

R-98

NATIONAL BUREAU OF STANDARDS REPORT

8936

A COMPUTER PROGRAM FOR INVESTIGATING
COMPUTER AND OPERATING SYSTEM TURN-AROUND
FOR PROBLEMS EMPLOYING MULTIDIMENSIONAL ARRAYS

by

George W. Reitwiesner

GPO PRICE \$ _____

CFSTI PRICE(S) \$ _____

Hard copy (HC) 2.00

Microfiche (MF) .50

ff 653 July 65

N66 38708
(ACCESSION NUMBER)
46
(PAGES)
CR-78504
(NASA CR OR TMX OR AD NUMBER)

(THRU)
1
(CODE)
08
(CATEGORY)



U.S. DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS

THE NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards is a principal focal point in the Federal Government for assuring maximum application of the physical and engineering sciences to the advancement of technology in industry and commerce. Its responsibilities include development and maintenance of the national standards of measurement, and the provisions of means for making measurements consistent with those standards; determination of physical constants and properties of materials; development of methods for testing materials, mechanisms, and structures, and making such tests as may be necessary, particularly for government agencies; cooperation in the establishment of standard practices for incorporation in codes and specifications; advisory service to government agencies on scientific and technical problems; invention and development of devices to serve special needs of the Government; assistance to industry, business, and consumers in the development and acceptance of commercial standards and simplified trade practice recommendations; administration of programs in cooperation with United States business groups and standards organizations for the development of international standards of practice; and maintenance of a clearinghouse for the collection and dissemination of scientific, technical, and engineering information. The scope of the Bureau's activities is suggested in the following listing of its four Institutes and their organizational units.

Institute for Basic Standards. Applied Mathematics. Electricity. Metrology. Mechanics. Heat. Atomic Physics. Physical Chemistry. Laboratory Astrophysics.* Radiation Physics. Radio Standards Laboratory.* Radio Standards Physics; Radio Standards Engineering. Office of Standard Reference Data.

Institute for Materials Research. Analytical Chemistry. Polymers. Metallurgy. Inorganic Materials. Reactor Radiations. Cryogenics.* Materials Evaluation Laboratory. Office of Standard Reference Materials.

Institute for Applied Technology. Building Research. Information Technology. Performance Test Development. Electronic Instrumentation. Textile and Apparel Technology Center. Technical Analysis. Office of Weights and Measures. Office of Engineering Standards. Office of Invention and Innovation. Office of Technical Resources. Clearinghouse for Federal Scientific and Technical Information.**

Central Radio Propagation Laboratory.* Ionospheric Telecommunications. Tropospheric Telecommunications. Space Environment Forecasting. Aeronomy.

* Located at Boulder, Colorado 80301.

** Located at 5285 Port Royal Road, Springfield, Virginia 22171.

NATIONAL BUREAU OF STANDARDS REPORT

NBS PROJECT

2050630

August 2, 1965

NBS REPORT

8936

A COMPUTER PROGRAM FOR INVESTIGATING
COMPUTER AND OPERATING SYSTEM TURN-AROUND
FOR PROBLEMS EMPLOYING MULTIDIMENSIONAL ARRAYS

by

George W. Reitwiesner

Computation Laboratory

IMPORTANT NOTICE

NATIONAL BUREAU OF STANDARDS REPORTS are usually preliminary or progress accounting documents intended for use within the Government. Before material in the reports is formally published it is subjected to additional evaluation and review. For this reason, the publication, reprinting, reproduction, or open-literature listing of this Report, either in whole or in part, is not authorized unless permission is obtained in writing from the Office of the Director, National Bureau of Standards, Washington 25, D.C. Such permission is not needed, however, by the Government agency for which the Report has been specifically prepared if that agency wishes to reproduce additional copies for its own use.



U. S. DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS

A COMPUTER PROGRAM FOR INVESTIGATING
COMPUTER AND OPERATING SYSTEM TURN-AROUND
FOR PROBLEMS EMPLOYING MULTIDIMENSIONAL ARRAYS

by

George W. Reitwiesner

ABSTRACT

This writing describes, in considerable detail, a computer program which was prepared to measure the turn-around capability of a computer operating system in an atmosphere heavily weighted by problems which demand the referencing of data in multidimensional arrays.

This writing describes, in considerable detail, a computer program which was prepared to measure the turn-around capability of a computer operating system in an atmosphere heavily weighted by problems which demand the referencing of data in multidimensional arrays.

The program operates cyclically, with two phases per cycle, and each succeeding cycle performs under the government of two indices, m and n , which advance and withdraw according to a pattern which will be described below. The execution of the program through several cycles is intended reasonably to simulate a representative assortment of mixed checkout and short production runs over a few days of normal computer use in the postulated atmosphere, and several cycles may fit into a few days of normal batch-processed workload.

In blunt deference to its relative universality, Fortran is employed as the programming language.

Each of the two phases of each cycle involves a separate computer program, as follows. Phase 1 is a short production program which deals with alphameric data; it reads a skeletal Fortran program and punches a completed Fortran program which it develops therefrom under the guidance of the parameters m and n and of other data passed on from the previous cycle. (Of course, on the first cycle, phase 1 involves also the compilation of the production program.) Phase 2 consists of the compilation and running of the program developed in phase 1; however, the compilation and/or running of the phase 2 program may be impossible on some of the cycles, depending upon conditions among the parameters involved and upon the computer and the programming and operating systems employed.

To effect the transition from each cycle to the next, a key card, recording the parameters m and n and other data, is punched both during the operation of phase 1 (the failing key card) and after successful completion of the operation of phase 2 (the passing key card). One of these key cards is read (for vital control data) at the beginning of phase 1 of the next cycle: the passing key card, if available, to "advance" the indices m and n ; but the failing key card, otherwise, to "withdraw" them.

The phase 2 program solves a particular problem: The Laplace equation in two rectangular dimensions:

$$\delta^2 T / \delta x^2 + \delta^2 T / \delta y^2 = 0$$

for constant uniform boundary around an n -by- n internal mesh. In particular, for a positive integer parameter $Level$, and for the uniform boundary and uniform initial interior conditions

$$T(1,k) = T(n+2,k) = T(k,1) = T(k,n+2) = 1+2^{-(Level+1)} \quad k=1,2,3\dots n+2$$

$$\text{Initially } T(1,j) = 1+2^{-1} \quad 1,j=2,3,4\dots n+1$$

solution is approached by the Liebmann process of reiterating the replacement of each of the n^2 interior mesh values, in turn, by the average of its four nearest neighbors:

$$T(1,j) = (T(i-1,j) + T(i+1,j) + T(1,j-1) + T(1,j+1))/4 \quad 1,j=2,3,4\dots n+1$$

until the interior values all arrive within some prescribed tolerance of stability. Clearly, upon sufficient reiteration, the ideal solution for this particular problem must stabilize at

$$\text{All } T(1,j) = 1+2^{-(Level+1)}$$

however the numerical problem solving aspects are inconsequential here. Indeed, through an iteration counting parameter $Itct$, the programming deliberately limits the number of iterations of mesh value replacements so that the total time spent attempting to get a solution can be limited to a few seconds, regardless of numerical values at the interior mesh points.

What is under scrutiny here is the capacity of the computer operating system to provide turn-around, from programmer through machine and back to programmer, in an atmosphere of relatively heavy indexing throughout arrays.

The primary unit of data in the phase 2 program is the $(n+2)$ -by- $(n+2)$ array of $4(n+1)$ boundary values and n^2 interior mesh values, and each reiteration of the n^2 replacements involved in the solution procedure creates a new display of data over that $(n+2)$ -by- $(n+2)$ array. To complicate the indexing requirements, however, more than one such array is employed: a set of m^2 such arrays is employed in an m -by- m array of such arrays, and the storage of the numerical values at the mesh points for each new iteration (of n^2 replacements) is effected, except when $m=1$, in a separate $(n+2)$ -by- $(n+2)$ array, continuing cyclically over all the m^2 unit arrays. Consequently, in referencing each interior mesh value, four indicies are involved: those of the mesh point within its unit array and those of the unit array within the array of arrays; and each of the two (row and column) indicies which locate the mesh point within the overall array of arrays must be computed as a combination of two others: that (row or column, respectively) of the mesh point within its unit array and that (row or column, respectively) of the unit array within the array of arrays.

Thus, the parameter m provides for the complication of the indexing employed in effecting the solution; and the value assigned to m , in concert with that of n , governs the amount of numerical data storage required. Clearly, an upper limit on the product of m times $n+2$ is provided by the storage capacity of the computer, reduced according to the demands made by the operating system and by the phase 2 programming.

In the phase 2 programming, the repeated replacement of each interior mesh value by the average of its four nearest neighbors would be programmed most easily in a doubled n -fold loop: for example, a single replacement statement in an n -fold loop on column indicies within an n -fold loop on row indicies. However a different technique is used. To reveal emphatically any awkwardness which the compiler introduces into the programming when indicies must be switched frequently, and to probe for the maximum program size which the compiler and the computer can accommodate, that single replacement is employed n^2 times: the generating program of phase 1 generates it n^2 times in n sets, one for each row, of n employments, one for each column.

If each of these n^2 generations were to yield a single (Fortran) statement, that statement would employ five pairs of (row and column) indicies: one pair for the mesh value being replaced and four pairs for its four neighbors; and each index for each pair would be a sum of two entities: a variable which identifies a (row or column) base reference value for its particular one of the m^2 unit arrays within the overall array of arrays, and a constant which identifies its particular row or column within the unit array. However, the generator has been designed to yield two (Fortran) statements per mesh value replacement: the first one replaces the mesh value with the sum of three of its neighbors, and the second replaces that with the completed average; thus, the number of pairs of (row and column) indicies employed is raised from five to seven per mesh value replaced: four pairs for each first statement and three pairs for each second statement.

Further, since two $(n+2)$ -by- $(n+2)$ unit arrays must be referred to by each pair of such statements (the "new" one into which the replacement value is stored and from which are drawn such neighbors as already have been replaced, and the "old" one from which are drawn such neighbors as have not yet been replaced), the two replacement statements for each mesh value must refer to four variable indicies: two old and two new. The generator attempts to maximize the interlacing of references to the old and new unit arrays.

While the numerical convergence to the solution of the Laplace equation is of but passing interest here, a certain tolerance examination is provided in the generated phase 2 program---in manner to introduce another indexing complexity: jumping back and forth between counting loops. This examination consists simply of continuously counting the number of replaced mesh values which lie within $2^{-(level+1)}$ of the known

ideal solution. The counting sequence is included n times in the phase 2 programming: immediately after each set of n mesh point replacements. It consists of a pair of loops which act as one, scanning the values over the mesh row just replaced. One loop scans over consecutive mesh values which fail to meet the tolerance examination and counts and switches to the other loop to continue the scan of the row whenever it encounters a mesh value which does meet tolerance; the other loop scans over and counts consecutive mesh values which do meet the tolerance examination and switches back to the first to continue the scan of the row whenever it encounters a mesh value which fails to meet tolerance.

The store demand of the compiled generated phase 2 program, then, is measured as a quadratic polynomial in the parameters m and n . The number of statements in the programming is a quadratic in n ; and the numerical data storage requirement is a quadratic in m and n : precisely, the square of the product of m times $n+2$. The sequence of cycles, advancing and withdrawing over m and n , has been designed to maintain strain against the limit of available storage while gradually exchanging from a starting condition of small n and large m (representing moderate programming demand and large data demand) to a final condition of large n and minimum m ($= 1$) (representing moderate data demand and extreme programming demand). In the context of a computer with a 32,768-word store capacity, representative initial parameter values are considered to be $n = 8$ and $m = 16$ (demanding 25,600 words for storing $m^2 = 256$ unit arrays of $(n+2)^2 = 100$ values each, and requiring the compilation of a program which contains $n^2 = 64$ pairs of replacement statements and $n = 8$ tolerance checks), whereafter, with each cycle: n is increased ("advanced") by 8 each time phase 2 yields a passing key card; but m is halved ("withdrawn") to allow n to increase further whenever phase 2 fails to yield a passing key card; and after $m = 1$ is reached, n is increased and/or decreased by 8, 4, 2, 1, in turn, to ascertain the maximum value of n for which solution is attainable.

