

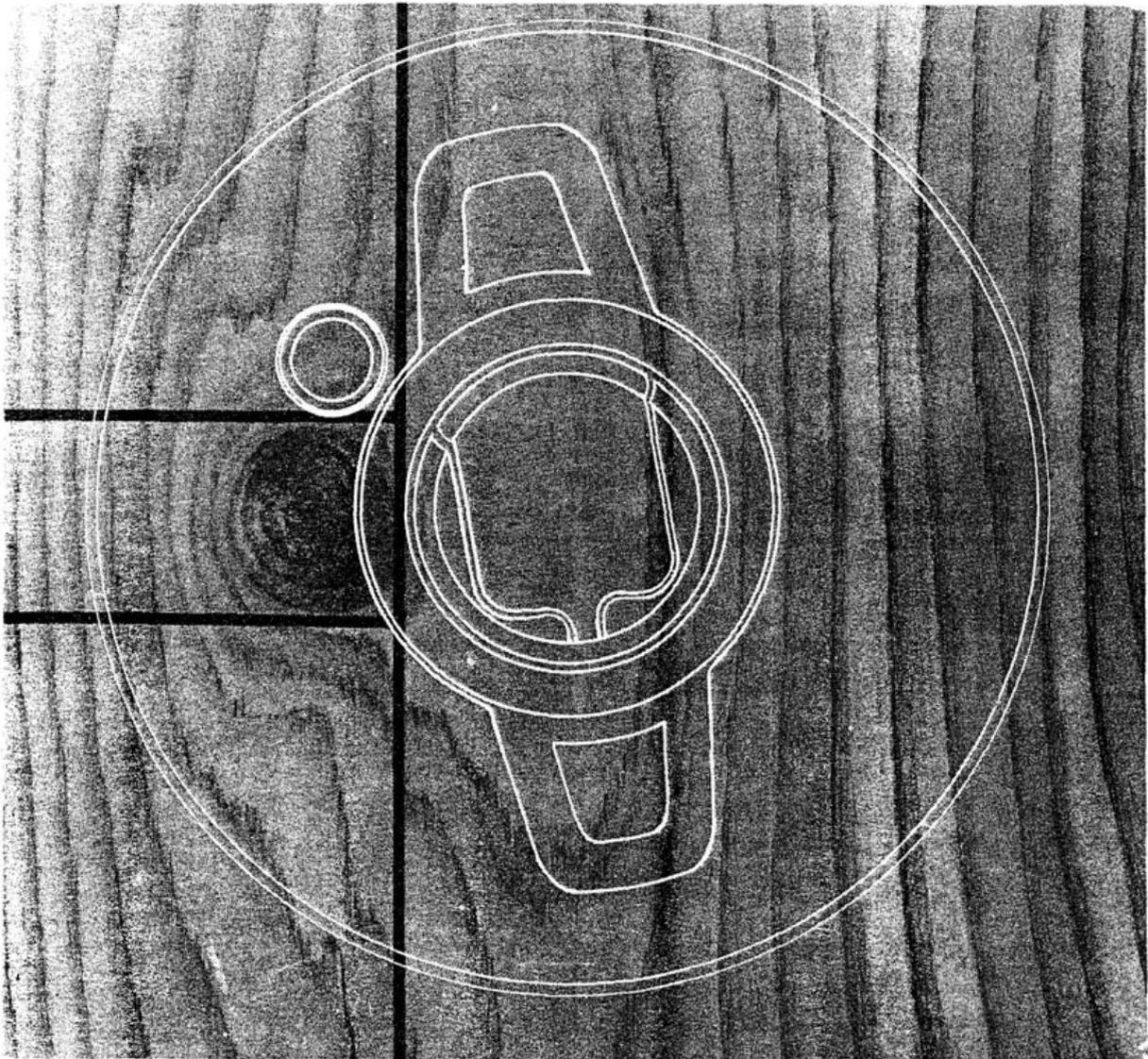
# **COMPUTER PROGRAM FOR GRADING HARDWOOD LUMBER**

*Prepared by*

**LYNN GALIGER  
and  
HIRAM HALLOCK**

**JULY 1971**

**U.S. Department of Agriculture \* Forest Service \* Forest Products Laboratory \* Madison, Wis.**



#### ACKNOWLEDGMENT

The authors acknowledge the work of Claudia Wodzinski for her part in setting up general procedures in the program, writing subroutine MCA and also coding the parts of subroutine BCC which form combinations of cuttings and resolve overlap between combinations of two cuttings.

It is recognized that all possible alternatives of the program have not been tested by execution.

COMPUTER PROGRAM FOR GRADING HARDWOOD LUMBER

By

Lynn Galiger, Mathematician  
Hiram Hallock, Forest Products Technologist

Forest Products Laboratory,<sup>1</sup> Forest Service  
U.S. Department of Agriculture

AN EXPLANATION--

Information contained in this publication should be used along with USDA Forest Service Research Paper FPL 157, "Grading Hardwood Lumber by Computer." (FPL 157 is available upon request from the Forest Products Laboratory, Madison, Wis.)

Research Paper FPL 157 describes the background and development of this information. This current publication goes on to detail the variables used in the program, lists program GRADE, and uses flow-charts to illustrate the program and subroutines.

---

<sup>1</sup> The Laboratory is maintained at Madison, Wis., in cooperation with the University of Wisconsin.

## VARIABLES

Most of the variables used in the program are defined in this section, Since a variable may be used in more than one area of the program with more than one meaning, the list of variables has been divided into three major groups. The variables in GRADE are further subdivided into two groups according to use. The variables in subroutine BCC are subdivided into four groups. Each group of variables identifies the area of the program in which the variables will be found.

Variables with a single meaning that are used in GRADE, MCA, and BCC are defined below:

ACL - array in which the minimum lengths permitted for a given potential grade are stored  
ACW - array in which the minimum widths permitted for a given potential grade are stored  
BL - board length  
BLX - upper x coordinate of the board  
BLY - upper y coordinate of the board  
BUX - lower x coordinate of the board  
BUY - lower y coordinate of the board  
BW - board width  
COMP - number of cutting units required for a combination of n cuttings to grade  
FG - array for storing the grade of each face of the board  
FLX - lower x coordinate of a defect  
FLY - lower y coordinate of a defect  
FUX - upper x coordinate of a defect  
FUY - upper y coordinate of a defect  
GCL - array that stores the minimum lengths for all grades  
GCW - array that stores the minimum widths for all grades  
IND = 0, a solution has not been found  
= 1, a solution has been found  
MNC - maximum number of cuttings permitted  
NBN - board number  
ND - number of defects  
NDT - temporary storage of the original number of defects  
NF - face number (1 or 2)  
NFD - array for storing the defects of each face  
NPG - number of the potential grade  
NTI - limits the number of times BCC is executed to 2 for each potential grade  
SM - surface measure

## VARIABLES USED IN GRADE

### Variables Used Only in the Defect Check

The variables defined in a specific way for their use in the defect check routine are listed below:

A50 - fifty percent of the area in the end foot  
BLHH - one half the length of the board  
EFA - area of the end foot  
EFAL - area of the defects in the left end foot  
EFAR - area of the defects in the right end foot  
EFL - upper x coordinate of the left end foot  
EFR - lower x coordinate of the right end foot  
ICK - number of knot defects  
ICP - number of pith defects  
ICS - number of split and shake defects  
ICW - number of wane defects  
ID - defect code number  
KID - array for storing knot defects  
PID - array for storing pith defects  
PL - total length of splits and shakes permitted  
SID - array for storing split and shake defects  
SL - total length of splits and shakes  
WID - array for storing wane defects  
WOB - total length of the wane on the bottom edge of the board  
WOT - total length of the wane on the top edge of the board

### Variables Used Only in the "Quick Check"

The variables defined in a specific way for their use in the defect check routine are listed below:

BCA - cutting units in a single cutting or a combination of cuttings which meet the requirements for the next lower grade  
BC1 - cutting units in the largest single cutting which meet the requirements for the next lower grade  
BC2 - cutting units in the best resolved combination of two cuttings which meet the requirements for the next lower grade  
BC3 - cutting units in the best resolved combination of three cuttings which meet the requirements for the next lower grade  
BC4 - cutting units in the best resolved combination of four cuttings which meet the requirements for the next lower grade  
CCOMP - cutting units required to grade the next lower grade  
COMP3 - a recalculation of CCOMP due to an increase in the number of cuttings in the combination

## MCA VARIABLES

The variables defined in a specific way for their use in the subroutine MCA are listed below.

- AMG - upper x coordinate of the clear area
- BMG - lower x coordinate of the clear area
- CA - array for storing the area of cuttings in cutting units
- CL - array for storing the length of cuttings
- CW - array for storing the width of cuttings
- FI - lower x coordinate of the defect whose upper y is the lower boundary of the clear area
- FJ - lower x coordinate of the defect whose lower y is the upper boundary of the clear area
- FX - length of a clear area
- N - width of a clear area
- IC - index of the defect whose upper y is the lower boundary of the clear area
- IDM - total number of indices stored in either the IOX or IOY array
- II - index used to reference either x's or y's when the defects are being ordered in the IOX or IOY arrays respectively
- IK - defect subscript used as an index of the W array to scan y coordinates
- IOX - array in which the lower and upper x coordinates are ordered together and stored; this array name is used to order the y's in the IOY array
- IOY - array in which the lower and upper y coordinates are ordered together and stored
- IW - used to set up the subscript for storing ordered defect coordinates in either the IOX or IOY array
  - defect subscript used as an index of the W array to scan x coordinates
- IX - subscript in the IOY array which indicates the lower y boundary of the clear area
- IY - subscript in the IOY array which indicates the upper y boundary of the clear area
- JA - subscript of the IOY array for a defect which is a potential upper y of the clear area
- JC - subscript of the defect whose lower y is the upper boundary of the clear area
- JD - subscript of the IOY array used to scan all y coordinates for clear area
- K - index for positioning defect scanning in either the x or y coordinates using the FLX array name
- KF - subscript of the IOX array for the left boundary of the clear area
- KI - subscript of the FUY array of the defect to be checked for its relative y position to the clear area
- KK - index for positioning defect storage in either the IOX or IOY array using the IOX array name
- LF - subscript of the IOX array for the right boundary of the clear area
- MNEB = 0, a single cutting large enough to meet the grading requirements has not been found
  - ≥ 1, the above-mentioned cutting was found
- NC - the number of cuttings in the board
- PW - relative position of defects to the clear area with respect to their x coordinates

SLX - array for storing the lower x coordinate of a cutting with sufficient cutting units to grade  
 SLY - array for storing the lower y coordinate of a cutting with sufficient cutting units to grade  
 SUX - array for storing the upper x coordinate of a cutting with sufficient cutting units to grade  
 SUY - array for storing the upper y coordinate of a cutting with sufficient cutting units to grade  
 TLX - array for storing the lower x coordinate of cuttings  
 TLY - array for storing the lower y coordinate of cuttings  
 TUX - array for storing the upper x coordinate of cuttings  
 TUY - array for storing the upper y coordinate of cuttings  
 UI - upper x coordinate of the defect which bounds the clear area on the bottom  
 UJ - upper x coordinate of the defect which bounds the clear area on the top  
 W - array for checking the relative position of defects to the clear area  
 WGT - area of the cutting in cutting units

### BCC VARIABLES

The variables with a single meaning that are used in more than one routine in BCC are defined below:

CU - array for temporary storage of the number of cutting units in a resolved combination of cuttings  
 IJ - lowest numbered cutting in a combination  
 IK - second lowest numbered cutting in a Combination  
 IXYZ - index which identifies the exit of a four-cutting combination from the two-cutting and three-cutting routine  
 KOUNT - array for storing the total number of combinations of n cuttings available for resolution  
 NS1 = 0, a no-solution condition has not been determined  
       = 1, a no-solution condition has been determined  
 OL - array in which the overlap area of all pairs of cuttings are stored  
 RCO = 0, overlap recalculation loop is being used by a three-cutting classification 2 combination  
       = 1, overlap for a pair of cuttings being processed by solution type 11, alternate crosscut routine, is being calculated  
 T = 4 (used to convert width measurements to inches)  
       = 48 (used to convert length measurements to feet)  
 X - temporary storage of the length of the overlap in feet  
 XZ - length or width of the overlap depending upon whether x's or y's are being used in the calculation

### Combination-Forming Routine

The variables defined in a specific way for their use in the combination-forming routine are listed below.

AI - array which stores the total number of cutting units for a combination after the overlap is subtracted

I - index of the first cutting in a combination being formed  
 IA - array which stores all combinations of cuttings with enough cutting units to be considered for resolution  
 IB - array used to temporarily store the numbers of the cuttings being formed into combinations  
 IJ - number of cuttings permitted in the combination being formed  
 IL - subscript in the cutting arrays which indicates the first cutting in the combination  
 IM - subscript in the IB array which represents the first cutting in the combination  
 IN - index which represents the second cutting in the combination  
 IOL - subscript of the cutting arrays which indicates the cutting in the middle of a combination of three cuttings with triple overlap  
 IX - index used to get into the proper area of the IA array  
 J - index of the second cutting in a combination being formed  
 JI - subscript of the overlap array which indicates the first cutting in a combination  
 JK - subscript of the IB array which indicates the second cutting in a combination  
 JL - subscript of the overlap array which indicates the second cutting in a combination  
 JM = 0, index used to reference the TLX array  
       = 25, index used to reference the TLY array  
 JN - subscript of the cutting arrays which indicates the second cutting in a combination  
 K - index of the third cutting in a combination being formed  
 L - index of the fourth cutting in a combination being formed  
 LC - the number of combinations of n cuttings stored in the IA array at a given time  
 LK - identifies each cutting in a combination of n cuttings which must be moved in the ordering of the IA array  
 LM - section of the IA array to be referenced for a given combination  
 LN - section of the AI array to be referenced for a given combination  
 M - index of the fifth cutting in a combination being formed  
 MI - position of a cutting within a section of the IA array  
 MJ - index of the seventh cutting in a combination being formed  
 N - index of the sixth cutting in a combination being formed  
 NK - number of the second cutting in a pair of cuttings referenced by the overlap array  
 NL - subscript of the IB array for the second cutting referenced by the overlap array  
 NM - number of the second cutting in the pair of cuttings which has the smallest overlap in a combination of three cuttings with triple overlap  
 NN - number of the first cutting in the pair of cuttings which has the smallest overlap in a combination of three cuttings with triple overlap  
 OIL - number of pairs of cuttings which overlap in a combination of three or more cuttings  
 OLL - length or width of the overlap of two cuttings depending upon whether x's or y's are being used in the calculation  
 ZCA - area of the triple overlap of a combination of three cuttings  
 ZLX - lower x coordinate of the triple overlap area  
 ZLY - lower y coordinate of the triple overlap area

ZUX - upper x coordinate of the triple overlap area  
ZUY - upper y coordinate of the triple overlap area

### Two-Cutting Routine

The variables defined in a specific way for their use in the two-cutting routine for resolving overlap are listed below.

AA - area of one cutting lost by crosscutting  
AB - area of the other cutting lost by crosscutting  
AC - area of one cutting lost by ripping  
AD - area of the other cutting lost by ripping  
AF - area of one cutting resolved by the alternate crosscut method in solution type 11  
AG - area of the other cutting resolved by the alternate crosscut method in solution type 11  
AH - total area lost in both cuttings by the alternate crosscut method in solution type 11  
CHECK - ACW, ACL, WMIN, or XLMIN value used to compute WX  
EXL - length of the cutting which would benefit by the execution of the alternate crosscut method in solution type 11  
II - position of crosscut  
    = 1, to the left of or at the left edge of the overlap  
    = 2, within the overlap with the narrow cutting on the left  
    = 3, within the overlap with the narrow cutting on the right  
    = 4, to the right of or at the right edge of the overlap  
IJJ - first of two cuttings being resolved by solution types 12, 13, and 14  
IKK - second of two cuttings being resolved by solution types 12, 13, and 14  
INI - limits the number of cuttings processed by the two-cutting routine  
JJ - position of rip  
    = 1, below or at the lower edge of the overlap  
    = 2, within the overlap with the short cutting below  
    = 3, within the overlap with the short cutting above  
    = 4, above or at the upper edge of the overlap  
K - references the proper constant stored in the ACW array  
    - width of the first cutting in a combination (see L)  
KK - controls processing of solution types 10 and 11  
    = 1, crosscut routine should be executed  
    = 2, areas lost by placing the rip and crosscut should be calculated and then compared  
L - width of the second cutting in a combination  
    = 1, the smaller minimum width permitted  
    = 2, the larger minimum width permitted  
M - index used to reference either the cutting length or cutting width array when the rip or crosscut routine is executed  
    = 0, cutting length array is used  
    = 25, cutting width array is used  
MM - subscript of the SAVE array for solution types less than or equal to 11  
    = 1, cutting IK is on top  
    = 2, cutting IK is on the bottom

MMS - temporary storage of MM  
MN - subscript of the SAVE array for solution types greater than 11  
    = 1, cutting IK is the narrow cutting  
    = 2, cutting IK is not the narrow cutting  
N = 1, a rip resolution  
    = 2, a crosscut resolution  
NCA - number of the cutting on the right for a crosscut  
    - number of the cutting above for a rip  
NCAS - temporary storage of NCA  
NCB - number of the cutting on the left for a crosscut  
    - number of the cutting below for a rip  
NCBS - temporary storage of NCB  
NLC - number of the longer cutting  
NM - subscript of the SAVE array for solution types greater than 11  
    = 1, cutting IJ is the narrow cutting  
    = 2, cutting IJ is not the narrow cutting  
NN - subscript of the SAVE array for solution types less than or equal to 11  
    = 1, cutting IJ is on the bottom  
    = 2, cutting IJ is on top  
NNC - number of the narrow cutting  
NNS - temporary storage for NN  
NSC - number of the short cutting  
NST - number of the solution type  
NTYPE - classification of two-cutting overlap  
    = 1, neither cutting separates the other two into two nonoverlapping  
    pieces with respect to either the entire length or width of that  
    cutting  
    = 2, one or both cuttings will separate the other cutting into two  
    nonoverlapping pieces with respect to either the entire length or  
    width of that cutting  
NWC - number of the wide cutting  
SAVE - array for storing the original cutting coordinates temporarily that  
    are redefined due to the overlap resolution  
TA - cutting units in a resolved combination of two cuttings  
TL - total length of the two overlapping cuttings  
WMIN - minimum width  
WR - length or width of the remaining cutting once the coordinate of the cross-  
    cut or rip is determined  
WX - array for storing the coordinate of the crosscut or rip  
XL - difference of the lower x coordinates of cuttings IJ and IK  
XLMIN - minimum length  
XU - difference of the upper x coordinates of cuttings IJ and IK  
YL - difference of the lower y coordinates of cuttings IJ and IK  
YU - difference of the upper y coordinates of cuttings IJ and IK

### Three-Cutting Routine

The variables defined in a specific way for their use in the three-cutting routine for resolving overlap are listed below.

