SIMPLE PRECISION
C
USO
C CALL URN28(IX,X,NBATCH)
C
DESCRIPCION DE PARAMETROS
C IX.....VALOR INICIAL DEL GENERADOR, PUEDE SER: 1,..., 2**32-2
C X.....ES EL VECTOR DE LOS NUMEROS ALEATORIOS EN (0.0, 1.0)

```

```

C   NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA
GENERAR
C           MEDIANTE ESTA SUBROUTINA
C
C OBSERVACIONES
C   IX NO DEBE SER ALTERADO POR EL USUARIO DURANTE LA EJECUCION DEL
C   PROGRAMA SALVO QUE SE DESEE REPETIR UNA SUCESION DE NUMEROS
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C METODO
C   UN GENERADOR CONGRUENCIAL MULTIPLICATIVO
C   CON CONSTANTE MULTIPLICATIVA 16807 Y MODULO 2**31 -1
C   REFERENCIAS:
C   SCHRAGE, LINUS. "A MORE PROTABLE FORTRAN RANDOM NUMBER GENERATOR."
C   ACM TRANSACTIONS ON MATHEMATICAL SOFTWARE, JUNIO 1979, PAG.132-138.
C*****
C
SUBROUTINE URN28(IX,X,NBATCH)
DIMENSION X(10000)
INTEGER TWO15,TWO16,P
MULT = 16807
TWO15 = 32768
TWO16 = 65536
P = 2147483647
IIX=IX
DO 10 I=1,NBATCH
KXHI=( IIX- MOD( IIX,TWO16 ))/TWO16
KXALO=( IIX-KXHI*TWO16 )*MULT
KLEFTLO=(KXALO-MOD(KXALO,TWO16))/TWO16
KFHI=KXHI*MULT+KLEFTLO
K=(KFHI-MOD(KFHI,TWO15))/TWO15
IIX=(( (KXALO-KLEFTLO*TWO16) -P)+(KFHI-K*TWO15)*TWO16)+K
IF (IIX.LT.0) THEN
IIX=IIX+P
END IF
X(I)=IIX*4.656612875E-10
10 CONTINUE
IX=IIX
RETURN
END

C
C
C*****
C   SUBROUTINE RRN30(X,NBATCH,TIMES)
C*****
C
C   ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN30.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN30I E IMPRIME LOS VALORES INICIALES.
C   URN30 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIAR URN30
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
SUBROUTINE RRN30(X,NBATCH,TIMES)
DIMENSION X(10000)

COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN

```



```

COMMON/COMUN30/ISTRM,IZ

IF(NEWGEN.NE.0) GOTO 10
C
CALL URN30I
WRITE(*,990) ISTRM,IZ
C
10 IF(TIMES.EQ.YES)GO TO 20
C
C          GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
CALL URN30(X,NBATCH)
GOTO 30
C
C          GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20 KTIME=0
C   CALL TIMER(ITIME)
C   CALL URN30 (X,NBATCH)
C   CALL TIMER(ITIME)
C   KTIME=KTIME+ITIME
C
30 IF(NEWGEN.NE.0) GOTO 999
WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
* ' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
WRITE(6,990) ISTRM,IZ
990 FORMAT(1X,'ISTRM=',I20,5X,'IZ=',I20)
999 RETURN
END
C
C*****
C   SUBROUTINE URN30I
C*****
C
C EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION DEFAULT.(IOPT=0)
C NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
SUBROUTINE URN30I
DIMENSION X(10000)

COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
COMMON/COMUN30/ISTRM,IZ

ISTRM =0
IZ=0
RETURN
END
C
C*****
C   SUBROUTINE URN30(X,NBATCH)
C*****
C
C PROPOSITO
C GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C PRECISION EN (0.0,1.0)
C
C USO
C CALL URN30(X,NBATCH)
C Xi+1 = 630360016 Xi (MOD. 2**31 -1)

```

```

C
C DESCRIPCION DE PARAMETROS
C   ISTRM...SEMILLA NUMERO ENTERO ENTRE 1 Y 21474
C   IZ...VALOR ACTUAL DEL ARREGLO ZRNG(ISTRM) DE 100 POSICIONES
C   NBATCH.....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENE-
C           RAR MEDIANTE ESTA SUBROUTINA
C
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C METODO
C
C   REFERENCIAS:
C   MARSE Y ROBERTS (1983)
C
C*****
C
C   SUBROUTINE URN30(X,NBATCH)
C   DIMENSION X(10000)
C   INTEGER B2E15,B2E16,HI15,HI31,OVFLOW,ZI
C   DATA MULT1,MULT2 /24112,26143/
C   DATA B2E15,B2E16,MODLUS /32768,65536,2147483647/
C   COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C   COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C   COMMON/COMUN30/ISTRM,IZ
C
C   ISTRM2=ISTRM + 1
C   ZI=NXSEED(ISTRM2)
C   DO 20 I = 1 ,NBATCH
C   HI15=ZI / B2E16
C   LOWPRD=(ZI-HI15*B2E16)*MULT1
C   LOW15=LOWPRD/B2E16
C   HI31= HI15*MULT1 + LOW15
C   OVFLOW = HI31 / B2E15
C   ZI = (((LOWPRD-LOW15*B2E16)-MODLUS)+
C   *(HI31-OVFLOW*B2E15)*B2E16)+OVFLOW
C
C   IF (ZI.LT.0)ZI=ZI+MODLUS
C   HI15= ZI/B2E16
C   LOWPRD= (ZI-HI15*B2E16)*MULT2
C   LOW15=LOWPRD/B2E16
C   HI31=HI15*MULT2+LOW15
C   OVFLOW= HI31/B2E15
C   ZI = (((LOWPRD-LOW15*B2E16)-MODLUS)+
C   *(HI31-OVFLOW*B2E15)*B2E16)+OVFLOW
C   IF (ZI.LT.0)ZI=ZI+MODLUS
C   XR = (2*(ZI/256)+1)/16777216.0
C   X(I)=XR
20  CONTINUE
C   IZ=ZI
C   ISTRM=ISTRM2
C   RETURN
C   END
C
C   INTEGER FUNCTION NXSEED(ISTRM2)
C   INTEGER SEED
C   DOUBLE PRECISION Z
C   Z=1973272912
C   DO 10 I = 1,ISTRM2
C   Z = DMOD(715.D0*Z, 2147483647.D0)

```

```

      Z = DMOD(1058.D0*Z, 2147483647.D0)
      Z = DMOD(1385.D0*Z, 2147483647.D0)
10  CONTINUE
      NXSEED=IDINT(Z)
      RETURN
      END

C
C
C*****
C      SUBROUTINE RRN31(X,NBATCH,TIMES)
C*****
C
C      ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C      GENERADOR URN31.
C      EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C      A URN31I E IMPRIME LOS VALORES INICIALES.
C      URN31 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C      QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN31
C      DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
      SUBROUTINE RRN31(X,NBATCH,TIMES)
      DIMENSION X(10000)

      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
      COMMON /COMUN3132/IXX

      IF(NEWGEN.NE.0) GOTO 10

C
      CALL URN31I
      WRITE(*,990) IXX

C
10  IF(TIMES.EQ.YES)GO TO 20

C
C      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
      CALL URN31(X,NBATCH)
      GOTO 31

C
C      GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20  KTIME=0
C      CALL TIMER(ITIME)
      CALL URN31(X,NBATCH)
C      CALL TIMER(ITIME)
      KTIME=KTIME+ITIME
C
31  IF(NEWGEN.NE.0) GOTO 999
      WRITE(6,400) NBATCH
400  FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
      *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
      WRITE(6,990) IXX
990  FORMAT(1X,'IXX=',I20)
999  RETURN
      END