The advancing and withdrawing of m and n yield, in the final value for n , a measure of the problem-size capacity of the computer and its programming and operating systems. And the cyclic progression over production runs and over compilation attempts and runs yields a capacity to measure turn-around in an atmosphere of complex indexing demands over multidimensional arrays. Also, the generated phase 2 programs, through their storage requirements for programming and through their (Itct parameter) controlled solution times, afford a measure of computer and compiler adroitness in handling large and complicated indexing situations.

And, as a final complication to the turn-around requirement, each program (phases 1 and 2) duplicates its key card on magnetic tape for demounting and for subsequent remounting and reading and examination for total agreement against the key card itself, as a condition for beginning the phase 1 generation of the next cycle.

The programming for phase 1 (the generator) and phase 2 (for $n = 8$ and $m = 8$) is shown in the appendix hereto.

APPENDIX

Programming for

Phase 1 Generator

Phase 2 $n=8, m=8$

C	/*	SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM	TEST	1
		JOB JOB JOB JOB JOB JOB JOB JOB JOB JOB JOB JOB JOB JOB JOB JOB CARD	TEST	2
		FOR	TEST	3
C		THIS PROGRAM IS A DRAFT OF A TEST OF COMPUTER OPERATING PROCEDURE.	TEST	4
C		IT IS WRITTEN IN FORTRAN	TEST	5
C		IT IS A FORM OF GENERATOR PROGRAM.	TEST	6
C		IT EMPLOYS AN INPUT DATA DECK WHICH ITSELF IS A FORTRAN PROGRAM.	TEST	7
C		A DESCRIPTION OF THE TEST AND INSTRUCTIONS FOR CONDUCTING IT ARE	TEST	8
C		CONTAINED ON COMMENT CARDS IN THE GENERATOR PROGRAM.	TEST	9
C		THE LAST LINE OF THIS PROGRAM LISTING CONTAINS (CFF) IN COLS.1-3.	TEST	10
C		GENERATOR DECK	TEST	11
C		THIS PROGRAM OPERATES RECURSIVELY OVER A RUN INDEX NRUN=1,2,3,...	TEST	12
C		THIS PROGRAM READS BOTH A CARD DECK AND ONE TAPE	TEST	13
C		THE CARDS ARE THE DATA DECK FOLLOWING BELOW.	TEST	14
C		THE 2ND CARD OF THE DATA DECK IS A KEY CARD - TO BE REPLACED FOR TEST	TEST	15
C		EACH RUN.	TEST	16
C		THE REMAINING CARDS OF THE DATA DECK ARE UNCHANGED FOR EACH RUN.	TEST	17
C		THE TAPE CONTAINS A DUPLICATE OF THE KEY CARD - FOR READING AND	TEST	18
C		CHECKING AGAINST THE KEY CARD (FOR NRUN=1 THIS TAPE IS OMITTED).	TEST	19
C		THE OUTPUT OF THIS PROGRAM IS-	TEST	20
C		1. A (FAILING) KEY CARD FOR POTENTIAL READING AS ABOVE.	TEST	21
C		2. A TAPE CONTAINING THE DUPLICATE OF THAT CARD.	TEST	22
C		3. A FORTRAN DECK FOR ASSEMBLY AND SOLUTION OF A CERTAIN PROBLEM.	TEST	23
C		THAT PROBLEM HAS NO DATA INPUT AND HAS OUTPUT AS FOLLOWS-	TEST	24
C		1. A (PASSING) KEY CARD FOR READING AS ABOVE.	TEST	25
C		2. A TAPE CONTAINING THE DUPLICATE OF THAT CARD.	TEST	26
C		3. A TAPE CONTAINING SOLUTION DATA FOR OFF-LINE PRINTING.	TEST	27
C		FOR EACH VALUE OF NRUN=1,2,3,... ONE OF TWO THINGS HAPPENS, EITHER	TEST	28
C		A. THE FORTRAN DECK IS GENERATED AND ASSEMBLES AND YIELDS OUTPUT-	TEST	29
C		IN THIS CASE THE PASSING KEY CARD AND ACCOMPANYING TAPE ARE	TEST	30
C		USED AS INPUT FOR THE NEXT GENERATION, OR	TEST	31
C		B. ALTERNATIVE A. DOES NOT OCCUR- IN THIS CASE THE FAILING KEY	TEST	32
C		CARD AND ACCOMPANYING TAPE ARE SUBSTITUTED.	TEST	33

C	/*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM	TEST	34
C	THE CERTAIN PROBLEM IS THE LAPLACE EQUATION FOR FIXED BOUNDARY IN	TEST	35
C	TWO RECTANGULAR DIMENSIONS, ITERATING THE REPLACEMENT OF EACH	TEST	36
C	GRID VALUE BY THE AVERAGE OF ITS 4 NEIGHBORS. SOLUTION TIME IS	TEST	37
C	HELD ESSENTIALLY CONSTANT BY A PARAMETER ITXP AS THE NUMBER N OF	TEST	38
C	GRID POINTS PER SIDE INCREASES UNDER A PRESCRIBED PROCEDURE OVER	TEST	39
C	INCREASING NRUN=1,2,3,....	TEST	40
C	UNDER AN AUXILIARY PARAMETER M (BIG FOR SMALL N - SMALL FOR BIG N)	TEST	41
C	THE NUMERICAL STORAGE REQUIREMENTS ARE HELD AS HIGH AS POSSIBLE	TEST	42
C	FOR EACH RUN- VERY HIGH FOR SMALL N BUT MUCH LOWER FOR LARGER N.	TEST	43
C	THE PROBLEM SOLUTION DECK IS SO GENERATED THAT SEPARATE STATEMENT	TEST	44
C	ARE USED FOR EACH GRID REPLACEMENT - THUS SPACE REQUIREMENTS FOR	TEST	45
C	PROGRAMMING INCREASE ULTIMATELY AS THE SQUARE OF N. ALSO A LINEAR	TEST	46
C	INCREASE RESULTS FROM CERTAIN CONVERGENCE CHECKING.	TEST	47
C	ULTIMATELY, FURTHER ITERATION BECOMES IMPOSSIBLE FOR SOME LARGE N.	TEST	48
C	INPUT-OUTPUT EQUIPMENT IDENTIFICATION IS MADE ONLY SCHEMATICALLY-	TEST	49
C	TO BE MADE PRECISE UPON SELECTION OF PARTICULAR EQUIPMENT.	TEST	50
C	INPUT-OUTPUT STATEMENTS ARE IDENTIFIED BY DIGITS PUNCHED IN COLS	TEST	51
C	1 AND 6 ACCORDING TO THE FOLLOWING SCHEDULE-	TEST	52
C	(COL.6)=3 TAPE ASSIGNMENT MADE BY GENERATOR FROM KEY CARD DATA,	TEST	53
C	(COL.6)=4 TAPE ASSIGNMENT MADE BY GENERATOR FROM KEY CARD DATA,	TEST	54
C	(COL.1)=0 CARD READING OR PUNCHING OR ON-LINE PRINTING,	TEST	55
C	(COL.1)=1 INPUT AND OUTPUT OF KEY CARD IMAGE VIA TAPE NTAPE,	TEST	56
C	(COL.1)=2 OFF LINE OUTPUT VIA TAPE NPRINT.	TEST	57
C	NOTE- THE ASSIGNMENTS OF NTAPE AND NPRINT ON THE KEY CARD MUST	TEST	58
C	MATCH THOSE MADE BY PROGRAMMING STATEMENTS WITH (COL.1)=1,2.	TEST	59
C	THE COMPOSITION OF THE KEY CARD IS SHOWN IN SEVERAL PLACES BELOW.	TEST	60
C	AN EXPLANATION OF FLAGGING (C,F,F) IN COLS. 1-3 IS SHOWN IN THE	TEST	61
C	DATA DECK BELOW.	TEST	62

C	/**	SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM	TEST	94
C		ADMINISTRATION - DIMENSIONS,EQUIVALENCES,FORMATS	TEST	95
C	/**/	FOLLOWING INPUT-OUTPUT STATEMENT IS FOR TAPE READ	TEST	96
		LTI= 5	TEST	97
C	/**/	FOLLOWING INPUT-OUTPUT STATEMENT IS FOR TAPE WRITE	TEST	98
		LTO=15	TEST	99
C		LIMIT ON R(,) AND S(,) IS FIXED AT 2 ABOVE MAXIMUM ALLOWABLE N=48	TEST	100
		LIMIT=48	TEST	101
		LIMITX=LIMIT+2	TEST	102
C		LIMIT ON T(,) IS FIXED AT 15+1=16	TEST	103
		DIMENSION R(50,72),IR(50,72)	TEST	104
		DIMENSION S(50,72),IS(50,72)	TEST	105
		DIMENSION T(16,72),IT(16,72)	TEST	106
		EQUIVALENCE (R,IR),(S,IS),(T,IT)	TEST	107
		DIMENSION FLAG(18)	TEST	108
		DIMENSION DIGIT(11)	TEST	109
		EQUIVALENCE (FLAG(1),C),(FLAG(2),F),(DIGIT(11),BLANK)	TEST	110
		DIMENSION KEY(13)	TEST	111
		DIMENSION LEY(13)	TEST	112
		DIMENSION TAPECD(5)	TEST	113
		DIMENSION TAPEPR(5)	TEST	114
		DIMENSION INTENS(5)	TEST	115
		EQUIVALENCE (KEY(1),NFLAG)	TEST	116
		EQUIVALENCE (KEY(2),NTAPE)	TEST	117
		EQUIVALENCE (KEY(3),NPRINT)	TEST	118
		EQUIVALENCE (KEY(4),MAYBE)	TEST	119
		EQUIVALENCE (KEY(5),NRUN)	TEST	120
		EQUIVALENCE (KEY(6),MARKER)	TEST	121
		EQUIVALENCE (KEY(7),NOUSE)	TEST	122
		EQUIVALENCE (KEY(8),N)	TEST	123
		EQUIVALENCE (KEY(9),M)	TEST	124
		EQUIVALENCE (KEY(10),ITXP)	TEST	125
		EQUIVALENCE (KEY(11),ITCT)	TEST	126
		EQUIVALENCE (KEY(12),LEVEL)	TEST	127
		EQUIVALENCE (KEY(13),CLOSE)	TEST	128
300		FORMAT(A1,I5,11I6,F8.0)	TEST	129
310		FORMAT(1H1I5,11I6,F8.0)	TEST	130
320		FORMAT(5A1,I1,66A1)	TEST	131
330		FORMAT(1H ,71A1,F7.0)	TEST	132
340		FORMAT(55H1PARAMETERS ON KEY CARD ARE ZERO OR LESS OR FAIL CHECKS)	TEST	133
350		FORMAT(53H WITH TAPE IMAGE - NEW KEY CARD HAS NOT BEEN PUNCHED)	TEST	134
360		FORMAT(48H1OPERATION TERMINATED - IMPASSE DEVELOPED ON RUNI4)	TEST	135
370		FORMAT(55H1THIS RUN PUNCHES A KEY CARD AND A DECK TO BE ASSEMBLED)	TEST	136
380		FORMAT(21H THIS RUN WRITES TAPE I3,19H FOR CONTINGENT USE)	TEST	137
390		FORMAT(52H PAUSE TO CHECK TAPE MOUNTING - THEN PUSH START)	TEST	138
400		FORMAT(31H GENERATION IS COMPLETE FOR RUNI3, 5H, N =I3, 5H, M =I3)	TEST	139
410		FORMAT(51H CONTINGENT KEY CARD IS PUNCHED AND WRITTEN ON TAPEI3)	TEST	140
420		FORMAT(52H GENERATED PROGRAM IS PUNCHED AND READY FOR ASSEMBLY)	TEST	141
430		FORMAT(55H PAUSE TO DISMOUNT TAPES, THEN PUSH START TO GET OFF)	TEST	142
440		FORMAT(1H)	TEST	143