A1 - indicates which combination of two cuttings has no overlap  
 AREA1 - area of the cutting IE which has been reduced by ripping due to the placement of the minimum-size cutting  
 AREA2 - area of the cutting IE which has been reduced by crosscutting due to the placement of the minimum-size cutting  
 AREA3 - area of the cutting IH which has been reduced by ripping due to the placement of the minimum-size cutting  
 AREA4 - area of the cutting IH which has been reduced by crosscutting due to the placement of the minimum-size cutting  
 AT - number of cutting units in three-cutting classification 1 or classification 3 resolution  
 C3 = 0, a combination of three cuttings is not being processed  
     = 1, a combination of three cuttings is being processed or a combination of three cuttings from a four-cutting classification 2 combination is being processed  
 CAN - cutting units in a combination of three cuttings which has been resolved by the New Routine  
 CAO - cutting units in a combination of three cuttings which has been resolved by the Old Routine  
 CB = 0, New Routine was not executed  
     = 1, New Routine was executed and the result should be compared with the result of the Old Routine  
     = 2, only the Old Routine has a resolution to be considered  
 CHN = 0, the first ten cuttings are restored to the TLX arrays from the WLX arrays  
     = 1, the three original cuttings are restored in the three-cutting combination to the TLX arrays  
 CN = 1, a three-cutting classification 2 combination will have a pair of original cuttings resolved by the two-cutting routine  
     = 2, a three-cutting classification 2 combination will select one of three pairs of cuttings to be resolved by the two-cutting routine  
 CN1 - number of the largest cutting found by the expansion pass of MCA  
 D1 - difference of the extreme upper y's in a classification 2 combination  
 D2 - difference of the extreme lower y's in a classification 2 combination  
 E = 0, cutting IE is at least minimum size  
     = 1, cutting IE is not at least minimum size  
 FX - length of cutting IG  
 FX1 - adjusted length of cutting IE or IH  
 FY1 - adjusted width of cutting IE or IH  
 H = 0, cutting IH is at least minimum size  
     = 1, cutting IH is not at least minimum size  
 I3 - index for the three-cutting routine loop  
 IE - cutting which overlaps two cuttings in a three-cutting classification 1 combination  
     - cutting whose upper y is one of the two cuttings which determine the widest exterior area in a three-cutting classification 2 combination  
 IE1 - temporary assignment of the cutting to be IE in a classification 2 Combination whose upper y's are being checked for the exterior area  
 IE2 - temporary assignment of the cutting to be IE in a classification 2 combination whose lower y's are being checked for the exterior area

**IG** - cutting in which the minimum-size cutting is placed  
**IG1** - temporary IG - see IE1  
**IG2** - temporary IG - see IE2  
**IH** - remaining cutting in a combination of three cuttings  
**IH1** - temporary IH - see IE1  
**IH2** - temporary IH - see IE2  
**IJ1** - temporary IJ - see IE1  
**IJ2** - temporary IJ - see IE2  
**IK1** - temporary IK - see IE1  
**IK2** - temporary IK - see IE2  
**IKK** - remaining cutting in a combination of three cuttings  
**ILL** - index determines to which part of BCC the control should be returned after the minimum-size check on the adjusted cutting  
     = 1, cutting IE has been reduced by ripping  
     = 2, cutting LE has been reduced by crosscutting  
     = 3, cutting IH has been reduced by ripping  
     = 4, cutting IH has been reduced by crosscutting  
**K1** = 0, first pair of cuttings in the combination do not overlap  
     = 1, first pair of cuttings in the combination do overlap  
**K2** = 0, second pair of cuttings in the combination do not overlap  
     = 1, second pair of cuttings in the combination do overlap  
**K3** = 0, third pair of cuttings in the combination do not overlap  
     = 1, third pair of cuttings in the combination do overlap  
**K3C** - limits the number of combinations processed by the three-cutting routine  
**KB** - number of the middle cutting in the New Routine  
**L1** - type of three-cutting overlap  
     = 1 (classification 1--only one cutting overlaps two cuttings)  
     = 2 (classification 2--each cutting overlaps two cuttings)  
     = 3 (classification 3--one cutting does not overlap either of the other two cuttings)  
**M1** - limits the number of times the overlap calculation loop is executed for recalculation of overlap for adjusted cuttings  
**MSC** - refers to the adjusted cutting IH  
     = 0, neither rip nor crosscut adjustment is at least minimum size  
     = 1, rip adjustment is at least minimum size  
     = 2, crosscut adjustment is at least minimum size  
     = 3, both the rip and crosscut adjustments are at least minimum size  
**N1** = 0, all recalculation of overlap of adjusted cuttings is complete  
     = 1, overlap must be calculated if the residual of IG is a cutting of at least minimum size  
**N11** - first cutting in a triple overlapping combination  
**N12** - second cutting in a triple overlapping combination  
**N13** - third cutting in a triple overlapping combination  
**N1J** - subscript of the IB array which represents the first cutting  
**N1L** - subscript of the IB array which represents the second cutting  
**NCT** - temporary storage of the original number of cuttings found by MCA  
**NFF** - number of cuttings stored as defects  
     = 2, two cuttings are stored for a combination of three cuttings  
     = 3, three cuttings are stored for a combination of four cuttings  
**NK1** = 1, cutting IE has been adjusted  
     = 2, cutting IH has been adjusted  
     = 3, both cuttings have been adjusted

NL1 - indicates how cutting IE was reduced in order to place a minimum-size cutting within IG most advantageously  
 = 1, rip at +bottom of IE  
 = 2, rip at top of IE  
 = 3, crosscut at left end of IE  
 = 4, crosscut at right end of IE  
 When IE is being adjusted by ripping, NL1 is set equal to 1 or 2. Then the crosscut adjustment of IE is made and either 1 or 3 is added to the present value of NL1. The result is a sum of 2, 3, 4, or 5. Once a choice is made between ripping and crosscutting NL1 can be reset and stored in NLIS as 1, 2, 3, or 4 for later reference using the following information:  
 A sum of 2 indicates: rip--NL1 = 1, or crosscut--NL1 = 3  
 A sum of 3 indicates: rip--NL1 = 2, or crosscut--NL1 = 3  
 A sum of 4 indicates: rip--NL1 = 1, or crosscut--NL1 = 4  
 A sum of 5 indicates: rip--NL1 = 2, or crosscut--NL1 = 4  
 NL1 is then used in the same way for adjusting IH

NLIS - temporary storage of NL1

NN1 - number of the independent cutting in a classification 3 combination  
 - number of the expanded cutting in a classification 1 or 2 combination

NO1 = 0, cutting of IE and IH have been adjusted  
 = 1, cutting IE has not been adjusted  
 = 2, cutting IH has not been adjusted  
 = 3, cuttings IE and IH have not been adjusted

SPC - total number of cutting units in a classification 2 solution

T1 - relative position of cuttings IG and IE by value which is the difference of the upper y of IE and the lower y of IG

T2 - difference of the upper y of IG and the lower y of IE

T3 - difference of the upper x of IE and the lower x of IG

T4 - difference of the upper x of IG and the lower x of IE

WA - array for storing the area of cuttings used in the three-cutting routine

WL - array for storing the length of cuttings used in the three-cutting routine

WLX - array for storing the lower x coordinates of cuttings used in the three-cutting routine

WLY - array for storing the lower y coordinates of cuttings used in the three-cutting routine

WUX - array for storing the upper x coordinates of cuttings used in the three-cutting routine

WUY - array for storing the upper y coordinates of cuttings used in the three-cutting routine

WW - array for storing the width of cuttings used in the three-cutting routine

XMIN - minimum length

YMIN - minimum width

ZW = 0, solution combination has been found  
 = 1, resolution combination has insufficient cutting units to meet the grading requirements

## Four-Cutting Routine

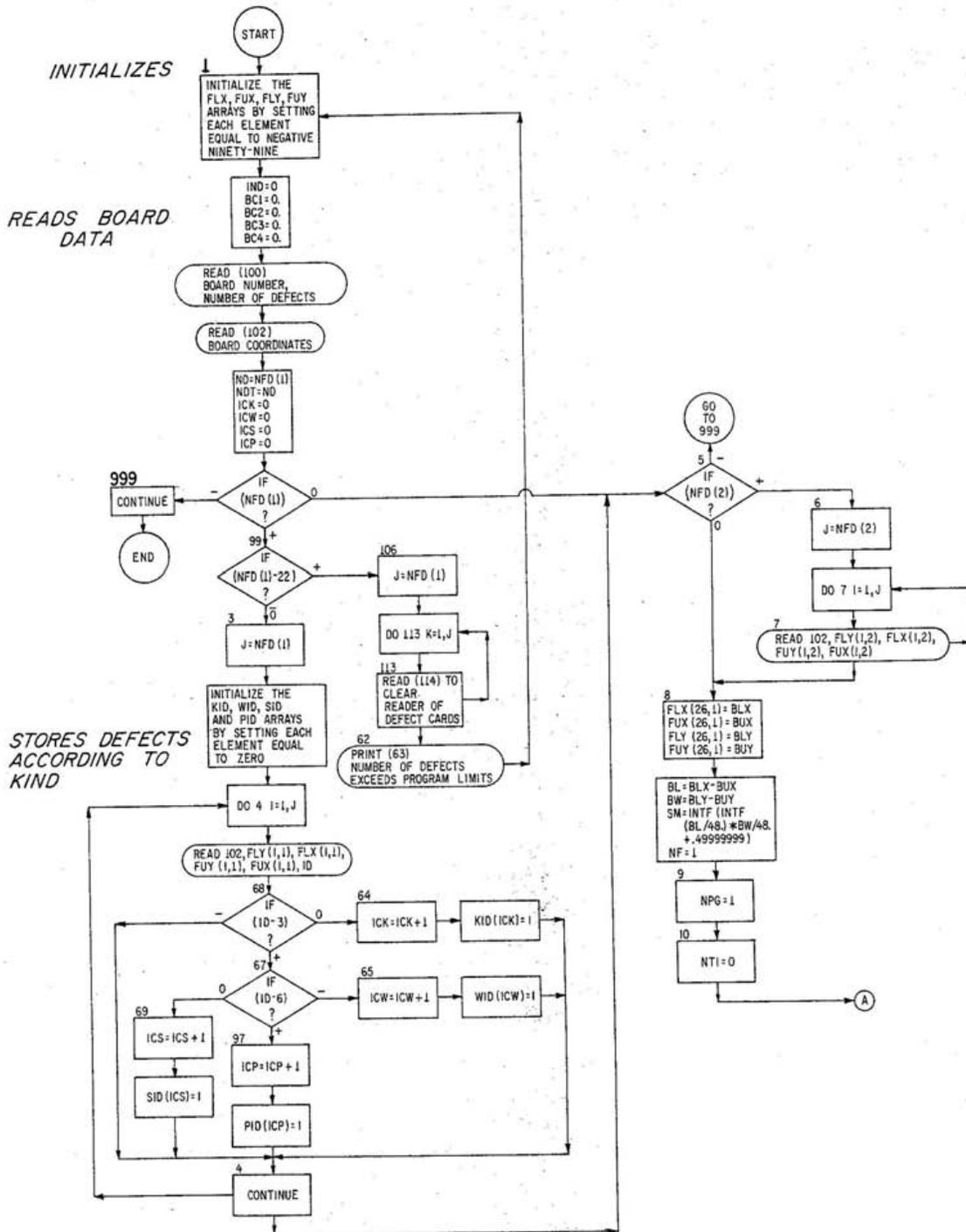
The variables defined in a specific way for their use in the four-cutting routine to resolve overlap are listed below.

- C4 = 0, combination of four cuttings is not being processed
- = 1, combination of four cuttings is being processed
- CR = 2, classification 1 combination is being processed
- = 3, classification 2 combination is being processed
- I4C - number of classification 1 combinations stored
- I4D - number of classification 2 combinations stored
- IAA - array in which classification 1 combinations are stored
- IAB - array in which classification 2 combinations are stored
- IC4T - total number of four cutting combinations stored
- ICC - number of pairs of overlapping cuttings in a given combination
- IDA - array for storing the independent cuttings in a classification 1 combination
- IDB - array for storing the independent cuttings in a classification 2 combination
- IDC - first independent cutting in a classification 1 or classification 2 combination
- IDD - second independent cutting in a classification 1 combination
- IKK - third lowest numbered cutting in a combination
- IQ - first cutting in a combination
- IQC - cuttings with which IQ overlaps; the constants 1, 2, and 4 are added to the counter for each cutting first through third, that is paired with IQ and overlaps it
- IR - second cutting in a combination
- IRC - cuttings with which IR overlaps (see IQC)
- IS - third cutting in a combination
- ISC - cuttings with which IS overlaps (see IQC)
- IT - fourth cutting in a combination
- ITC - cuttings with which IT overlaps (see IQC)
- ITT - subscript of the IDA array
- IU - temporary storage of an independent cutting in a combination
- IV - temporary storage of the second independent cutting in a combination
- JPI - number of classification 1 combinations available for resolution
- KPI - number of classification 2 combinations available for resolution
- M4 - types of four-cutting overlap
- = 1 (classification 1--the combination has two independent cuttings)
- = 2 (classification 2--the combination has one independent cutting)
- N4C - number of four-cutting combinations stored for resolution
- SC - independent cuttings in classification 1 combinations
- A constant is added for each cutting that is independent
- IQ = +1, IR = +2, IS = +4, IT = +7
- SC = 3, independent cuttings are IQ and IR
- = 5, independent cuttings are IQ and IS
- = 6, independent cuttings are IR and IS
- = 8, independent cuttings are IQ and IT
- = 9, independent cuttings are IR and IT
- = 11, independent cuttings are IS and IT

XCA - array for storing the area of cuttings used in the four-cutting routine  
XCL - array for storing the length of cuttings used in the four-cutting routine  
XCW - array for storing the width of cuttings used in the four-cutting routine  
XLX - array for storing the lower x coordinates of cuttings used in the four-cutting routine  
XLY - array for storing the lower y coordinates of cuttings used in the four-cutting routine  
XUX - array for storing the upper x coordinates of cuttings used in the four-cutting routine  
XUY - array for storing the upper y coordinates of cuttings used in the four-cutting routine

