Historic, Archive Document

Do not assume content reflects current scientific knowledge, policies, or practices.

A99.9
F7632 U
Cp. 3

EXT.

PREDICTING SOFTWOOD CUTTING YIELDS BY COMPUTER

by Bernard J. Erickson and
Donald C. Markstrom

U. S. DEPT. OF AGRICULTURE
NATIONAL AGRICULTURAL LIBRARY
RECEIVED

APR 10 1973

PROCUREMENT SECTION
CURRENT SERIAL RECORDS

USDA Forest Service
Research Paper RM-98

December 1972

Rocky Mountain Forest and
Range Experiment Station

Forest Service

U.S. Department of Agriculture

Fort Collins, Colorado

Preface and Acknowledgments

The computer program presented here was developed in a peripheral study in support of an analysis of the feasibility of roll laminating fiber overlays on low-grade ponderosa pine lumber. A part of that study involved determining economically and technically feasible methods for eliminating open defects in lumber to be used as overlay substrates. The overall study, done in cooperation with the Duke City Lumber Company, Albuquerque, New Mexico, and financed by the Economic Development Administration, U.S. Department of Commerce, is reported in USDA Forest Service Research Paper RM-97.

Several people, in addition to the authors, contributed to the development of the program. Arthur Bourke and John McElfresh, consultants to the Rocky Mountain Station on the roll laminating study, participated both in formulating the problem and writing the initial program. Hiram Hallock, wood technologist with the USDA Forest Products Laboratory, helped formulate the problem. Jacob Kovner, biometrician, and Lincoln Mueller and Roland Barger, wood technologists, Rocky Mountain Station, also provided guidance during the course of the study. Barger is now at the Intermountain Forest and Range Experiment Station, and Mueller is retired.

Abstract

A computer program, written in FORTRAN predicts the maximum yield of cuttings for softwood cut-up and edge- and end-gluing operation. The program calculates cutting recovery (given cutting width and length constraints and defect locations on the board), and locates ripping saw kerfs.

Keywords: Programming (computers), softwood cut-up, end and edge gluing, dimension stock.

Predicting Softwood Cutting Yield by Computer

by

Bernard J. Erickson, Computer Programmer,

and

Donald C. Markstrom, Associate Wood Technologist

Rocky Mountain Forest and Range Experiment Station¹

¹*Central headquarters maintained at Fort Collins, in cooperation with Colorado State University.*

Contents

	Page
Program CUTUP in Brief	1
Characterizing Boards and Defects	1
Using the Program	2
Program CUTUP	2
Program PERMU	3
Program CONST	3
Program RECVRY	4
Appendix 1: Sample Output from Programs	5
PERMU Output	5
CONST Output	6
RECVRY Output	8
Appendix 2: Source Listings for Programs	10
Program PERMU	10
Program CONST	11
Program RECVRY	13
Program DEFEDIT	15
Program CTDFCT	15

Predicting Softwood Cutting Yield by Computer

Bernard J. Erickson and Donald C. Markstrom

Program CUTUP in Brief

Currently, major emphasis in improving wood conversion efficiency is concentrated in three areas: (1) improving yields from the resource, (2) reducing waste and residues, and (3) reducing processing errors resulting from repetitive human decisions. The program described here can contribute to greater efficiency in all of these areas. Automated defect-sensing methods now being developed offer the possibility of feeding continuous defect data to a computer as input to the type of optimizing program described, thereby automating the whole cut-up decision process.

A computer program CUTUP to predict the maximum yield of cuttings obtainable from softwood lumber is described in this paper. The program calculates cutting recovery (given cutting width and length constraints and defect location on the board), and will predict the maximum yield of cuttings of a specified grade, for a softwood cut-up and edge- and end-gluing operation.

The conventional practice in softwood cut-up operations is to first rip the board into prescribed cutting widths in a manner to maximize recovery, and then crosscut the full-board-length cuttings to remove defects. The CUTUP computer program simulates this practice systematically, (1) locating full-board-length cuttings of specified widths and sequence to maximize recovery, (2) crosscutting to remove defects, and (3) measuring the areas of all cuttings of a specified length or longer. The recovered area excludes (1) the area of defects, (2) areas between defects, or between a defect and the end of the board, where the recoverable length is less than the specified minimum, and (3) areas of all saw kerfs. The maximum number of rips is also specified in the program.

This program differs fundamentally from computer programs developed to predict yield of cuttings from hardwood cut-stock operations, where the usual sawing practice is to alternately crosscut and rip.^{2 3} Hardwood programs systematically scan the board to locate and measure

cuttings of prescribed width and length between defect areas, with the largest specified cutting considered first. The cuttings can then be removed from the board by starting with either a crosscut or rip, and alternating back and forth, taking as many cuts as possible during each operation. Softwood and hardwood programs are similar, however, in that the same format is used to record board and defect data on the input cards.

Characterizing Boards and Defects

Each board is considered to have X and Y axes, with its lower left corner at (0,0), its length along the X axis, and its width along the Y axis. The board size is described and defects are located by 1/4-inch coordinates, which designate the lower left and upper right corners of the areas (fig. 1). All measurements are expressed in 1/4-inch units, which can be used directly in calculating recovery of cut-up stock.

Data cards describing the boards and defects are punched in the following manner: The first card lists the board grade code in columns 6 and 7, the board number in columns 32 through 36, and total number of defects in columns 69 through 72. The second card gives the coordinates of the board, and each of the remaining cards gives the coordinates of a single defect. The lower left Y coordinates of the board and the defects are punched in columns 1 through 3 and the X coordinates in columns 5 through 7. The upper right Y coordinates are punched in columns 13 through 15 and the X coordinates in columns 17 through 19. The sequence of the defect cards in the deck must be the same as that of the defects, from left to right across the X axis of the board.

The total number of cards for each board equals the number of defects plus two.

The program was run with a sample of 600 boards selected from surfaced yard lumber at the Duke City Lumber Company. One hundred boards were randomly selected in each of six categories: grades 3 Common and 4 Common, in nominal widths of 4, 8, and 12 inches.

The boards were examined and defects outlined. As a convenience, the boards were then photographed with color transparency film. The slides were projected at one-half scale on rear-projection translucent glass to facilitate measuring board and defect data.

²Englerth, G. H., and D.E. Dunmire. *Programing for lumber yield. For. Prod. J.* 16(9): 67-69. 1966.

³Wodzinski, Claudia, and Eldona Hahm. *A computer program to determine yields of lumber. U.S. Dep. Agric., For. Prod. Lab., 33 p. Madison, Wis.* 1966.

Figure 1.--Diagrammatic sketch of a grade 4 Common 1- by 12-inch board, with associated program output data, illustrates the program developed for selective cut-up evaluation. Shaded areas represent defects that have been located by X-Y coordinates and blocked out. Lined areas represent additional material lost in cutting out defects.

Using the Program

Boards of different species, sizes, and grades could be sampled and the program used to evaluate the following:

1. Area recovered using alternative rip combinations.
2. Efficiency (in percent recovery) obtained from alternative widths of boards, for specified rip widths.
3. Effects of grade on rip recovery, for given species and specified rip widths.
4. Effects of adding or deleting rip widths from the array now used.
5. Alternative costs of input stock of various species, grades, and widths, based on predicted potential.

The program can undoubtedly be used to evaluate many other situations.

Program CUTUP

Program CUTUP is a set of three main programs — PERMU, CONST, and RECVRY. Program PERMU prepares a table of all possible cutting combinations for the N cutting widths (with M saw blades) to be considered for a given board-width class. User-selected cutting combinations are input to program RECVRY, which fits the cuts to individual boards to obtain maximum recovery area. Program CONST has one subroutine, XOUT, which outlines or

“x’s out” the defective areas of a board by relating defect, X, Y coordinates to computer core storage locations. The program prepares an input file of nonrecoverable board-defect areas, by specified cutting widths, for use in the RECVRY program. For boards 7.5 inches and wider, CONST requires considerable computer time. If narrower widths are to be processed, programs CONST and RECVRY may be run together as one program, with the output from CONST used directly as input to RECVRY. This eliminates the necessity of saving CONST output (Tape) for further processing by program RECVRY.

The board-defect data cards which are input to program CONST should be edited and error checked before use. If a large amount of data is to be processed, the data should be tape filed for ease of handling. A sample edit and card-to-tape program is included with the program listings in appendix 2.

The basic content and purpose of each program and subroutine is described in the sections that follow. The order and format of input control cards for each program are also specified. Variable names are defined in the source listings for each program. A sample output of an application of program CUTUP is shown in appendix 1, which also further explains the overall program operation.

Program PERMU, CONST, and RECVRY were originally written in the BASIC programming language for use on a time-sharing computer

system. The programs were modified, expanded, and rewritten in FORTRAN extended for use on a Control Data Corp. 6400 computer,⁴ and have been tested on the CDC 6400 computer system at the Colorado State University Computing Center.

Program PERMU

This program produces a table of possible cutting combinations for a specified board width, cutting or ripping widths, and cutting positions (kerfs). The computational procedure is based upon the principle that if one event A can take place in m ways, and a second event B can take place in n independent ways, then the number of ways in which both events can take place is mn. This principle can be extended to any number of independent events. The procedure was applied to the problem of cutting boards into different widths where each saw kerf is an event which can happen in m positions of the saw blade over a board. By expanding this technique, all combinations previously derived for narrower cutting widths and fewer cutting blades are saved for possible application in dealing with wider cutting widths and more cutting blades.

The procedure was used to process data involving three cutting widths of four, eight, and twelve 1/4-inch units (Q.I.U.), and five kerfs or cutting positions. Sample board widths ranged from 14 to 46 Q.I.U. To accommodate additional cutting widths and/or cutting positions, test parameters and array dimensions would need to be changed, as specified by the analysis problem.