C	/*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM	TEST	63
C	THE ENTIRE TEST PROCEDURE IS AS FOLLOWS-	TEST	64
C	1. SELECT THREE TAPE DRIVES-	TEST	65
C	A. ONE FOR OFF-LINE PRINTING ----- CALL THIS ONE NPRINT	TEST	66
C	B. ONE FOR RECORDING THE KEY CARD DUPLICATE- CALL THIS ONE NTAPE	TEST	67
C	C. ONE FROM WHICH TO READ THAT DUPLICATE --- CALL THIS ONE NDUPL.	TEST	68
C	(SELECT NTAPE AND NDUPL TO BE ON SEPARATE INPUT-OUTPUT TRUNKS.)	TEST	69
C	2. ADJUST THE FOLLOWING PROGRAMMING TO DESIGNATE-	TEST	70
C	A. NPRINT WHERE A (2) APPEARS IN COL.2,	TEST	71
C	B. NTAPE WHERE A (1) APPEARS IN COL.1 FOR OUTPUT,	TEST	72
C	C. NDUPL WHERE A (1) APPEARS IN COL.1 FOR INPUT.	TEST	73
C	3. PREPARE AN INITIAL KEY CARD (SEE FORMAT IN DATA DECK COMMENTS)	TEST	74
C	WITH THE INTEGER 1 IN EACH FIELD (BUT CLOSE=1.) EXCEPT SEVEN-	TEST	75
C	NTAPE = DECISION ABOVE	TEST	76
C	NPRINT = DECISION ABOVE	TEST	77
C	MARKER = 2 (SEE M BELOW)	TEST	78
C	N = 8	TEST	79
C	M = 32 (MARKER=2 WILL REDUCE THIS TO M=16 ON THE FIRST RUN)	TEST	80
C	ITXP = 4 (OR LESS) TO CONTROL SOLUTION TIME	TEST	81
C	LEVEL = 8 (OR SO)	TEST	82
C	(INFLAG=9999 IS SET BY PROGRAMMING).	TEST	83
C	4. FOR EACH NEW RUN (EXCEPT THE FIRST) DO THE FOLLOWING-	TEST	84
C	A. USE ONE OF TWO NEW KEY CARDS AND ITS TAPE DUPLICATE	TEST	85
C	(USE THE PRECEDING PASSING KEY CARD IF IT WAS PRODUCED -	TEST	86
C	(USE THE PRECEDING FAILING KEY CARD OTHERWISE	TEST	87
C	(BEGIN WITH THE INITIAL KEY CARD WITHOUT ITS TAPE DUPLICATE	TEST	88
C	TO PRODUCE A CONTINGENT (FAILING) KEY CARD AND A FORTRAN DECK,	TEST	89
C	B. --- IF POSSIBLE --- USE THAT FORTRAN DECK TO PRODUCE OFF-LINE	TEST	90
C	SOLUTION DATA AND A (PASSING) KEY CARD.	TEST	91
C	5. OBSERVE ON-LINE PRINTING AND QUIT WHEN AN IMPASSE IS REPORTED.	TEST	92
C	6. SAVE ALL KEY-CARDS AND ON-LINE AND OFF-LINE PRINTINGS.	TEST	93

C	/*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM	TEST 183
C	PREPARE NEW KEY DATA	TEST 184
	NRUN=NRUN+1	TEST 185
	ITCT=1	TEST 186
	CLOSE=0.0	TEST 187
C	INSURE NFLAG=9999 TO DISTINGUISH KEY CARD FROM GENERATED PROGRAM	TEST 188
	NFLAG=9999	TEST 189
C	COMPUTE K=8,4,2,1 AS HIGHEST POWER OF 2 WHICH DIVIDES N EXACTLY	TEST 190
	K=8	TEST 191
	DO 1134 I=1,3	TEST 192
	J=N/K	TEST 193
	IF(N-K*J)	1134, 1135, 1134 TEST 194
1134	K=K/2	TEST 195
C	QUIT IF N PASSES LIMIT FOR MARKER=1 OR IF M=K=1	TEST 196
1135	IF(N-LIMIT)	1137, 1136, 1410 TEST 197
1136	IF(MARKER-1)	1136, 1410, 1137 TEST 198
1137	IF((M-1)+(K-1))	1137, 1410, 1138 TEST 199
C	ESTABLISH NEW N,M	TEST 200
1138	IF(2-MARKER)	1138, 1139, 1140 TEST 201
1139	IF(M-1)	1139, 1142, 1149 TEST 202
1140	IF(M-1)	1140, 1143, 1141 TEST 203
C	FOR LAST RUN PASSING FOR M ABOVE 1, RAISE N BY 8	TEST 204
1141	K=8	TEST 205
	GO TO 1145	TEST 206
C	FOR LAST RUN FAILING FOR M EQUAL 1, LOWER N BY K/2 AND SET NOUSE=2	TEST 207
1142	K=-K/2	TEST 208
	NOUSE=2	TEST 209
	GO TO 1145	TEST 210
C	FOR LAST RUN PASSING FOR M EQUAL 1, RAISE N BY K OR K/2	TEST 211
1143	IF(NOUSE -1)	1143, 1145, 1144 TEST 212
C	FOR NOUSE=2, RAISE N BY K/2	TEST 213
1144	K=K/2	TEST 214
C	FOR NOUSE=1, RAISE N BY K	TEST 215
1145	N=N+K	TEST 216
	GO TO 1150	TEST 217
C	FOR LAST RUN FAILING FOR M ABOVE 1, DIVIDE M BY 2	TEST 218
1149	M=M/2	TEST 219
C	INSERT HERE ANY FURTHER CHECKING, SUCH AS MAYBE=F(NRUN) FOR SOME F	TEST 220
1150	CONTINUE	TEST 221
C	SET MARKER TO (CONTINGENT) FAILURE	TEST 222
	MARKER=2	TEST 223
C	PUNCH AND WRITE CONTINGENT FAILURE KEY AND PRINT IT ON LINE	TEST 224
C	/*/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR CARD PUNCH	TEST 225
01161	WRITE OUTPUT TAPE 14, 310, (KEY(I),I=1,13)	TEST 226
C	/*/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR TAPE WRITE	TEST 227
11165	REWIND LTO	TEST 228
C	/*/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR TAPE WRITE	TEST 229
11166	WRITE TAPE LTO, (KEY(I),I=1,13)	TEST 230
C	/*/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR TAPE WRITE	TEST 231
11167	END FILE LTO	TEST 232
C	/*/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 233
01168	WRITE OUTPUT TAPE 81, 300, BLANK,(KEY(I),I=1,13)	TEST 234

C /**	SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM	TEST 144
C	READ FLAG,DIGIT,KEY	TEST 145
C	**** FOLLOWING INPUT-OUTPUT STATEMENT IS FOR CARD READ	TEST 146
00500	READ 320, (FLAG(I),I=1,18),(DIGIT(I),I=1,11)	TEST 147
C	**** FOLLOWING INPUT-OUTPUT STATEMENT IS FOR CARD READ	TEST 148
00501	READ 300, CLOSE,(KEY(I),I=1,13)	TEST 149
C	PRINT ON-LINE REQUEST TO CHECK TAPE MOUNTING AND ROLL UP PRINTER	TEST 150
C	**** FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 151
00550	WRITE OUTPUT TAPE 81, 370	TEST 152
C	**** FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 153
00551	WRITE OUTPUT TAPE 81, 380, NPRINT	TEST 154
C	**** FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 155
00552	WRITE OUTPUT TAPE 81, 390	TEST 156
	DO 559 I=1,10	TEST 157
C	**** FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 158
00559	WRITE OUTPUT TAPE 81, 440	TEST 159
C	PAUSE PAUSE PAUSE PAUSE PAUSE PAUSE PAUSE PAUSE PAUSE PAUSE	TEST 160
	PAUSE	TEST 161
C	WRITE OLD KEY BOTH ON AND OFF LINE	TEST 162
C	**** FOLLOWING INPUT-OUTPUT STATEMENT IS FOR OFF-LINE PRINT	TEST 163
00598	PRINT 300, BLANK,(KEY(I),I=1,13)	TEST 164
C	**** FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 165
00599	WRITE OUTPUT TAPE 81, 300, BLANK,(KEY(I),I=1,13)	TEST 166
C	(EXCEPT FOR NRUN=1) READ TAPE IMAGE OF KEY	TEST 167
	IF(NRUN=1) 1400, 1107,11105	TEST 168
C	**** FOLLOWING INPUT-OUTPUT STATEMENT IS FOR TAPE READ	TEST 169
11105	REWIND LTI	TEST 170
C	**** FOLLOWING INPUT-OUTPUT STATEMENT IS FOR TAPE READ	TEST 171
11106	READ TAPE LTI, (LEY(I),I=1,13)	TEST 172
C	CHECK KEY AND (EXCEPT FOR NRUN=1) COMPARE TAPE IMAGE	TEST 173
11107	DO 1111 I=1,12	TEST 174
	IF(KEY(I)) 1400, 1400, 1109	TEST 175
11109	IF(NRUN=1) 1109, 1111, 1110	TEST 176
11110	IF(KEY(I)-LEY(I)) 1400, 1111, 1400	TEST 177
11111	CONTINUE	TEST 178
C	GENERATE INTENS = 10000,1000,100,10,1	TEST 179
	INTENS(1)=10000	TEST 180
	DO 1114 I=1,4	TEST 181
11114	INTENS(I+1)=INTENS(I)/10	TEST 182

C /*/	SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM	TEST 235
C	PREPARE KEY FOR GENERATED (PASSING) PROGRAM	TEST 236
	MARKER=1	TEST 237
C	CREATE DECIMAL FORM FOR KEY IN S(,) AND SET TAPECD AND TAPEPR	TEST 238
	NN=N+2	TEST 239
	KEY(13)=NN*M	TEST 240
	DO 1184 I=1,13	TEST 241
	LEY(I)=KEY(I)	TEST 242
	DO 1180 J=1,5	TEST 243
	K=LEY(I)/INTENS(J)	TEST 244
	L=K+1	TEST 245
	S(I,J)=DIGIT(L)	TEST 246
1180	LEY(I)=LEY(I)-K*INTENS(J)	TEST 247
	DO 1183 J=1,5	TEST 248
	IF(S(I,J)-DIGIT(1))	1184, 1183, 1184TEST 249
1183	S(I,J)=BLANK	TEST 250
1184	CONTINUE	TEST 251
	DO 1187 I=1,5	TEST 252
	TAPECD(I)=S(2,I)	TEST 253
1187	TAPEPR(I)=S(3,I)	TEST 254

C	/*	SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM	TEST 255
C		READ, COMPLETE, PUNCH NON-LOOP PROGRAMMING	TEST 256
C	/*	FOLLOWING INPUT-OUTPUT STATEMENT IS FOR CARD READ	TEST 257
01200	READ	320, (R(1,J),J=1,72)	TEST 258
	IF(R(1,1)-C)	1204, 1202, 1204	TEST 259
1202	IF(R(1,2)-F)	1200, 1203, 1200	TEST 260
1203	IF(R(1,3)-F)	1300, 1430, 1300	TEST 261
C		SORT CASES	TEST 262
1204	IF(IR(1,6)-1)	1290, 1210, 1205	TEST 263
1205	IF(IR(1,6)-3)	1220, 1230, 1240	TEST 264
C		CASE 1	TEST 265
1210	DO 1213 J=1,5		TEST 266
	R(1,J+13)=S(1,J)		TEST 267
	DO 1213 I=1,12		TEST 268
1213	S(I,J)=S(I+1,J)		TEST 269
	GO TO 1290		TEST 270
C		CASE 2	TEST 271
1220	DO 1222 I=1,3		TEST 272
	R(1,I+18)=S(13,I+2)		TEST 273
1222	R(1,I+22)=S(13,I+2)		TEST 274
	GO TO 1290		TEST 275
C		CASE 3	TEST 276
1230	DO 1231 I=1,3		TEST 277
1231	R(1,I+27)=TAPECD(I+2)		TEST 278
	GO TO 1290		TEST 279
C		CASE 4	TEST 280
1240	DO 1241 I=1,3		TEST 281
1241	R(1,I+27)=TAPEPR(I+2)		TEST 282
C		ALL CASES	TEST 283
1290	R(1,6)=BLANK		TEST 284
	CLOSE=CLOSEF+1.0		TEST 285
C	/*	FOLLOWING INPUT-OUTPUT STATEMENT IS FOR CARD PUNCH	TEST 286
01295	WRITE OUTPUT TAPE 14,	330, (R(1,J),J=2,72) ,CLOSE	TEST 287
	GO TO 1200		TEST 288