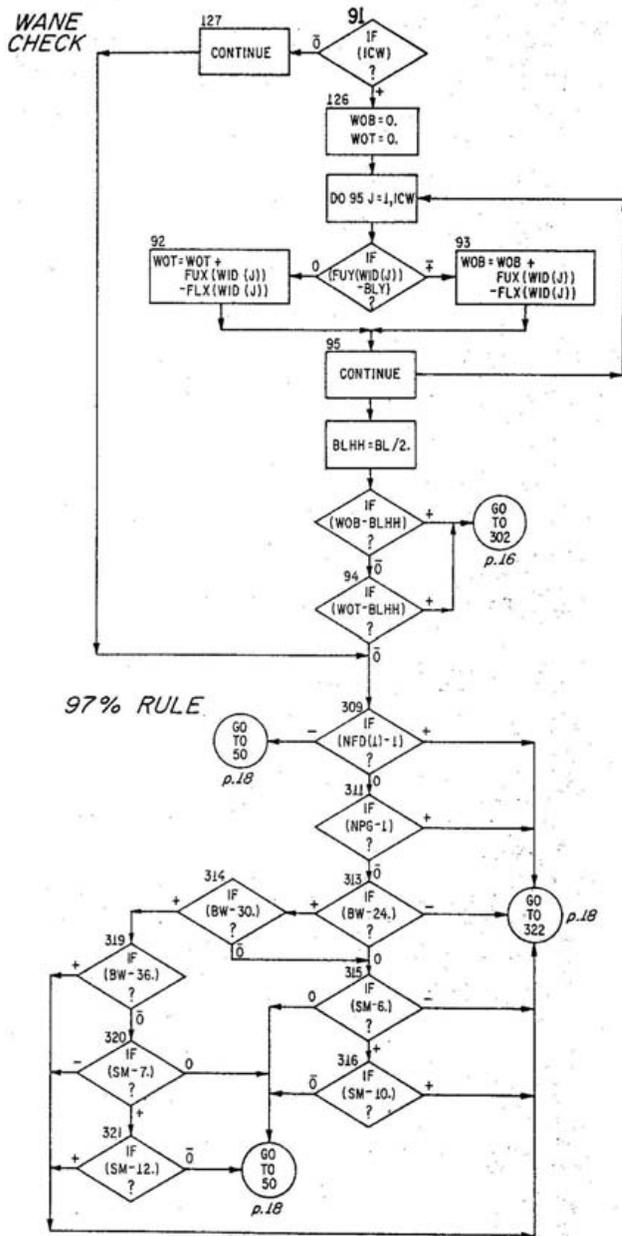
**PROGRAM GRADE ➡**

# MAIN PROGRAM

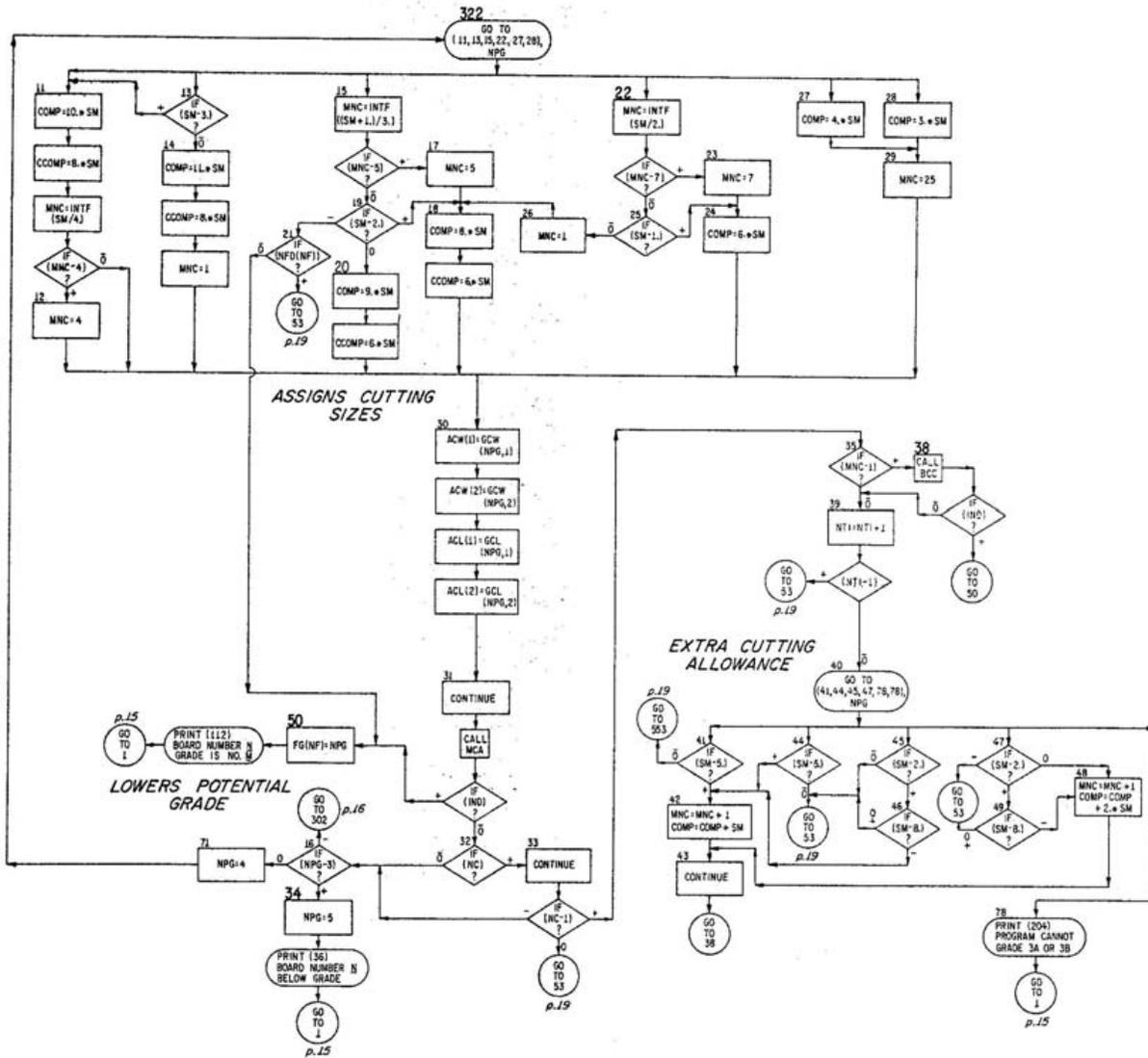




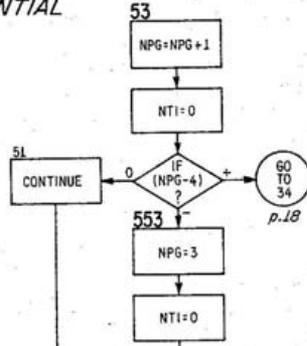
WANE CHECK



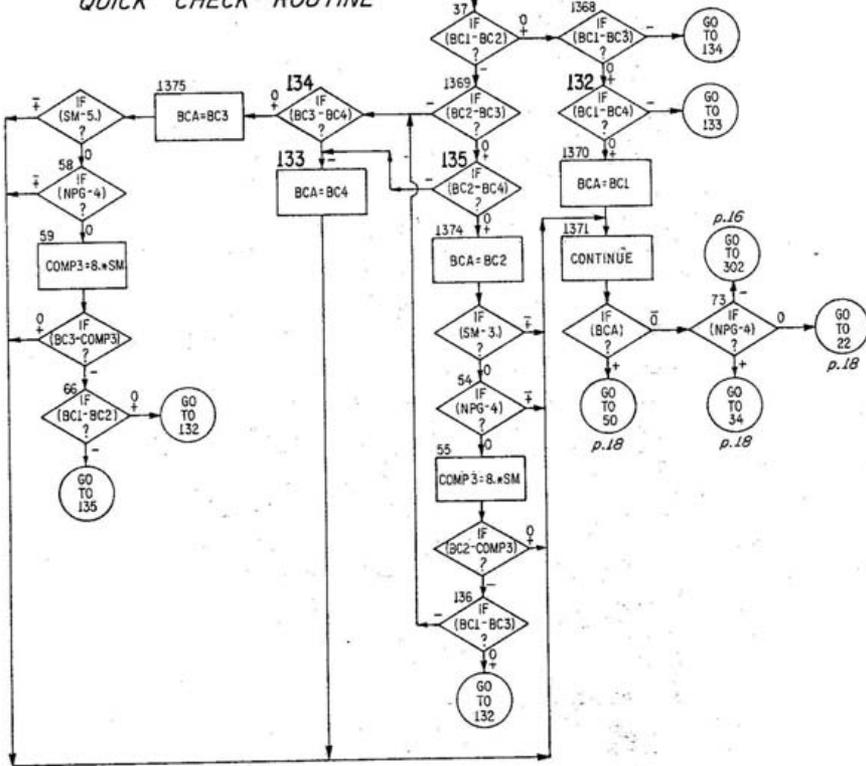
**CALCULATES THE NUMBER OF CUTTING UNITS  
REQUIRED AND THE NUMBER OF CUTTINGS PERMITTED  
FOR THE POTENTIAL GRADE ASSIGNED**



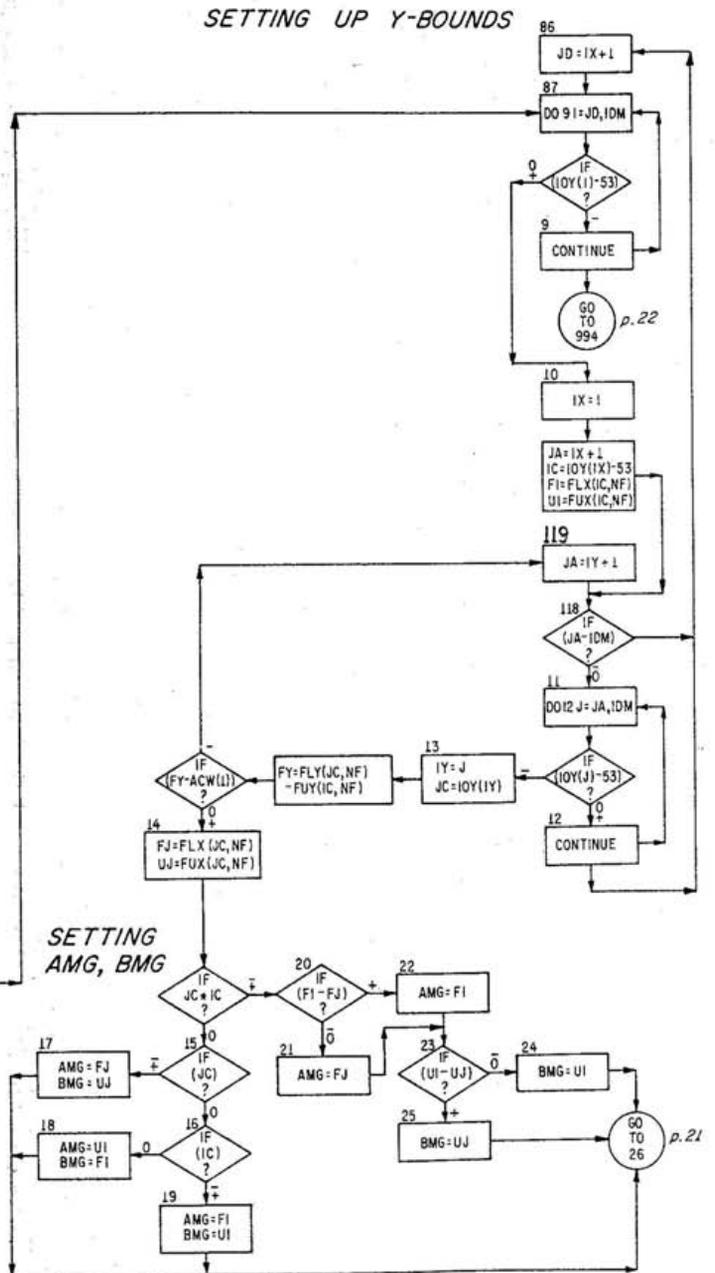
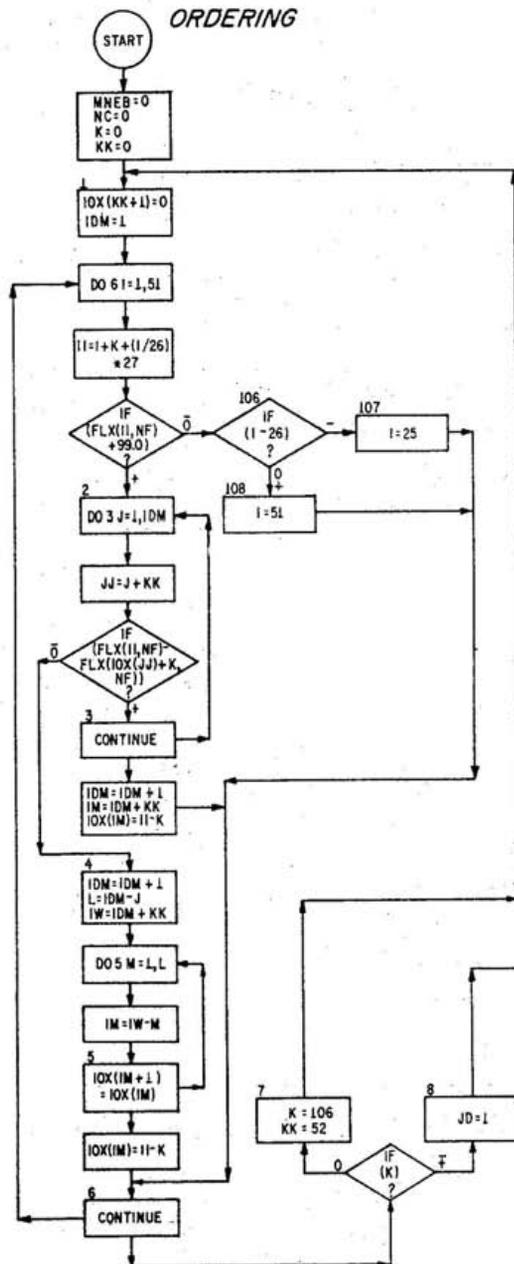
LOWERS POTENTIAL GRADE