Program Control Card Type 1. FORMAT (315)

Column

1-5	IBWDT	Overall board width in 1/4-inch units (Q.I.U.)
6-10	NSIZE	Number of different cutting sizes or rip widths to consider, MAX = 5.
11-15	NSWBL	Number of machine saw blades available, MAX = 6.

Program Control Card Type 2. FORMAT (515)

Column

1- 5	KUT(1)	Size of cutting width 1, in Q.I.U.
6-10	KUT(2)	Size of cutting width 2, in Q.I.U.

⁴The use of trade and company names is for the benefit of the reader; such use does not constitute an official endorsement or approval of any service or product by the U. S. Department of Agriculture to the exclusion of others that may be suitable.

11-15	KUT(3)	Size of cutting width 3, in Q.I.U.
16-20	KUT(4)	Size of cutting width 4, in Q.I.U.
21-25	KUT(5)	Size of cutting width 5, in Q.I.U.

Program CONST

This program constructs, in computer core storage, a board of variable width and length dimensions, and positions the location of defects in the board according to their coordinates. Board and defect coordinates are expressed in Q.I.U. to the X, Y coordinate system.

The amount of core storage available in the computer determines the size of boards which can be analyzed by this program. The array LWB(I,J), Length and Width of Board, is dimensioned according to the length (J) and width (I) of the largest boards to be analyzed in Q.I.U. If the largest boards to be processed are 11.5 inches by 16 feet, the array LWB(I,J) would be dimensioned I = 46, J = 768, resulting in 35,328 computer core locations to store all possible locations of defects.

Input data are contained in the set of cards which specify the board and defect coordinate values for one board (described in Characterizing Boards and Defects). The defects are positioned in LWB according to increasing defect coordinates in the J, or length, dimension of LWB(I,J).

If a large amount of data is to be processed, and is arranged or sorted by some criteria such as increasing board numbers within a board-width class, an option is provided to analyze a selected sub-set from the complete file. This option is exercised by two control parameters labeled NBSTR and NBEND, where NBSTR is the number of the first board in the sub-set and NBEND is the number of the last board in the sub-set to be processed.

A user input parameter labeled LBTD is a test variable used for setting a minimum acceptable length between defects. If an area between two defects (or between a defect and the end or start of a board) is less than LBTD (in Q.I.U.), the area is blocked out and treated as a defect area in the final scan of the board to locate all defect areas.

For each input board, program CONST operates in the following manner:

1. The board ID or Header Card is read, setting the variables NGRD, (board grade class), NBRD (board number), and NDEF (number of defects in board).
2. A board coordinate card is read defining the X-dimension (Width) as the ordinate and the Y-dimension (Length) the abscissa of the board to be processed. This notation is

reversed from the normal X, Y notation because of computer core storage procedure.

3. A set of defect coordinate cards is read until an END-OF-BOARD data card is encountered (IDXS=999).

The coordinates for each defect are positioned in an array by a pointer according to their order of occurrence within the board. Using the minimum length between defects test (LBTD), the defect array is scanned to find areas between defects which are less than LBTD. Defect location variables NX, LDS, and LDE are determined and passed to subroutine XOUT, along with the board array LWB, by the use of a COMMON statement. Subroutine XOUT then blocks out or "x's-out" the defined area of the defect in the board. This process continues until all defect areas have been positioned in the LWB array.

Indexes are now determined for a scan of the board to accumulate total defect area for NSIZE desired cuttings over a range of possible starting cut positions determined by board width. The total defect area for the board is written to an output file (Disk or Tape) for input to program RECVRY. The next board ID or Header Card is read and the process is repeated until an end-of-file is encountered in the board input data.

Program Control Card. FORMAT (315)

Column

1- 5	LBTD	Minimum length between defects limitation (Q.I.U.)
6-10	NBSTR	Board number to start analysis
11-15	NBEND	Board number to end analysis

Board Defect Data Card Deck

NDEF + two cards for each board to be processed plus an END-OF-BOARD Card.

Program RECVRY

This program computes the nondefective areas in a board, and simulates a number of possible cuttings, limited by the width of the board and the combinations of cutting sizes desired, to determine the best combination to use for maximum recovery.

Input for the program is the board-defect area file (Tape or Disk) from program CONST, and a user-selected set of cutting combinations generated by program PERMU to be evaluated. The cutting combinations are input to the program in a DATA statement, array KPCT. The coding order is explained in the program listing.

The program reads a title card and an input parameter control card. The parameter LPRM

is the lower limit of combined cuttings to be evaluated from the combination sets in array KPCT. Only combined cuttings greater than LPRM are used to evaluate the best fit, thus eliminating cutting combinations which do not use the entire width of the board. The variable NSIZE and the array KUT are set from values coded in DATA KSZE.

The board ID and board coordinates are read from the input file, along with board defect areas. Areas of nondefect are computed from defect area and area of the board, and saved. Indexes are set for selecting cutting combinations from array KPCT to obtain maximum recovery for the board. The number of saw kerfs which can be placed within the bounds of the board is determined. A saw kerf may be adjacent to the lower or upper edges of a board, or it may lie within the board itself. There is no provision in this program for straightening the edges and ends of the board to assure a straight board. A warped board may be utilized, however, by crosscutting into shorter lengths or ripping into narrow widths, by assigning appropriate defect coordinates. The sum of the cutting widths and saw kerfs cannot exceed the board width plus one unit.

A total recovery area is computed from the nondefective area data for a selected combination of cuttings. This value is stored as the "best" until a larger recovery area is obtained from a different combination of cuts, which then replaces it as the "best." After all acceptable combinations have been evaluated for best fit, the individual board recovery summary is printed out. Totals are accumulated for total recovery output by board grade and size class until an end-of-file is encountered in the input data file.

Program Control Card Type 1. FORMAT (4A10)

Column

1-40	TITLE	Short title to identify data being processed.
------	-------	---

Program Control Card Type 2. FORMAT (415)

Column

1- 5	LBTD	Minimum length between defects control: same value as in CONST.
6-10	LPRM	Lower limit of combined cuttings control: used to select combinations of cutting sizes.
11-15	NBSTR	Board number to start analysis for recoverable area.
16-20	NBEND	Board number to end analysis for recoverable area.

Board numbers (NBRD) to be included in the analysis should be $\geq$ NBSTR and $\leq$ NBEND.

Appendix 1: Sample Output From Programs

The sample output following is a set of computer printout pages obtained from running the PERMU and CONST/RECVRY programs. The board input data consisted of a set of 10 boards from the 14 Q.I.U. (3.5-inch) board width class. Five boards, numbered 291-295, were grade 3 Common with a length of 768 Q.I.U. (16 feet), and five boards, numbered 301-305, were grade 4 Common with a length of 576 Q.I.U. (12 feet). The minimum acceptable length between defects was set at 36 Q.I.U. (9 inches). Only those cutting combinations which resulted in a total cut (width) greater than 8 Q.I.U. and less than or equal to 14 Q.I.U. (total board width) were evaluated. Cutting combinations evaluated included (numbers in Q.I.U.), (4, 8), (8, 4), (12). The only other possible combination (4, 4, 4) was excluded to decrease the number of 1-inch-wide cuttings.

PERMU Output

The first line is a summary of the input values assigned to the program control parameters IBWDT, NSIZE, NSWBL, and the array KUT.

IBWDT = 46 Q.I.U.	Maximum width of sample boards, 11.5-inches
NSIZE = 3	Number of cutting sizes to consider
NSWBL = 4	Number of machine saw blades available
KUT(1) = 4 Q.I.U.	Consider a 1-inch cutting
KUT(2) = 8 Q.I.U.	Consider a 2-inch cutting
KUT(3) = 12 Q.I.U.	Consider a 3-inch cutting
KUT(4) = 0	Not used this run
KUT(5) = 0	Not used this run

The total number of possible combinations for three sizes and four blades is the computed value 120. Only those combinations which can be made using four positions within the limits of the board are printed out for user consideration.

Column 1 of the output table is a combination (COMBO) number used to reference the cutting combination. Columns 2-7 contain cutting combinations. The number 6 is used to fill out the table, and has no significance as far as combinations are concerned. The numbers 1, 2, and 3 in the table are referenced to the cutting sizes stored in array KUT so that 1 = 4, 2 = 8, and 3 = 12 Q.I.U. The number in column 8 is the total units cut for the given cutting combination and is used as a test limit in program RECVRY. The table output is ordered by increasing values in column 8.

	46	3	4	4	8	12	0	0
NBR OF DIFFERENT CUT COMBINATIONS =								120
1	1	6	6	6	6	6	6	4
2	2	6	6	6	6	6	6	8
4	1	1	6	6	6	6	6	8
3	3	6	6	6	6	6	6	12
5	2	1	6	6	6	6	6	12
7	1	2	6	6	6	6	6	12
13	1	1	1	6	6	6	6	12
6	3	1	6	6	6	6	6	16
8	2	2	6	6	6	6	6	16
10	1	3	6	6	6	6	6	16
14	2	1	1	6	6	6	6	16
16	1	2	1	6	6	6	6	16
22	1	1	2	6	6	6	6	16
40	1	1	1	1	6	6	6	16
9	3	2	6	6	6	6	6	20
11	2	3	6	6	6	6	6	20
15	3	1	1	6	6	6	6	20
17	2	2	1	6	6	6	6	20
19	1	3	1	6	6	6	6	20
23	2	1	2	6	6	6	6	20
25	1	2	2	6	6	6	6	20
31	1	1	3	6	6	6	6	20
41	2	1	1	1	6	6	6	20
43	1	2	1	1	6	6	6	20
49	1	1	2	1	6	6	6	20
67	1	1	1	2	6	6	6	20
12	3	3	6	6	6	6	6	24
18	3	2	1	6	6	6	6	24
20	2	3	1	6	6	6	6	24
24	3	1	2	6	6	6	6	24
26	2	2	2	6	6	6	6	24
28	1	3	2	6	6	6	6	24
32	2	1	3	6	6	6	6	24
34	1	2	3	6	6	6	6	24
42	3	1	1	1	6	6	6	24
44	2	2	1	1	6	6	6	24
46	1	3	1	1	6	6	6	24
50	2	1	2	1	6	6	6	24
52	1	2	2	1	6	6	6	24
58	1	1	3	1	6	6	6	24
68	2	1	1	2	6	6	6	24
70	1	2	1	2	6	6	6	24
76	1	1	2	2	6	6	6	24
94	1	1	1	3	6	6	6	24
21	3	3	1	6	6	6	6	28
27	3	2	2	6	6	6	6	28
29	2	3	2	6	6	6	6	28
33	3	1	3	6	6	6	6	28
35	2	2	3	6	6	6	6	28
37	1	3	3	6	6	6	6	28
45	3	2	1	1	6	6	6	28
47	2	3	1	1	6	6	6	28
51	3	1	2	1	6	6	6	28
53	2	2	2	1	6	6	6	28