C	/**/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM	TEST 289
C	READ LOOP	TEST 290
C	/**/ FOLLOWING INPUT-OUTPUT STATEMENT IS FOR CARD READ	TEST 291
01300	READ 320, (R(1,J),J=1,72)	TEST 292
	IF(R(1,1)-C) 1302, 1300, 1302	TEST 293
C	/**/ FOLLOWING INPUT-OUTPUT STATEMENT IS FOR CARD READ	TEST 294
01302	READ 320, (S(1,J),J=1,72)	TEST 295
	IF(S(1,1)-C) 1304, 1302, 1304	TEST 296
1304	I=0	TEST 297
1305	I=I+1	TEST 298
C	/**/ FOLLOWING INPUT-OUTPUT STATEMENT IS FOR CARD READ	TEST 299
01306	READ 320, (T(I,J),J=1,72)	TEST 300
	IF(T(I,1)-C) 1305, 1308, 1305	TEST 301
1308	IF(T(I,2)-F) 1306, 1320, 1306	TEST 302
C	EXPAND R,S	TEST 303
1320	DO 1324 I=2,LIMITX	TEST 304
	DO 1324 J=1,72	TEST 305
	R(I,J)=R(1,J)	TEST 306
	S(I,J)=S(1,J)	TEST 307
1324	R(I,17)=R(1,37)	TEST 308
C	CREATE BASIC COLUMN INDICIES IN (R,)	TEST 309
	DO 1338 I=1,10	TEST 310
	DO 1338 J=1,10	TEST 311
	K=10*I+J-10	TEST 312
	IF(K-LIMITX) 1334, 1334, 1340	TEST 313
1334	IF(I-1) 1334, 1335, 1336	TEST 314
1335	IF(J-1) 1335, 1338, 1337	TEST 315
1336	R(K,18)=DIGIT(I)	TEST 316
1337	R(K,19)=DIGIT(J)	TEST 317
1338	CONTINUE	TEST 318
C	SPREAD COLUMN INDICIES IN R,S	TEST 319
1340	DO 1347 I=1,LIMIT	TEST 320
	DO 1347 J=1,3	TEST 321
	R(I+1,J+32)=R(I,J+16)	TEST 322
	R(I+1,J+47)=R(I+2,J+16)	TEST 323
	R(I+1,J+62)=R(I+1,J+16)	TEST 324
	S(I+1,J+16)=R(I+1,J+16)	TEST 325
	S(I+1,J+32)=R(I+1,J+16)	TEST 326
1347	S(I+1,J+47)=R(I+1,J+16)	TEST 327

C	/*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM	TEST 328
C	COMPLETE AND PUNCH LOOP PROGRAMMING FOR ROWS=1,2,3,---N	TEST 329
	DO 1390 IROW=1,N	TEST 330
C	SPREAD ROW INDICIES IN R,S	TEST 331
	DO 1359 I=1,LIMIT	TEST 332
	DO 1359 J=1,3	TEST 333
	R(I+1,J+10)=R(IROW+1,J+16)	TEST 334
	R(I+1,J+26)=R(IROW+1,J+16)	TEST 335
	R(I+1,J+41)=R(IROW+1,J+16)	TEST 336
	R(I+1,J+56)=R(IROW ,J+16)	TEST 337
	S(I+1,J+10)=R(IROW+1,J+16)	TEST 338
	S(I+1,J+26)=R(IROW+2,J+16)	TEST 339
1359	S(I+1,J+41)=R(IROW+1,J+16)	TEST 340
C	SPREAD ROW INDEX KEY IN T	TEST 341
	DO 1375 J=1,2	TEST 342
	T(7,J+16)=R(2,J+11)	TEST 343
	T(13,J+16)=R(2,J+11)	TEST 344
	T(4,J+11)=R(2,J+11)	TEST 345
	IF(T(4,J+11)-BLANK)	1364, 1363, 1364 TEST 346
1363	T(4,J+11)=DIGIT(1)	TEST 347
1364	T(11,J+11)=T(4,J+11)	TEST 348
	DO 1366 I=13,15	TEST 349
1366	T(I,J+2)=T(4,J+11)	TEST 350
	DO 1368 I= 6,10	TEST 351
1368	T(I,J+2)=T(4,J+11)	TEST 352
	T(3,J+2)=T(4,J+11)	TEST 353
	DO 1375 K=57,69,6	TEST 354
	L=J+K	TEST 355
	DO 1373 I=5,13	TEST 356
1373	T(I,L)=T(4,J+11)	TEST 357
	DO 1375 I=8,11	TEST 358
1375	T(I,L)=BLANK	TEST 359
C	PUNCH	TEST 360
	DO 1382 I=1,N	TEST 361
	R(I+1,6)=BLANK	TEST 362
	CLOSE=CLOSE+1.0	TEST 363
C	/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR CARD PUNCH	TEST 364
01381	WRITE OUTPUT TAPE 14, 330, (R(I+1,J),J=2,72) ,CLOSE	TEST 365
	S(I+1,6)=BLANK	TEST 366
	CLOSE=CLOSE+1.0	TEST 367
C	/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR CARD PUNCH	TEST 368
01382	WRITE OUTPUT TAPE 14, 330, (S(I+1,J),J=2,72) ,CLOSE	TEST 369
	DO 1385 I=1,15	TEST 370
	T(I,6)=BLANK	TEST 371
	CLOSE=CLOSE+1.0	TEST 372
C	/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR CARD PUNCH	TEST 373
01385	WRITE OUTPUT TAPE 14, 330, (T(I,J),J=2,72) ,CLOSE	TEST 374
1390	CONTINUE	TEST 375
	GO TO 1200	TEST 376

C /**/	SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM	TEST 377
C	ALARMS	TEST 378
C	ERROR ALARMS	TEST 379
C	/***/ FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 380
01400	WRITE OUTPUT TAPE 81, 340	TEST 381
C	/***/ FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 382
01401	WRITE OUTPUT TAPE 81, 350	TEST 383
	GO TO 1420	TEST 384
C	TERMINAL ALARMS	TEST 385
C	/***/ FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 386
01410	WRITE OUTPUT TAPE 81, 360	TEST 387
C	PRINT KEY DATA	TEST 388
C	/***/ FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 389
01420	WRITE OUTPUT TAPE 81, 300, BLANK,(KEY(I),I=1,13)	TEST 390
C	/***/ FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 391
01421	WRITE OUTPUT TAPE 81, 300, BLANK,(LEY(I),I=1,13)	TEST 392
	GO TO 1440	TEST 393
C	SUCCESSFUL GENERATION	TEST 394
C	/***/ FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 395
01430	WRITE OUTPUT TAPE 81, 400, NRUN,N,M	TEST 396
C	/***/ FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 397
01431	WRITE OUTPUT TAPE 81, 410, NPRINT	TEST 398
C	/***/ FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 399
01432	WRITE OUTPUT TAPE 81, 420	TEST 400
C	PRINT ON-LINE STATEMENT OF COMPLETION AND ROLL UP PRINTER	TEST 401
C	/***/ FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 402
01440	WRITE OUTPUT TAPE 81, 430	TEST 403
C	***** END FILE ON PUNCH TAPE ***** END FILE ON PUNCH TAPE *****	TEST 404
C	/***/ FOLLOWING INPUT-OUTPUT STATEMENT IS FOR CARD PUNCH	TEST 405
01443	END FILE 14	TEST 406
	DO 1445 I=1,10	TEST 407
C	/***/ FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 408
01445	WRITE OUTPUT TAPE 81, 440	TEST 409
C	PAUSE PAUSE PAUSE PAUSE PAUSE PAUSE PAUSE PAUSE PAUSE PAUSE	TEST 410
	PAUSE	TEST 411
	CALL SYSTEM	TEST 412

```

C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM TEST 413
  END TEST 414
  LOAD BATCH TEST 415
  TRA TEST 416
CFLAG DATA DECK -(0123456789 )- REPLACE THE NEXT (KEY) CARD FOR EACH RUNTEST 417
1 9999 15 16 1 1 2 1 8 32 13 1 7TEST 418
C COMPOSITION OF PRECEDING (KEY) CARD IS DETAILED BELOW TEST 419
C ALL COMMENT CARDS ARE IGNORED BY GENERATOR EXCEPT SOME WHICH SERVETEST 420
C AS FLAGS BY VIRTUE OF (F) IN COL 2 (TERMINAL FLAG HAS F IN COL 3)TEST 421
C 1ST FLAG = DATA DECK HEADER TEST 422
C 2ND FLAG = LOOP BEGINNING TEST 423
C 3RD FLAG = LOOP ENDING TEST 424
C 4TH FLAG = TERMINAL FLAG TEST 425
C COMPOSITION OF KEY CARD AS FOLLOWS TEST 426
C COL 1 = = DIGIT 1 FOR PAGE ADVANCE IN PRINTING TEST 427
C COLS 2- 6 = NFLAG = 9999 TO BE DISTINCT FROM STATEMENT NUMBERS)TEST 428
C COLS 7-12 = NTAPE = NUMBER OF TAPE HOLDING COPY OF KEY TEST 429
C COLS 13-18 = NPRINT = NUMBER OF TAPE CONTAINING OUTPUT FOR PRINTERTEST 430
C COLS 19-24 = MAYBE = FOR POTENTIAL SPECIAL USE TEST 431
C COLS 25-30 = NRUN = RUN NUMBER TEST 432
C COLS 31-36 = MARKER = (1) FOR RUN COMPLETED OR (2) FOR FAILURE TEST 433
C COLS 37-42 = NOUSE = (1) FOR NO M=1 FAILURES OR (2) OTHERWISE TEST 434
C COLS 43-48 = N = NUMBER OF INTERIOR GRIDS PER BLOCK SIDE TEST 435
C COLS 49-54 = M = NUMBER OF BLOCKS PER SIDE OF TOTAL ARRAY TEST 436
C COLS 55-60 = ITXP = TIME CONTROL PARAMETER - SEE BELOW TEST 437
C COLS 61-66 = ITCT = NUMBER OF BLOCK ITERATIONS -SEE BELOW TEST 438
C COLS 67-72 = LEVEL = POWER OF 2 SEPARATING GRID FROM BOUNDARY TEST 439
C COLS 73-80 = CLOSE = NUMBER OF TIMES GRIDS LIE WITHIN TOLERANCE TEST 440
C ITCT = ((2**ITXP)/(M*M))/(N*N) TEST 441
C NUMBER OF GRIDS REPLACED = ITCT*(M*M)*(N*N)TEST 442