QUICK CHECK ROUTINE



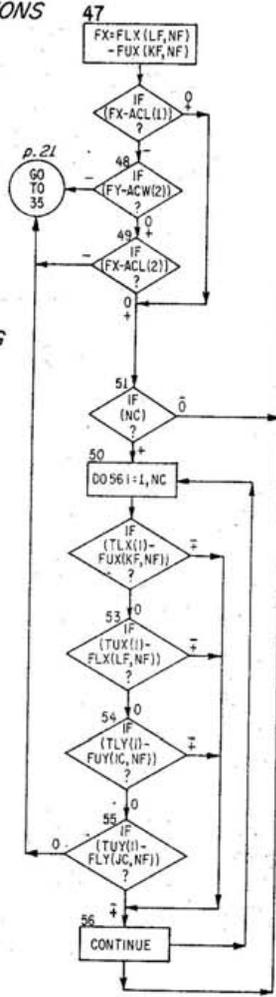
# SUBROUTINE MCA



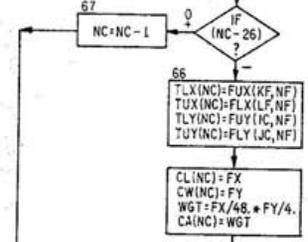


FINDING DIMENSIONS OF CUTTING

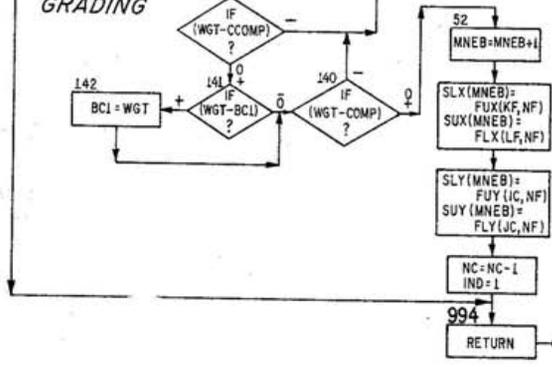
CHECKS TO SEE IF CUTTING HAS ALREADY BEEN FOUND



STORES CUTTING



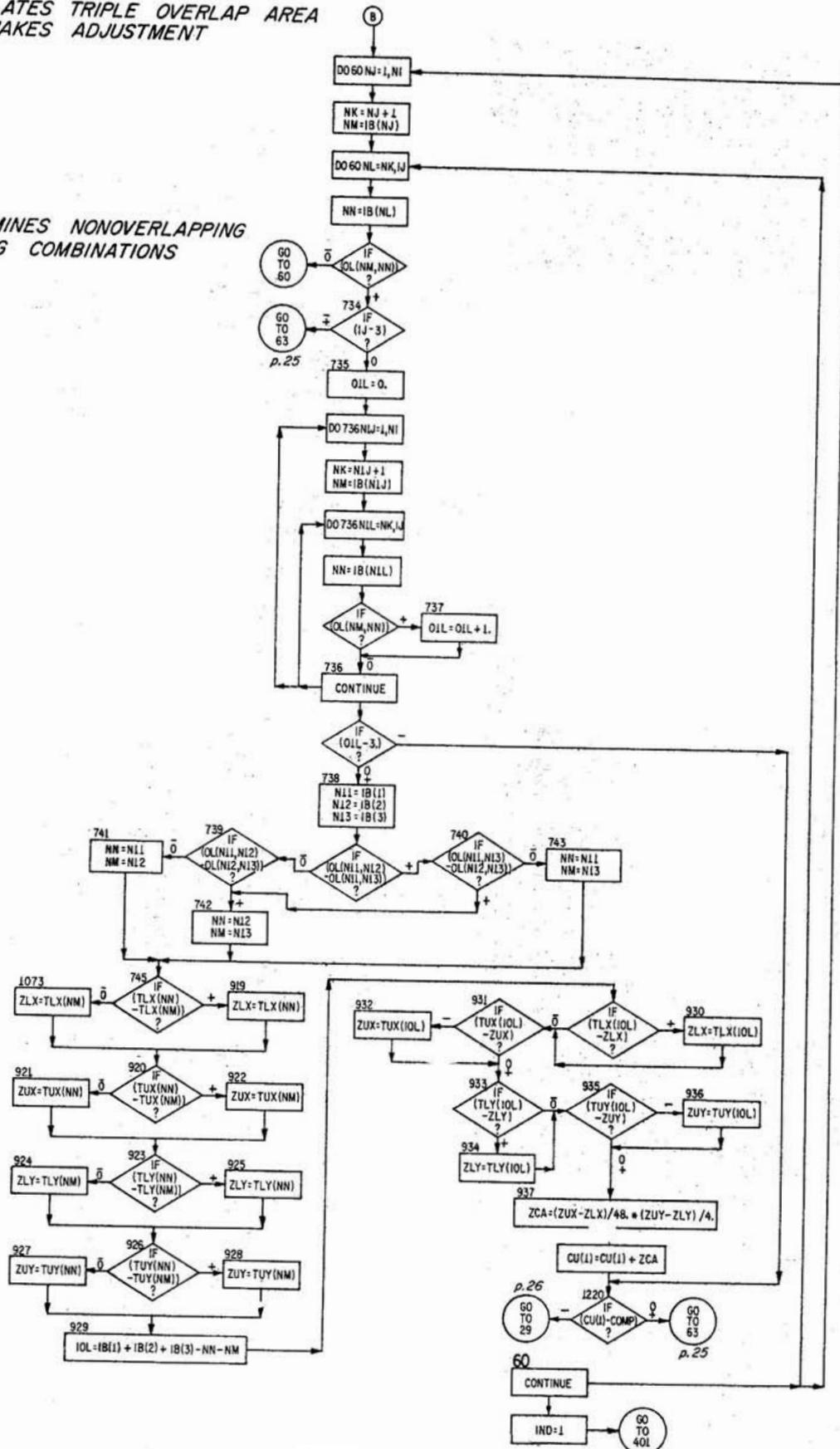
SINGLE CUTTING CHECK FOR GRADING



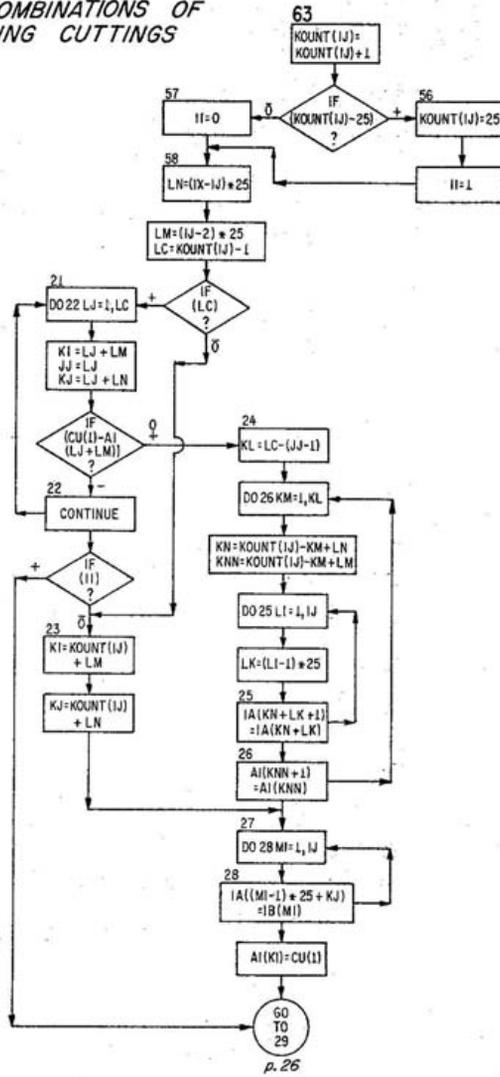


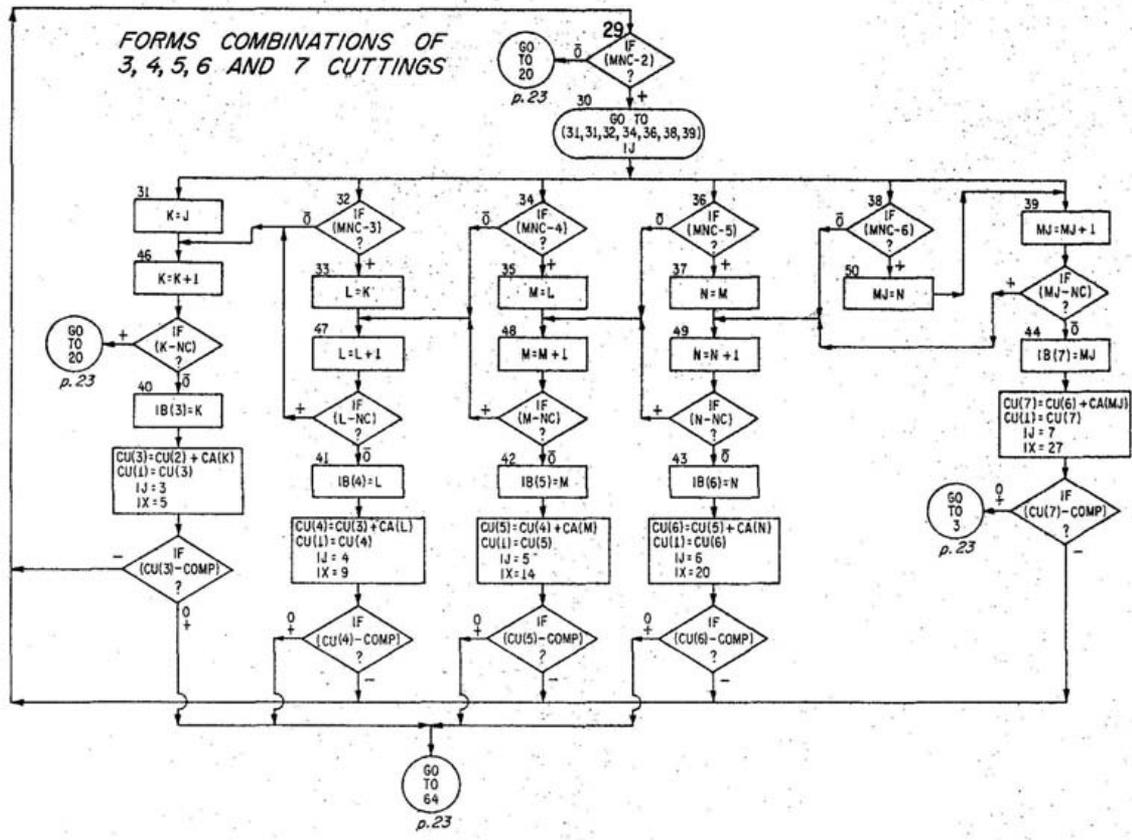
CALCULATES TRIPLE OVERLAP AREA  
AND MAKES ADJUSTMENT

DETERMINES NONOVERLAPPING  
CUTTING COMBINATIONS

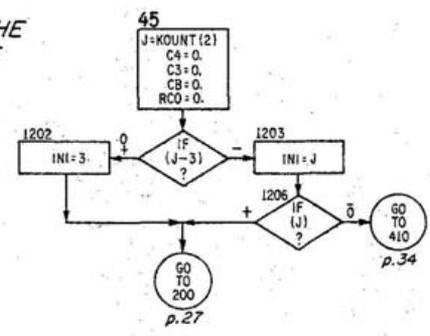


STORES COMBINATIONS OF  
OVERLAPPING CUTTINGS

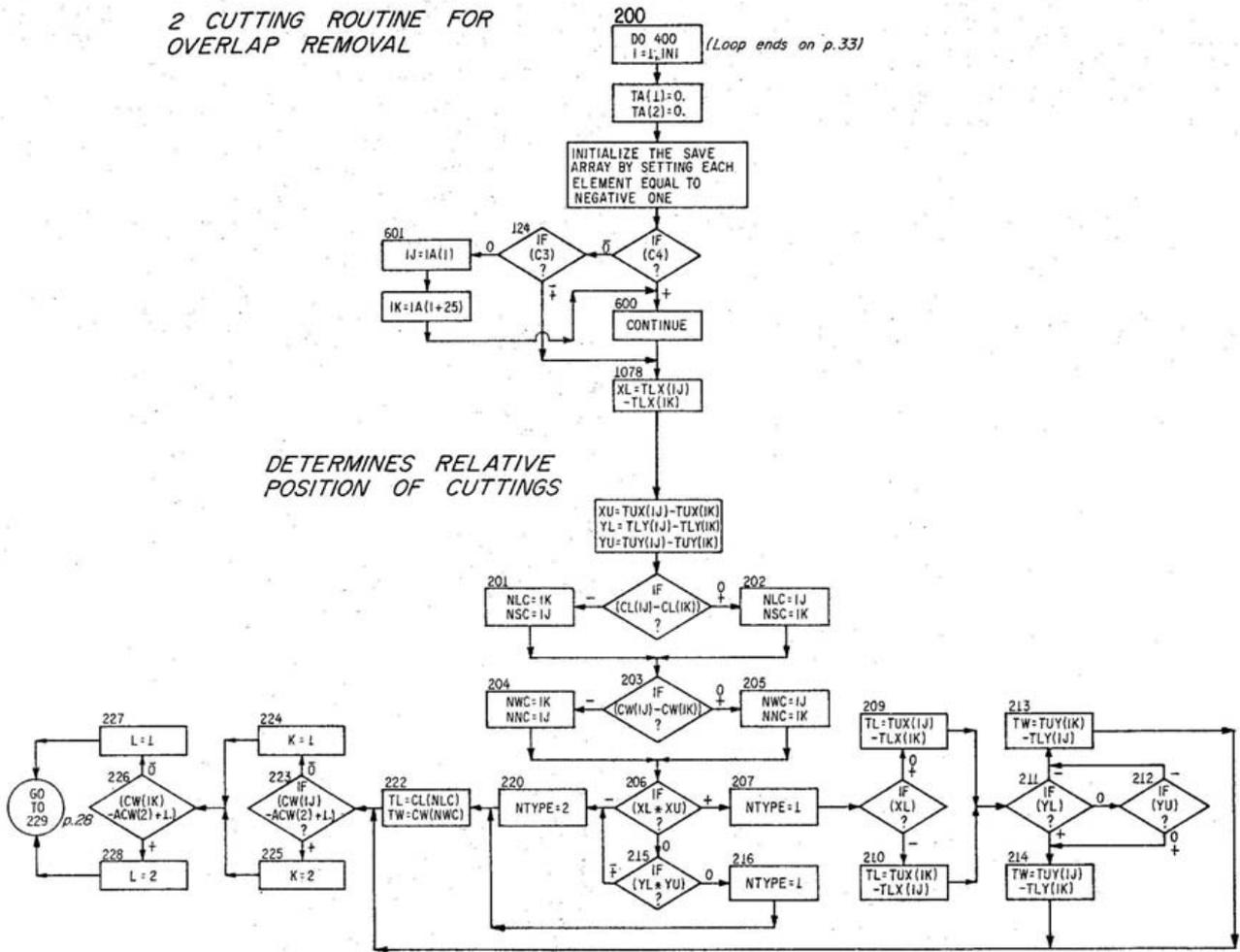




**INITIALIZES FOR THE  
2 CUTTING ROUTINE**