55	1	3	2	1	6	6	28
59	2	1	3	1	6	6	28
61	1	2	3	1	6	6	28
69	3	1	1	2	6	6	28
71	2	2	1	2	6	6	28
73	1	3	1	2	6	6	28
77	2	1	2	2	6	6	28
79	1	2	2	2	6	6	28
85	1	1	3	2	6	6	28
95	2	1	1	3	6	6	28
97	1	2	1	3	6	6	28
103	1	1	2	3	6	6	28
30	3	3	2	6	6	6	32
36	3	2	3	6	6	6	32
38	2	3	3	6	6	6	32

48	3	3	1	1	6	6	32
54	3	2	2	1	6	6	32
56	2	3	2	1	6	6	32
60	3	1	3	1	6	6	32
62	2	2	3	1	6	6	32
64	1	3	3	1	6	6	32
72	3	2	1	2	6	6	32
74	2	3	1	2	6	6	32
78	3	1	2	2	6	6	32
80	2	2	2	2	6	6	32
82	1	3	2	2	6	6	32
86	2	1	3	2	6	6	32
88	1	2	3	2	6	6	32
96	3	1	1	3	6	6	32
98	2	2	1	3	6	6	32
100	1	3	1	3	6	6	32
104	2	1	2	3	6	6	32
106	1	2	2	3	6	6	32
112	1	1	3	3	6	6	32
39	3	3	3	6	6	6	36
57	3	3	2	1	6	6	36
63	3	2	3	1	6	6	36
65	2	3	3	1	6	6	36

75	3	3	1	2	6	6	36
81	3	2	2	2	6	6	36
83	2	3	2	2	6	6	36
87	3	1	3	2	6	6	36
89	2	2	3	2	6	6	36
91	1	3	3	2	6	6	36
99	3	2	1	3	6	6	36
101	2	3	1	3	6	6	36
105	3	1	2	3	6	6	36
107	2	2	2	3	6	6	36
109	1	3	2	3	6	6	36
113	2	1	3	3	6	6	36
115	1	2	3	3	6	6	36
66	3	3	3	1	6	6	40
84	3	3	2	2	6	6	40
90	3	2	3	2	6	6	40
92	2	3	3	2	6	6	40
102	3	3	1	3	6	6	40
108	3	2	2	3	6	6	40
110	2	3	2	3	6	6	40
114	3	1	3	3	6	6	40
116	2	2	3	3	6	6	40
118	1	3	3	3	6	6	40

CONST Output

The first line is a statement of the input values assigned to the program control parameters LBTB = 36, NBSTR = 291, and NBEND = 305. The next 10 lines are the input data for the first board, followed by the computed defect area for three cutting sizes for that board. Complete output for all the test boards is printed out for the sample run. When a large number of boards is to be evaluated, complete output, for only the first board in the input set is printed to reduce the number of output pages required. The remaining boards in the input set are indicated only by their ID, which contains the grade class, board number, number of defects in the board, and the overall coordinates of the board. The last line is a statement indicating the run was successfully completed, and gives the total number of boards processed.

MIN. LENGTH BETWEEN DEFECTS CONTROL FOR
THIS RUN IS 36 291 - 305

3	291	8	
0	0	14	768
11	366	14	374
0	411	7	421
0	444	1	473
4	451	12	463
0	537	2	547
0	574	3	582
8	667	13	671
3	697	13	710
1	480	960	1896
2	440	1120	1776
3	292	824	1332
4	260	760	0
5	260	824	0
6	380	824	0
7	380	824	0
8	220	0	0
9	252	0	0
10	252	0	0
11	252	0	0
12	0	0	0
13	0	0	0
14	0	0	0
3	292	5	
0	0	14	768
0	208	9	232
4	299	13	316
0	514	4	525
1	533	10	550
5	637	10	642
1	240	656	984
2	308	656	984
3	328	656	984
4	328	656	0
5	252	504	0

6	252	504	0
7	252	504	0
8	252	0	0
9	252	0	0
10	156	0	0
11	68	0	0
12	0	0	0
13	0	0	0
14	0	0	0

3	293	7	
0	0	14	768
0	31	3	37
8	113	14	123
2	508	7	514
1	525	5	530
0	543	14	544
9	609	14	618
10	711	14	716

1	292	584	1164
2	292	664	1164
3	292	736	1164
4	144	480	0
5	144	480	0
6	184	480	0
7	220	480	0
8	100	0	0
9	100	0	0
10	100	0	0
11	100	0	0
12	0	0	0
13	0	0	0
14	0	0	0

3	294	5	
0	0	14	768
10	245	14	252
0	295	14	355
10	359	14	381
9	646	14	658
9	708	14	717

1	240	480	1368
2	240	480	1368
3	240	648	1368
4	240	912	0
5	240	912	0
6	240	912	0
7	324	912	0
8	456	0	0
9	456	0	0
10	456	0	0
11	456	0	0
12	0	0	0
13	0	0	0
14	0	0	0

3	295	7	
0	0	14	768
11	75	14	84
0	175	4	190
5	431	12	439
6	545	13	554
8	661	14	669
0	679	5	689
0	737	10	768

1	224	584	1200
2	224	728	1200
3	256	728	1200
4	292	728	0
5	232	680	0
6	224	520	0
7	224	520	0
8	224	0	0
9	260	0	0
10	260	0	0
11	136	0	0
12	0	0	0
13	0	0	0
14	0	0	0

4	301	7	
0	0	14	576
11	0	14	26
2	95	7	100
0	146	5	153
2	181	9	186
7	322	11	327
0	498	2	576
3	547	9	555

1	492	1024	1848
2	492	1024	1848
3	296	632	1260
4	296	632	0
5	316	840	0
6	176	560	0
7	176	560	0
8	156	0	0
9	260	0	0
10	124	0	0
11	124	0	0
12	0	0	0
13	0	0	0
14	0	0	0

4	302	12	
0	0	14	576
0	40	5	47
0	226	4	237
10	281	14	293
3	291	9	298
12	307	14	346
9	348	14	361
9	384	14	391
0	438	6	448
6	475	9	479
0	505	7	512
0	544	6	576
12	551	14	576

1	424	1304	2592
2	424	1304	3192
3	424	1648	3192
4	652	1728	0
5	608	1640	0
6	580	1984	0
7	348	1376	0
8	256	0	0
9	256	0	0
10	540	0	0
11	540	0	0
12	0	0	0

13	0	0	0
14	0	0	0

4	303	4	
0	0	14	576
2	161	7	167
0	290	3	329
6	420	14	476
3	502	11	514
1	228	1112	1668
2	228	1112	1668
3	228	1112	1668
4	400	800	0
5	400	800	0
6	400	800	0
7	400	800	0
8	376	0	0
9	376	0	0
10	376	0	0
11	376	0	0
12	0	0	0
13	0	0	0
14	0	0	0

4	304	8	
0	0	14	576
9	84	14	93
7	124	13	132
0	208	9	222
9	236	14	246
0	305	3	312
4	346	13	358
11	466	14	472
6	511	14	522
1	84	688	1872
2	268	688	1872
3	268	1200	1872
4	148	872	0
5	180	920	0
6	180	920	0
7	436	920	0
8	436	0	0
9	460	0	0
10	348	0	0
11	348	0	0
12	0	0	0
13	0	0	0
14	0	0	0

4	305	5	
0	0	14	576
1	48	5	54
6	156	11	165
11	320	14	328
11	491	14	498
0	517	3	576
1	260	592	1296
2	260	592	1296
3	260	592	1296
4	60	120	0
5	60	240	0
6	36	192	0
7	36	192	0
8	36	0	0
9	96	0	0

10	96	0	0
11	96	0	0
12	0	0	0
13	0	0	0
14	0	0	0

END-OF-RUN. TOTAL BOARDS PROCESSED = 10

RECVRY Output

The first line is a short title to describe the type of boards processed by the program, and a summary of input values assigned to the program control parameters LBTD = 36, LPRM = 8, NBSTR = 291, and NBEND = 305. Output consists of percent recovery for each individual board, percent total recovery for all boards, percent total recovery for grade 3 Common boards, percent total recovery for grade 4 Common boards, and a summary of the number of times each available cutting combination was selected for the individual boards.

Output for the first board, number 291, shows that combination number 5 generated the "best" recovery area. From the PERMU output sample, combination number 5 is a (2, 1) cutting or an 8 and 4 Q.I.U. cut. Next is a set of six numbers which position the saw kerfs for combination number 5. The first saw kerf is located at unit 0 of the board. Units 1 through 8 are used for the first cut (8 Q.I.U.). The second saw kerf is located at unit 9. Units 10, 11, 12, and 13 are used for the second cut (4 Q.I.U.) and the third saw kerf is located at unit 14. A saw kerf location of zero indicates a position adjacent to the bottom edge of the board. If the saw kerfs and cutting widths do not account for all available units in the board as shown by boards 292, 295, and 302, two adjacent saw kerfs of one unit each are indicated, and only the upper saw kerf location is printed. The total clear area recovered is 8,004 Q.I.U., the total area of the board is 10,752 Q.I.U. (14 x 768), and the percent recovery for the board is 74.442 (8,004/10,752 x 100). This format is repeated for each board in the input file.