```

```

C /**/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM TEST 443
C MAJOR BLOCK STORAGE DISPLAY TEST 444
C TEST 445
C 123*Z ETC AS AT LEFT MAJOR COL INDEX Z TEST 446
C 111*1222*2*****MMM*M BLOCK COL INDEX DENOTESTEST 447
C 123*Z123*Z*****123*Z LOCAL COL INDEX NN=N+2TEST 448
C TEST 449
C 1 1 1 XXXXXXXXXXXX*****XXXXX TEST 450
C 2 1 2 X XX X* *X X TEST 451
C 3 1 3 X 1 XX 2 X* *X M X TEST 452
C * * * X XX X* *X X TEST 453
C NN=N+2 1 N XXXXXXXXXXXX*****XXXXX TEST 454
C NN+1 2 1 XXXXXXXXXXXX*****XXXXX TEST 455
C NN+2 2 2 X XX X* *X X TEST 456
C NN+3 2 3 XM+1XXM+2X* *X2*MX X DENOTES BOUNDARY TEST 457
C * * * X XX X* *X X * SHOWS TRANSITION TEST 458
C 2*NN 2 NN XXXXXXXXXXXX*****XXXXX TEST 459
C * * * ***** TEST 460
C * * * * ** ** ** * NUMBERS SHOWN IN BLOCKS TEST 461
C * * * * ** ** ** * INDICATE BLOCK SEQUENCE TEST 462
C * * * * ** ** ** * TEST 463
C * * * ***** TEST 464
C (M-1)*NN+1 M 1 XXXXXXXXXXXX*****XXXXX TEST 465
C (M-1)*NN+2 M 2 X XX X* X X TEST 466
C (M-1)*NN+3 M 3 X XX X* XM*MX TEST 467
C * * * X XX X* X X TEST 468
C M*NN M NN XXXXXXXXXXXX*****XXXXX TEST 469
C TEST 470
C COLUMNS AT LEFT SHOW ( LOCAL ROW INDEX AT RIGHT TEST 471
C ( BLOCK ROW INDEX IN CENTERTEST 472
C ( MAJOR ROW INDEX AT LEFT TEST 473

```

C	/*	SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM	TEST 474
C		ADMINISTRATION - DIMENSIONS EQUIVALENCES, FORMATS	TEST 475
C		DIMENSION OF T(,) IS SET BY GENERATOR AT T(Y,Y) FOR $Y=M*NN=M*(N+2)$	TEST 476
	2	DIMENSION T(,)	TEST 477
		DIMENSION KEY(13)	TEST 478
		EQUIVALENCE (KEY(1),NFLAG)	TEST 479
		EQUIVALENCE (KEY(2),NTAPE)	TEST 480
		EQUIVALENCE (KEY(3),NPRINT)	TEST 481
		EQUIVALENCE (KEY(4),MAYBE)	TEST 482
		EQUIVALENCE (KEY(5),NRUN)	TEST 483
		EQUIVALENCE (KEY(6),MARKER)	TEST 484
		EQUIVALENCE (KEY(7),NOUSE)	TEST 485
		EQUIVALENCE (KEY(8),N)	TEST 486
		EQUIVALENCE (KEY(9),M)	TEST 487
		EQUIVALENCE (KEY(10),ITXP)	TEST 488
		EQUIVALENCE (KEY(11),ITCT)	TEST 489
		EQUIVALENCE (KEY(12),LEVEL)	TEST 490
		EQUIVALENCE (KEY(13),CLOSE)	TEST 491
300		FORMAT(1H1I5,11I6,F8.0)	TEST 492
310		FORMAT(8F15.9)	TEST 493
320		FORMAT(21H1THIS RUN WRITES TAPE13,23H FOR NEXT GENERATOR RUN)	TEST 494
330		FORMAT(21H THIS RUN WRITES TAPE13,22H FOR OFF-LINE PRINTING)	TEST 495
340		FORMAT(52H PAUSE TO CHECK TAPE MOUNTING - THEN PUSH START)	TEST 496
350		FORMAT(30H1SOLUTION IS COMPLETED FOR RUN13, 5H, N =13, 5H, M =13)	TEST 497
360		FORMAT(43H KEY CARD IS PUNCHED AND IS WRITTEN ON TAPE13)	TEST 498
370		FORMAT(40H OUTPUT FOR OFF-LINE PRINTING IS ON TAPE13)	TEST 499
380		FORMAT(55H PAUSE TO DISMOUNT TAPES, THEN PUSH START TO GET OFF)	TEST 500
390		FORMAT(1H)	TEST 501
400		FORMAT(1H 15,11I6,F8.0)	TEST 502

C	/*	SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM	TEST 503
C		PRELIMINARY - ALL BUT TWO (BELOW) KEY VALUES ARE SET BY GENERATOR	TEST 504
	1NFLAG =		TEST 505
C	*/	FOLLOWING INPUT-OUTPUT STATEMENT IS FOR TAPE WRITE	TEST 506
	1NTAPE =		TEST 507
C	*/	FOLLOWING INPUT-OUTPUT STATEMENT IS FOR OFF-LINE PRINT SPCL.	TEST 508
	1NPRINT=		TEST 509
	1MAYBE =		TEST 510
	1NRUN =		TEST 511
	1MARKER=		TEST 512
	1NOUSE =		TEST 513
	1N =		TEST 514
	1M =		TEST 515
	1ITXP =		TEST 516
	1ITCT =		TEST 517
	1LEVEL =		TEST 518
C		CLOSE IS INITIALIZED LATER	TEST 519
C		ITCT IS INITIALIZED LATER	TEST 520
C		PRINT ON-LINE REQUEST TO CHECK TAPE MOUNTING AND ROLL UP PRINTER	TEST 521
C	*/	FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 522
00450	WRITE	OUTPUT TAPE 81, 320, NTAPE	TEST 523
C	*/	FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 524
00451	WRITE	OUTPUT TAPE 81, 330, NPRINT	TEST 525
C	*/	FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 526
00452	WRITE	OUTPUT TAPE 81, 340	TEST 527
	DO	455 KK=1,10	TEST 528
C	*/	FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 529
00455	WRITE	OUTPUT TAPE 81, 390	TEST 530
C		PAUSE PAUSE PAUSE PAUSE PAUSE PAUSE PAUSE PAUSE PAUSE PAUSE	TEST 531
	PAUSE		TEST 532
C	*/	FOLLOWING INPUT-OUTPUT STATEMENT IS FOR OFF-LINE PRINT	TEST 533
00458	PRINT	400, (KEY(I),I=1,13)	TEST 534
C	*/	FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 535
00459	WRITE	OUTPUT TAPE 81, 400, (KEY(I),I=1,13)	TEST 536

C	/*	SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM	TEST 537
C		INITIALIZATION OF STORAGE AND COUNTERS	TEST 538
		NN=N+2	TEST 539
		MM=M*M	TEST 540
	DO	590 IM=1,M	TEST 541
	DO	590 JM=1,M	TEST 542
	DO	590 IO=1,NN	TEST 543
	DO	590 JO=1,NN	TEST 544
		IN=(IM-1)*NN+IO	TEST 545
		JN=(JM-1)*NN+JO	TEST 546
C		SET EACH (INTERIOR AND BOUNDARY) GRID VALUE AT $1+(1/2)**1$	TEST 547
		T(IN,JN)=1.5	TEST 548
C		RESET EACH BOUNDARY VALUE AT ----- $1+(1/2)**(LEVEL+1)$	TEST 549
550	DO	590 KK=1,LEVEL	TEST 550
		IF((IO-1)*(JO-1)*(NN-IO)*(NN-JO))	550, 580, 590 TEST 551
580		T(IN,JN)=0.5+T(IN,JN)/2.0	TEST 552
590		CLOSE=0	TEST 553
C		CLOSE WILL COUNT THE NUMBER OF TIMES A COMPUTED VALUE FOR INTERIOR	TEST 554
C		GRIDS IS EQUAL TO OR BELOW $1+(1/2)**LEVEL$ -WITHIN $(1/2)**(LEVEL+1)$	TEST 555
C		OF THE FINAL SOLUTION OF ALL INTERIOR GRIDS EQUAL TO BOUNDARY.	TEST 556
		ITCT=1	TEST 557
	DO	690 KK=1,ITXP	TEST 558
690		ITCT=ITCT+ITXP	TEST 559
		ITCT=(ITCT/MM)/(N*N)	TEST 560
		IF(ITCT-1)	695, 696, 696 TEST 561
695		ITCT=1	TEST 562
696		CONTINUE	TEST 563
C		ITCT ESTABLISHES THE NUMBER OF ITERATIONS OVER THE ENTIRE ARRAY OF	TEST 564
C		MM BLOCKS	TEST 565
		TOLER=2.0*T(1,1)-1.0	TEST 566
C		TOLER = TOLERANCE VALUE OF $1+(1/2)**LEVEL$	TEST 567
C		CONTROL	TEST 568
C		ITERATE MM BLOCK ITERATIONS ITCT TIMES	TEST 569
	DO	9000 ITER=1,ITCT	TEST 570
C		ITERATE M SETS OF M BLOCK ITERATIONS INTO BLOCKS ALONG ROWS IM	TEST 571
	DO	9000 IM=1,M	TEST 572
		IN=(IM-1)*NN+1	TEST 573
C		ITERATE M BLOCK ITERATIONS INTO BLOCKS COL JM ON BLOCK ROW IM	TEST 574
	DO	9000 JM=1,M	TEST 575
		JO=JN	TEST 576
		JN=(JM-1)*NN+1	TEST 577
C		SETTINGS OF IO AND JO ARE INCORRECT ONLY FOR JM=2 AND IM=JM=ITER=1	TEST 578
		IF(JM-2)	850, 860, 880 TEST 579
850		IF((IM-1)+(ITER-1))	850, 860, 880 TEST 580
860		IO=IN	TEST 581
		JO=1.0	TEST 582
880		CONTINUE	TEST 583
C		JO = INDEX OF FIRST (BOUNDARY) COL OF OLD BLOCK	TEST 584
C		IO = INDEX OF FIRST (BOUNDARY) ROW OF OLD BLOCK	TEST 585
C		JN = INDEX OF FIRST (BOUNDARY) COL OF NEW BLOCK	TEST 586
C		IN = INDEX OF FIRST (BOUNDARY) ROW OF NEW BLOCK	TEST 587
C		ITERATE	TEST 588

C	/*	SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM	TEST 589
CF		DATA DECK FOR GENERATOR - RECURRING PROGRAMMING BEGINNING	TEST 590
C		GENERATOR REPEATS ALL THIS PROGRAMMING N TIMES	TEST 591
C		GENERATOR REPEATS NEXT TWO STATEMENTS N TIMES (TOTAL OF N*N TIMES)	TEST 592
		T(IN ,JN)= T(IN ,JN)+T(IO ,JO)+T(IN ,JN)	TEST 593
		T(IN ,JN)=(T(IO ,JO)+T(IN ,JN))/4.0	TEST 594
		KK=JN-1	TEST 595
		KW=KK+NN	TEST 596
1	1	KA=KK+1	TEST 597
C		REPEAT LOOP AS LONG AS CLOSENESS CONDITION FAILS EXCEPT FOR KK=JN	TEST 598
		DO 1 4 KK=KA,KW	TEST 599
		IF(KK-KW)	1 2, 1 9, 1 1TEST 600
1	2	IF(KK-JN)	1 2, 1 6, 1 3TEST 601
1	3	IF((T(IN+ ,KK)-TOLER))	1 5, 1 5, 1 4TEST 602
1	4	CONTINUE	TEST 603
1	5	CLOSE=CLOSE+1.0	TEST 604
1	6	KA=KK+1	TEST 605
C		REPEAT LOOP AS LONG AS CLOSENESS CONDITION DOES NOT FAIL	TEST 606
		DO 1 8 KK=KA,KW	TEST 607
		IF(KW-KK)	1 6, 1 9, 1 7TEST 608
1	7	IF(-(T(IN+ ,KK)-TOLER))	1 1, 1 8, 1 8TEST 609
1	8	CLOSE=CLOSE+1.0	TEST 610
1	9	CONTINUE	TEST 611
CF		DATA DECK FOR GENERATOR - RECURRING PROGRAMMING ENDING	TEST 612
C		KA = FIRST (BOUNDARY) INDEX FOR TOLERANCE SEARCH	TEST 613
C		KK = INDEX (VARIABLE) VALUE FOR TOLERANCE SEARCH	TEST 614
C		KW = FINAL (BOUNDARY) INDEX FOR TOLERANCE SEARCH	TEST 615