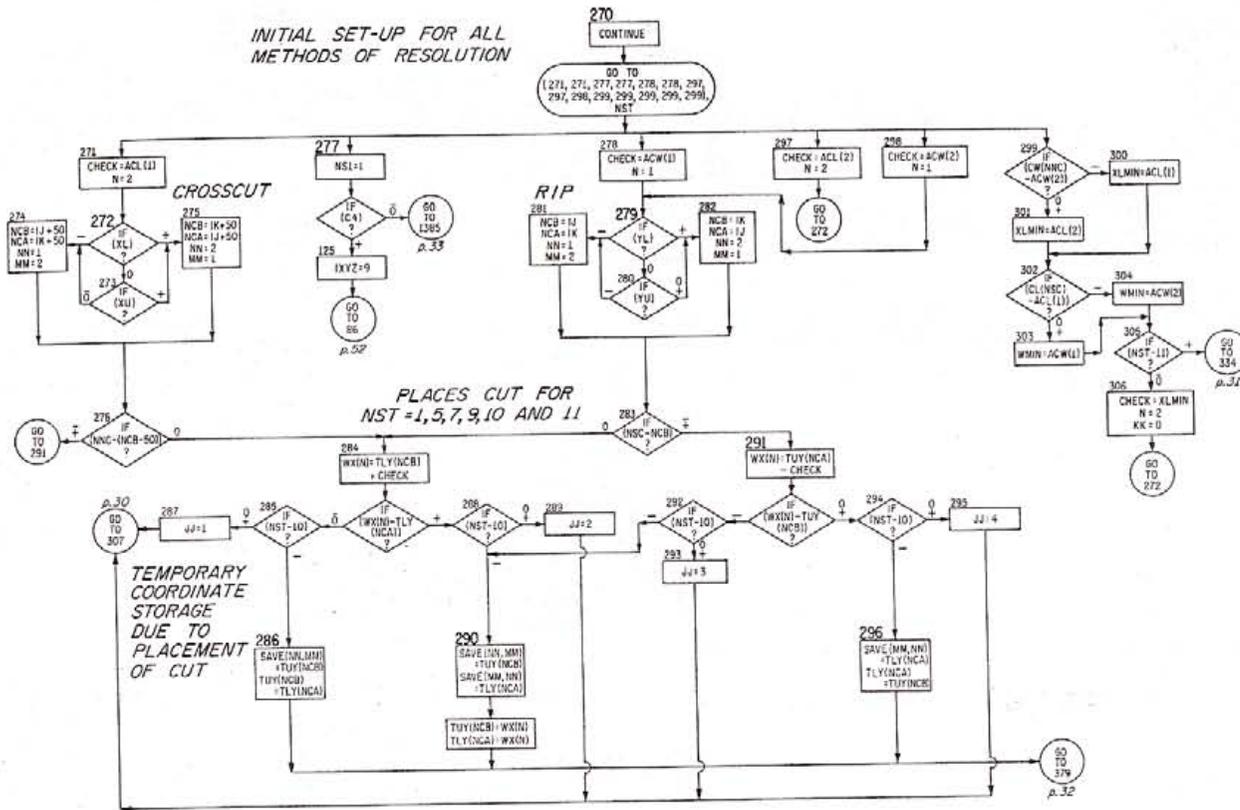


2 CUTTING ROUTINE FOR  
OVERLAP REMOVAL





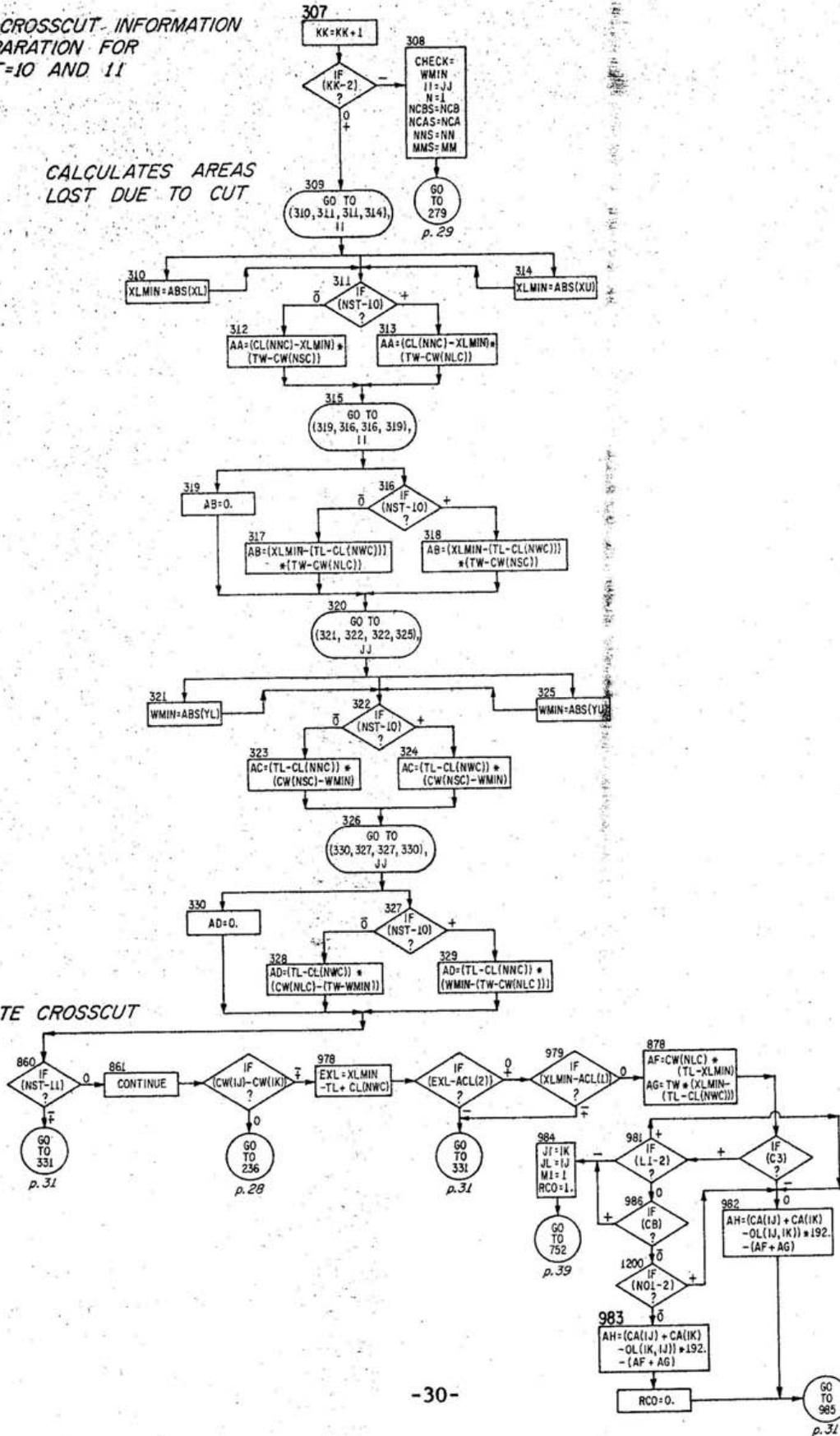
INITIAL SET-UP FOR ALL METHODS OF RESOLUTION



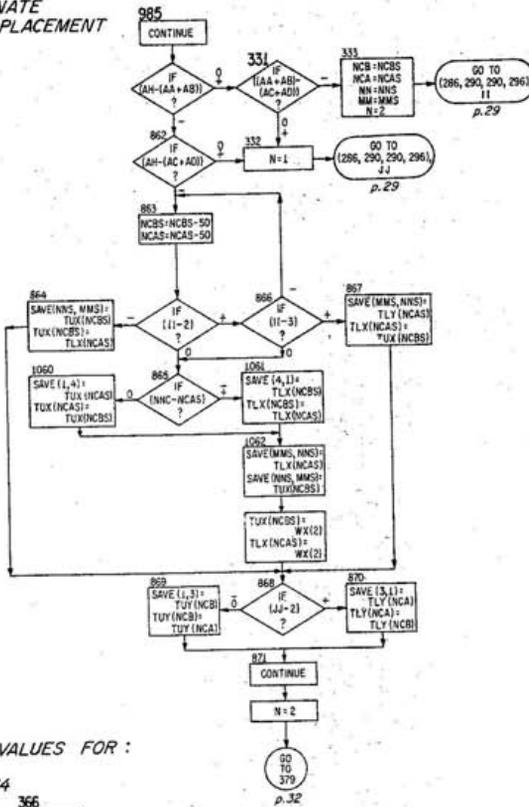
STORES CROSSCUT INFORMATION  
IN PREPARATION FOR  
RIP NST=10 AND 11

CALCULATES AREAS  
LOST DUE TO CUT

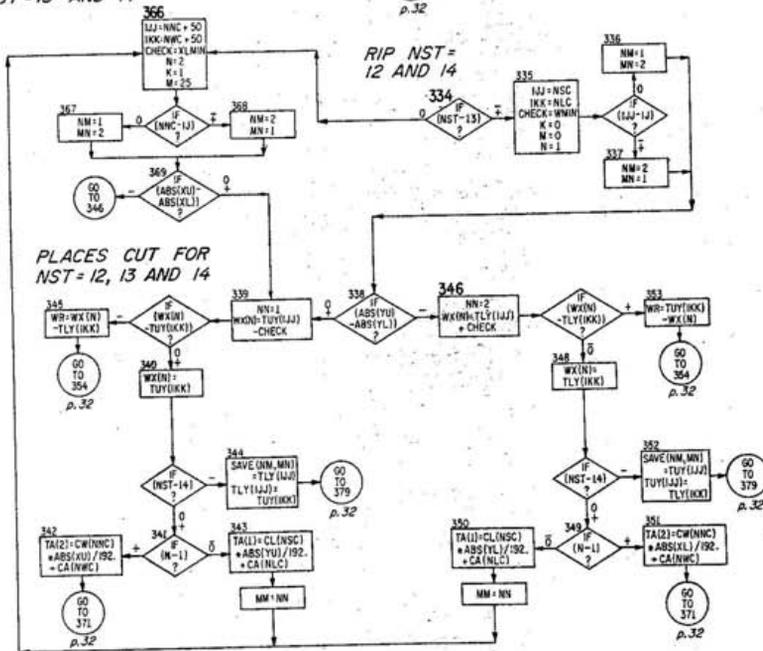
NST=11  
ALTERNATE CROSSCUT

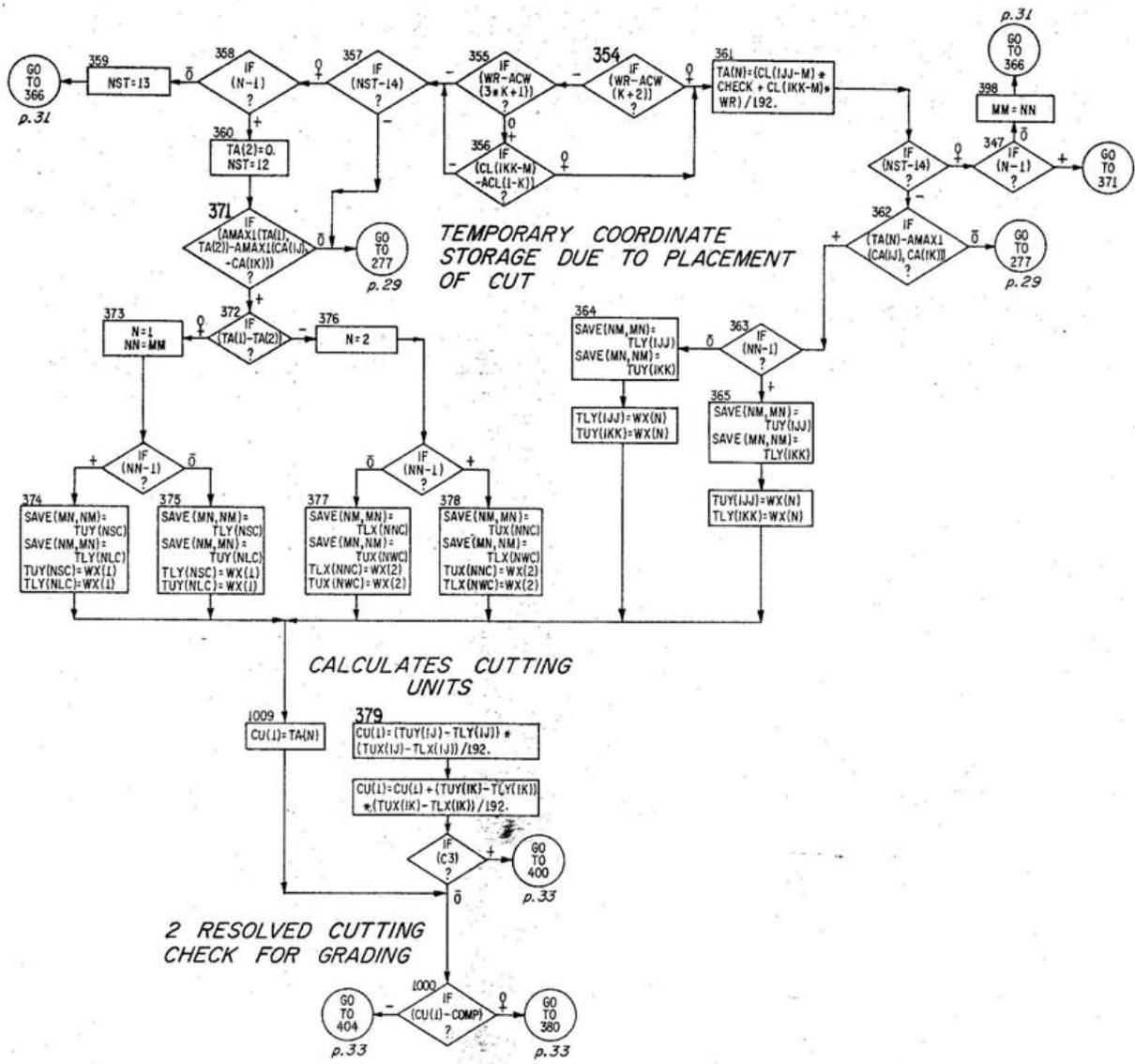


TEMPORARY COORDINATE  
STORAGE DUE TO PLACEMENT  
OF CUT

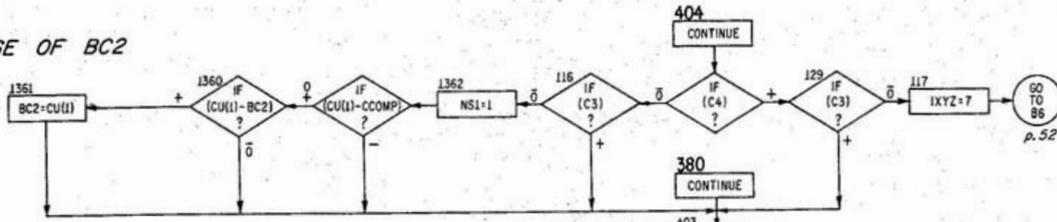


SETS UP VALUES FOR:  
CROSSCUT NST=13 AND 14

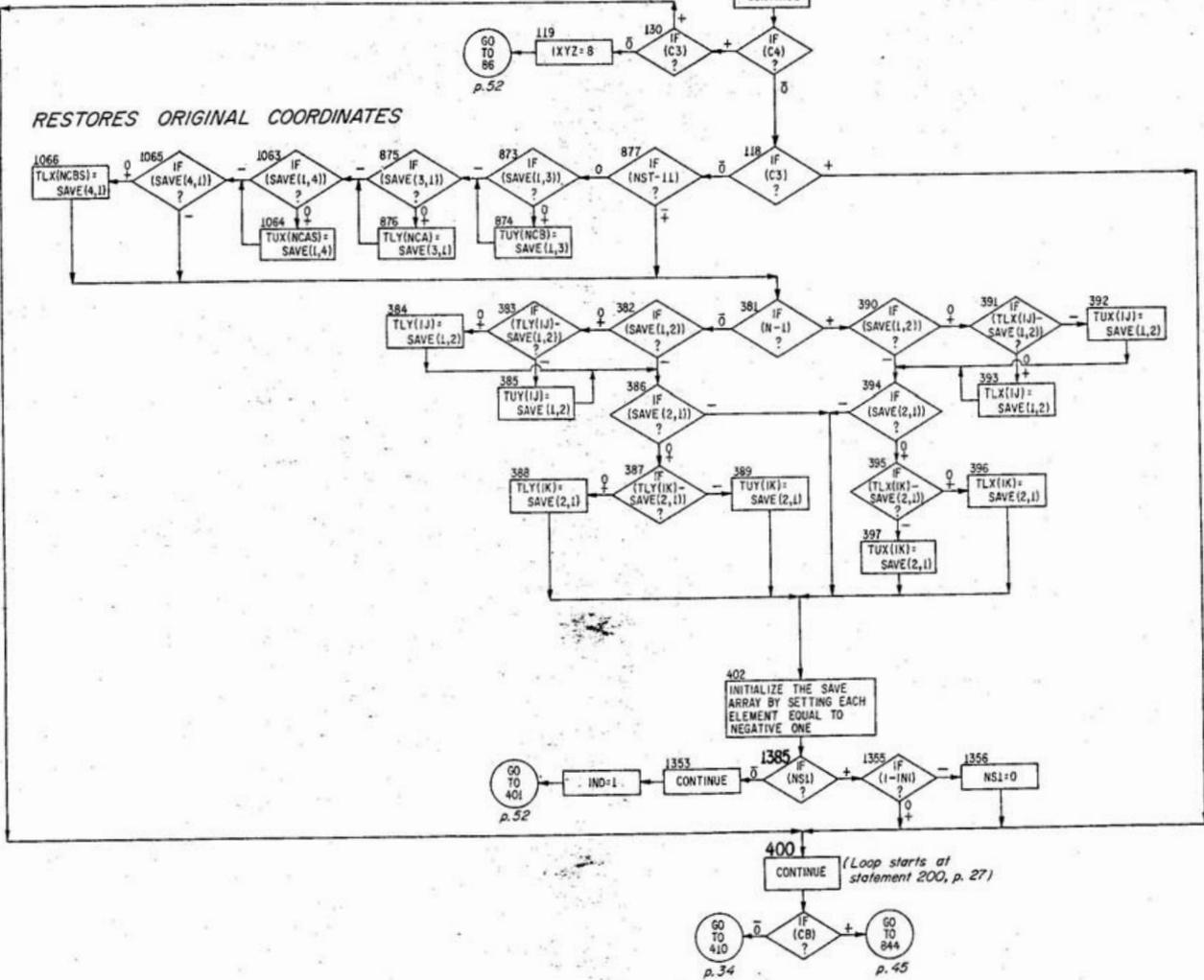




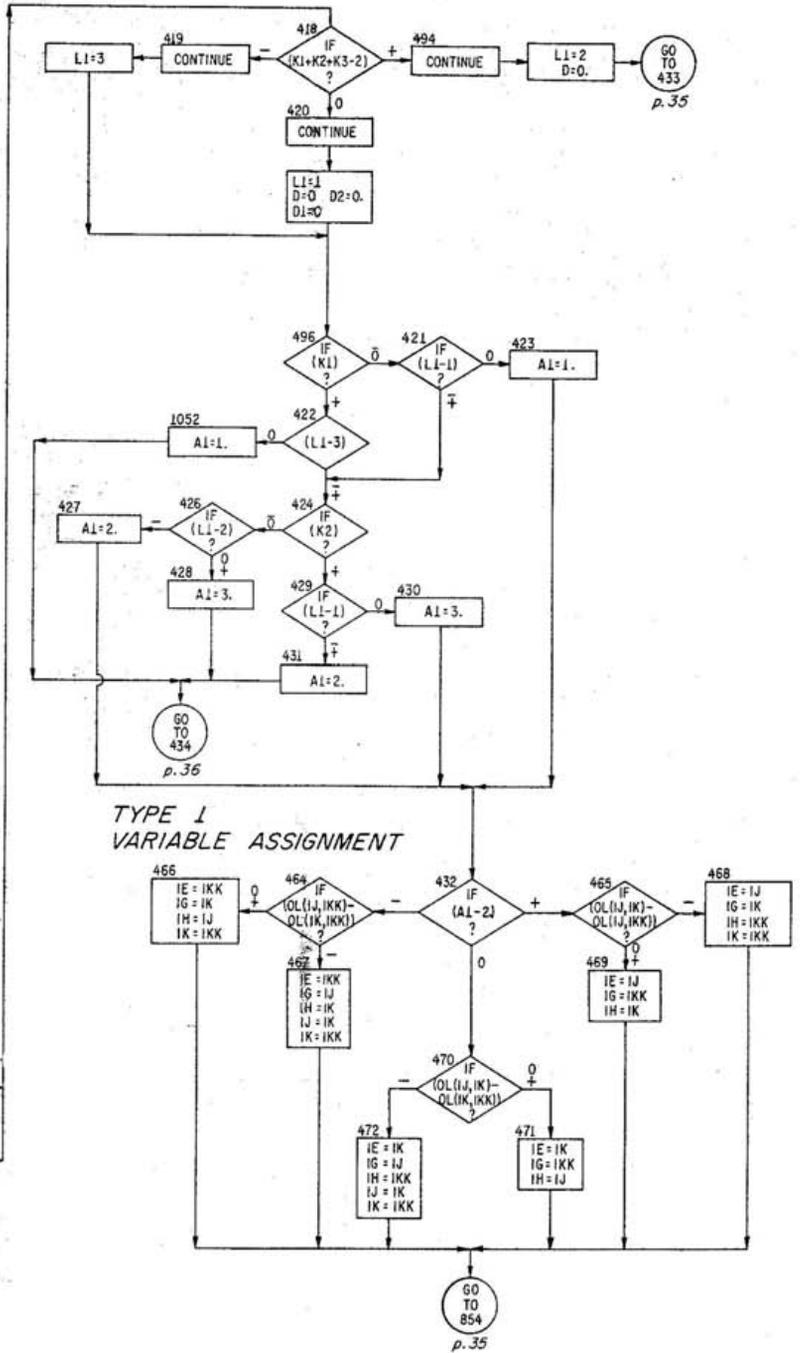
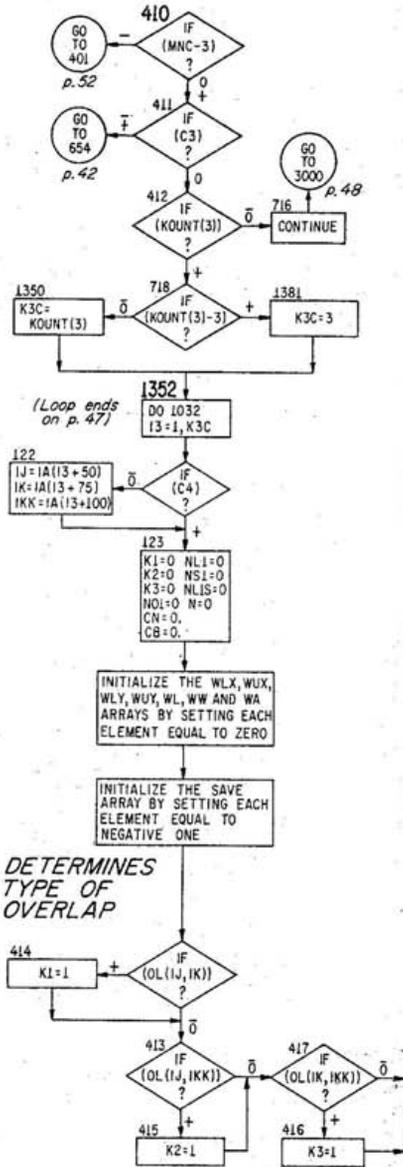
STORAGE OF BC2