TEST 4-INCH 10 BRDS MIN LGT=36 11/27/72
36 8 291 305

BOARD NBR = 291
COMBO NBR = 5
0 9 14 15 16 17
AREA RECOVERED = 8004
AREA OF BOARD = 10752
PRCNT RECOVERY = 74.4420

BOARD NBR = 292
COMBO NBR = 5

0 10 15 16 17 18
 AREA RECOVERED = 8492
 AREA OF BOARD = 10752
 PRCNT RECOVERY = 78.9807

BOARD NBR = 293
 COMBO NBR = 5
 0 9 14 15 16 17
 AREA RECOVERED = 8532
 AREA OF BOARD = 10752
 PRCNT RECOVERY = 79.3527

BOARD NBR = 294
 COMBO NBR = 5
 0 9 14 15 16 17
 AREA RECOVERED = 8280
 AREA OF BOARD = 10752
 PRCNT RECOVERY = 77.0089

BOARD NBR = 295
 COMBO NBR = 5
 0 10 15 16 17 18
 AREA RECOVERED = 8496
 AREA OF BOARD = 10752
 PRCNT RECOVERY = 79.0179

BOARD NBR = 301
 COMBO NBR = 7
 0 5 14 15 16 17
 AREA RECOVERED = 5860
 AREA OF BOARD = 8064
 PRCNT RECOVERY = 72.6687

BOARD NBR = 302
 COMBO NBR = 7
 0 6 15 16 17 18
 AREA RECOVERED = 5112
 AREA OF BOARD = 8064
 PRCNT RECOVERY = 63.3929

BOARD NBR = 303
 COMBO NBR = 7
 0 5 14 15 16 17
 AREA RECOVERED = 5884
 AREA OF BOARD = 8064
 PRCNT RECOVERY = 72.9663

BOARD NBR = 304
 COMBO NBR = 7
 0 5 14 15 16 17
 AREA RECOVERED = 5908
 AREA OF BOARD = 8064
 PRCNT RECOVERY = 73.2639

BOARD NBR = 305
 COMBO NBR = 7
 0 5 14 15 16 17
 AREA RECOVERED = 6460
 AREA OF BOARD = 8064
 PRCNT RECOVERY = 80.1091

TOTAL AREA RECOVERED = 71028
 TOTAL AREA OF BOARDS = 94080
 PRCNT TOTAL RECOVERY = 75.4974

TOTAL AREA RECOVERED-3C = 41804
 TOTAL AREA OF BOARDS-3C = 53760
 PCNT RECOVERY BOARDS-3C = 77.7604

TOTAL AREA RECOVERED-4C = 29224
 TOTAL AREA OF BOARDS-4C = 40320
 PCNT RECOVERY BOARDS-4C = 72.4802

PRCNT RECOVERY PER CUTTING SIZE BOARDS-3C
 26.68899 51.07143 0.00000 0.00000

PRCNT RECOVERY PER CUTTING SIZE BOARDS-4C
 24.88095 47.59921 0.00000 0.00000

COMBO NBR NBR TIMES USED
 5 5
 7 5

END OF RUN. BOARDS 291 - 305 PROCESSED

An example of the System and Program Control Cards for running program CONST/RECVRY which generated the sample output follows:

JLIMITS, CM55000, T30, PR30. ACOUNT NBR., NAME.
 FTN.
 REQUEST, TAPE7, D0000, 01. CONST OUPUT (DISK)
 LGO.
 REWIND (TAPE7,LGO) RECVRY INPUT (DISK)
 RFL(43000)
 FTN.
 LGO.
 7/8/9
 CONST Source Deck.
 7/8/9
 CONST Program Control Card (LBTD,NBSTR,NBEND).
 Board Defect Data Cards (NDEF+3 Cards/Board).
 7/8/9
 RECVRY Source Deck.
 7/8/9
 RECVRY Program Control Card Type 1 (TITLE).
 RECVRY Program Control Card Type 2 (LBTD, LPRM,
 NBSTR, NBEND).
 6/7/8/9 END-OF-JOB Card.

If Programs CONST and RECVRY are to be run as two separate programs, request TAPE7 to be a magnetic tape output file from CONST and specify it as a tape input file to Program RECVRY.

Appendix 2: Source Listings for Programs

Program PERMU

```

PROGRAM PERMU
1(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1)
C
C CONVERSION OF PROG NAME J1 (BASIC) TO FORTRAN. THIS PROGRAM CONSTRUCTS
C THE PERMUTATION, COMBINATION TABLE FOR NSIZE(NBR OF BRO WOTH SIZES)
C AND NSWBL(NBR OF MACHINE SAW BLADES). INITIAL INPUT IS TAKEN FROM TWO
C CAROS. OUTPUT IS A PRINT LISTING AND IF DESIRED A TAPE FILE(TAPE1).
C O. MARKSTROM 10/72 RMFRES BJE
C
C--INPUT CARO 1 (315)
C IBWDT=BOARD WIOTH 14,30,46 UNITS
C NSIZE=NBR OF OIFFERENT SIZES TO CONSIDER. MAX=5
C NSWBL=NBR OF SAW BLADES FOR YOUR MACHINE. MAX=6
C--INPUT CARO 2 (515)
C KUT(I)=SIZES FOR NSIZE SIZES ON CARO 1. MAX=5
C
C--PROGRAM VARIABLE NAMES
C P(I,J)=PERMUTATION ARRAY FOR 6 SAW BLADES, 5 CUTTING WIOTHS
C C(I)=COMBINATION ARRAY FOR 6 SAW BLADES, 5 CUTTING WIOTHS
C LSUM=TOTAL NBR OF OIFFERENT CUT COMBINATIONS IN ARRAY C(KK)
C NSWB=SAW BLADE NBR., 1 TO NSWBL+1
C NCMB=COMBINATION (COMBO) NBR., 1 TO 1023. (FOR 3 SIZES,5 BLADES)
C KERF=ACCUMULATOR FOR 1/4-INCH SAW KERFS
C MWOH=ACCUMULATOR FOR CUTTING WIOTHS IN 1/4-INCH UNITS
C
C INTEGER P(7,1023),C(10)
C DIMENSION KUT(10)
C REWIND 1
C
C--INITIALIZE ARRAYS AND VARIABLES
C O0 5 I=1,10
C C(I)=0
C KUT(I)=0
C 5 CONTINUE
C O0 8 N1=1,7
C O0 7 N2=1,1023
C P(N1,N2)=0
C 7 CONTINUE
C 8 CONTINUE
C LSUM=0
C
C--READ AND PRINT OUT INPUT CAROS 1 AND 2. WRITE TAPE FILE 1 IO RECORD
C READ(5,100) IBWOT,NSIZE,NSWBL
C 100 FORMAT(5I5)
C KK=NSWBL + 1
C O0 15 N1=1,KK
C O0 10 N2=1,1023
C P(N1,N2)=6
C 10 CONTINUE
C 15 CONTINUE
C READ(5,100) (KUT(I),I=1,5)
C WRITE(6,102) IBWOT,NSIZE,NSWBL,(KUT(I),I=1,5)
C 102 FORMAT(1H1,3I5,5X,5I5,/)
C WRITE(1,103) NSIZE,(KUT(I),I=1,5)
C 103 FORMAT(6I5)
C
C--DETERMINE COMBINATIONS FOR NSIZE CUTS AND KK SAW BLADES
C K=0
C C(2)=NSIZE
C LSUM=LSUM + NSIZE
C I=0
C O0 20 K=2,KK
C I=K + 1
C C(I)=NSIZE**K + C(I-1)
C LSUM=LSUM + NSIZE**K
C 20 CONTINUE
C
C NCUT=C(I-1)
C WRITE(6,104) NCUT
C 104 FORMAT(1H ,*NBR OF OIFFERENT CUT COMBINATIONS =*I6,/)
C
C--FILL ARRAY P FOR NSIZE CUTS AND KK SAW BLADES
C N=1
C I=0
C C(1)=1
C O0 50 NSWB=1,KK
C K=C(NSWB)
C I=NSWB - 1
C IF(I.EQ.0) KZ=1
C IF(I.EQ.0) GO TO 25
C KZ=NSIZE**I
C K=K + 1
C 25 KCZ=K + KZ
C O0 30 NCMB=K,KCZ
C IF(NCMB.GT.1023) GO TO 35
C P(NSWB,NCMB)=N
C 30 CONTINUE
C 35 N=N + 1
C K=K + KZ
C IF(N.GT.NSIZE) GO TO 40
C IF(K.GE.LSUM) GO TO 45
C GO TO 25
C 40 N=1
C GO TO 25
C 45 N=1
C 50 CONTINUE
C
C--COMBINE CUT WIOTH AND SAW KERF SELECTIONS AND DETERMINE MIN BOARD
C--WIOTH FROM WHICH CUT CAN BE MADE
C KERF=0
C MWOH=0
C O0 75 NCMB=1,1023
C O0 55 NSWB=1,KK
C L=P(NSWB,NCMB)
C IF(L.EQ.6) GO TO 60
C MWOH=MWOH + KUT(L)
C KERF=KERF + 1
C 55 CONTINUE
C GO TO 70
C 60 MSUM=MWOH + KERF
C IF(MSUM.LE.IBWOT) GO TO 65
C P(7,NCMB)=99
C GO TO 70
C 65 P(7,NCMB)=MWOH
C 70 KERF=0
C MWOH=0
C 75 CONTINUE
C
C--PRINT OUT PERM-COMB TABLE, ARRAY P, OF POSSIBLE CUTS FROM A BOARD
C--4 UNITS(1-INCH) TO IBWOT UNITS WIOE.
C KTMP=0
C O0 85 N=4,IBWOT
C O0 80 NCMB=1,1023
C IF(P(7,NCMB).NE.N) GO TO 80
C MWOH=P(7,NCMB)
C KTMP=P(6,NCMB)
C IF(KTMP.NE.0) GO TO 78
C KTMP=6
C 78 WRITE(6,106) NCMB,(P(I,NCMB),I=1,5),KTMP,MWOH
C 106 FORMAT(1H ,8I5)
C WRITE(1,108) NCMB,(P(I,NCMB),I=1,5),KTMP,MWOH
C 108 FORMAT(8I5)
C 80 CONTINUE
C 85 CONTINUE
C
C ENOFIE 1
C REWIND 1
C CALL EXIT
C ENO