C	/*	SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM	TEST 616
C		TERMINATION	TEST 617
C		FOLLOWING CONTINUE STATEMENT TERMINATES OVERALL BLOCK-ITERATION	TEST 618
	9000	CONTINUE	TEST 619
C		REWIND TAPES AND PRINT AND WRITE KEY	TEST 620
C		/*/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR TAPE WRITE	TEST 621
	190013	REWIND LTO	TEST 622
C		/*/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR OFF-LINE PRINT SPCL.	TEST 623
	290024	REWIND LTP	TEST 624
C		/*/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR TAPE WRITE	TEST 625
	190033	WRITE TAPE LTO, (KEY(KK),KK=1,13)	TEST 626
C		/*/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR OFF-LINE PRINT SPCL.	TEST 627
	290044	WRITE OUTPUT TAPE LTP, 300, (KEY(KK),KK=1,13)	TEST 628
C		WRITE MAJOR DIAGONALS	TEST 629
		KW=NN*M	TEST 630
		DO 9100 KK=1,KW	TEST 631
	9100	T(1,KK)=T(KK,KK)	TEST 632
		DO 9200 KA=1,NN	TEST 633
		KW=NN*(M-1)+KA	TEST 634
C		/*/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR OFF-LINE PRINT SPCL.	TEST 635
	292004	WRITE OUTPUT TAPE LTP, 310, (T(1,KK),KK=KA,KW,NN)	TEST 636
C		TERMINATE TAPES AND PUNCH KEY CARD	TEST 637
C		/*/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR TAPE WRITE	TEST 638
	193003	END FILE LTO	TEST 639
C		/*/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR OFF-LINE PRINT SPCL.	TEST 640
	293014	END FILE LTP	TEST 641
C		/*/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR CARD PUNCH	TEST 642
	09309	WRITE OUTPUT TAPE 14, 300, (KEY(KK),KK=1,13)	TEST 643
C		***** END FILE ON PUNCH TAPE ***** END FILE ON PUNCH TAPE *****	TEST 644
C		/*/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR CARD PUNCH	TEST 645
	09310	END FILE 14	TEST 646
C		PRINT ON-LINE STATEMENT OF COMPLETION AND ROLL UP PRINTER	TEST 647
C		/*/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 648
	09400	WRITE OUTPUT TAPE 81, 350, NRUN,N,M	TEST 649
C		/*/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 650
	09402	WRITE OUTPUT TAPE 81, 360, NTAPE	TEST 651
C		/*/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 652
	09403	WRITE OUTPUT TAPE 81, 370, NPRINT	TEST 653
C		/*/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 654
	09405	WRITE OUTPUT TAPE 81, 380	TEST 655
		DO 9407 KK=1,10	TEST 656
C		/*/*/* FOLLOWING INPUT-OUTPUT STATEMENT IS FOR ON-LINE PRINT	TEST 657
	09407	WRITE OUTPUT TAPE 81, 390	TEST 658
C		PAUSE PAUSE PAUSE PAUSE PAUSE PAUSE PAUSE PAUSE PAUSE PAUSE	TEST 659
		PAUSE	TEST 660
		CALL SYSTEM	TEST 661
		END	TEST 662
CFF		DATA DECK FOR GENERATOR - TERMINAL FLAG	TEST 663
		REMARK END OF RUN	TEST 664

C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM

JOB			1.
FOR			2.
DIMENSION T(80, 80)			3.
DIMENSION KEY(13)			4.
EQUIVALENCE (KEY(1),NFLAG)			5.
EQUIVALENCE (KEY(2),NTAPE)			6.
EQUIVALENCE (KEY(3),NPRINT)			7.
EQUIVALENCE (KEY(4),MAYBE)			8.
EQUIVALENCE (KEY(5),NRUN)			9.
EQUIVALENCE (KEY(6),MARKER)			10.
EQUIVALENCE (KEY(7),NOUSE)			11.
EQUIVALENCE (KEY(8),N)			12.
EQUIVALENCE (KEY(9),M)			13.
EQUIVALENCE (KEY(10),ITXP)			14.
EQUIVALENCE (KEY(11),ITCT)			15.
EQUIVALENCE (KEY(12),LEVEL)			16.
EQUIVALENCE (KEY(13),CLOSE)			17.
300 FORMAT(1H1I5,11I6,F8.0)			18.
310 FORMAT(8F15.9)			19.
320 FORMAT(21H1THIS RUN WRITES TAPEI3,23H FOR NEXT GENERATOR RUN)			20.
330 FORMAT(21H THIS RUN WRITES TAPEI3,22H FOR OFF-LINE PRINTING)			21.
340 FORMAT(52H PAUSE TO CHECK TAPE MOUNTING - THEN PUSH START)			22.
350 FORMAT(30H1SOLUTION IS COMPLETED FOR RUNI3, 5H, N =I3, 5H, M =I3)			23.
360 FORMAT(43H KEY CARD IS PUNCHED AND IS WRITTEN ON TAPEI3)			24.
370 FORMAT(40H OUTPUT FOR OFF-LINE PRINTING IS ON TAPEI3)			25.
380 FORMAT(55H PAUSE TO DISMOUNT TAPES, THEN PUSH START TO GET OFF)			26.
390 FORMAT(1H)			27.
400 FORMAT(1H I5,11I6,F8.0)			28.
NFLAG = 9999			29.
NTAPE = 15			30.
NPRINT= 16			31.
MAYBE = 1			32.
NRUN = 2			33.
MARKER= 1			34.
NOUSE = 1			35.
N = 8			36.
M = 8			37.
ITXP = 13			38.
ITCT = 1			39.
LEVEL = 7			40.
0450 WRITE OUTPUT TAPE 81, 320, NTAPE			41.
0451 WRITE OUTPUT TAPE 81, 330, NPRINT			42.
0452 WRITE OUTPUT TAPE 81, 340			43.
DO 455 KK=1,10			44.
0455 WRITE OUTPUT TAPE 81, 390			
PAUSE			

C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM

0458 PRINT	400,	(KEY(I),I=1,13)	45.	
0459 WRITE OUTPUT TAPE 81,	400,	(KEY(I),I=1,13)	46.	
NN=N+2			47.	
MM=M*M			48.	
DO 590 IM=1,M			49.	
DO 590 JM=1,M			50.	
DO 590 IO=1,NN			51.	
DO 590 JO=1,NN			52.	
IN=(IM-1)*NN+IO			53.	
JN=(JM-1)*NN+JO			54.	
T(IN,JN)=1.5			55.	
550 DO 590 KK=1,LEVEL			56.	
IF((IO-1)*(JO-1)*(NN-IO)*(NN-JO))	550,	580,	590	57.
580 T(IN,JN)=0.5+T(IN,JN)/2.0				58.
590 CLOSE=0				59.
ITCT=1				60.
DO 690 KK=1,ITXP				61.
690 ITCT=ITCT+ITCT				62.
ITCT=(ITCT/MM)/(N*N)				63.
IF(ITCT-1)	695,	696,	696	64.
695 ITCT=1				65.
696 CONTINUE				66.
TOLER=2.0*T(1,1)-1.0				67.
DO 9000 ITER=1,ITCT				68.
DO 9000 IM=1,M				69.
IN=(IM-1)*NN+1				70.
DO 9000 JM=1,M				71.
JO=JN				72.
JN=(JM-1)*NN+1				73.
IF(JM-2)	850,	860,	880	74.
850 IF((IM-1)+(ITER-1))	850,	860,	880	75.
860 IO=IN				76.
JO=1.0				77.
880 CONTINUE				78.

C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM

T(IN+ 1,JN+ 1)= T(IN+ 1,JN)+T(IO+ 1,JO+ 2)+T(IN ,JN+ 1)	79.
T(IN+ 1,JN+ 1)=(T(IO+ 2,JO+ 1)+T(IN+ 1,JN+ 1))/4.0	80.
T(IN+ 1,JN+ 2)= T(IN+ 1,JN+ 1)+T(IO+ 1,JO+ 3)+T(IN ,JN+ 2)	81.
T(IN+ 1,JN+ 2)=(T(IO+ 2,JO+ 2)+T(IN+ 1,JN+ 2))/4.0	82.
T(IN+ 1,JN+ 3)= T(IN+ 1,JN+ 2)+T(IO+ 1,JO+ 4)+T(IN ,JN+ 3)	83.
T(IN+ 1,JN+ 3)=(T(IO+ 2,JO+ 3)+T(IN+ 1,JN+ 3))/4.0	84.
T(IN+ 1,JN+ 4)= T(IN+ 1,JN+ 3)+T(IO+ 1,JO+ 5)+T(IN ,JN+ 4)	85.
T(IN+ 1,JN+ 4)=(T(IO+ 2,JO+ 4)+T(IN+ 1,JN+ 4))/4.0	86.
T(IN+ 1,JN+ 5)= T(IN+ 1,JN+ 4)+T(IO+ 1,JO+ 6)+T(IN ,JN+ 5)	87.
T(IN+ 1,JN+ 5)=(T(IO+ 2,JO+ 5)+T(IN+ 1,JN+ 5))/4.0	88.
T(IN+ 1,JN+ 6)= T(IN+ 1,JN+ 5)+T(IO+ 1,JO+ 7)+T(IN ,JN+ 6)	89.
T(IN+ 1,JN+ 6)=(T(IO+ 2,JO+ 6)+T(IN+ 1,JN+ 6))/4.0	90.
T(IN+ 1,JN+ 7)= T(IN+ 1,JN+ 6)+T(IO+ 1,JO+ 8)+T(IN ,JN+ 7)	91.
T(IN+ 1,JN+ 7)=(T(IO+ 2,JO+ 7)+T(IN+ 1,JN+ 7))/4.0	92.
T(IN+ 1,JN+ 8)= T(IN+ 1,JN+ 7)+T(IO+ 1,JO+ 9)+T(IN ,JN+ 8)	93.
T(IN+ 1,JN+ 8)=(T(IO+ 2,JO+ 8)+T(IN+ 1,JN+ 8))/4.0	94.
KK=JN-1	95.
KW=KK+NN	96.
1011 KA=KK+1	97.
DO 1014 KK=KA,KW	98.
IF(KK-KW)	1012, 1019, 1011 99.
1012 IF(KK-JN)	1012, 1016, 1013 100.
1013 IF((T(IN+ 1,KK)-TOLER))	1015, 1015, 1014 101.
1014 CONTINUE	102.
1015 CLOSE=CLOSE+1.0	103.
1016 KA=KK+1	104.
DO 1018 KK=KA,KW	105.
IF(KW-KK)	1016, 1019, 1017 106.
1017 IF(-(T(IN+ 1,KK)-TOLER))	1011, 1018, 1018 107.
1018 CLOSE=CLOSE+1.0	108.
1019 CONTINUE	109.