C
C*****
C      SUBROUTINE URN31I
C*****
C
C      EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION DEFAULT.(IOPT=0)

```

```

C NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
SUBROUTINE URN31I
  DIMENSION X(10000)

  COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
  COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
  COMMON /COMUN3132/IXX

  IXX =1234567
  RETURN
  END
C
C*****
C SUBROUTINE URN31(X,NBATCH)
C*****
C
C PROPOSITO
C GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C PRECISION EN (0.0,1.0)
C
C USO
C CALL URN31(X,NBATCH)
C
C DESCRIPCION DE PARAMETROS
C IX...SEMILLA NUMERO ENTERO
C NBATCH.....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENE-
C RAR MEDIANTE ESTA SUBRUTINA
C
C
C SUBRUTINAS Y FUNCIONES REQUERIDAS
C NINGUNA
C
C METODO
C
C  $X_{i+1} = 2456949 X_i \pmod{2^{24}}$ 
C
C REFERENCIAS:
C H. MIYAZAKI (1987)
C
C*****
*
C
SUBROUTINE URN31(X,NBATCH)
  DIMENSION X(10000)

  COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
  COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
  COMMON /COMUN3132/IXX

C
MM=2**24
DO 10 I=1,NBATCH
  IY= 2456949*IXX
  IY=MOD(IY,MM)
  IF(IY)5,6,6
5 IY=(IY+2147483647)+1
6 IY2=IY*128
  IXX =ABS(IY2/128)
  XX =FLOAT(IXX)/MM
  X(I) = XX

```

```

10 CONTINUE
RETURN
END

C
C
C*****
C      SUBROUTINE RRN32(X,NBATCH,TIMES)
C*****
C
C      ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C      GENERADOR URN32.
C      EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C      A URN32I E IMPRIME LOS VALORES INICIALES.
C      URN32 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C      QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN32
C      DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
      SUBROUTINE RRN32(X,NBATCH,TIMES)
      DIMENSION X(10000)

      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
      COMMON /COMUN3132/IXX

      IF(NEWGEN.NE.0) GOTO 10

C
      CALL URN32I
      WRITE(*,990)IXX

C
10 IF(TIMES.EQ.YES)GO TO 20

C
      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
      CALL URN32(X,NBATCH)
      GOTO 32

C
      GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20 KTIME=0
C      CALL TIMER(ITIME)
C      CALL URN32 (X,NBATCH)
C      CALL TIMER(ITIME)
C      KTIME=KTIME+ITIME
C
32 IF(NEWGEN.NE.0) GOTO 999
      WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
      *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
      WRITE(6,990)IXX
990 FORMAT(1X,'IXX=',I20)
999 RETURN
      END

C
C*****
C      SUBROUTINE URN32I
C*****
C
C      EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION DEFAULT.(IOPT=0)
C      NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
      SUBROUTINE URN32I

```

```

DIMENSION X(10000)

COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
COMMON /COMUN3132/IXX

IXX =1234567
RETURN
END
C*****
C SUBROUTINE URN32(X,NBATCH)
C*****
C
C PROPOSITO
C GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C PRECISION EN (0.0,1.0)
C
C USO
C CALL URN32(X,NBATCH)
C
C DESCRIPCION DE PARAMETROS
C IX...SEMILLA NUMERO ENTERO
C NBATCH.....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENE-
C RAR MEDIANTE ESTA SUBROUTINA
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C NINGUNA
C
C METODO
C  $X_{i+1} = 3513383 X_i \pmod{2^{25}}$ 
C
C REFERENCIAS:
C H. MIYAZAKI (1987)
C
C*****
C SUBROUTINE URN32(X,NBATCH)
DIMENSION X(10000)

COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
COMMON /COMUN3132/IXX

C
MM=2**25
DO 10 I=1,NBATCH
IY= 3513383*IXX
IY=MOD(IY,MM)
IF(IY)5,6,6
5 IY=(IY+2147483647)+1
6 IY2=IY*64
IXX =ABS(IY2/64)
XX =FLOAT(IXX)/MM
X(I) =XX
10 CONTINUE
RETURN
END
C
C
C*****

```

```

C      SUBROUTINE RRN33(X,NBATCH,TIMES)
C*****
C      ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C      GENERADOR URN33.
C      EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C      A URN33I E IMPRIME LOS VALORES INICIALES.
C      URN33 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C      QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN33
C      DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
      SUBROUTINE RRN33(X,NBATCH,TIMES)
      DIMENSION X(10000)

      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
      COMMON /COMUN33/XI

      IF(NEWGEN.NE.0) GOTO 10

C
      CALL URN33I
      WRITE(*,990)XI

C
      10 IF(TIMES.EQ.YES)GO TO 20

C
      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
      CALL URN33(X,NBATCH)
      GOTO 33

C
      GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
      20 KTIME=0
      CALL TIMER(ITIME)
      CALL URN33 (X,NBATCH)
      CALL TIMER(ITIME)
      KTIME=KTIME+ITIME

C
      33 IF(NEWGEN.NE.0) GOTO 999
      WRITE(6,400) NBATCH
      400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
      * ' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
      WRITE(6,990)XI
      990 FORMAT(1X,'XI=',G15.10)
      999 RETURN
      END

C
C*****
C      SUBROUTINE URN33I
C*****
C      EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION DEFAULT.(IOPT=0)
C      NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
      SUBROUTINE URN33I
      DIMENSION X(10000)

      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
      COMMON /COMUN33/XI

      XI =12345678.0

```

```

RETURN
END
C
C*****
C   SUBROUTINE URN33(X,NBATCH)
C*****
C   PROPOSITO
C   GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C   PRECISION EN (0.0,1.0)
C
C   USO
C   CALL URN33(X,NBATCH)
C
C   DESCRIPCION DE PARAMETROS
C   XI...SEMILLA NUMERO ENTERO
C   NBATCH....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA
GENE-
RAR MEDIANTE ESTA SUBROUTINA
C
C
C
C   SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C   METODO
C   COMBINA DOS SUCESSIONES PARA FORMAR :
C    $X_i = Y_{2i+2}$ 
C   SIENDO:  $Y_{2i+1} = 11257 Y_{2i} \pmod{10^{**}8 + 7}$ 
C            $Y_{2i+2} = 12553 Y_{2i+1} \pmod{10^{**}8 + 37}$ 
C
C
C   REFERENCIAS:
C   H. MIYAZAKI (1987)
C
C*****
C
C   SUBROUTINE URN33(X,NBATCH)
C   DIMENSION X(10000)

COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
COMMON /COMUN33/XI

C
XMD1=100000007.0
XMD2=100000037.0
DO 10 I=1,NBATCH
Y1=11257.0*XI
Y1=AMOD(Y1,XMD1)
IF(Y1)5,6,6
5 Y1=(Y1+2147483647)+1
6 Y2=12553.0*Y1
Y2=AMOD(Y2,XMD2)
IF(Y2)7,8,8
7 Y2=(Y2+2147483647)+1
8 XI=Y2
Y2=Y2/XMD2
X(I)=Y2
10 CONTINUE
END
C
C

```