```

Program CONST

```

PROGRAM CONST
1(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7)

C
C CONVERSION OF PROG NAME CONST (BASIC) TO FORTRAN. THIS PROGRAM INPUTS
C INITIAL X,Y DEFECT COORDINATES(FROM CARDS) AND OUTPUTS MODIFIED X,Y
C COORDINATES(ON FILE 7) USING A MIN LENGTH BETWEEN DEFECTS LIMITATION
C WHICH IS CARD INPUT(VAR. LBTD). THE MODIFIED X,Y FOR DEFECTS IS INPUT
C TO PROG RECVRY WHICH SELECTS THE CUT COMBO FOR EACH INDIVIDUAL BRO.
C D. MARKSTROM 10/72 RMFRES BJE MODF. 07/72
C
C--INPUT CONTROL CARD (315)
C LBTD=MIN. LENGTH BETWEEN DEFECTS CONTROL VARIABLE (IN UNITS)
C NBSTR=NBR OF THE BRD TO START ANALYSIS EXP. 201
C NBEND=NBR OF THE BRD TO END ANALYSIS EXP. 299
C
C--DATA CARDS. BRO IDENT., BRO COORD., DEF COORD., END OF BRO INPUT.
C
C--BOARD IDENTIFICATION CARD. ONE/BOARD (5X,12,24X,15,32X,14)
C NGRD=BOARD GRADE CODE
C NBRD=BOARD NUMBER
C NDEF=NUMBER OF DEFECTS IN BOARD
C--BOARD COORDINATE CARD. ONE/BOARD (13,1X,13,5X,13,1X,13)
C IBXS=BOARD X-DIR(WIDTH) START COORD. IBXS=0
C IBYS=BOARD Y-DIR(LNGTH) START COORD. IBYS=0
C IBXE=BOARD X-DIR(WIDTH) END COORD. IBXE=14,30, OR 46
C IBYE=BOARD Y-DIR(LNGTH) END COORD. IBYE=576,672, OR 768
C--DEFECT COORDINATE CARD(S). NDEF/BOARD (13,1X,13,5X,13,1X,13)
C IOXS=DEFECT X-DIR(WIDTH) START COORD. IOXS GE IBXS
C IOYS=DEFECT Y-DIR(LNGTH) START COORD. IOYS GE IBYS
C IOXE=DEFECT X-DIR(WIDTH) END COORD. IOXE LE IBXE
C IOYE=DEFECT Y-DIR(LNGTH) END COORD. IOYE LE IBYE
C--END OF BOARD DATA CARD. ONE/BOARD (13,1X,13,5X,13,1X,13)
C IDXS=999
C IDYS=0
C IOXE=0
C IOYE=0
C--A SET OF DATA CARDS FOR ONE BOARD WILL CONTAIN NDEF+3 CARDS.
C
C--PROGRAM VARIABLE NAMES
C T(1,44)=TABLE OF START AND END DEFECT COORD. FOR A BRO. I=IBXE
C DFCT(3)=ACCUMULATOR FOR DEFECT AREA FOR 3 CUTTINGS, 4-8-12 UNITS.
C P(1)=POINTER WHICH POSITIONS DEFECT COORD. IN TABLE T. I=IBXE
C KUT(3)=THREE CUTTING SIZES. SET BY PROG STATEMENTS.
C LWB(1,768)=BRO DIMENSION ARRAY FOR COMPUTER SIMULATION OF ALL
C DEFECTS IN A BRO. I=IBXE
C MIND(1)=MIN DEFECT COORD. FOR LWB SCAN OF DEFECT AREA. I=IBXE
C MAXD(1)=MAX DEFECT COORD. FOR LWB SCAN OF DEFECT AREA. I=IBXE
C LDS=LENGTH DEFECT START COORD. VALUE, USED TO X-OUT DEFECT IN LWB
C LOE=LENGTH DEFECT END COORD. VALUE, USED TO X-OUT DEFECT IN LWB
C NX=WIDTH DEFECT COORD. VALUE, USED TO X-OUT DEFECT IN LWB
C
C--THE I DIMENSION IS SET BY THE USER AND IS DETERMINED BY THE WIDTH
C--SIZE OF BOARDS TO BE PROCESSED. THIS HELPS TO REDUCE CORE STORAGE
C--REQUIREMENTS AND RUN COSTS.
C
C INTEGER T(14,44),DFCT(3),P(14)
C DIMENSION KUT(3)
C COMMON LWB(14,768),MIND(14),MAXD(14),LDS,LDE,NBRD,NX
C REWIND 7
C--INITIALIZE ARRAYS. SET THE UPPER LIMIT OF THE I INDEX TO IBXE.
C KFRST=0
C KOUNT=0
C KUT(1)=3
C KUT(2)=7
C KUT(3)=11
C DFCT(1)=0
C DFCT(2)=0
C DFCT(3)=0
C DO 2 I=1,14
C DO 2 J=1,768
C 2 LWB(I,J)=0
C DO 4 I=1,14
C DO 4 J=1,44
C 4 T(I,J)=0
C DO 6 I=1,14
C 6 P(I)=2
C--READ INPUT CONTROL CARD
C READ(5,100) LBTD,NBSTR,NBEND
C 100 FORMAT(315)
C WRITE(6,101) LBTD,NBSTR,NBEND
C 101 FORMAT(1H1,5X,*MIN. LENGTH BETWEEN DEFECTS CONTROL FOR THIS RUN IS
C 1 *14,2X,14,* - *,14,/)
C
C 10 DO 8 I=1,14
C MIND(I)=900
C 8 MAXD(I)=0
C--READ BOARD ID CARD
C 12 READ(5,102) NGRD,NBRD,NDEF
C 102 FORMAT(5X,12,24X,15,32X,14)
C IF(EDF(5)) GO TO 15
C 15 IF(NBRD.LT.NBSTR) GO TO 87
C IF(NBRD.GT.NBEND) GO TO 90
C WRITE(6,104) NGRD,NBRD,NDEF
C 104 FORMAT(1H0,13,1X,13,5X,13)
C--READ BOARD COORD. CARD
C READ(5,106) IBXS,IBYS,IBXE,IBYE
C 106 FORMAT(13,1X,13,5X,13,1X,13)
C--WRITE FILE 7 FOR RECVRY INPUT
C WRITE(7) NGRD,NBRD,NDEF
C WRITE(7) IBXS,IBYS,IBXE,IBYE
C WRITE(6,107) IBXS,IBYS,IBXE,IBYE
C 107 FORMAT(1H ,13,1X,13,5X,13,1X,13)

C--SET BRD Y-DIR START COORD. = 1 IBYS
C IBYS=1
C--READ BOARD DEFECT COORD. CARD(S)
C 20 READ(5,106) IDXS,IOYS,IOXE,IOYE
C IF(IDXS.EQ.999) GO TO 35
C IF(KFRST.NE.0) GO TO 22
C WRITE(6,107) IOXS,IOYS,IOXE,IOYE
C 22 LDS=IDYS + 1
C LDE=IOYE
C IF(IDXE.LE.IBXE) GO TO 25
C IOXE=IBXE
C 25 KK=IDXE - 1
C IF(IOXS.EQ.0.AND.KK.EQ.0) GO TO 32
C--STORE DEFECT COORD. IN ARRAY T AND SIMULATE DEFECT IN ARRAY LWB
C DO 30 I=IDXS,KK
C N=I + 1
C K=P(N)
C T(N,K+1)=IDYS
C T(N,K+2)=IDYE
C P(N)=P(N) + 2
C NX=N
C CALL XOUT
C 30 CONTINUE
C GO TO 20
C
C 32 K=P(1)
C T(1,K+1)=IOYS
C T(1,K+2)=IDYE
C P(1)=P(1) + 2
C NX=1
C CALL XOUT
C GO TO 20
C
C--DETERMINE IF MIN LENGTH BETWEEN DEFECTS EXISTS. IF NO, BLOCK OUT
C--(X-OUT) CLEAR AREA BETWEEN DEFECTS IN ARRAY LWB.
C 35 CONTINUE
C KEND=IBXE
C DO 50 N=1,KEND
C K=P(N)
C P(N)=P(N) + 1
C T(N,K+1)=IBYE
C T(N,K+2)=IBYE
C KK=P(N)
C DO 45 M=3,KK,2
C IF(T(N,M) - T(N,M-1).EQ.0) GO TO 45
C IF(T(N,M) - T(N,M-1).LT.LBTD) GO TO 40
C IF(T(N,M).EQ.IBYS) GO TO 45
C GO TO 45
C 40 LOS=T(N,M-1) + 1
C LOE=T(N,M+1)
C NX=N
C CALL XOUT
C 45 CONTINUE
C P(N)=2
C 50 CONTINUE
C
C--SET INDEXES FOR BOARD SCAN TO FIND DEFECT AREAS
C DO 85 N=1,KEND
C KK=0
C IS=N
C DO 80 K=3,5
C I=K - KK
C KK=KK + 2
C IE=N + KUT(I)
C LIMIT=IBXE - KUT(I)
C IF(N.GT.LIMIT) GO TO 80
C
C--FIND MIN-MAX DEFECT COORD. FOR BOARD SCAN. IF MAXD(M)=0, NO DEFECTS
C--FOR RANGE OF SCAN.
C
C DO 54 M=IS,IE
C IF(MAXD(M).EQ.0) GO TO 54
C GO TO 56
C 54 CONTINUE
C GO TO 80
C 56 IBYS=MIND(IS)
C IBYE=MAXD(IS)
C DO 60 M=IS,IE
C IF(MIND(M).LT.IBYS) IBYS=MIND(M)
C IF(MAXD(M).GT.IBYS) IBYE=MAXD(M)
C 60 CONTINUE
C
C--MAKE X,Y SCAN OF BOARD ARRAY LWB TO FIND DEFECT AREA. IF DEFECT FOUND
C--ACCUMULATE DEFECT AREA IN DFCT(I).
C NCLRY=0
C DO 75 J=IBYS,IBYE
C DO 65 M=IS,IE
C IF(LWB(M,J).EQ.NBRD) GO TO 68
C 65 CONTINUE
C NCLRY=NCLRY + 1
C GO TO 75
C--TEST FOR MIN DISTANCE BETWEEN DEFECTS IN LENGTH(Y) DIRECTION
C 68 IF(NCLRY.EQ.0) GO TO 70
C IF(NCLRY.GE.LBTD) NCLRY=0
C IF(NCLRY.GE.LBTD) GO TO 70
C MULT=KUT(I) + 1
C IADD=NCLRY * MULT
C DFCT(I)=DFCT(I) + IADD
C NCLRY=0
C
C 70 DFCT(I)=DFCT(I) + KUT(I) + 1
C 75 CONTINUE
C 80 CONTINUE
C P(1)=2
C