C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM

T(IN+ 2,JN+ 1)= T(IN+ 2,JN)+T(IO+ 2,JO+ 2)+T(IN+ 1,JN+ 1)	110.
T(IN+ 2,JN+ 1)=(T(IO+ 3,JO+ 1)+T(IN+ 2,JN+ 1))/4.0	111.
T(IN+ 2,JN+ 2)= T(IN+ 2,JN+ 1)+T(IO+ 2,JO+ 3)+T(IN+ 1,JN+ 2)	112.
T(IN+ 2,JN+ 2)=(T(IO+ 3,JO+ 2)+T(IN+ 2,JN+ 2))/4.0	113.
T(IN+ 2,JN+ 3)= T(IN+ 2,JN+ 2)+T(IO+ 2,JO+ 4)+T(IN+ 1,JN+ 3)	114.
T(IN+ 2,JN+ 3)=(T(IO+ 3,JO+ 3)+T(IN+ 2,JN+ 3))/4.0	115.
T(IN+ 2,JN+ 4)= T(IN+ 2,JN+ 3)+T(IO+ 2,JO+ 5)+T(IN+ 1,JN+ 4)	116.
T(IN+ 2,JN+ 4)=(T(IO+ 3,JO+ 4)+T(IN+ 2,JN+ 4))/4.0	117.
T(IN+ 2,JN+ 5)= T(IN+ 2,JN+ 4)+T(IO+ 2,JO+ 6)+T(IN+ 1,JN+ 5)	118.
T(IN+ 2,JN+ 5)=(T(IO+ 3,JO+ 5)+T(IN+ 2,JN+ 5))/4.0	119.
T(IN+ 2,JN+ 6)= T(IN+ 2,JN+ 5)+T(IO+ 2,JO+ 7)+T(IN+ 1,JN+ 6)	120.
T(IN+ 2,JN+ 6)=(T(IO+ 3,JO+ 6)+T(IN+ 2,JN+ 6))/4.0	121.
T(IN+ 2,JN+ 7)= T(IN+ 2,JN+ 6)+T(IO+ 2,JO+ 8)+T(IN+ 1,JN+ 7)	122.
T(IN+ 2,JN+ 7)=(T(IO+ 3,JO+ 7)+T(IN+ 2,JN+ 7))/4.0	123.
T(IN+ 2,JN+ 8)= T(IN+ 2,JN+ 7)+T(IO+ 2,JO+ 9)+T(IN+ 1,JN+ 8)	124.
T(IN+ 2,JN+ 8)=(T(IO+ 3,JO+ 8)+T(IN+ 2,JN+ 8))/4.0	125.
KK=JN-1	126.
KW=KK+NN	127.
1021 KA=KK+1	128.
DO 1024 KK=KA,KW	129.
IF(KK-KW)	1022, 1029, 1021 130.
1022 IF(KK-JN)	1022, 1026, 1023 131.
1023 IF((T(IN+ 2,KK)-TOLER))	1025, 1025, 1024 132.
1024 CONTINUE	133.
1025 CLOSE=CLOSE+1.0	134.
1026 KA=KK+1	135.
DO 1028 KK=KA,KW	136.
IF(KW-KK)	1026, 1029, 1027 137.
1027 IF(-(T(IN+ 2,KK)-TOLER))	1021, 1028, 1028 138.
1028 CLOSE=CLOSE+1.0	139.
1029 CONTINUE	140.

C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM

T(IN+ 3,JN+ 1)= T(IN+ 3,JN)+T(IO+ 3,JO+ 2)+T(IN+ 2,JN+ 1)	141.
T(IN+ 3,JN+ 1)=(T(IO+ 4,JO+ 1)+T(IN+ 3,JN+ 1))/4.0	142.
T(IN+ 3,JN+ 2)= T(IN+ 3,JN+ 1)+T(IO+ 3,JO+ 3)+T(IN+ 2,JN+ 2)	143.
T(IN+ 3,JN+ 2)=(T(IO+ 4,JO+ 2)+T(IN+ 3,JN+ 2))/4.0	144.
T(IN+ 3,JN+ 3)= T(IN+ 3,JN+ 2)+T(IO+ 3,JO+ 4)+T(IN+ 2,JN+ 3)	145.
T(IN+ 3,JN+ 3)=(T(IO+ 4,JO+ 3)+T(IN+ 3,JN+ 3))/4.0	146.
T(IN+ 3,JN+ 4)= T(IN+ 3,JN+ 3)+T(IO+ 3,JO+ 5)+T(IN+ 2,JN+ 4)	147.
T(IN+ 3,JN+ 4)=(T(IO+ 4,JO+ 4)+T(IN+ 3,JN+ 4))/4.0	148.
T(IN+ 3,JN+ 5)= T(IN+ 3,JN+ 4)+T(IO+ 3,JO+ 6)+T(IN+ 2,JN+ 5)	149.
T(IN+ 3,JN+ 5)=(T(IO+ 4,JO+ 5)+T(IN+ 3,JN+ 5))/4.0	150.
T(IN+ 3,JN+ 6)= T(IN+ 3,JN+ 5)+T(IO+ 3,JO+ 7)+T(IN+ 2,JN+ 6)	151.
T(IN+ 3,JN+ 6)=(T(IO+ 4,JO+ 6)+T(IN+ 3,JN+ 6))/4.0	152.
T(IN+ 3,JN+ 7)= T(IN+ 3,JN+ 6)+T(IO+ 3,JO+ 8)+T(IN+ 2,JN+ 7)	153.
T(IN+ 3,JN+ 7)=(T(IO+ 4,JO+ 7)+T(IN+ 3,JN+ 7))/4.0	154.
T(IN+ 3,JN+ 8)= T(IN+ 3,JN+ 7)+T(IO+ 3,JO+ 9)+T(IN+ 2,JN+ 8)	155.
T(IN+ 3,JN+ 8)=(T(IO+ 4,JO+ 8)+T(IN+ 3,JN+ 8))/4.0	156.
KK=JN-1	157.
KW=KK+NN	158.
1031 KA=KK+1	159.
DO 1034 KK=KA,KW	160.
IF(KK-KW)	1032, 1039, 1031 161.
1032 IF(KK-JN)	1032, 1036, 1033 162.
1033 IF((T(IN+ 3,KK)-TOLER))	1035, 1035, 1034 163.
1034 CONTINUE	164.
1035 CLOSE=CLOSE+1.0	165.
1036 KA=KK+1	166.
DO 1038 KK=KA,KW	167.
IF(KW-KK)	1036, 1039, 1037 168.
1037 IF(-(T(IN+ 3,KK)-TOLER))	1031, 1038, 1038 169.
1038 CLOSE=CLOSE+1.0	170.
1039 CONTINUE	171.

C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM

T(IN+ 4,JN+ 1)= T(IN+ 4,JN	)	+T(IO+ 4,JO+ 2)+T(IN+ 3,JN+ 1)	172.
T(IN+ 4,JN+ 1)=(T(IO+ 5,JO+ 1)+T(IN+ 4,JN+ 1))	/4.0		173.
T(IN+ 4,JN+ 2)= T(IN+ 4,JN+ 1)+T(IO+ 4,JO+ 3)+T(IN+ 3,JN+ 2)			174.
T(IN+ 4,JN+ 2)=(T(IO+ 5,JO+ 2)+T(IN+ 4,JN+ 2))	/4.0		175.
T(IN+ 4,JN+ 3)= T(IN+ 4,JN+ 2)+T(IO+ 4,JO+ 4)+T(IN+ 3,JN+ 3)			176.
T(IN+ 4,JN+ 3)=(T(IO+ 5,JO+ 3)+T(IN+ 4,JN+ 3))	/4.0		177.
T(IN+ 4,JN+ 4)= T(IN+ 4,JN+ 3)+T(IO+ 4,JO+ 5)+T(IN+ 3,JN+ 4)			178.
T(IN+ 4,JN+ 4)=(T(IO+ 5,JO+ 4)+T(IN+ 4,JN+ 4))	/4.0		179.
T(IN+ 4,JN+ 5)= T(IN+ 4,JN+ 4)+T(IO+ 4,JO+ 6)+T(IN+ 3,JN+ 5)			180.
T(IN+ 4,JN+ 5)=(T(IO+ 5,JO+ 5)+T(IN+ 4,JN+ 5))	/4.0		181.
T(IN+ 4,JN+ 6)= T(IN+ 4,JN+ 5)+T(IO+ 4,JO+ 7)+T(IN+ 3,JN+ 6)			182.
T(IN+ 4,JN+ 6)=(T(IO+ 5,JO+ 6)+T(IN+ 4,JN+ 6))	/4.0		183.
T(IN+ 4,JN+ 7)= T(IN+ 4,JN+ 6)+T(IO+ 4,JO+ 8)+T(IN+ 3,JN+ 7)			184.
T(IN+ 4,JN+ 7)=(T(IO+ 5,JO+ 7)+T(IN+ 4,JN+ 7))	/4.0		185.
T(IN+ 4,JN+ 8)= T(IN+ 4,JN+ 7)+T(IO+ 4,JO+ 9)+T(IN+ 3,JN+ 8)			186.
T(IN+ 4,JN+ 8)=(T(IO+ 5,JO+ 8)+T(IN+ 4,JN+ 8))	/4.0		187.
KK=JN-1			188.
KW=KK+NN			189.
1041 KA=KK+1			190.
DO 1044 KK=KA,KW			191.
IF(KK-KW)		1042, 1049, 1041	192.
1042 IF(KK-JN)		1042, 1046, 1043	193.
1043 IF((T(IN+ 4,KK)-TOLER))		1045, 1045, 1044	194.
1044 CONTINUE			195.
1045 CLOSE=CLOSE+1.0			196.
1046 KA=KK+1			197.
DO 1048 KK=KA,KW			198.
IF(KW-KK)		1046, 1049, 1047	199.
1047 IF(-(T(IN+ 4,KK)-TOLER))		1041, 1048, 1048	200.
1048 CLOSE=CLOSE+1.0			201.
1049 CONTINUE			202.

C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM

T(IN+ 5,JN+ 1)= T(IN+ 5,JN)+T(IO+ 5,JO+ 2)+T(IN+ 4,JN+ 1)	203.
T(IN+ 5,JN+ 1)=(T(IO+ 6,JO+ 1)+T(IN+ 5,JN+ 1))/4.0	204.
T(IN+ 5,JN+ 2)= T(IN+ 5,JN+ 1)+T(IO+ 5,JO+ 3)+T(IN+ 4,JN+ 2)	205.
T(IN+ 5,JN+ 2)=(T(IO+ 6,JO+ 2)+T(IN+ 5,JN+ 2))/4.0	206.
T(IN+ 5,JN+ 3)= T(IN+ 5,JN+ 2)+T(IO+ 5,JO+ 4)+T(IN+ 4,JN+ 3)	207.
T(IN+ 5,JN+ 3)=(T(IO+ 6,JO+ 3)+T(IN+ 5,JN+ 3))/4.0	208.
T(IN+ 5,JN+ 4)= T(IN+ 5,JN+ 3)+T(IO+ 5,JO+ 5)+T(IN+ 4,JN+ 4)	209.
T(IN+ 5,JN+ 4)=(T(IO+ 6,JO+ 4)+T(IN+ 5,JN+ 4))/4.0	210.
T(IN+ 5,JN+ 5)= T(IN+ 5,JN+ 4)+T(IO+ 5,JO+ 6)+T(IN+ 4,JN+ 5)	211.
T(IN+ 5,JN+ 5)=(T(IO+ 6,JO+ 5)+T(IN+ 5,JN+ 5))/4.0	212.
T(IN+ 5,JN+ 6)= T(IN+ 5,JN+ 5)+T(IO+ 5,JO+ 7)+T(IN+ 4,JN+ 6)	213.
T(IN+ 5,JN+ 6)=(T(IO+ 6,JO+ 6)+T(IN+ 5,JN+ 6))/4.0	214.
T(IN+ 5,JN+ 7)= T(IN+ 5,JN+ 6)+T(IO+ 5,JO+ 8)+T(IN+ 4,JN+ 7)	215.
T(IN+ 5,JN+ 7)=(T(IO+ 6,JO+ 7)+T(IN+ 5,JN+ 7))/4.0	216.
T(IN+ 5,JN+ 8)= T(IN+ 5,JN+ 7)+T(IO+ 5,JO+ 9)+T(IN+ 4,JN+ 8)	217.
T(IN+ 5,JN+ 8)=(T(IO+ 6,JO+ 8)+T(IN+ 5,JN+ 8))/4.0	218.
KK=JN-1	219.
KW=KK+NN	220.
1051 KA=KK+1	221.
DO 1054 KK=KA,KW	222.
IF(KK-KW)	1052, 1059, 1051 223.
1052 IF(KK-JN)	1052, 1056, 1053 224.
1053 IF((T(IN+ 5,KK)-TOLER))	1055, 1055, 1054 225.
1054 CONTINUE	226.
1055 CLOSE=CLOSE+1.0	227.
1056 KA=KK+1	228.
DO 1058 KK=KA,KW	229.
IF(KW-KK)	1056, 1059, 1057 230.
1057 IF(-(T(IN+ 5,KK)-TOLER))	1051, 1058, 1058 231.
1058 CLOSE=CLOSE+1.0	232.
1059 CONTINUE	233.