```

C*****
C      SUBROUTINE RRN34(X,NBATCH,TIMES)
C*****
C
C      ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C      GENERADOR URN34.
C      EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C      A URN34I E IMPRIME LOS VALORES INICIALES.
C      URN34 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C      QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIAR URN34
C      DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C      SUBROUTINE RRN34(X,NBATCH,TIMES)
C      DIMENSION X(10000)
C
C      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C      COMMON/COMUN3435/KW1,KY1,KZ1,XR1
C
C      IF(NEWGEN.NE.0) GOTO 10
C
C      CALL URN34I
C      WRITE(*,990)KW1,KY1,KZ1,XR1
C
C      10 IF(TIMES.EQ.YES)GO TO 20
C
C      GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C      CALL URN34(X,NBATCH)
C      GOTO 30
C
C      GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C      20 KTIME=0
C      CALL TIMER(ITIME)
C      CALL URN34 (X,NBATCH)
C      CALL TIMER(ITIME)
C      KTIME=KTIME+ITIME
C
C      30 IF(NEWGEN.NE.0) GOTO 999
C      WRITE(6,400) NBATCH
C      400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
C      *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C      WRITE(6,990)KW1,KY1,KZ1,XR1
C      990
C      FORMAT(1X,'KW=' ,I15,3X,'KY=' ,I15,3X,'KZ=' ,I15,3X,/1X,'XR=' ,F12.10)
C      999 RETURN
C      END
C
C*****
C      SUBROUTINE URN34I
C*****
C      EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION DEFAULT.(IOPT=0)
C      NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
C      SUBROUTINE URN34I
C      DIMENSION X(10000)
C
C      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C      COMMON /COMUN3435/KW1,KY1,KZ1,XR1

```

```

KW1 =1
KY1 =1
KZ1 =1
XR1 = 0.453
RETURN
END

C
C*****
C      SUBROUTINE URN34(X,NBATCH)
C*****
C
C      PROPOSITO
C      GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C      PRECISION EN (0.0,1.0)
C
C      USO
C      CALL URN34(X,NBATCH)
C
C      DESCRIPCION DE PARAMETROS
C      KW1...SEMILLA NUMERO ENTERO PARA LA SUCESION {Wi}
C      KY1...SEMILLA DE LA SUCESION {Yi}
C      KZ1...SEMILLA DE LA SUCESION {Zi}
C      XR1...VALOR INICIAL REAL PARA EL CALCULO DE INDICES DE LA TABLA
C      NBATCH....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENE-
C              RAR MEDIANTE ESTA SUBROUTINA
C
C
C      SUBROUTINAS Y FUNCIONES REQUERIDAS
C      NINGUNA
C
C      METODO
C      UTILIZA TRES SUCESIONES :
C       $W_{i+1} = 44 W_i \pmod{2039}$ 
C       $Y_{i+1} = 45 Y_i \pmod{2037}$ 
C       $Z_{i+1} = 41 Z_i \pmod{2003}$ 
C      LA SUCESION {Xi} SE OBTIENE DESPUES DE PERMUTACIONES EN UNA TABLA
C       $X_i = W_i/2039 + Y_i/2039*2037 + Z_i/2039*2037*2003$ 
C
C      REFERENCIAS:
C      F. SEZGIN (1987)
C
C*****
C
C      SUBROUTINE URN34(X,NBATCH)
C      DIMENSION X(10000)
C
C      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C      COMMON /COMUN3435/KW1,KY1,KZ1,XR1
C
C
C      DIMENSION TW(10),TY(10),TZ(10)
C      KW=KW1
C      KY=KY1
C      KZ=KZ1
C      XR=XR1
C      DO 1 I=1,10
C      KW=MOD(44*KW,2039)
C      TW(I)=KW
C      KY=MOD(45*KY,2027)

```

```

        TY(I)=KY
        KZ=MOD(41*KZ,2003)
        TZ(I)=KZ
1  CONTINUE
        DO 2 J=1,NBATCH
        KW=MOD(44*KW,2039)
        KY=MOD(45*KY,2027)
        KZ=MOD(41*KZ,2003)
        IU=1000*XR
        I1=IU/100
        I=IU-I1*100
        I2=I/10
        I3=I-10*I2+1
        I1=I1+1
        I2=I2+1
        IW=TW(I1)
        IY=TY(I2)
        IZ=TZ(I3)
        TW(I1)=KW
        TY(I2)=KY
        TZ(I3)=KZ
        NMBR=MOD(IW,6)+1
        GO TO(10,20,30,40,50,60),NMBR
10  XR=FLOAT(IW)/2038.0 - FLOAT(IY)/4128988.0 +
    FLOAT(IZ)/8266233976.0
        GO TO 100
20  XR=FLOAT(IW)/2038.0 - FLOAT(IZ)/4080076.0 +
    FLOAT(IY)/8266233976.0
        GO TO 100
30  XR=FLOAT(IY)/2026.0 - FLOAT(IZ)/4056052.0 +
    FLOAT(IW)/8266233976.0
        GO TO 100
40  XR=FLOAT(IY)/2026.0 - FLOAT(IW)/4128988.0 +
    FLOAT(IZ)/8266233976.0
        GO TO 100
50  XR=FLOAT(IZ)/2002.0 - FLOAT(IW)/4080076.0 +
    FLOAT(IY)/8266233976.0
        GO TO 100
60  XR=FLOAT(IZ)/2002.0 - FLOAT(IY)/4056052.0 +
    FLOAT(IW)/8266233976.0
100 X(J)=XR
2  CONTINUE
        KW1=KW
        KY1=KY
        KZ1=KZ
        XR1=XR
        RETURN
        END
C
C
C*****
C      SUBROUTINE RRN35(X,NBATCH,TIMES)
C*****
C
C      ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C      GENERADOR URN35.
C      EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C      A URN35I E IMPRIME LOS VALORES INICIALES.
C      URN35 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C      QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN35
C      DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.

```

```

C*****
SUBROUTINE RRN35(X,NBATCH,TIMES)
DIMENSION X(10000)

COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON /COMUN3435/KW1,KY1,KZ1,XR1

IF(NEWGEN.NE.0) GOTO 10
C
CALL URN35I
WRITE(*,990)KW1,KY1,KZ1,XR1
C
10 IF(TIMES.EQ.YES)GO TO 20
C
GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
CALL URN35(X,NBATCH)
GOTO 30
C
GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20 KTIME=0
CALL TIMER(ITIME)
CALL URN35 (X,NBATCH)
CALL TIMER(ITIME)
KTIME=KTIME+ITIME
C
30 IF(NEWGEN.NE.0) GOTO 999
WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
*' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
WRITE(6,990)KW1,KY1,KZ1,XR1
990
FORMAT(1X,'KW=',I15,3X,'KY=',I15,3X,'KZ=',I15,3X,/1X,'XR=',F12.10)
999 RETURN
END
C
C*****
C SUBROUTINE URN35I
C*****
C EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION DEFAULT.(IOPT=0)
C NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
SUBROUTINE URN35I
DIMENSION X(10000)
CHARACTER*3 YES
COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
COMMON /COMUN3435/KW1,KY1,KZ1,XR1