```



```

C--LIST DEFECT AREAS FOR FIRST BOARD AND WRITE FILE 7 FOR ALL BOARDS
C--FOR RECVRY INPUT.  BRANCH AND DO NEXT BOARD.
C      IF(KFRST.NE.0) GO TO 82
      WRITE(6,108) N,DFCT(1),DFCT(2),DFCT(3)
108  FORMAT(1H ,14,3(2X,15))
      82 WRITE(7) DFCT(1),DFCT(2),DFCT(3)
        DFCT(1)=0
        DFCT(2)=0
        DFCT(3)=0
      85 CONTINUE
        KDUNT=KDUNT + 1
        KFRST=1
        GO TO 10
C
      87 KRECS=NDEF + 2
        DD 88 NSKP=1,KRECS
        READ(5,106) IDUM1,IDUM2,IDUM3,IDUM4
      88 CONTINUE
        GO TO 12
C
      90 WRITE(6,110) KDUNT
110  FORMAT(1H0,5X,*END-OF-RUN. TOTAL BOARDS PROCESSED =*14)
      ENOFIL 7
      REWIND 7
      ENO

      SUBROUTINE XDUT
C
C ROUTINE GO SUB 490 FROM PROG CONST (BASIC)
C THIS ROUTINE SIMULATES THE BRO DEFECT AREAS BY STORING THE BRO NUMBER
C (NBRD) IN ALL X,Y LOCATIONS OF A DEFECT. IT IS X-ING OUT, OR BLOCKING
C OUT THE AREA OF DEFECT AS SPECIFIED BY THE INDICATORS NX,LOS, AND LDE.
C MAX AND MIN DEFECT LIMITS IN THE Y DIRECTION ARE ALSO DETERMINED.
C
      COMMON LWB(14,768),MIND(14),MAXD(14),LDS,LDE,NBRD,NX
      DD 10 I=LDS,LDE
      LWB(NX,I)=NBRD
      10 CONTINUE
C
      IF(LDE.GT.MAXD(NX)) MAXD(NX)=LDE
      IF(LOS.LT.MIND(NX)) MIND(NX)=LOS
      RETURN
      ENO

```

Program RECVRY

```

PROGRAM RECVRY
1(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7)
C
C CONVERSION OF PROG NAME G0001 (BASIC) TO FORTRAN. THIS PROGRAM INPUTS
C THE MODIFIED X,Y BOARD COORDINATES (DEFECT AREA) FROM FILE TAPE7
C GENERATED IN PROG CONST AND OUTPUTS THE BEST CUT TO MAXIMIZE RECOVER-
C ABLE AREA FOR INDIVIDUAL BOARDS. THE PERMU-COMBO TABLE TO BE USED TO
C EVALUATE SELECTED BRO SIZES IS CODED IN DATA ARRAY KPCT. THESE VALUES
C WILL CHANGE DEPENDING ON BRO SIZE AND A NEW DATA LIST MUST BE INSERTED
C IN SOURCE DECK. D. MARKSTROM 10/72 RMFRES BJE MODF. 06/72
C
C--INPUT CONTROL CARD TYPE 1 (4A10)
C   TITLE(I)=IDENTIFICATION CARD FOR RUN
C
C--INPUT CONTROL CARD TYPE 2 (4I5)
C   LBTD=MIN. LENGTH BETWEEN DEFECTS, SAME AS IN PROG CONST.
C   LPRM=LOWER LIMIT OF COMBINED CUTTINGS TO BE CONSIDERED FROM THE
C   PERMU-COMBO SETS IN ARRAY KPCT. CUTTINGS GT LPRM ARE USED
C   TO EVALUATE BEST FIT UNTILL A STOP INDICATOR, 999 IS FOUND.
C   NBSTR=NBR OF THE BRO TO START ANALYSIS EXP. 201
C   NBEND=NBR OF THE BRO TO END ANALYSIS EXP. 299
C
C--PROGRAM VARIABLE NAMES.
C
C   NSIZE=NBR OF CUTTING SIZES TO CONSIDER MAX OF 5
C   KUT(I)=CUT SIZES IN QTR UNITS FOR NSIZE CUTS EXP. 4,8,12,D,D
C   NGRD=BRO GRADE CODE
C   NBRO=BRO NUMBER
C   NDEF=NBR OF DEFECTS IN BRO
C   IBXS=BRO X-DIR(WIDTH) START COORD. IBXS=0
C   IBYS=BRO Y-DIR(LNGTH) START COORD. IBYS=0
C   IBXE=BRO X-DIR(WIDTH) END COORD. IBXE=14,3D,OR 46
C   IBYE=BRO Y-DIR(LNGTH) END COORD. IBYE=576,672,OR 768
C   NCMB=COMBINATION(COMBO) NUMBER
C   KKS1=COMBO CUT SIZE 1 CODE 1=4, 2=8, 3=12 UNITS
C   KKS2=COMBO CUT SIZE 2 CODE 1=4, 2=8, 3=12 UNITS
C   KKS3=COMBO CUT SIZE 3 CODE 1=4, 2=8, 3=12 UNITS
C   KKS4=COMBO CUT SIZE 4 CODE 1=4, 2=8, 3=12 UNITS
C   KKS5=COMBO CUT SIZE 5 CODE 1=4, 2=8, 3=12 UNITS
C   KKS6=COMBO CUT SIZE 6 CODE 1=4, 2=8, 3=12 UNITS
C   NTCUT=TOTAL UNITS CUT FOR COMBO NCMB(SAWKERFS NOT INCLUDED)
C
C   CLEAR(I,J)=WORK ARRAY TO DETERMINE CLEAR AREA OF BOARD
C   RGRSZ(I,J)=ARRAY TO ACCUMULATE RECOVERY PER BOARD GRADE AND SIZE
C   SCMBN(I)=ARRAY OF SELECTED COMBO NBRS
C   IGRO(I)=TOTAL AREA RECOVERED BY GRADE
C   JGRD(I)=TOTAL AREA OF BOARDS BY GRADE
C
C   INTEGER CLEAR(6,5D),RGRSZ(5,6),SCMBN(13D),TITLE(4),P1,P2,P3,P4,P5,
C   IP6,CN
C   DIMENSION KUT(9),IGRO(9),JGRD(9),KSIZE(6)
C   DIMENSION KPCT(56)
C
C--DATA LIST STATEMENTS. KPCT IS DEPENDENT ON SIZE OF BROS EVALUATED.
C--INCLUDE ONE MORE COMBO SET THEN DESIRED TO TERMINATE SEARCH WITH
C--NCMB (KPCT(IDX)) SET = 999. KPCT IS FOR IBXE=14 BOARD WIDTH.
C
C   DATA KSIZE/3,4,8,12,D,0/
C   DATA KPCT/1,1,6,6,6,6,6,4,2,2,6,6,6,6,6,8,4,1,1,6,6,6,6,8,3,3,6,6,
C   16,6,6,12,5,2,1,6,6,6,6,12,7,1,2,6,6,6,6,12,999,1,1,1,1,1,1,99/
C
C--ZERO OUT ARRAYS AND READ CONTROL CARDS 1 AND 2
C   KA=0
C   KSAR=0
C   KSAB=0
C   KSLT=1
C   REWIND 7
C   DO 2 I=1,6
C   DO 2 J=1,50
C 2 CLEAR(I,J)=0
C   DO 4 I=1,5
C   DO 4 J=1,6
C 4 RGRSZ(I,J)=0
C   DO 6 I=1,130
C 6 SCMBN(I)=0
C   DO 10 I=1,9
C   KUT(I)=0
C   IGRO(I)=0
C 10 JGRD(I)=0
C   READ(5,1D0) TITLE
C 100 FORMAT(4A10)
C   READ(5,1D2) LBTD,LPRM,NBSTR,NBEND
C 102 FORMAT(4I5)
C   WRITE(6,104) TITLE,LBTD,LPRM,NBSTR,NBEND
C 104 FORMAT(1H1,9X,4A10,4I5,/)
C   NSIZE=KSIZE(1)
C   DO 15 I=1,5
C 15 KUT(I)=KSIZE(I+1)
C
C--READ INPUT GENERATED BY PROG CONST. IO, BRO COORD., DEFECT AREAS
C 20 IOX=1
C   READ(7) NGRD,NBRO,NDEF
C   IF(EOF(7)) GO TO 25
C 25 READ(7) IBXS,IBYS,IBXE,IBYE
C   IF(NBRO.LT.NBSTR) GO TO 27
C   IF(NBRO.GT.NBEND) GO TO 90
C   GO TO 28
C 27 KSLT=0
C
C 28 M1=IBXE
C   DO 30 J=1,M1
C   READ(7) CLEAR(1,J),CLEAR(2,J),CLEAR(3,J)
C
C 30 CONTINUE
C   IF(KSLT.EQ.0) GO TO 80
C
C--USING DEFECT AREAS FROM PROG CONST, COMPUTE CLEAR AREAS OF BOARD
C   DO 40 I=1,NSIZE
C   MULT=KUT(I)*IBYE
C   M1=((IBXE-KUT(I))+1)
C   DO 35 J=1,M1
C   CLEAR(I,J)=MULT - CLEAR(I,J)
C 35 CONTINUE
C 40 CONTINUE
C   GO TO 50
C
C   DO 45 J=1,IBXE
C   WRITE(6,112) J,CLEAR(1,J),CLEAR(2,J),CLEAR(3,J)
C 112 FORMAT(1H ,4I8)
C 45 CONTINUE
C
C--SELECT A CUTTING COMBINATION FROM ARRAY KPCT
C 50 CONTINUE
C 52 NCMB=KPCT(IDX)
C   KKS1=KPCT(IDX+1)
C   KKS2=KPCT(IDX+2)
C   KKS3=KPCT(IDX+3)
C   KKS4=KPCT(IDX+4)
C   KKS5=KPCT(IDX+5)
C   KKS6=KPCT(IDX+6)
C   NTCUT=KPCT(IDX+7)
C   IF(NTCUT.EQ.99) GO TO 75
C   IOX=IOX+8
C   IF(NTCUT.LE.LPRM) GO TO 52
C   KX=IBXE - NTCUT
C
C--KX IS NBR OF SAW KERF CUTS WITHIN THE BOARD DIMENSIONS IBXS-IBXE
C--EVALUATE SELECTED CUTTING COMBINATION FOR MAX CLEAR AREA, STORE IN KA
C   DO 70 N=1,KX
C   I1=(KUT(KKS1) + N + 1)
C   J1=I1 + (KX-N)
C   KS=CLEAR(KKS1,N)
C   DO 68 N1=1,J1
C   KS1=CLEAR(KKS2,N1) + KS
C   I2=(KUT(KKS2) + N1 + 1)
C   J2=I2 + (KX-((J1-I1)+N))
C   DO 66 N2=1,J2
C   KS2=CLEAR(KKS3,N2) + KS1
C   I3=(KUT(KKS3) + N2 + 1)
C   J3=I3 + (KX-((J2-I2)+(J1-I1)+N))
C   DO 64 N3=1,J3
C   KS3=CLEAR(KKS4,N3) + KS2
C   I4=(KUT(KKS4) + N3 + 1)
C   J4=I4 + (KX-((J3-I3)+(J2-I2)+(J1-I1)+N))
C   DO 62 N4=1,J4
C   KS4=CLEAR(KKS5,N4) + KS3
C   I5=(KUT(KKS5) + N4 + 1)
C   J5=I5 + (KX-((J4-I4)+(J3-I3)+(J2-I2)+(J1-I1)+N))
C   DO 60 N5=1,J5
C   KS5=CLEAR(KKS6,N5) + KS4
C   IF(KS5.GT.KA) GO TO 85
C 58 KS5=0
C 60 CONTINUE
C 62 CONTINUE
C 64 CONTINUE
C 66 CONTINUE
C 68 CONTINUE
C 70 CONTINUE
C   GO TO 52
C
C--OUTPUT INDIVIDUAL BOARD RESULTS, PCNT RECOVERY BY BRO
C 75 IF(KA.EQ.0) GO TO 80
C   WRITE(6,116) NBRO
C 116 FORMAT(1H ,9X,*BOARD NBR = *I4)
C   WRITE(6,118) CN
C 118 FORMAT(1H ,9X,*COMBO NBR = *I4)
C   WRITE(6,120) P1,P2,P3,P4,P5,P6
C 120 FORMAT(1H ,9X,6I4)
C   WRITE(6,122) KA
C 122 FORMAT(1H ,9X,*AREA RECOVERED = *I6)
C   KB=IBXE*IBYE
C   WRITE(6,124) KB
C 124 FORMAT(1H ,9X,*AREA OF BOARD = *I6)
C   AR=FLOAT(KA)
C   AB=FLOAT(KB)
C   PR=(AR/AB) * 100.
C   WRITE(6,126) PR
C 126 FORMAT(1H ,9X,*PRCNT RECOVERY =*F8.4,/)
C   P1=P1 + 1
C   P2=P2 + 1
C   P3=P3 + 1
C   P4=P4 + 1
C   P5=P5 + 1
C   P6=P6 + 1
C
C--ACCUMULATE RESULTS BY GRADE AND CUTTING SIZES
C   SCMBN(CN)=SCMBN(CN) + 1
C   KSAB=KSAB + KB
C   KSAR=KSAR + KA
C   IGRO(NGRD)=IGRO(NGRD) + KA
C   JGRD(NGRD)=JGRD(NGRD) + KB
C   RGRSZ(NGRD,L1)=RGRSZ(NGRD,L1) + CLEAR(L1,P1)
C   RGRSZ(NGRD,L2)=RGRSZ(NGRD,L2) + CLEAR(L2,P2)
C   RGRSZ(NGRD,L3)=RGRSZ(NGRD,L3) + CLEAR(L3,P3)
C   RGRSZ(NGRD,L4)=RGRSZ(NGRD,L4) + CLEAR(L4,P4)
C   RGRSZ(NGRD,L5)=RGRSZ(NGRD,L5) + CLEAR(L5,P5)
C   RGRSZ(NGRD,L6)=RGRSZ(NGRD,L6) + CLEAR(L6,P6)