RESTORES ORIGINAL COORDINATES

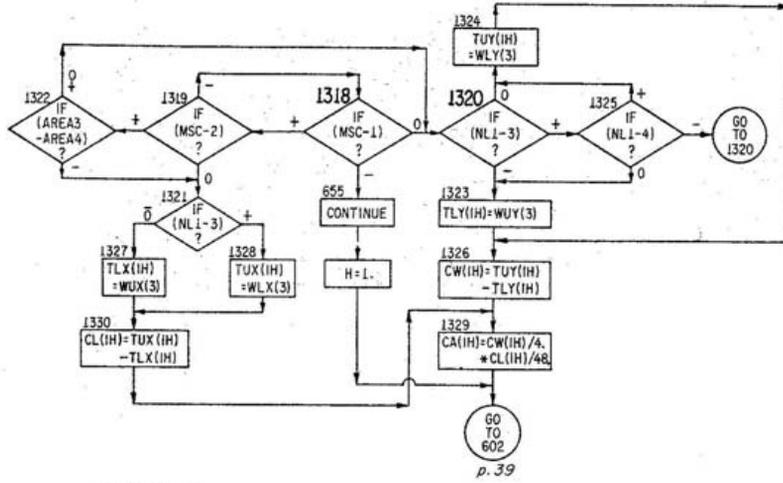


### 3 CUTTING ROUTINE FOR OVERLAP REMOVAL

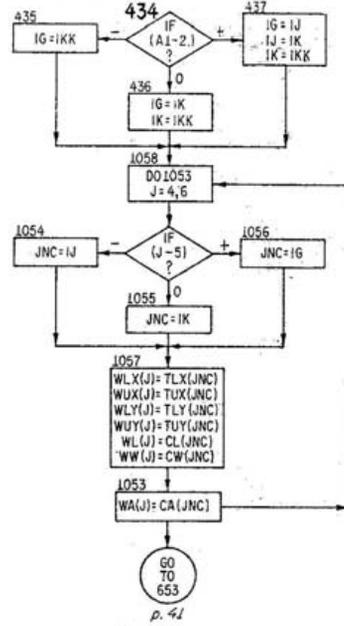




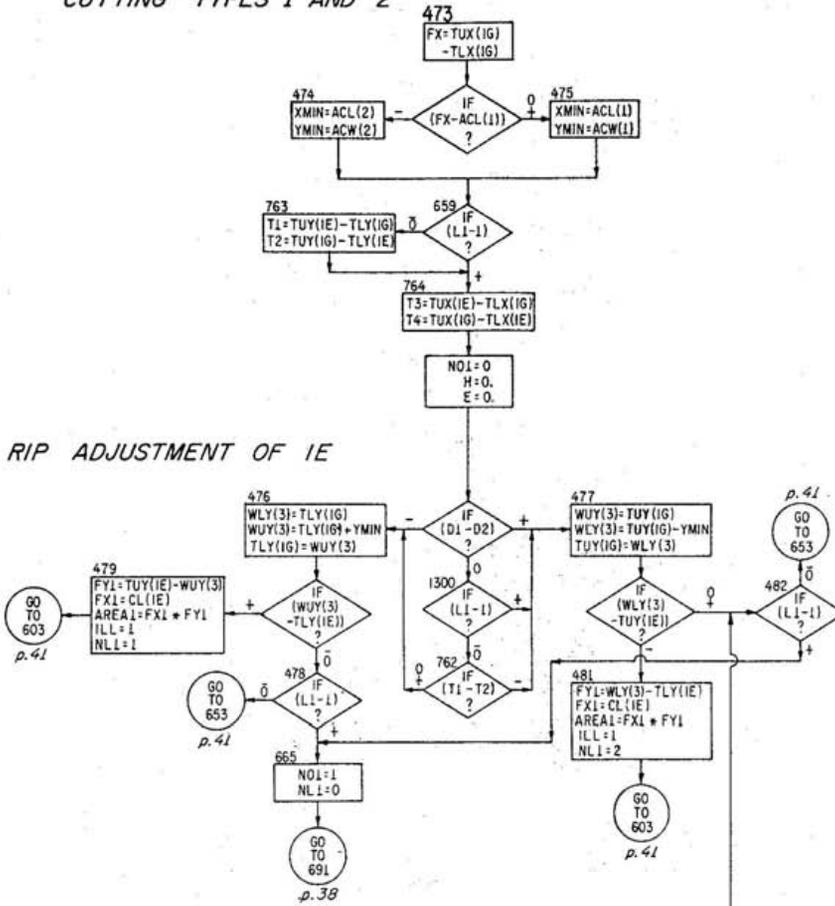
COMPARES RIP AND CROSSCUT  
ADJUSTMENTS OF IH



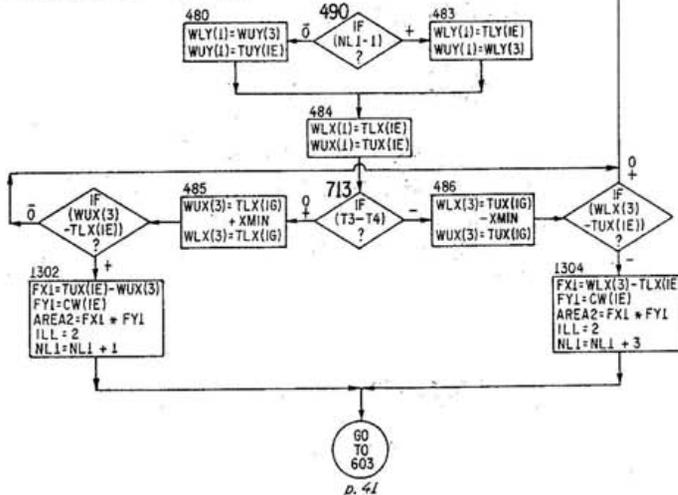
TYPE 3  
VARIABLE ASSIGNMENT



PLACES MINIMUM SIZE  
CUTTING - TYPES 1 AND 2

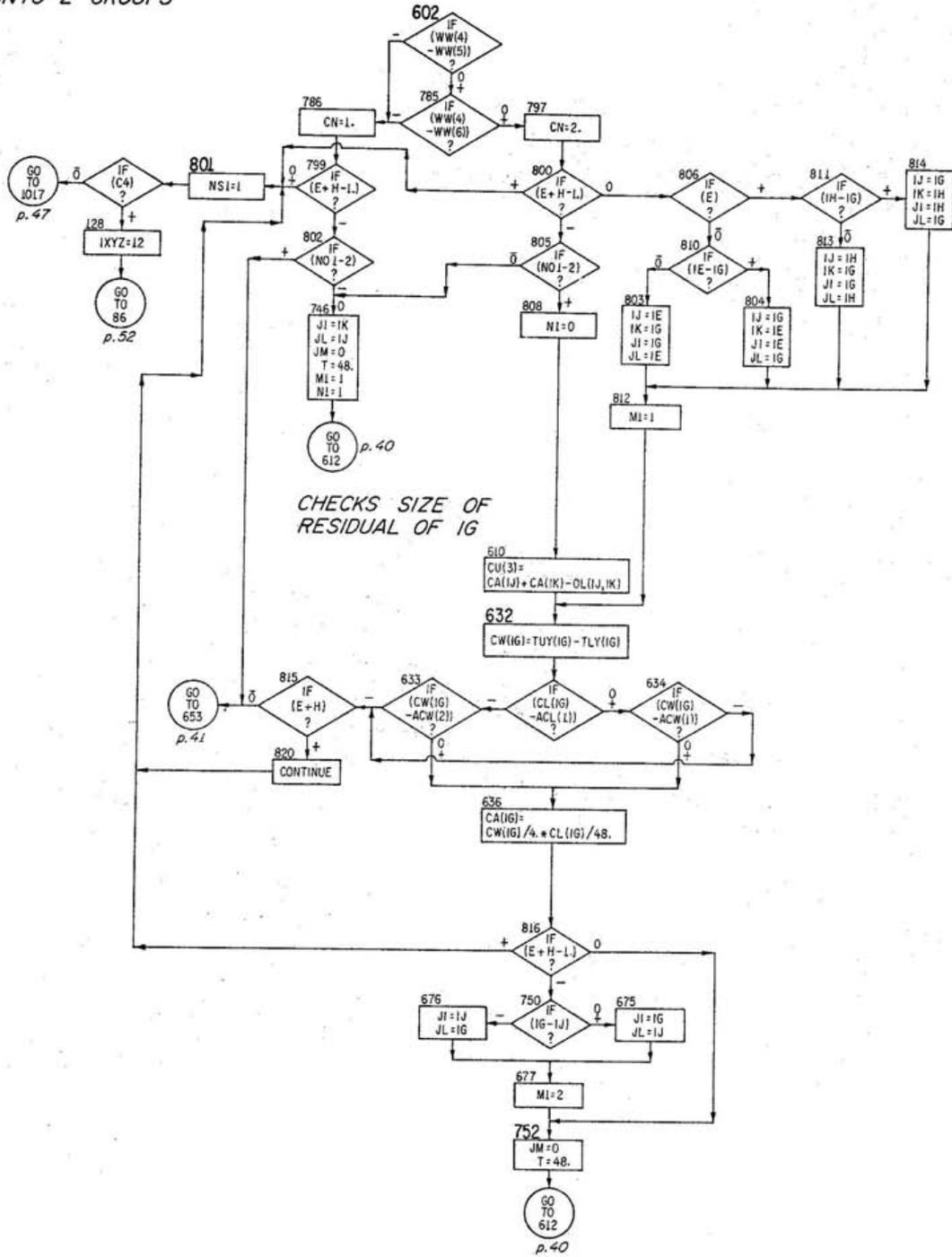


CROSSCUT ADJUSTMENT OF IE



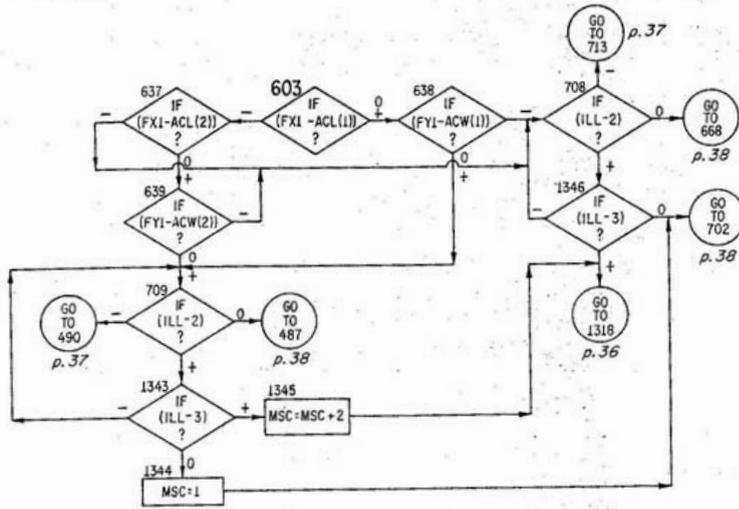


SEPARATES TYPE 2 COMBINATIONS  
INTO 2 GROUPS

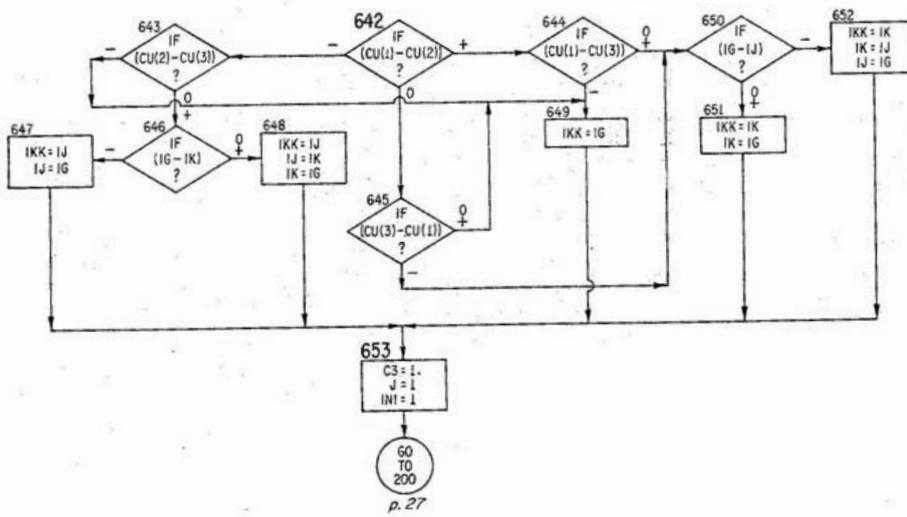




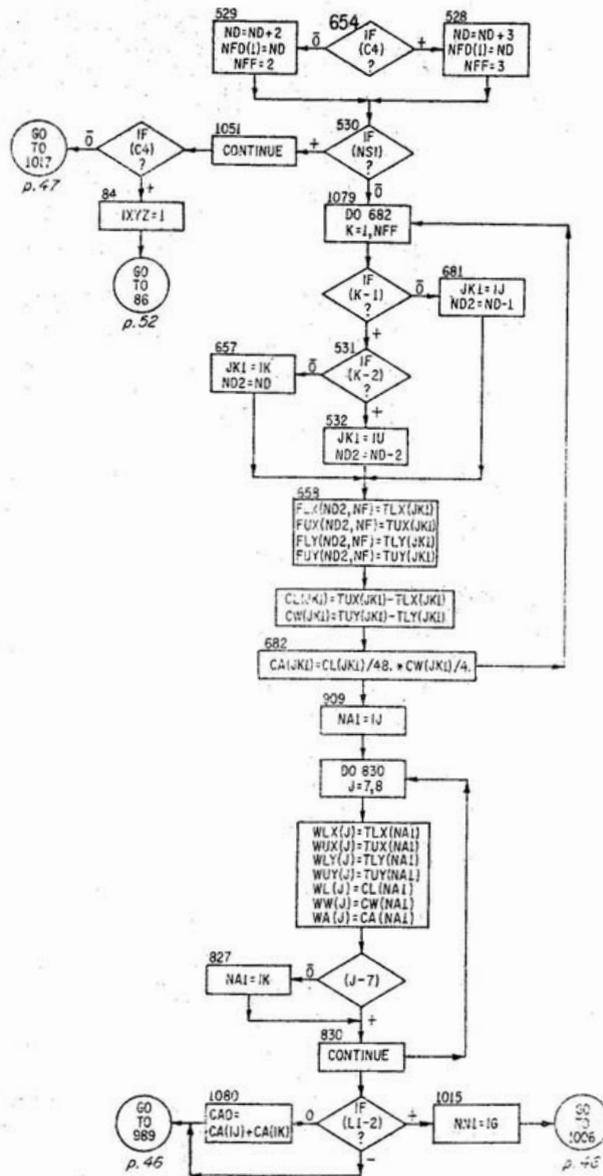
CHECKS ADJUSTED CUTTINGS FOR SIZE  
 TYPE 2 AND 2 CUTTING NST=11 ALTERNATE CROSSCUT