KW1 =1
KY1 =1
KZ1 =1
XR1=0.0
RETURN
END
C
C*****
C SUBROUTINE URN35(X,NBATCH)

```

```

C*****
C
C   PROPOSITO
C   GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C   PRECISION EN (0.0,1.0)
C
C   USO
C   CALL URN35(X,NBATCH)
C
C   DESCRIPCION DE PARAMETROS
C   KW1...SEMILLA NUMERO ENTERO PARA LA SUCESION {Wi}
C   KY1...SEMILLA DE LA SUCESION {Yi}
C   KZ1...SEMILLA DE LA SUCESION {Zi}
C   XR1...VALOR INICIAL REAL
C   NBATCH....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENE-
C   RAR MEDIANTE ESTA SUBROUTINA
C
C
C   SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA
C
C   METODO
C   UTILIZA CUATRO SUCESIONES :
C    $W_{i+1} = 157 W_i \pmod{32363}$ 
C    $Y_{i+1} = 146 Y_i \pmod{31727}$ 
C    $Z_{i+1} = 142 Z_i \pmod{31657}$ 
C    $X_i = (W_i + Y_i + Z_i - 3) \pmod{32362}$ 
C   LUEGO:  $U_i = (X_i + 1) / 32363$ 
C
C   REFERENCIAS:
C   L' ECUYER P. (1987)
C
C*****
C
C   SUBROUTINE URN35(X,NBATCH)
C   DIMENSION X(10000)
C
C   COMMON/ TSTTIM/KTIME,C TIME(1000),TIMLOW,LOWSAM
C   COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
C   COMMON /COMUN3435/KW1,KY1,KZ1,XR1
C
C   DIMENSION TW(10),TY(10),TZ(10)
C   KW=KW1
C   KY=KY1
C   KZ=KZ1
C   XR=XR1
C   DO 2 J=1,NBATCH
C   KW=MOD(157*KW,32363)
C   KY=MOD(146*KY,31727)
C   KZ=MOD(142*KZ,31657)
C   XR=MOD(KW+KY+KZ-3,32362)
C   XR=(XR+1)/32363
C   X(J)=XR
C 2 CONTINUE
C   KW1=KW
C   KY1=KY
C   KZ1=KZ
C   XR1=XR
C   RETURN
C   END
C

```

```

C
C*****
C   SUBROUTINE RRN36
C*****
C
C   ESTA SUBRUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN36.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN36I.
C   RRN36 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN36
C   DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C
C   SUBROUTINE RRN36(X,NBATCH,TIMES)
C     INTEGER INTIN,JEN,L16CB2,R20SUM,R5B1C1,B1,B2,C1,C2
C     DIMENSION X(10000)
C
C     COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
C     COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C     COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C
C     IF(NEWGEN.NE.0) GOTO 10
C     CALL URN36I
C     WRITE(6,990)INTIN(1),INTIN(2),INTIN(3),INTIN(4),INTIN(5)
C
C   10 IF(TIMES.EQ.YES)GO TO 20
C
C     GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C     CALL URN36(X,NBATCH)
C     GOTO 30
C
C     GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
C   20 KTIME=0
C     CALL TIMER(ITIME)
C     CALL URN36 (X,NBATCH)
C     CALL TIMER(ITIME)
C     KTIME=KTIME+ITIME
C
C   30 IF(NEWGEN.NE.0) GOTO 999
C     WRITE(6,400) NBATCH
C   400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
C     *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C     WRITE(6,990)INTIN(1),INTIN(2),INTIN(3),INTIN(4),INTIN(5)
C   990 FORMAT(1X,'INTIN(1)=' ,I15,5X,'INTIN(2)=' ,I15
C /1X,'INTIN(3)=' ,I15,
C     *5X,'INTIN(4)=' ,I15,/1X,'INTIN(5)=' ,I15)
C   999 RETURN
C     END
C
C*****
C   SUBROUTINE URN36I
C*****
C
C   EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION DEFAULT.(IOPT=0)
C   NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
C   SUBROUTINE URN36I
C     INTEGER INTIN,JEN,L16CB2,R20SUM,R5B1C1,B1,B2,C1,C2

```

```

DIMENSION X(10000)

COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT, ICNTKS, ICNIMT,NEWGEN

IA=5
IR=0
MULT = 1220703125
INTIN(1) = 32767
INTIN(2) = INTIN(1) / 32768
J= INTIN(1)*65536
IF(J.LT.0) J = J+2147483647+1
INTIN(3) = J/65536
INTIN(4) = MULT/32768
J = MULT*65536
IF(J.LT.0) J = J+2147483647+1
INTIN(5) = J/65536
RETURN
END

C
C*****
C SUBROUTINE URN36
C*****
C
C PROPOSITO
C GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE PRECI-
C SION EN (0.0, 1.0)
C
C USO
C CALL URN36 (X,NBATCH)
C
C DESCRIPCION DE PARAMETROS
C IA.....DIMENSION VECTOR DE NUMEROS ENTEROS
C IR.....DIMENSION VECTOR DE NUMEROS REALES
C NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR
C MEDIANTE ESTA SUBRUTINA
C INTIN(1),INTIN(2),INTIN(3),INTIN(4),INTIN(5) VALORES ENTEROS
C
C
C SUBRUTINAS Y FUNCIONES REQUERIDAS
C NINGUNA
C
C METODO
C UN GENERADOR CONGRUENCIAL MULTIPLICATIVO
C CON CONSTANTE MULTIPLICATIVA 5**13 Y MODULO 2**35
C  $X_{i+1}=5^{13}X_i \pmod{2^{35}}$ 
C
C REFERENCIAS:
C DUDEWICZ E., KARIAN Z. Y MARSHALL R. , "RANDOM NUMBER GENERATION
C ON MICROCOMPUTERS', THE SOCIETY FOR COMPUTER SIMULATION,1985,
C PAG.9-14
C
C*****
SUBROUTINE URN36 (X,NBATCH)
INTEGER INTIN,JEN,L16CB2,R20SUM,R5B1C1,B1,B2,C1,C2
DIMENSION X(10000)

COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT, ICNTKS, ICNIMT,NEWGEN

```

```

C
C1=INTIN(4)
C2=INTIN(5)
B1=INTIN(2)
B2=INTIN(3)
C
DO 10 I =1 ,NBATCH
J2= (B2*C1 + B1*C2)*2048
IF (J2.LT.0) J2=J2+ 2147483647 + 1
R20SUM=J2/2048
J3= B1*C1*67108864
IF (J3.LT.0) J3=J3+ 2147483647 + 1
R5B1C1 = J3/67108864
L16CB2 = (C2*B2)/32768
J4= ((L16CB2 + R20SUM + R5B1C1 * 32768)* 2048)
IF (J4.LT.0) J4=J4+ 2147483647 + 1
B1 = J4/2048
JEN = B1/32
J5= C2*B2*65536
IF (J5.LT.0) J5=J5+ 2147483647 + 1
B2 = J5/65536
X(I) = FLOAT(JEN)/ 32768.0
10 CONTINUE
INTIN(2)=B1
INTIN(3)=B2
RETURN
END
C
C
C*****
C SUBROUTINE RRN37(X,NBATCH,TIMES)
C*****
C
C ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C GENERADOR URN37.
C EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C A URN37I E IMPRIME LOS VALORES INICIALES.
C URN37 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C QUIERE)E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN37
C DESPUES DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C SUBROUTINE RRN37(X,NBATCH,TIMES)
C INTEGER INTIN
C DIMENSION X(10000)
C
C COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
C COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C
C IF(NEWGEN.NE.0) GOTO 10
C
C CALL URN37I
C WRITE(6,990)INTIN(1)
C
C 10 IF(TIMES.EQ.YES)GO TO 20
C
C GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C CALL URN37(X,NBATCH)
C GOTO 30

```