C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM

T(IN+ 6,JN+ 1)= T(IN+ 6,JN)+T(IO+ 6,JO+ 2)+T(IN+ 5,JN+ 1)	234.
T(IN+ 6,JN+ 1)=(T(IO+ 7,JO+ 1)+T(IN+ 6,JN+ 1))/4.0	235.
T(IN+ 6,JN+ 2)= T(IN+ 6,JN+ 1)+T(IO+ 6,JO+ 3)+T(IN+ 5,JN+ 2)	236.
T(IN+ 6,JN+ 2)=(T(IO+ 7,JO+ 2)+T(IN+ 6,JN+ 2))/4.0	237.
T(IN+ 6,JN+ 3)= T(IN+ 6,JN+ 2)+T(IO+ 6,JO+ 4)+T(IN+ 5,JN+ 3)	238.
T(IN+ 6,JN+ 3)=(T(IO+ 7,JO+ 3)+T(IN+ 6,JN+ 3))/4.0	239.
T(IN+ 6,JN+ 4)= T(IN+ 6,JN+ 3)+T(IO+ 6,JO+ 5)+T(IN+ 5,JN+ 4)	240.
T(IN+ 6,JN+ 4)=(T(IO+ 7,JO+ 4)+T(IN+ 6,JN+ 4))/4.0	241.
T(IN+ 6,JN+ 5)= T(IN+ 6,JN+ 4)+T(IO+ 6,JO+ 6)+T(IN+ 5,JN+ 5)	242.
T(IN+ 6,JN+ 5)=(T(IO+ 7,JO+ 5)+T(IN+ 6,JN+ 5))/4.0	243.
T(IN+ 6,JN+ 6)= T(IN+ 6,JN+ 5)+T(IO+ 6,JO+ 7)+T(IN+ 5,JN+ 6)	244.
T(IN+ 6,JN+ 6)=(T(IO+ 7,JO+ 6)+T(IN+ 6,JN+ 6))/4.0	245.
T(IN+ 6,JN+ 7)= T(IN+ 6,JN+ 6)+T(IO+ 6,JO+ 8)+T(IN+ 5,JN+ 7)	246.
T(IN+ 6,JN+ 7)=(T(IO+ 7,JO+ 7)+T(IN+ 6,JN+ 7))/4.0	247.
T(IN+ 6,JN+ 8)= T(IN+ 6,JN+ 7)+T(IO+ 6,JO+ 9)+T(IN+ 5,JN+ 8)	248.
T(IN+ 6,JN+ 8)=(T(IO+ 7,JO+ 8)+T(IN+ 6,JN+ 8))/4.0	249.
KK=JN-1	250.
KW=KK+NN	251.
1061 KA=KK+1	252.
DO 1064 KK=KA,KW	253.
IF(KK-KW)	1062, 1069, 1061 254.
1062 IF(KK-JN)	1062, 1066, 1063 255.
1063 IF((T(IN+ 6,KK)-TOLER))	1065, 1065, 1064 256.
1064 CONTINUE	257.
1065 CLOSE=CLOSE+1.0	258.
1066 KA=KK+1	259.
DO 1068 KK=KA,KW	260.
IF(KW-KK)	1066, 1069, 1067 261.
1067 IF(-(T(IN+ 6,KK)-TOLER))	1061, 1068, 1068 262.
1068 CLOSE=CLOSE+1.0	263.
1069 CONTINUE	264.

C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM

T(IN+ 7,JN+ 1)= T(IN+ 7,JN)+T(IO+ 7,JO+ 2)+T(IN+ 6,JN+ 1)	265.
T(IN+ 7,JN+ 1)=(T(IO+ 8,JO+ 1)+T(IN+ 7,JN+ 1))/4.0	266.
T(IN+ 7,JN+ 2)= T(IN+ 7,JN+ 1)+T(IO+ 7,JO+ 3)+T(IN+ 6,JN+ 2)	267.
T(IN+ 7,JN+ 2)=(T(IO+ 8,JO+ 2)+T(IN+ 7,JN+ 2))/4.0	268.
T(IN+ 7,JN+ 3)= T(IN+ 7,JN+ 2)+T(IO+ 7,JO+ 4)+T(IN+ 6,JN+ 3)	269.
T(IN+ 7,JN+ 3)=(T(IO+ 8,JO+ 3)+T(IN+ 7,JN+ 3))/4.0	270.
T(IN+ 7,JN+ 4)= T(IN+ 7,JN+ 3)+T(IO+ 7,JO+ 5)+T(IN+ 6,JN+ 4)	271.
T(IN+ 7,JN+ 4)=(T(IO+ 8,JO+ 4)+T(IN+ 7,JN+ 4))/4.0	272.
T(IN+ 7,JN+ 5)= T(IN+ 7,JN+ 4)+T(IO+ 7,JO+ 6)+T(IN+ 6,JN+ 5)	273.
T(IN+ 7,JN+ 5)=(T(IO+ 8,JO+ 5)+T(IN+ 7,JN+ 5))/4.0	274.
T(IN+ 7,JN+ 6)= T(IN+ 7,JN+ 5)+T(IO+ 7,JO+ 7)+T(IN+ 6,JN+ 6)	275.
T(IN+ 7,JN+ 6)=(T(IO+ 8,JO+ 6)+T(IN+ 7,JN+ 6))/4.0	276.
T(IN+ 7,JN+ 7)= T(IN+ 7,JN+ 6)+T(IO+ 7,JO+ 8)+T(IN+ 6,JN+ 7)	277.
T(IN+ 7,JN+ 7)=(T(IO+ 8,JO+ 7)+T(IN+ 7,JN+ 7))/4.0	278.
T(IN+ 7,JN+ 8)= T(IN+ 7,JN+ 7)+T(IO+ 7,JO+ 9)+T(IN+ 6,JN+ 8)	279.
T(IN+ 7,JN+ 8)=(T(IO+ 8,JO+ 8)+T(IN+ 7,JN+ 8))/4.0	280.
KK=JN-1	281.
KW=KK+NN	282.
1071 KA=KK+1	283.
DO 1074 KK=KA,KW	284.
IF(KK-KW)	1072, 1079, 1071 285.
1072 IF(KK-JN)	1072, 1076, 1073 286.
1073 IF((T(IN+ 7,KK)-TOLER))	1075, 1075, 1074 287.
1074 CONTINUE	288.
1075 CLOSE=CLOSE+1.0	289.
1076 KA=KK+1	290.
DO 1078 KK=KA,KW	291.
IF(KW-KK)	1076, 1079, 1077 292.
1077 IF(-(T(IN+ 7,KK)-TOLER))	1071, 1078, 1078 293.
1078 CLOSE=CLOSE+1.0	294.
1079 CONTINUE	295.

C /*/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM

T(IN+ 8,JN+ 1)= T(IN+ 8,JN	)	+T(IO+ 8,JO+ 2)+T(IN+ 7,JN+ 1)	296.
T(IN+ 8,JN+ 1)=(T(IO+ 9,JO+ 1)+T(IN+ 8,JN+ 1))/4.0			297.
T(IN+ 8,JN+ 2)= T(IN+ 8,JN+ 1)+T(IO+ 8,JO+ 3)+T(IN+ 7,JN+ 2)			298.
T(IN+ 8,JN+ 2)=(T(IO+ 9,JO+ 2)+T(IN+ 8,JN+ 2))/4.0			299.
T(IN+ 8,JN+ 3)= T(IN+ 8,JN+ 2)+T(IO+ 8,JO+ 4)+T(IN+ 7,JN+ 3)			300.
T(IN+ 8,JN+ 3)=(T(IO+ 9,JO+ 3)+T(IN+ 8,JN+ 3))/4.0			301.
T(IN+ 8,JN+ 4)= T(IN+ 8,JN+ 3)+T(IO+ 8,JO+ 5)+T(IN+ 7,JN+ 4)			302.
T(IN+ 8,JN+ 4)=(T(IO+ 9,JO+ 4)+T(IN+ 8,JN+ 4))/4.0			303.
T(IN+ 8,JN+ 5)= T(IN+ 8,JN+ 4)+T(IO+ 8,JO+ 6)+T(IN+ 7,JN+ 5)			304.
T(IN+ 8,JN+ 5)=(T(IO+ 9,JO+ 5)+T(IN+ 8,JN+ 5))/4.0			305.
T(IN+ 8,JN+ 6)= T(IN+ 8,JN+ 5)+T(IO+ 8,JO+ 7)+T(IN+ 7,JN+ 6)			306.
T(IN+ 8,JN+ 6)=(T(IO+ 9,JO+ 6)+T(IN+ 8,JN+ 6))/4.0			307.
T(IN+ 8,JN+ 7)= T(IN+ 8,JN+ 6)+T(IO+ 8,JO+ 8)+T(IN+ 7,JN+ 7)			308.
T(IN+ 8,JN+ 7)=(T(IO+ 9,JO+ 7)+T(IN+ 8,JN+ 7))/4.0			309.
T(IN+ 8,JN+ 8)= T(IN+ 8,JN+ 7)+T(IO+ 8,JO+ 9)+T(IN+ 7,JN+ 8)			310.
T(IN+ 8,JN+ 8)=(T(IO+ 9,JO+ 8)+T(IN+ 8,JN+ 8))/4.0			311.
KK=JN-1			312.
KW=KK+NN			313.
1081 KA=KK+1			314.
DO 1084 KK=KA,KW			315.
IF(KK-KW)	1082, 1089, 1081		316.
1082 IF(KK-JN)	1082, 1086, 1083		317.
1083 IF((T(IN+ 8,KK)-TOLER))	1085, 1085, 1084		318.
1084 CONTINUE			319.
1085 CLOSE=CLOSE+1.0			320.
1086 KA=KK+1			321.
DO 1088 KK=KA,KW			322.
IF(KW-KK)	1086, 1089, 1087		323.
1087 IF(-(T(IN+ 8,KK)-TOLER))	1081, 1088, 1088		324.
1088 CLOSE=CLOSE+1.0			325.
1089 CONTINUE			326.

C /**/ SIGNAL COMMENT CARD TO BEGIN NEW SHEET UNDER AUTO-LISTING PROGRAM

9000	CONTINUE			327.
9001	REWIND	15		328.
9002	REWIND	16		329.
9003	WRITE TAPE	15,	(KEY(KK),KK=1,13)	330.
9004	WRITE OUTPUT TAPE	16, 300,	(KEY(KK),KK=1,13)	331.
	KW=NN*M			332.
	DO 9100 KK=1,KW			333.
9100	T(1,KK)=T(KK,KK)			334.
	DO 9200 KA=1,NN			335.
	KW=NN*(M-1)+KA			336.
9200	WRITE OUTPUT TAPE	16, 310,	(T(1,KK),KK=KA,KW,NN)	337.
9300	END FILE	15		338.
9301	END FILE	16		339.
9309	WRITE OUTPUT TAPE 14,	300,	(KEY(KK),KK=1,13)	340.
9310	END FILE	14		341.
9400	WRITE OUTPUT TAPE 81,	350,	NRUN,N,M	342.
9402	WRITE OUTPUT TAPE 81,	360,	NTAPE	343.
9403	WRITE OUTPUT TAPE 81,	370,	NPRINT	344.
9405	WRITE OUTPUT TAPE 81,	380		345.
	DO 9407 KK=1,10			346.
9407	WRITE OUTPUT TAPE 81,	390		347.
	PAUSE			348.
	CALL SYSTEM			349.
	END			350.
	LOAD BATCH			
	TRA			
	REMARK	END OF RUN		