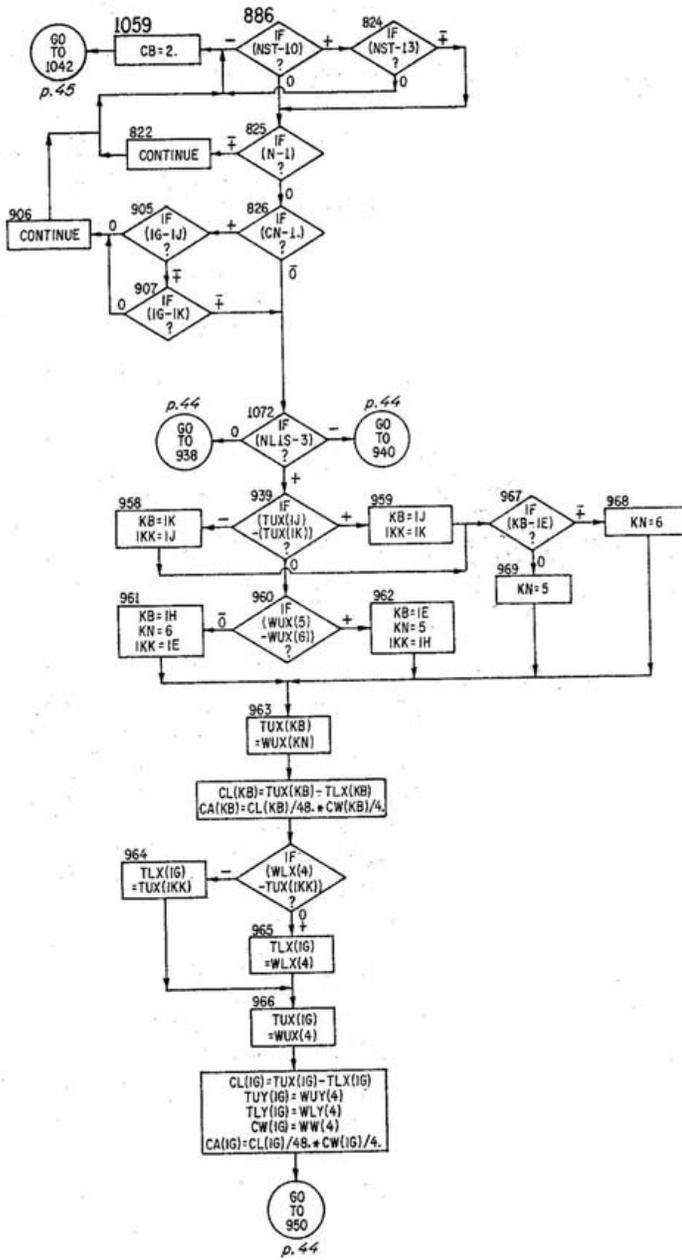
SELECTS BEST ADJUSTED COMBINATION - TYPE 2



# STORAGE OF 2 RESOLVED CUTTINGS



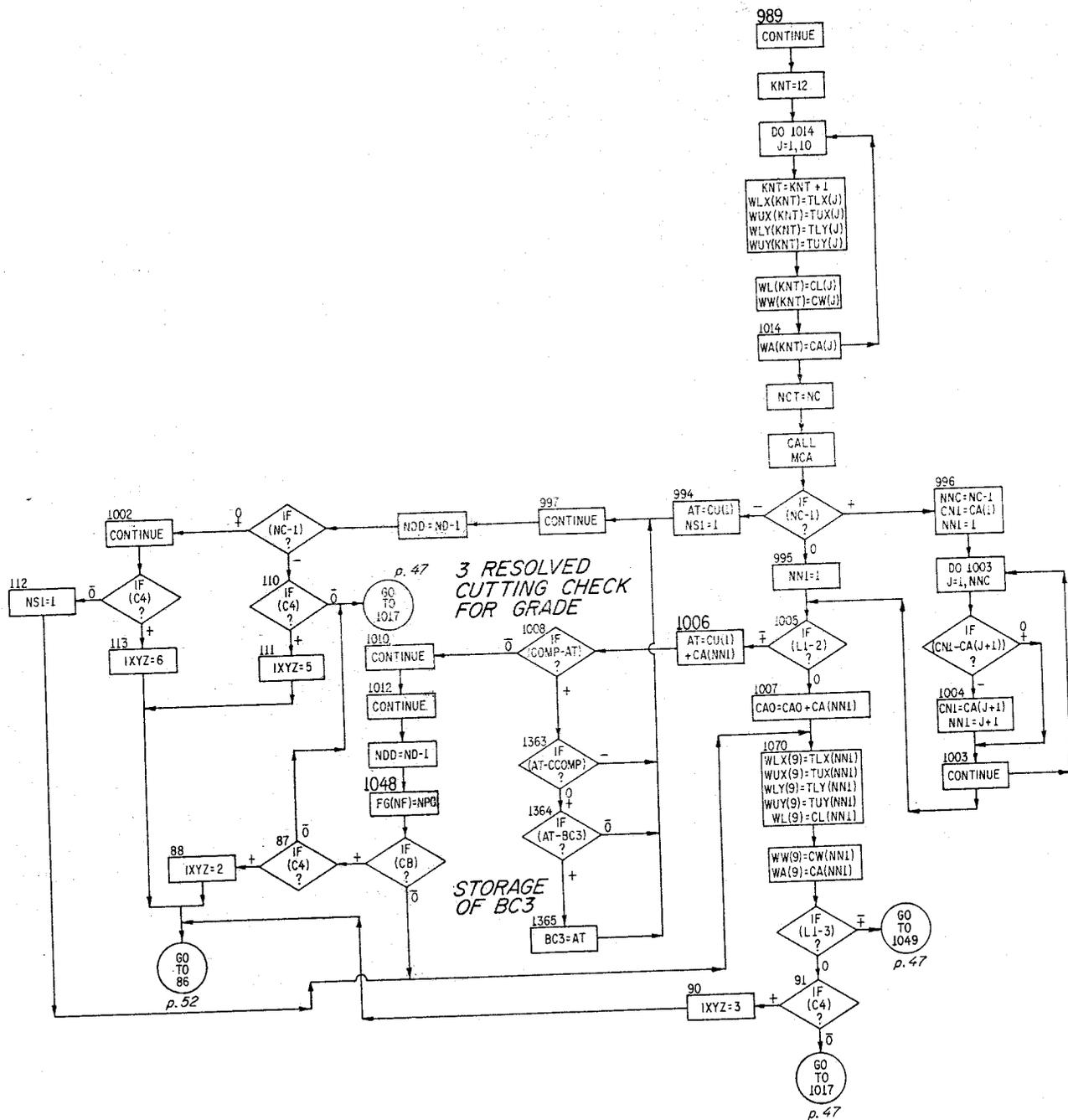
NEW ROUTINE





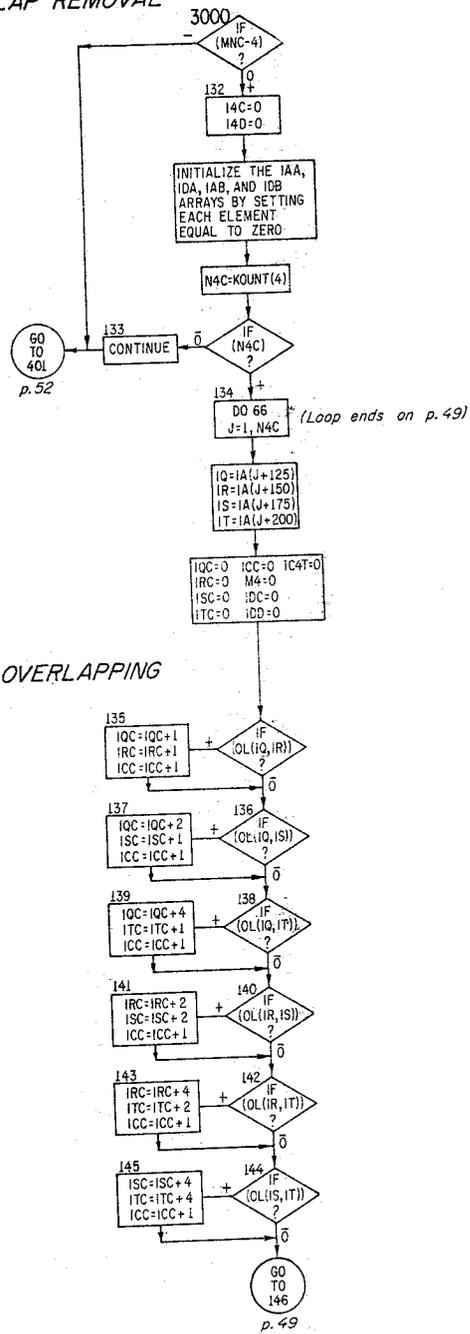


EXPANSION OF MINIMUM SIZE CUTTING - TYPES 1 AND 2



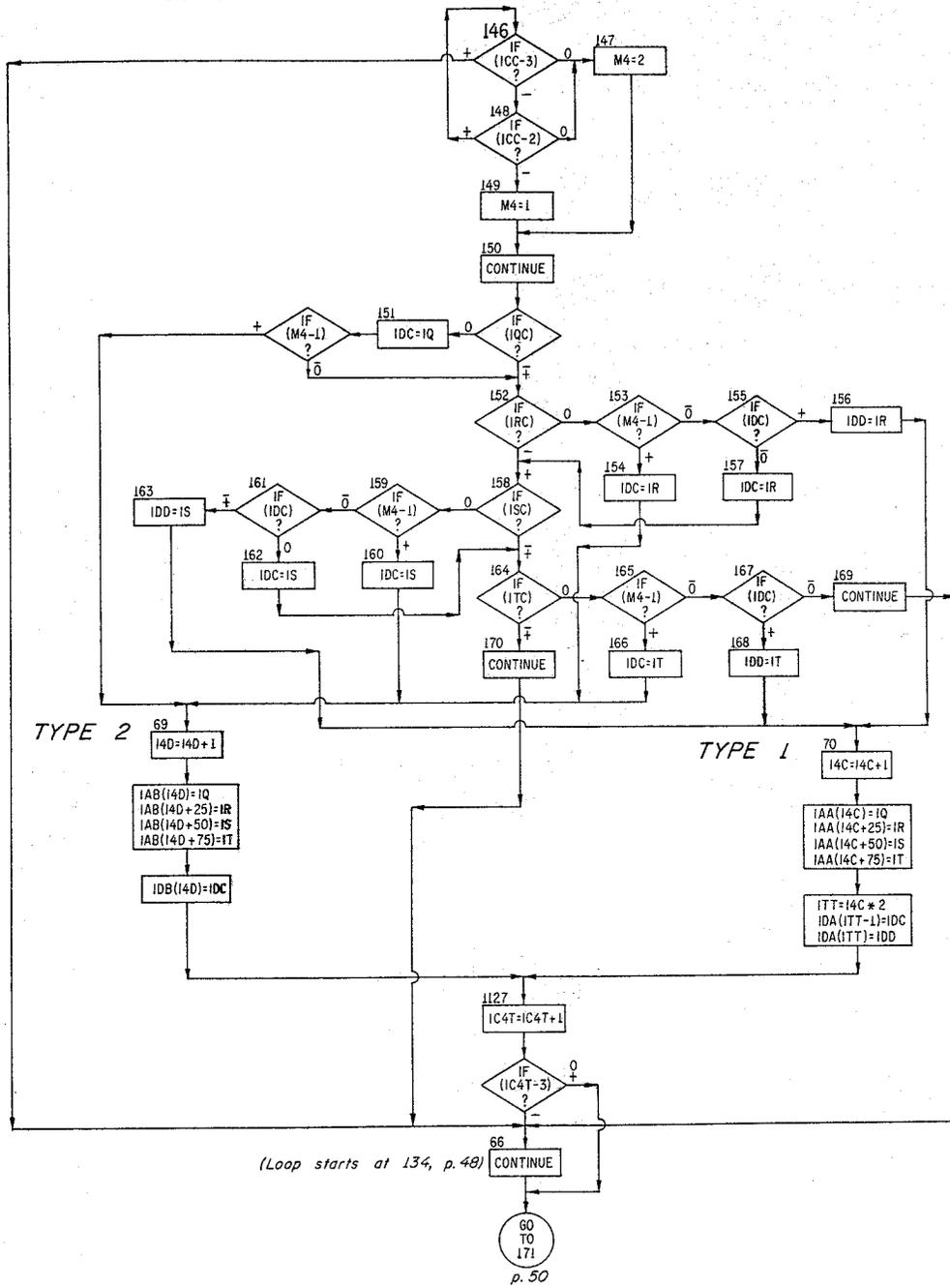


4 CUTTING ROUTINE  
FOR OVERLAP REMOVAL

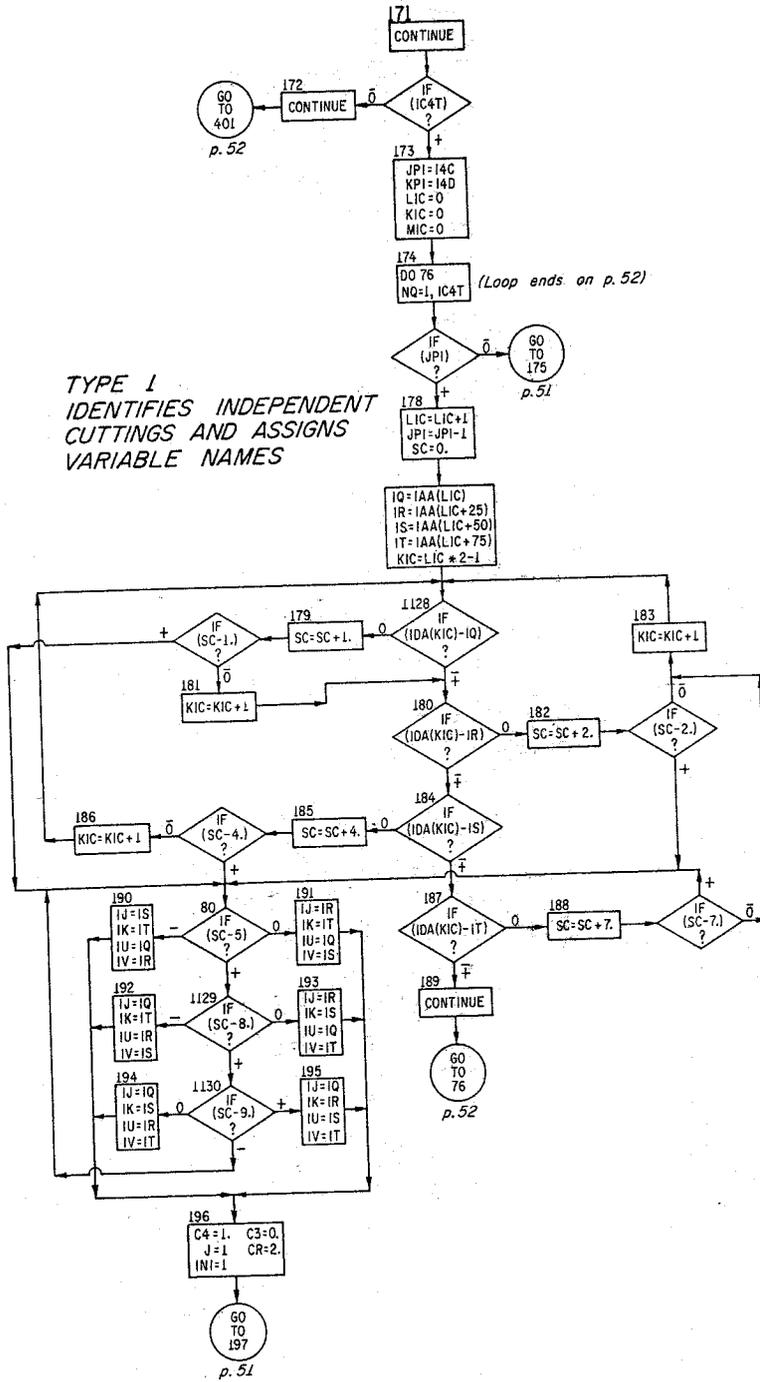


IDENTIFIES OVERLAPPING  
CUTTINGS

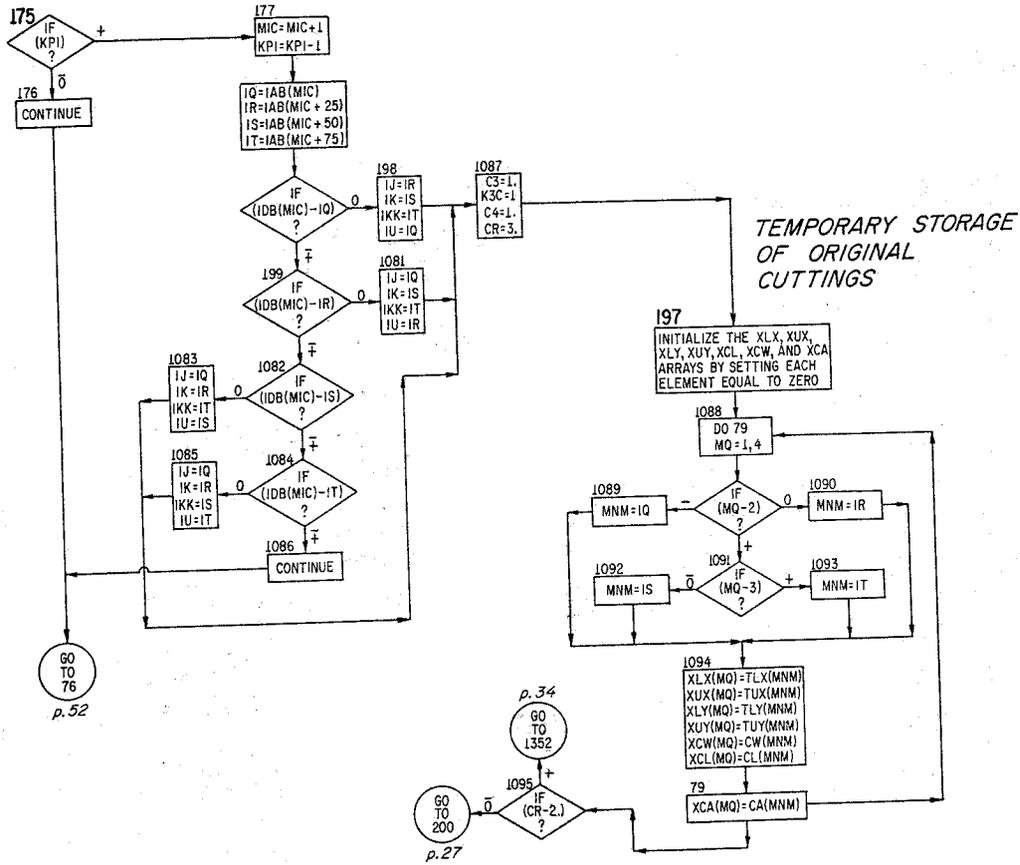
CLASSIFIES AND STORES  
2 TYPES OF COMBINATIONS



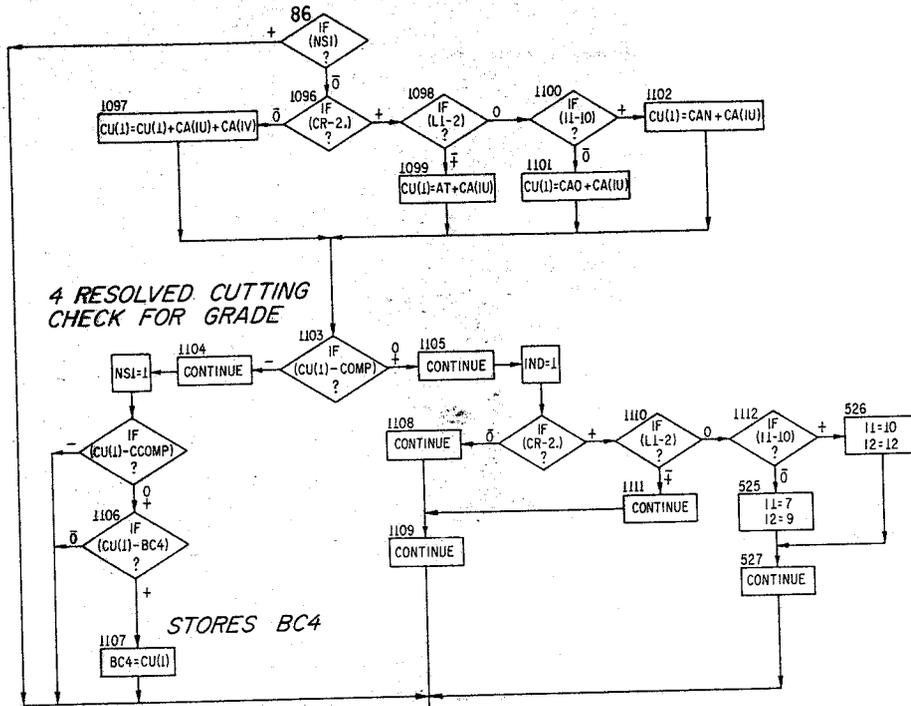
**TYPE 1  
IDENTIFIES INDEPENDENT  
CUTTINGS AND ASSIGNS  
VARIABLE NAMES**



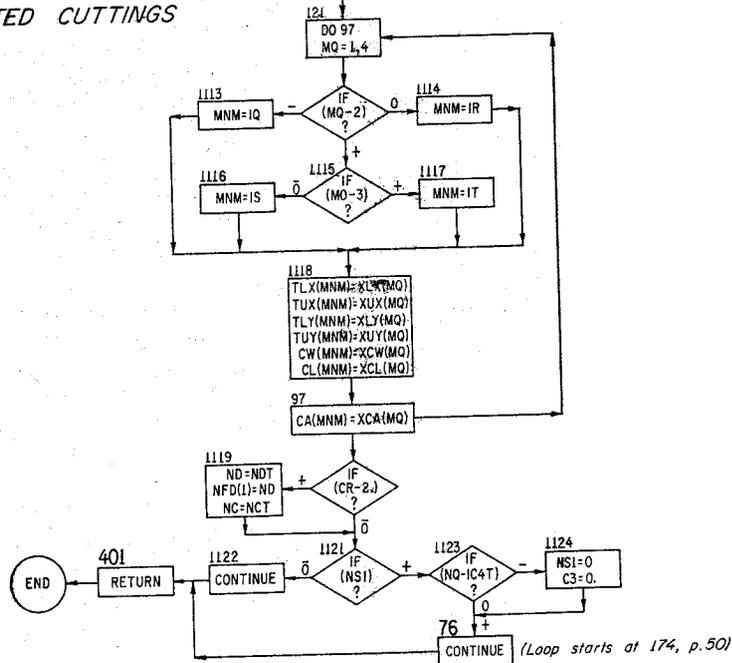
TYPE 2  
IDENTIFIES INDEPENDENT  
CUTTING AND ASSIGNS  
VARIABLE NAMES



**CALCULATES CUTTING UNITS  
IN RESOLVED COMBINATION**



**RESTORES ORIGINAL COORDINATES  
TO ADJUSTED CUTTINGS**



# PROGRAM LISTING

```

PROGRAM GRADE
TO TERMINATE JOB LET NFD(1)=-1, FOLLOWED BY A BLANK CARD
DIMENSION FLX(26,2),FUX(26,2),FLY(26,2),FUY(26,2),TLY(25),TUY(25),
1TLX(25),TUX(25),CL(25),CW(25),ACW(2),ACL(2),NFD(2),IOX(52),
2IOY(52),W(25),CA(25),CU(7),IA(675),AI(150),IB(7),OL(25,25),
3KOUNT(7),SLX(25),SUX(25),SLY(25),SUY(25),SAVE(7,7),GCW(6,2),
4GCL(6,2),FG(2),WLX(25),WUX(25),WLY(25),WUY(25),TA(2),WX(2),WL(25),
5WW(25),WA(25),KID(25),WID(25),SID(25),PID(25),IAA(100),IDA(50),
6IAB(100),IDB(25),XLX(4),XUX(4),XLY(4),XUY(4),XCL(4),XCW(4),XCA(4)
COMMON BLX,FLX,BUX,FUX,BLY,FLY,BUY,FUY,TLY,TUY,TLX,TUX,CL,CW,ACW,
1ACL,NFD,MNEB,NC,IND,IOX,IOY,DDW,W,NF,CA,CU,COMP,IA,AI,IB,OL,KOUNT,
2MNC,SLX,SUX,SLY,SUY,SAVE,NTI,FG,WLX,WUX,WLY,WUY,WL,WW,WA,NS1,IG,
3ND,CB,CAO,CAN,NPG,NBN,BC1,BC2,BC3,BC4,CCOMP,NDT
DATA(((GCW(I,J),I=1,6),J=1,2)=5(12.),6.,3(16.),2(12.),6.))
DATA(((GCL(I,J),I=1,6),J=1,2)=2(336.),144.,3(96.),2(240.),4(96.))
1 DO 2 I=1,211
2 FLX(I,1)=-99.0
IND=0
BC1=0.
BC2=0.
BC3=0.
BC4=0.
READ 100,NBN,NFD(2),NFD(1)
100 FORMAT(31X,15,3X,14,25X,14)
READ 102,BUY,BUX,BLY,BLX
102 FORMAT(F3.0,1X,F3.0,5X,F3.0,1X,F3.0,21X,13)
ND=NFD(1)
NDT=ND
ICK=0
ICW=0
ICS=0
ICP=0
IF(NFD(1))999,5,99
99 IF(NFD(1)-22)3,3,106
106 J=NFD(1)
DO 113 K=1,J
113 READ 114
114 FORMAT(40X,40X)
62 PRINT 63
63 FORMAT(40HNUMBER OF DEFECTS EXCEEDS PROGRAM LIMITS)
GO TO 1
3 J=NFD(1)
DO 115 K=1,25
115 KID(K)=0
DO 121 K=1,75
121 WID(K)=0.
DO 4 I=1,J
READ 102,FLY(I,1),FLX(I,1),FUY(I,1),FUX(I,1),ID
68 IF(ID-3)4,64,67
67 IF(ID-6)65,69,97
64 ICK=ICK+1
KID(ICK)=I
GO TO 4
65 ICW=ICW+1

```

```

WID(ICW)=I
GO TO 4
69 ICS=ICS+1
   SID(ICS)=I
   GO TO 4
97 ICP=ICP+1
   PID(ICP)=I
4 CONTINUE
5 IF(NFD(2))999,8,6
6 J=NFD(2)
DO 7 I=1,J
7 READ 102,FLY(I,2),FLX(I,2),FUY(I,2),FUX(I,2)
8 FLX(26,1)=BLX
  FUX(26,1)=BUX
  FLY(26,1)=BLY
  FUY(26,1)=BUY
  BL=BLX-BUX
  BW=BLY-BUY
  SM=INTF (INTF (BL/48.)*BW/48.+ .49999999)
NF=1
9 NPG=1
10 NTI=0
   IF(BW-24.)308,300,300
300 IF(BL-384.)327,306,306
327 IF(BL-288.)326,303,303
326 IF(BL-192.)324,302,302
302 CONTINUE
   NPG=3
   GO TO 322
303 NPG=2
   GO TO 128
306 CONTINUE
128 EFL=BUX+48.
   EFR=BLX-48.
   IF(ICS)124,124,79
124 CONTINUE
   GO TO 90
79 DO 70 J=1,ICS
   IF(FUX(SID(J))-EFL)70,70,72
72 IF(FLX(SID(J))-EFL)75,75,74
74 IF(FLX(SID(J))-EFR)75,70,70
70 CONTINUE
   GO TO 90
75 SL=0.
   DO 76 J=1,ICS
76 SL=SL+FUX(SID(J))-FLX(SID(J))
   SL=SL/4.
   PL=2.*SM
   IF(SL-PL)90,90,302
90 IF(ND)137,137,139
137 CONTINUE
   GO TO 91
139 EFA=BW*48.
   A50=EFA*.50

```

```

EFAL=0.
EFAR=0.
DO 80 J=1,ND
  IF (FLX(J)-EFL)81,82,82
81 IF (FUX(J)-EFL)84,84,86
84 EFAL=EFAL+(FUX(J)-FLX(J))*(FUY(J)-FLY(J))
  GO TO 80
82 IF (FLX(J)-EFR)130,85,131
130 IF (FUX(J)-EFR)80,80,85
85 EFAR=EFAR+(FUX(J)-EFR)*(FUY(J)-FLY(J))
  GO TO 80
131 EFAR=EFAR+(FUX(J)-FLX(J))*(FUY(J)-FLY(J))
  GO TO 80
86 EFAL=EFAL+(EFL-FLX(J))*(FUY(J)-FLY(J))
87 IF (FUX(J)-EFR)80,80,85
80 CONTINUE
  IF (EFAL-A50)89,89,302
89 IF (EFAR-A50)91,91,302
91 IF (ICW)127,127,126
126 WOB=0.
  WOT=0.
  DO 95 J=1,ICW
    IF (FUY(WID(J))-BLY)93,92,93
92 WOT=WOT+FUX(WID(J))-FLX(WID(J))
    GO TO 95
93 WOB=WOB+FUX(WID(J))-FLX(WID(J))
95 CONTINUE
  BLHH=BL/2.
  IF (WOB-BLHH)94,94,302
94 IF (WOT-BLHH)309,309,302
127 CONTINUE
  GO TO 309
308 IF (BW-16.)323,327,327
323 IF (BW-12.)324,326,326
324 PRINT 325,NBN
325 FORMAT(1X,12HBOARD NUMBER,15,4X,13HIS NOT LUMBER)