```

C
C          GENERACION DE NUMEROS ALEATORIOS   CON CALCULO DE TIEMPOS
C
C    20 KTIME=0
C      CALL TIMER(ITIME)
C      CALL URN37 (X,NBATCH)
C      CALL TIMER(ITIME)
C      KTIME=KTIME+ITIME
C
C    30 IF(NEWGEN.NE.0) GOTO 999
C      WRITE(6,400) NBATCH
C  400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
C      *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C      WRITE(6,990)INTIN(1)
C  990 FORMAT(1X,'INTIN(1)=' ,I20)
C  999 RETURN
C      END
C
C*****
C      SUBROUTINE URN37I
C*****
C
C EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION DEFAULT.(IOPT=0)
C NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
C      SUBROUTINE URN37I
C      INTEGER INTIN
C      DIMENSION X(10000)
C
C      COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
C      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C
C      IA=1
C      IR=0
C      INTIN(1) = 32767
C      RETURN
C      END
C
C*****
C      SUBROUTINE URN37
C*****
C
C PROPOSITO
C   GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C   PRECISION EN (0.0,1.0)
C
C USO
C CALL URN37 (X,NBATCH)
C
C DESCRIPCION DE PARAMETROS
C   IA.....DIMENSION VECTOR DE NUMEROS ENTEROS
C   IR.....DIMENSION VECTOR DE NUMEROS REALES
C   NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA
C   GENERAR
C           MEDIANTE ESTA SUBROUTINA
C   INTIN(1) VALOR ENTERO
C
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   NINGUNA

```

```

C
C METODO
C   UN GENERADOR CONGRUENCIAL MULTIPLICATIVO
C   CON CONSTANTE MULTIPLICATIVA 5**13 Y MODULO 2**31
C     Xi+1=5**13*Xi (MOD 2**31)
C
C REFERENCIAS:
C   DUDEWICZ E., KARIAN Z. Y MARSHALL R. , "RANDOM NUMBER GENERATION
C   ON MICROCOMPUTERS', THE SOCIETY FOR COMPUTER SIMULATION,1985,
C   PAG.9-14
C
C*****
C   SUBROUTINE URN37 (X,NBATCH)
C     INTEGER INTIN
C     DIMENSION X(10000)
C     COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
C     COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C     COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C     J=INTIN(1)
C     DO 10 I =1 ,NBATCH
C       J=J*1220703125
C       IF (J.LT.0) J= J+ 2147483647 + 1
C       X(I) = FLOAT(J)* 0.4656613E-9
10  CONTINUE
C     INTIN(1)=J
C     RETURN
C     END
C
C
C*****
C   SUBROUTINE RRN38(X,NBATCH,TIMES)
C*****
C
C   ESTA SUBROUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C   GENERADOR URN38.
C   EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C   A URN38I E IMPRIME LOS VALORES INICIALES.
C   URN38 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C   QUIERE)
C   E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN38 DESPUES
C   DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C   SUBROUTINE RRN38(X,NBATCH,TIMES)
C     INTEGER INTIN
C     DIMENSION X(10000)
C
C     COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
C     COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C     COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C
C     IF(NEWGEN.NE.0) GOTO 10
C
C   CALL URN38I
C   WRITE(6,990)INTIN(1)
C
C   10 IF(TIMES.EQ.YES)GO TO 20
C
C     GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
C   CALL URN38(X,NBATCH)
C   GOTO 30

```

```

C
C          GENERACION DE NUMEROS ALEATORIOS   CON CALCULO DE TIEMPOS
C
C    20 KTIME=0
C      CALL TIMER(ITIME)
C      CALL URN38 (X,NBATCH)
C      CALL TIMER(ITIME)
C      KTIME=KTIME+ITIME
C
C    30 IF(NEWGEN.NE.0) GOTO 999
C      WRITE(6,400) NBATCH
400  FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
      *' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
C      WRITE(6,990)INTIN(1)
990  FORMAT(1X,'INTIN(1)=' ,I20)
999  RETURN
C      END
C
C*****
C      SUBROUTINE URN38I
C*****
C
C  EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION DEFAULT.(IOPT=0)
C  NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
C      SUBROUTINE URN38I
C      INTEGER INTIN
C      DIMENSION X(10000)
C
C      COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
C      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C      IA=1
C      IR=0
C      INTIN(1) = 32767
C      RETURN
C      END
C
C*****
C      SUBROUTINE URN38
C*****
C
C  PROPOSITO
C  GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C  PRECISION EN (0.0,1.0)
C
C  USO
C  CALL URN38 (X,NBATCH)
C
C  DESCRIPCION DE PARAMETROS
C  IA.....DIMENSION VECTOR DE NUMEROS ENTEROS
C  IR.....DIMENSION VECTOR DE NUMEROS REALES
C  NBATCH...ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENERAR
C           MEDIANTE ESTA SUBROUTINA
C  INTIN(1) VALOR ENTERO
C
C
C  SUBROUTINAS Y FUNCIONES REQUERIDAS
C  NINGUNA
C
C  METODO

```

```

C      UN GENERADOR CONGRUENCIAL MULTIPLICATIVO
C      CON CONSTANTE MULTIPLICATIVA 5**13 Y MODULO 2**15
C       $X_{i+1} = 5^{13} X_i \pmod{2^{15}}$ 
C
C      REFERENCIAS:
C      DUDEWICZ E., KARIAN Z. Y MARSHALL R. , "RANDOM NUMBER GENERATION
C      ON MICROCOMPUTERS', THE SOCIETY FOR COMPUTER SIMULATION,1985,
C      PAG.9-14
C
C*****
C      SUBROUTINE URN38 (X,NBATCH)
C      INTEGER INTIN
C      DIMENSION X(10000)
C
C      COMMON /OWNPRM/INTIN(200),REALIN(200),IA,IR
C      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C      COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
C      J = INTIN(1)
C      DO 10 I =1 ,NBATCH
C      J=J*15625
C      J=J*65536
C      IF (J.LT.0) J=J+ 2147483647+1
C      J = J/65536
C      J=J*15625
C      J=J*65536
C      IF (J.LT.0) J=J+ 2147483647+1
C      J=J/65536
C      J=J*5
C      J=J*65536
C      IF (J.LT.0) J= J+ 2147483647+1
C      J=J/65536
C      X(I) = FLOAT(J)/ 32768.0
C 10 CONTINUE
C      INTIN(1)=J
C      RETURN
C      END
C
C
C*****
C      SUBROUTINE RRN39 (X,NBATCH,TIMES)
C*****
C
C      ESTA SUBRUTINA ES USADA INTERNAMENTE POR GENTSTPC PARA LLAMAR AL
C      GENERADOR URN39.
C      EN PRIMER TERMINO INICIALIZA EL GENERADOR A TRAVES DE UN LLAMADO
C      A URN39I E IMPRIME LOS VALORES INICIALES.
C      URN39 CALCULA TAMBIEN LOS TIEMPOS DE PROCESAMIENTO (SI SE LO RE-
C      QUIERE)
C      E IMPRIME LOS VALORES NECESARIOS PARA REINICIALIZAR URN39 DESPUES
C      DE HABER GENERADO EL PRIMER LOTE DE NUMEROS ALEATORIOS.
C*****
C      SUBROUTINE RRN39 (X,NBATCH,TIMES)
C      DIMENSION X(10000)
C      DOUBLE PRECISION SEED,PASS,TEMP,XR
C
C      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C      COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
C      COMMON/COMUN39/SEED,ITOTAL39
C
C      IF(NEWGEN.NE.0) GOTO 10
C

```