```

```

C
80 KA=0
   KSLT=1
   00 B2 I=1,6
   00 B2 J=1,50
82 CLEAR(I,J)=0
   GO TO 20
C
85 CN=NCMB
   KA=KS5

   P1=N - 1
   P2=N1 - 1
   P3=N2 - 1
   P4=N3 - 1
   P5=N4 - 1
   P6=N5 - 1
   L1=KKS1
   L2=KKS2
   L3=KKS3
   L4=KKS4
   L5=KKS5
   L6=KKS6
   GO TO 58
C
C--OUTPUT ALL BOARD RESULTS, PCNT RECOVERY BY GRAOE, BY SIZE WITHIN GRO
90 WRITE(6,200) KSAR
200 FORMAT(1H0,9X,*TOTAL AREA RECOVERED = *IB)
   WRITE(6,202) KSAB
202 FDRMAT(1H ,9X,*TOTAL AREA OF BOARDS = *IB)
   AR=FLOAT(KSAR)
   AB=FLDAT(KSAB)
   PR=(AR/AB) * 100.
   WRITE(6,204) PR
204 FORMAT(1H ,9X,*PCNT TOTAL RECOVERY = *FB.4,/)
C
   WRITE(6,206) IGRO(3)
206 FORMAT(1H0,9X,*TOTAL AREA RECOVERED-3C = *IB)
   WRITE(6,208) JGRO(3)
208 FORMAT(1H ,9X,*TOTAL AREA OF BOARDS-3C = *IB)
   AR=FLOAT(JGRO(3))
   AB=FLOAT(IGRO(3))
   IF(AR.EQ.0.0) PR=0.0
   IF(AR.EQ.0.0) GO TO 92
   PR=(AB/AR) * 100.
92 WRITE(6,210) PR
210 FORMAT(1H ,9X,*PCNT RECOVERY BOARDS-3C = *FB.4,/)
C
   WRITE(6,212) IGRD(4)
212 FORMAT(1H0,9X,*TOTAL AREA RECOVERED-4C = *IB)
   WRITE(6,214) JGRD(4)
214 FORMAT(1H ,9X,*TOTAL AREA OF BOARDS-4C = *IB)
   AR=FLOAT(JGRD(4))
   AB=FLOAT(IGRD(4))
   IF(AR.EQ.0.0) PR=0.0
   IF(AR.EQ.0.0) GO TO 94
   PR=(AB/AR) * 100.
94 WRITE(6,216) PR
216 FORMAT(1H ,9X,*PCNT RECOVERY BOARDS-4C = *FB.4,/)
C
   WRITE(6,218)
218 FORMAT(1H0,9X,*PCNT RECOVERY PER CUTTING SIZE BOARDS-3C*)
   IF(JGRO(3).EQ.0) GO TO 96
   AR=FLOAT(JGRD(3))
   AB=FLOAT(RGRSZ(3,1))

   PR1=(AB/AR) * 100.
   AB=FLOAT(RGRSZ(3,2))
   PR2=(AB/AR) * 100.
   AB=FLDAT(RGRSZ(3,3))
   PR3=(AB/AR) * 100.
   AB=FLOAT(RGRSZ(3,4))
   PR4=(AB/AR) * 100.
   GO TO 98
96 PR1=0.0
   PR2=0.0
   PR3=0.0
   PR4=0.0
98 WRITE(6,220) PR1,PR2,PR3,PR4
220 FORMAT(1H ,7X,4(2X,F9.5))
C
   WRITE(6,222)
222 FORMAT(1H0,9X,*PCNT RECOVERY PER CUTTING SIZE BOARDS-4C*)
   IF(JGRD(4).EQ.0) GO TO 140
   AR=FLOAT(JGRD(4))
   AB=FLDAT(RGRSZ(4,1))
   PR1=(AB/AR) * 100.
   AB=FLOAT(RGRSZ(4,2))
   PR2=(AB/AR) * 100.
   AB=FLDAT(RGRSZ(4,3))
   PR3=(AB/AR) * 100.
   AB=FLOAT(RGRSZ(4,4))
   PR4=(AB/AR) * 100.
   GO TO 142
140 PR1=0.0
   PR2=0.0
   PR3=0.0
   PR4=0.0
142 WRITE(6,220) PR1,PR2,PR3,PR4
C
   WRITE(6,224)
224 FORMAT(1H0,9X,*COMBO NBR NBR TIMES USED *)
   DO 144 KC=1,130
   IF(SCMBN(KC).EQ.0) GO TO 144
   WRITE(6,226) KC,SCMBN(KC)
226 FORMAT(1H ,9X,I5,10X,I5)
144 CONTINUE
C
C--ENO DF RUN. REWIND TAPE7, PRINT MESSAGE, CALL EXIT
C
   WRITE(6,228) NBSTR,NBEND
228 FORMAT(1H0,9X,*END DF RUN. BOARDS*I5,* - *I5,* PROCESSED*)
   REWIND 7
   CALL EXIT
   ENO