```

CALL URN39I
WRITE(6,990)SEED,ITOTAL39
C
10 IF(TIMES.EQ.YES)GO TO 20
C
C GENERACION DE NUMEROS ALEATORIOS SIN CALCULO DE TIEMPOS
C
CALL URN39(X,NBATCH)
GOTO 30
C
C GENERACION DE NUMEROS ALEATORIOS CON CALCULO DE TIEMPOS
C
20 KTIME=0
C CALL TIMER(ITIME)
CALL URN39 (X,NBATCH)
C CALL TIMER(ITIME)
C KTIME=KTIME+ITIME
C
30 IF(NEWGEN.NE.0) GOTO 999
WRITE(6,400) NBATCH
400 FORMAT(1X,'LA INICIALIZACION PARA EL PROXIMO NUMERO ALEATORIO',
*' SIGUIENTE A LOS',I11/1X,'YA GENERADOS) ES')
WRITE(6,990)SEED,ITOTAL39
990 FORMAT(1X,'SEED=',D15.10,/1X, 'ITOTAL39 =',I20)
999 RETURN
END
C
C*****
C SUBROUTINE URN39I
C*****
C
C EN ESTE PROCEDIMIENTO SE UTILIZA LA INICIALIZACION DEFAULT.(IOPT=0)
C NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
SUBROUTINE URN39I
DIMENSION X(10000)
DOUBLE PRECISION SEED,PASS,TEMP,XR

COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
COMMON /COUNTS/ICNT,ICNTKS,ICNIMT,NEWGEN
COMMON/COMUN39/SEED,ITOTAL39
SEED = 2.0 /(1.0+SQRT(5.0))
ITOTAL39 = 0
RETURN
END
C
C*****
C SUBROUTINE URN39(X,NBATCH)
C*****
C
C PROPOSITO
C GENERAR NUMEROS PSEUDOALEATORIOS EN PUNTO FLOTANTE CON SIMPLE
C PRECISION EN (0.0,1.0)
C
C USO
C CALL URN39(X,NBATCH)
C
C DESCRIPCION DE PARAMETROS
C SEED .....SEMILLA NUMERO REAL SIMPLE PRECISION
C IGUAL A 2/(1+SQRT(5))
C TOTAL39 ...VALOR INICIAL REAL SIMPLE PRECISION

```

```

C  NBATCH.....ES LA CANTIDAD DE NUMEROS ALEATORIOS QUE SE DESEA GENE-
C          RAR MEDIANTE ESTA SUBROUTINA
C
C
C  SUBROUTINAS Y FUNCIONES REQUERIDAS
C    NINGUNA
C
C  METODO
C    UTILIZA EL ALGORITMO
C     $X_i = [X_{i-1} + (i+X_{i-1})\text{SEN}(i)] \pmod{1}$ 
C
C    REFERENCIAS:
C    REY (1990)
C
C*****
C  EN ESTA SUBROUTINA SE UTILIZA LA INICIALIZACION DEFAULT.(IOPT=0)
C  NO SE TOMAN LOS VALORES DE INICIALIZACION DE OTRA FUENTE.
C*****
C
C    SUBROUTINE URN39(X,NBATCH)
C      DIMENSION X(10000)
C      DOUBLE PRECISION SEED,PASS,TEMP,XR
C
C      COMMON/ TSTTIM/KTIME,CTIME(1000),TIMLOW,LOWSAM
C      COMMON /COUNTS/ICNT,ICNTKS,ICNTMT,NEWGEN
C      COMMON/COMUN39/SEED,ITOTAL39
C      DO 10 I= 1,NBATCH
C        PASS=(ITOTAL39+I)*1.0
C        TEMP=SEED+(PASS+SEED)*DSIN(PASS)
C        XR=TEMP-INT(TEMP)
C        IF (XR.LT.0.0) THEN
C          XR=XR+1.0
C        ENDIF
C        SEED=XR
C        X(I)=XR
10    CONTINUE
C      ITOTAL39=ITOTAL39+NBATCH
C      RETURN
C      END
C
C
C
C*****
C    SUBROUTINE PRINTSTS (GNOMBRE)
C*****
C
C    SUBROUTINA PARA IMPRIMIR RESULTADOS DEL TEST TST01 CON LA
C    OPCION CS2KS2
C    IMPRESION  DE UNA TABLA CON RESULTADOS DE LOS 19 TESTS.
C
C*****
C    SUBROUTINE PRINTSTS(GNOMBRE)
C      COMMON/GAMCHI2/TT(19,2)
C      CHARACTER*5 GNOMBRE
C
C    CALCULAR CANTIDAD DE GAMA APROX. ENTRE 0.50 ,095 Y 0.99
C    J50=0
C    J95=0
C    J99=0
C    DO 80 I = 1,19
C      IF (TT(I,2).GE.0.99) THEN
C        J99=J99+1

```

```

ELSE IF (TT(I,2).GE.0.95) THEN
J95=J95+1
ELSE IF (TT(I,2).GE.0.50) THEN
J50=J50+1
ENDIF
80  CONTINUE
J95= J99+J95
J50=J95+J50
C    IMPRESION RESULTADOS 19 TESTS Y CANT. PROBAB.
WRITE(*,7890)
7890 FORMAT(/,/,/22X,30('*'))
WRITE(*,123) GNOMBRE
123  FORMAT(25X,'G E N E R A D O R: ',1X,5A)
WRITE(*,5678)
5678 FORMAT(22X,30('*'),/,/)
WRITE(*,101)
101  FORMAT(1X,'T E S T',T32,'T(I)',T38,'P[CHI-CUAD<=T(I)]')
WRITE(*,102)
102  FORMAT(1X,'-----
')
WRITE(*,11)TT(1,1),TT(1,2)
11  FORMAT(1X,'DISTR.UNIF.(TST02)',T30,F6.2,T40,F5.2)
WRITE(*,12)TT(2,1),TT(2,2)
12  FORMAT(1X,'CUPONES D=5(TST04)',T30,F6.2,T40,F5.2)
WRITE(*,13)TT(3,1),TT(3,2)
13  FORMAT(1X,'CUPONES FR(100R)',T30,F6.2,T40,F5.2)
WRITE(*,14)TT(4,1),TT(4,2)
14  FORMAT(1X,'CUPONES D=10',T30,F6.2,T40,F5.2)
WRITE(*,15)TT(5,1),TT(5,2)
15  FORMAT(1X,'GAP ARRIBA MEDIA(TST05)',T30,F6.2,T40,F5.2)
WRITE(*,16)TT(6,1),TT(6,2)
16  FORMAT(1X,'GAP BAJO MEDIA',T30,F6.2,T40,F5.2)
WRITE(*,17)TT(7,1),TT(7,2)
17  FORMAT(1X,'GAP (0.33,0.667)',T30,F6.2,T40,F5.2)
WRITE(*,18)TT(8,1),TT(8,2)
18  FORMAT(1X,'PERMUTACIONES 3S(TST07)',T30,F6.2,T40,F5.2)
WRITE(*,19)TT(9,1),TT(9,2)
19  FORMAT(1X,'PERMUTACIONES 4S',T30,F6.2,T40,F5.2)
WRITE(*,20)TT(10,1),TT(10,2)
20  FORMAT(1X,'PERMUTACIONES 5S',T30,F6.2,T40,F5.2)
WRITE(*,21)TT(11,1),TT(11,2)
21  FORMAT(1X,'POKER (4,4) (TST08)',T30,F6.2,T40,F5.2)
WRITE(*,22)TT(12,1),TT(12,2)
22  FORMAT(1X,'POKER (5,6)',T30,F6.2,T40,F5.2)
WRITE(*,23)TT(13,1),TT(13,2)
23  FORMAT(1X,'POKER (5,4)',T30,F6.2,T40,F5.2)
WRITE(*,24)TT(14,1),TT(14,2)
24  FORMAT(1X,'CORRIDAS R(TST09)',T30,F6.2,T40,F5.2)
WRITE(*,25)TT(15,1),TT(15,2)
25  FORMAT(1X,'CORRIDAS FR(10R)',T30,F6.2,T40,F5.2)
WRITE(*,26)TT(16,1),TT(16,2)
26  FORMAT(1X,'CORRIDAS FR(100R)',T30,F6.2,T40,F5.2)
WRITE(*,27)TT(17,1),TT(17,2)
27  FORMAT(1X,'PARES SERIALES 3X3(TST10)',T30,F6.2,T40,F5.2)
WRITE(*,28)TT(18,1),TT(18,2)
28  FORMAT(1X,'PARES SERIALES 10X10',T30,F6.2,T40,F5.2)
WRITE(*,29)TT(19,1),TT(19,2)
29  FORMAT(1X,'PARES SERIALES 20X20',T30,F6.2,T40,F5.2)
WRITE(*,102)
WRITE(*,300)J99
300  FORMAT(1X,'CANTIDAD DE G`= P[CHI-CUAD<= T(I)]>= 0.99 ',2X,I6)

```

```

WRITE(*,301)J95
301  FORMAT(1X,'CANTIDAD DE G`= P[CHI-CUAD<= T(I)]>= 0.95 ',2X,I6)
WRITE(*,302)J50
302  FORMAT(1X,'CANTIDAD DE G`= P[CHI-CUAD<= T(I)]>= 0.50 ',2X,I6)
RETURN
END

C
C*****
C          F U N C I O N E S
C          CDFKS - CHIDF - DCDFN - DCDFNI
C*****
C
C*****
C          FUNCTION CDFKS(A)
C*****
C ESTE SUBPROGRAMA CALCULA LAS PROBABILIDADES PARA UNA VARIABLE A
C LEATORIA DE KOLMOGOROV-SMIRNOV.
C PROPOSITO:
C   CALCULAR LA PROBABILIDAD DE QUE X SEA MENOR QUE A, DONDE X ES UNA
C VARIABLE ALEATORIA KOLMOGOROV-SMIRNOV, PARA USAR EN TST01, TST06 Y
C TST11
C USO:
C   " P = CDFKS(A) "
C DESCRIPCION Y PARAMETROS:
C   A...ES EL PUNTO PARA EL CUAL LA PROBABILIDAD P(X<A) ES
C   CALCULADA
C METODO:
C   LA PROBABILIDAD DESEADA ESTA DADA POR:
C    $P(X<A)=COEFF*(EXP(-1*C)+EXP(-9*C)+...+EXP(-(2K-1)**2)*C)$ 
C   DONDE
C    $COEFF=SQRT(2*PI)/A$ 
C    $C=(PI**2)/(8*(A**2))$ 
C    $PI=3.1415926$ 
C Y LA SUMATORIA ES TERMINADA CUANDO EL SUMANDO ES MENOR QUE EXP(-20).
C
C REFERENCIA:
C   1. FELLER,W.: "ON THE KOLMOGOROV-SMIRNOV THEOREMS",ANNALS OF
C   MATHEMATICAL STATISTICS, VOL.19(1948), PP.177-189.
C*****
C
C          FUNCTION CDFKS(VALUE)
C          PI=3.141592654
C          CONST=(PI**2)/(8*(VALUE**2))
C          COEFF=SQRT(2*PI)/VALUE
C          SUM=0
C
C          DO 1 I=1,100
C          POWER=((2*I-1)**2)*CONST
C          IF(POWER.GT.20) GOTO 2
C 1 SUM=SUM+EXP(-POWER)
C
C          2 CDFKS=COEFF*SUM
C
C          RETURN
C          END
C
C*****
C          FUNCTION CHIDF(A,V)
C*****

```



```

C ESTA FUNCION CALCULA PROBABILIDADES PARA VARIABLES ALEATORIAS CHI
C CUADRADO CON "V" GRADOS DE LIBERTAD, V=1,2,...,21.
C
C PROPOSITO
C   GENERAR LA PROBABILIDAD DE QUE "X" SEA MENOR O IGUAL QUE "A",
C   DONDE X ES UNA VARIABLE ALEATORIA CHI-CUADRADO
C   CON V GRADOS DE LIBERTAD.
C
C USO
C   " P = CHIDF(A,V) "
C
C DESCRIPCION DE PARAMETROS
C   A...ES EL PUNTO PARA EL CUAL SE DESEA QUE LA PROBABILIDAD DE
C   X SEA MENOR O IGUAL QUE A
C   V...SON LOS GRADOS DE LIBERTAD PARA DICHA PROBABILIDAD
C
C OBSERVACIONES
C   ES UNA FUNCION EN DOBLE PRECISION Y DEBE OPERAR CON UN ARGUMENTO
C   DOBLE PRECISION. (CUANDO "A" NO LO ES PUEDEN PRODUCIRSE ERRORES
C   EN LA RUTINA QUE LA INVOCA)
C   LA EXACTITUD DE ESTE SUBPROGRAMA FUE VERIFICADO PARA V=1,2,...,21
C   LO CUAL ES SUFICIENTE PARA EL USO QUE SE HACE EN ESTE PAQUETE.
C (ES DECIR QUE LA FUNCION ES LLAMADA POR LOS PUNTOS EN ESE RANGO DE V
C Y LOS VALORES DE LA FUNCION DE DISTRIBUCION SON USADOS EN UN PROCESO
C DENOMINADO "PINCH" PARA HALLAR LOS PUNTOS PORCENTUALES DE CHI
C CUADRADO CADA 0.000001). LA EXACTITUD NO FUE EVALUADA PARA V>21.
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   DCDFN
C
C METODO
C   FORMULAS DE P.941, REFERENCIA 2.
C
C REFERENCIAS:
C   1. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C   GENERATION AND TESTING," AMERICAN SCIENCES PRESS,
C   COLUMBUS, OHIO, 1981.
C
C   2. ZELEN, M. AND SEVERO, N.C.: "PROBABILITY FUNCTIONS,"
C   HANDBOOK OF MATHEMATICAL FUNCTIONS WITH FORMULAS, GRAPHS, AND
C   MATHEMATICAL TABLES (EDITED BY M. ABRAMOWITZ AND I.A. STEGUN),
C   NATIONAL BUREAU OF STANDARDS, APPLIED MATHEMATICS SERIES, VOL.
C   55, JUNE 1964 (FOURTH PRINTING, DECEMBER 1965, WITH
C   CORRECTIONS), CHAPTER 26, PP. 925-995.
C
C*****
C
REAL*8 FUNCTION CHIDF(A,V)
REAL*8 SUM,B,A,EE,DCDFN
INTEGER V
IF(MOD(V,2).EQ.0) GO TO 120
IF(A.GT.0) GO TO 105
CHIDF=0.D0
RETURN
105 IF(V.NE.1) GO TO 107
CHIDF=2.D0*DCDFN(DSQRT(A))-1.D0
RETURN
107 SUM=0.D0
LIM2=(V-1)/2
B=1.D0/DSQRT(A)
DO 110 I=1,LIM2

```