```


Program DEFEDIT

```

PROGRAM DEFEDIT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C PROG TO EDIT DATA CARDS FOR CUTUP-REPAIR-RECOVERY STUDY D.MARKSTROM
C
    DIMENSION DLW(30),DLL(30),DUW(30),DUL(30),IDH(B)
C
    READ(5,200) IDH
    200 FORMAT(8A10)
    WRITE(6,1) IDH
    1 FORMAT(1H1,5X,8A10,/)
C--READ AND CHECK HEADER CARD.
    3 READ(5,10) CLASS, NUMBD,NDEF
    10 FORMAT(5X,R3,22X,I6,32X,I4)
    IF(EOF(5)) 100,9
    9 IF(CLASS .EQ. 3R 3C .OR. CLASS .EQ. 3R 4C) GOTO 11
    GO TO 14
    11 IF(NDEF .LT. 1) GO TO 14
    GO TO 13
    14 WRITE(6,2) NUMBD
    2 FORMAT(10*,10X,*ERROR IN THE FIRST CARD OF BOARD *,I4)
C--READ BOARD COORDINATE CARD. CHECK LWR COORD. LT UPR COORD..
    13 READ(5,20) PLW,PLL,PUW,PUL
    20 FORMAT(F3.0,1X,F3.0,5X,F3.0,1X,F3.0)
    IF(PLW .GE. PUW .OR. PLL .GE. PUL) GO TO 46
    IF(PUW .NE. 14.)GO TO 21
    GO TO 25
    21 IF(PUW .NE. 30.)GO TO 22
    GO TO 25
    22 IF(PUW .EQ. 46.)GO TO 25
    GO TO 46
    25 IF(PUL .NE. 576.)GO TO 40
    GO TO 35
    40 IF(PUL .NE. 672.)GO TO 41
    GO TO 35
    41 IF(PUL .EQ. 768.)GO TO 35
    46 WRITE(6,47) NUMBD,CLASS,PLW,PLL,PUW,PUL
    47 FORMAT(1H0,10X,*BOARD NBR.*I4,* CLASS *R3,* HAS AN ERROR IN THE BO
    1ARD COORD. CARD*,2(2X,F5.0,2X,F5.0))
C--READ DEFECT COORDINATE CARD(S).
    35 DO 36 I=1,NDEF
    READ(5,20) DLW(I),DLL(I),DUW(I),DUL(I)
    36 CONTINUE
C--CHECK LWR COORD. LT UPR COORD. AND FOR DEFECTS OUTSIDE BRD COORD..
    DO 60 I=1,NDEF
    IF(DLW(I)) 56,52,52
    52 IF(DLL(I)) 56,54,54
    54 IF(DLW(I).GT.PUW) GO TO 56
    IF(DUW(I).GT.PUW) GO TO 56
    IF(DLL(I).GT.PUL) GO TO 56
    IF(DUL(I).GT.PUL) GO TO 56
    IF(DLW(I).GE.DUW(I)) GO TO 56
    IF(DLL(I).GE.DUL(I)) GO TO 56
    GO TO 60
    56 WRITE(6,57) I,NUMBD,DLW(I),DLL(I),DUW(I),DUL(I)
    57 FORMAT(1H0,10X,*DEFECT CARD*I3,* BOARD NBR.*I4,* IS IN ERROR*,
    12(2X,F5.0,2X,F5.0))
    60 CONTINUE
C--CHECK FOR OVERLAP OF DEFECTS
    IF(NDEF-1) 3,3,62
    62 J1=NDEF - 1
    DO 74 J=1,J1
    J2=J + 1
    DO 74 K=J2,NDEF
    IF(DUL(J)-DLL(K)) 74,64,64
    64 IF(DLL(J)-DUL(K)) 66,66,74
    66 IF(DUW(J)-DLW(K)) 74,68,68
    68 IF(DLW(J)-DUW(K)) 70,70,74
    70 WRITE(6,72) J,NUMBD,DLW(J),DLL(J),DUW(J),DUL(J)
    72 FORMAT(1H0,10X,*DEFECT CARD*I3,* BOARD NBR.*I4,* HAS OVERLAP*,
    12(2X,F5.0,2X,F5.0))
    74 CONTINUE
    GO TO 3
C
    100 CONTINUE
    CALL EXIT
    END

```

Program CTDFCT

```

PROGRAM CTDFCT
    1(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1)
C
C PROG TO WRITE BOARD DEFECT DATA CARDS ON TAPE(FILE 1) FOR INPUT TO PROG
C CONST. A DUMMY RECORD,CONTAINING 999,0,0,0,IS WRITTEN AS AN END-OF-BOARD
C INDICATOR. INPUT IS A SET OF DEFECT CARDS GROUPED BY BRD-WIDTH CLASS
C SUCH AS ALL 4-INCH, 8-INCH, OR 12-INCH BRDS.
C D.MARKSTROM REPAIR STUDY 02/71 BJE
C
    DIMENSION LSTID(B)
C
    REWIND 1
    KDUMY=0
    KTREC=0
    KEND=0
    LAST=0
    NGRD=0
    NBRD=0
    NDEF=0
    IXS=0
    IYS=0
    IXE=0
    IYE=0
C--READ LIST ID CARD TO IDENTIFY OUTPUT (BA10)
    READ(5,100) LSTID
    100 FORMAT(BA10)
    WRITE(6,102) LSTID
    102 FORMAT(1H1,5X,BA10)
C
C READ BRD DEFECT CARDS AND OUTPUT TO TAPE1. LIST FOR MASTER RECORD.
C
    10 READ(5,104) NGRD,NBRD,NDEF
    104 FORMAT(5X,I2,24X,I5,32X,I4)
    IF(EOF(5)) 900,15
    15 LAST=NBRD
    WRITE(1,106) NGRD,NBRD,NDEF,KDUMY
    106 FORMAT(4I4)
    WRITE(6,108) NGRD,NBRD,NDEF,KDUMY
    108 FORMAT(1H0,10X,4I5)
    KEND=NDEF + 1
    DO 20 I=1,KEND
    READ(5,110) IXS,IYS,IXE,IYE
    110 FORMAT(I3,1X,I3,5X,I3,1X,I3)
    WRITE(1,106) IXS,IYS,IXE,IYE
    WRITE(6,112) IXS,IYS,IXE,IYE
    112 FORMAT(1H ,10X,4I5)
    20 CONTINUE
    IXS=999
    IYS=0
    IXE=0
    IYE=0
    WRITE(1,106) IXS,IYS,IXE,IYE
    WRITE(6,112) IXS,IYS,IXE,IYE
    IXS=0
    KTREC=KTREC + KEND + 2
    GO TO 10
C
C END OF CARD INPUT. TERMINATE RUN.
C
    900 ENDFILE 1
    REWIND 1
    WRITE(6,114) LAST,KTREC
    114 FORMAT(1H0,5X,*END-OF-RUN. NBR OF LAST BOARD ON FILE =*I4,* TOTAL
    1 RECS WRITTEN ON FILE =*I5)
    CALL EXIT
    END

```


Erickson, Bernard J., and Donald C. Markstrom.

1972. Predicting softwood cutting yield by computer. USDA
For. Serv. Res. Pap. RM-98, 15 p. Rocky Mt. For. and Range
Exp. Stn., Fort Collins, Colo.

A computer program, written in FORTRAN predicts the maximum yield of cuttings for softwood cut-up and edge- and end-gluing operation. The program calculates cutting recovery (given cutting width and length constraints and defect locations on the board), and locates ripping saw kerfs.

Keywords: Programming (computers), softwood cut-up, end and edge gluing, dimension stock.

Erickson, Bernard J., and Donald C. Markstrom.

1972. Predicting softwood cutting yield by computer. USDA
For. Serv. Res. Pap. RM-98, 15 p. Rocky Mt. For. and Range
Exp. Stn., Fort Collins, Colo.

A computer program, written in FORTRAN predicts the maximum yield of cuttings for softwood cut-up and edge- and end-gluing operation. The program calculates cutting recovery (given cutting width and length constraints and defect locations on the board), and locates ripping saw kerfs.

Keywords: Programming (computers), softwood cut-up, end and edge gluing, dimension stock.

Erickson, Bernard J., and Donald C. Markstrom.

1972. Predicting softwood cutting yield by computer. USDA
For. Serv. Res. Pap. RM-98, 15 p. Rocky Mt. For. and Range
Exp. Stn., Fort Collins, Colo.

A computer program, written in FORTRAN predicts the maximum yield of cuttings for softwood cut-up and edge- and end-gluing operation. The program calculates cutting recovery (given cutting width and length constraints and defect locations on the board), and locates ripping saw kerfs.

Keywords: Programming (computers), softwood cut-up, end and edge gluing, dimension stock.

Erickson, Bernard J., and Donald C. Markstrom.

1972. Predicting softwood cutting yield by computer. USDA
For. Serv. Res. Pap. RM-98, 15 p. Rocky Mt. For. and Range
Exp. Stn., Fort Collins, Colo.

A computer program, written in FORTRAN predicts the maximum yield of cuttings for softwood cut-up and edge- and end-gluing operation. The program calculates cutting recovery (given cutting width and length constraints and defect locations on the board), and locates ripping saw kerfs.

Keywords: Programming (computers), softwood cut-up, end and edge gluing, dimension stock.