```

        B=B*(A/(2*I-1))
        SUM=SUM+B
110 CONTINUE
        EE=DSQRT(2.D0/3.1415926535897932D0)*DEXP(-A/2.D0)
        CHIDF=2.D0*DCDFN(DSQRT(A))-1.D0-EE*SUM
        RETURN
120 IF(V.NE.2) GO TO 122
        CHIDF=1.D0-DEXP(-A/2.D0)
        RETURN
122 SUM=0.D0
        LIM2=(V-2)/2
        B=1.D0
        DO 210 I=1,LIM2
        B=B*(A/(2*I))
        SUM=SUM+B
210 CONTINUE
        EE=DEXP(-A/2.D0)
        CHIDF=1.D0-EE*(1.D0+SUM)
        RETURN
        END
C
C*****
C      FUNCTION DCDFN(X)
C
C*****
C
C PROPOSITO
C SUBPROGRAMA PARA CALCULAR PROBABILIDADES DE VARIABLES NORMALES
C ESTANDAR.
C ESTA FUNCION GENERA LA PROBABILIDAD DE QUE Z SEA MENOR O IGUAL
C QUE X, SIENDO Z UNA VARIABLE NORMAL ESTANDAR.
C
C USO
C      " P = DCDFN(X) "
C
C DESCRIPCION DE PARAMETROS
C      X...ES EL PUNTO PARA EL CUAL SE DESEA QUE LA PROBABILIDAD DE
C      Z SEA MENOR O IGUAL QUE X
C
C OBSERVACIONES
C      ES UNA FUNCION DOBLE PRECISION QUE SE AJUSTA A UN ARGUMENTO
C      EN DOBLE PRECISION
C
C SUBRUTINAS Y FUNCIONES REQUERIDAS
C      NINGUNA
C
C METODO
C      PARA EL ALGORITMO EMPLEADO Y SU EXACTITUD, REFERENCIA 2.
C      COMPARACIONES CON OTROS ALGORITMOS Y VELOCIDAD, REFERENCIA 1.
C
C REFERENCIAS
C      1. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C      GENERATION AND TESTING," AMERICAN SCIENCES PRESS,
C      COLUMBUS, OHIO, 1981.
C      2. MILTON,R.C. AND HOTCHKISS,R.: "COMPUTER EVALUATION OF THE
C      NORMAL AND INVERSE NORMAL DISTRIBUTION FUNCTIONS,"
C      TECHNOMETRICS, VOL.11(1969), PP. 817-822.
C
C*****
C
C      DOUBLE PRECISION FUNCTION DCDFN(X)

```

```

DIMENSION A(6),D(6),C(6,10)
REAL*8 A,C,D,X,Y,Z,SGNY
DATA A/.625D0,1.25D0,2.0D0,2.45D0,3.5D0,4.62D0/
DATA D/.3D0,.925D0,1.625D0,2.225D0,2.95D0,4.15D0/
DATA C/6.7982403291D-04,-1.2709753598D-03,6.7964525797D-04,
*-8.8500314297D-05,1.4791554152D-07,5.4879072878D-07,
*5.1028640888D-03,-1.4822649744D-03,-3.4589883732D-04,
*5.6826070991D-04,-6.1965013125D-05,-8.6656125337D-07,
*-6.0160862379D-03,8.7718965604D-03,-3.0883401043D-03,
*-4.5015531032D-04,2.2266710841D-04,6.6877769441D-07,
*-2.2725611373D-02,-2.0200415601D-03,7.2141082947D-03,
*-1.3872389725D-03,-4.4390018986D-04,-6.4084653798D-07,
*3.3555772127D-02,-3.4681674488D-02,-1.0677039440D-03,
*5.5422185916D-03,6.2978750848D-04,6.2471782478D-07,
*7.2703311825D-02,4.7642403207D-02,-2.4859750804D-02,
*-1.0221129286D-02,-6.6382630882D-04,-4.2105004094D-07,
*-1.4093895854D-01,5.6850293056D-02,5.7420315819D-02,
*1.1850195831D-02,5.1265678121D-04,2.0742133302D-07,
*-1.5468912970D-01,-2.2180615138D-01,-6.5383705170D-02,
*-8.8864030978D-03,-2.7657162532D-04,-7.6856040218D-08,
*5.1563045460D-01,2.3979043073D-01,4.0236129479D-02,
*3.9938885701D-03,9.3751413487D-05,1.8722020822D-08,
*1.6431337971D-01,4.0458836515D-01,4.8922186659D-01,
*4.9917416726D-01,4.9998489849D-01,4.999999779D-01/
Y=X*0.70710678119D0
SGNY=1.D0
IF(Y)2,1,3
1 DCDFN=.5D0
RETURN
2 SGNY=-1.D0
Y=-Y
3 DO 4 I=1,6
IF(Y-A(I))5,5,4
4 CONTINUE
Z=.5D0
GO TO 7
5 Y=Y-D(I)
Z=C(I,1)
DO 6 J=2,10
6 Z=Z*Y+C(I,J)
7 DCDFN=.5D0+SGNY*Z
RETURN
END
C
C*****
C FUNCTION DCFNI(P)
C
C*****
C
C PROPOSITO
C SUBPROGRAMA PARA CALCULAR PROBABILIDADES DE VARIABLES NORMALES
C ESTANDAR
C ESTA FUNCION GENERA 100P PUNTOS PORCENTUALES DE UNA VARIABLE
C NORMAL ESTANDAR.
C
C USO
C " X = DCFNI(P) "
C
C DESCRIPCION OF PARAMETROS
C P...ES LA PROPORCION PARA LA CUAL EL PUNTO PORCENTUAL 100P DE
C UNA DISTRIBUCION NORMAL ESTANDAR ES DESEADO

```

```

C
C OBSERVACIONES
C   ES UNA FUNCION DOBLE PRECISION QUE SE AJUSTA A UN ARGUMENTO
C   EN DOBLE PRECISION
C
C SUBROUTINAS Y FUNCIONES REQUERIDAS
C   DCDFN
C
C METODO
C   PARA EL ALGORITMO EMPLEADO Y SU EXACTITUD, REFERENCIA 2.
C   COMPARACIONES CON OTROS ALGORITMOS Y VELOCIDAD, REFERENCIA 1.
C   1. DUDEWICZ,E.J. AND RALLEY,T.: "THE HANDBOOK OF RANDOM NUMBER
C     GENERATION AND TESTING," AMERICAN SCIENCES PRESS,
C     COLUMBUS, OHIO, 1981.
C   2. MILTON,R.C. AND HOTCHKISS,R.: "COMPUTER EVALUATION OF THE
C     NORMAL AND INVERSE NORMAL DISTRIBUTION FUNCTIONS,"
C     TECHNOMETRICS, VOL.11(1969), PP. 817-822.
C
C*****
C
C   DOUBLE PRECISION FUNCTION DCDFNI(P)
C   REAL*8 X,Q,T,R,P,DCDFN
C   R=P
C   NN=1
C   Q=1.D0-P
C   IF(R)1,3,4
C 1  WRITE(6,2)P
C 2  FORMAT('0','ILLEGAL ARGUMENT IN DCDFNI. P=',D20.10)
C   STOP
C 3  DCDFNI=-7.D0
C   RETURN
C 4  IF(1.D0-R)1,5,6
C 5  DCDFNI=7.D0
C   RETURN
C 6  IF(R-.5D0)9,7,8
C 7  DCDFNI=0.D0
C   RETURN
C 8  R=1.D0-R
C   NN=2
C   Q=P
C 9  IF(R-1.D-10)10,11,11
C10  X=6.41D0
C   GO TO 14
C11  T=DSQRT(DLOG(1.D0/(R*R)))
C   X=T-
C   ((.010328D0*T+.802853D0)*T+2.515517D0)/(((.001308D0*T+.189269D
C   10)*T+1.432788D0)*T+1.D0)
C12  DO 13 I=1,2
C13  X=X-(DCDFN(X)-Q)/(.3989422804D0*DEXP(-.5D0*X*X))
C14  GO TO (15,16),NN
C15  DCDFNI=-X
C   RETURN
C16  DCDFNI=X
C   RETURN
C   END
C*****FIN GENTSTPC2*****
C *****4946 LINEAS CODIGO*****

```