  NPG=6
  GO TO 1
309 IF (NFD(1)-1)50,311,322
311 IF (NPG-1)313,313,322
313 IF (BW-24.)322,315,314
314 IF (BW-30.)315,315,319
315 IF (SM-6.)322,50,316
316 IF (SM-10.)50,50,322
319 IF (BW-36.)320,320,322
320 IF (SM-7.)322,50,321
321 IF (SM-12.)50,50,322
322 GO TO(11,13,15,22,27,28),NPG
  11 COMP=10.*SM
  CCOMP=8.*SM
  MNC=INTF(SM/4.)
  IF (MNC-4)30,30,12
  12 MNC=4
  GO TO 30

```

```

13 IF(SM-3.)14,14,11
14 COMP=11.*SM
   CCOMP=8.*SM
   MNC=1
   GO TO 30
15 MNC=INTF((SM+1.)/3.)
   IF(MNC-5)19,19,17
17 MNC=5
18 COMP=8.*SM
   CCOMP=6.*SM
   GO TO 30
19 IF(SM-2.)21,20,18
20 COMP=9.*SM
   CCOMP=6.*SM
   GO TO 30
21 IF(NFD(NF))50,50,53
22 MNC=INTF(SM/2.)
   IF(MNC-7)25,25,23
23 MNC=7
24 COMP=6.*SM
   GO TO 30
25 IF(SM-1.)26,26,24
26 MNC=1
   GO TO 18
27 COMP=4.*SM
   GO TO 29
28 COMP=3.*SM
29 MNC=25
30 ACW(1)=GCW(NPG,1)
   ACW(2)=GCW(NPG,2)
   ACL(1)=GCL(NPG,1)
   ACL(2)=GCL(NPG,2)
31 CONTINUE
   CALL MCA
   IF(IND)32,32,50
32 IF(NC)16,16,33
33 CONTINUE
   IF(NC-1)16,53,35
16 IF(NPG-3)302,71,34
71 NPG=4
   GO TO 322
34 NPG=5
   PRINT 36,NBN,SM
36 FORMAT(13H BOARD NUMBER,15,4X,11HBELOW GRADE,7X,3HSM=,F5.0)
   GO TO 1
35 IF(MNC-1)39,39,38
38 CALL BCC
   IF(IND)39,39,50
39 NTI=NTI+1
   IF(NTI-1)40,40,53
40 GO TO(41,44,45,47,78,78),NPG
41 IF(SM-5.)553,553,42
42 MNC=MNC+1
   COMP=COMP+SM

```

```

43 CONTINUE
   GO TO 38
44 IF(SM-5.)53,53,42
45 IF(SM-2.)53,53,46
46 IF(SM-8.)42,53,53
47 IF(SM-2.)53,48,49
48 MNC=MNC+1
   COMP=COMP+2.*SM
   GO TO 43
49 IF(SM-8.)48,53,53
50 FG(NF)=NPG
   PRINT 112,NBN,NPG,SM
112 FORMAT(13H BOARD NUMBER,15,4X,12HGRADE IS NO.,12,4X,3HSM=,F5.0)
   GO TO 1
53 NPG=NPG+1
   NTI=0
   IF(NPG-4)553,51,34
51 CONTINUE
   GO TO 37
553 NPG=3
   NTI=0
   37 IF(BC1-BC2)1369,1368,1368
1368 IF(BC1-BC3)134,132,132
   132 IF(BC1-BC4)133,1370,1370
1370 BCA=BC1
1371 CONTINUE
   IF(BCA)73,73,50
   73 IF(NPG-4)302,22,34
1369 IF(BC2-BC3)134,135,135
   135 IF(BC2-BC4)133,1374,1374
1374 BCA=BC2
   IF(SM-3.)1371,54,1371
54 IF(NPG-4)1371,55,1371
55 COMP3=8.*SM
   IF(BC2-COMP3)136,1371,1371
136 IF(BC1-BC3)134,132,132
134 IF(BC3-BC4)133,1375,1375
133 BCA=BC4
   GO TO 1371
1375 BCA=BC3
   IF(SM-5.)1371,58,1371
58 IF(NPG-4)1371,59,1371
59 COMP3=8.*SM
   IF(BC3-COMP3)66,1371,1371
66 IF(BC1-BC2)135,132,132
78 PRINT 204
204 FORMAT(1X,29HPROGRAM CANNOT GRADE 3A OR 3B)
   GO TO 1
999 CONTINUE
   END

```

```

SUBROUTINE MCA
  DIMENSION FLX(26,2),FUX(26,2),FLY(26,2),FUY(26,2),TLY(25),TUY(25),
  1TLX(25),TUX(25),CL(25),CW(25),ACW(2),ACL(2),NFD(2),IOX(52),
  2IOY(52),W(25),CA(25),CU(7),IA(675),AI(150),IB(7),OL(25,25),
  3KOUNT(7),SLX(25),SUX(25),SLY(25),SUY(25),SAVE(7,7),GCW(6,2),
  4GCL(6,2),FG(2),WLX(25),WUX(25),WLY(25),WUY(25),TA(2),WX(2),WL(25),
  5IAB(100),IDB(25),XLX(4),XUX(4),XLY(4),XUY(4),XCL(4),XCW(4),XCA(4)
  COMMON BLX,FLX,BUX,FUX,BLY,FLY,BUY,FUY,TLY,TUY,TLX,TUX,CL,CW,ACW,
  1ACL,NFD,MNEB,NC,IND,IOX,IOY,DDW,W,NF,CA,CU,COMP,IA,AI,IB,OL,KOUNT,
  2MNC,SLX,SUX,SLY,SUY,SAVE,NTI,FG,WLX,WUX,WLY,WUY,WL,WW,WA,NS1,IG,
  3ND,CB,CÃO,CAN,NPG,NBN,BC1,BC2,BC3,BC4,CCOMP,NDT
  MNEB=0
  NC=0
  K=0
  KK=0
  1 IOX(KK+1)=0
  IDM=1
  DO 6 I=1,51
    II=I+K+(I/26)*27
    IF(FLX(II,NF)+99,0)106,106,2
106 IF(I-26)107,108,108
107 I=25
    GO TO 6
108 I=51
    GO TO 6
  2 DO 3 J=1,IDM
    JJ=J+KK
    IF(FLX(II,NF)-FLX(IOX(JJ)+K,NF))4,4,3
  3 CONTINUE
    IDM=IDM+1
    IM=IDM+KK
    IOX(IM)=II-K
    GO TO 6
  4 IDM=IDM+1
    L=IDM-J
    IW=IDM+KK
    DO 5 M=1,L
    IM=IW-M
  5 IOX(IM+1)=IOX(IM)
    IOX(IM)=II-K
  6 CONTINUE
    IF(K)8,7,8
  7 K=106
    KK=52
    GO TO 1
  8 JD=1
    GO TO 87
86 JD=IX+1
87 DO 9 I=JD,IDM
    IF(IOY(I)-53)9,10,10
  9 CONTINUE
    GO TO 994
10 IX=I

```

```

    JA=IX+1
    IC=IOY(IX)-53
    FI=FLX(IC,NF)
    UI=FUX(IC,NF)
    GO TO 118
119 JA=IY+1
118 IF(JA-IDM)11,11,86
    DO 12 J=JA, IDM
    IF(IOY(J)-53)13,12,12
12 CONTINUE
    GO TO 86
13 IY=J
    JC=IOY(IY)
    FY=FLY(JC,NF)-FUY(IC,NF)
    IF(FY-ACW(1))119,14,14
14 FJ=FLX(JC,NF)
    UJ=FUX(JC,NF)
    IF(JC*IC)20,15,20
15 IF(JC)17,16,17
16 IF(IC)19,18,19
17 AMG=FJ
    BMG=UJ
    GO TO 26
18 AMG=UI
    BMG=FI
    GO TO 26
19 AMG=FI
    BMG=UI
    GO TO 26
20 IF(FI-FJ)21,21,22
21 AMG=FJ
    GO TO 23
22 AMG=FI
23 IF(UI-UJ)24,24,25
24 BMG=UI
    GO TO 26
25 BMG=UJ
26 DO 27 K=1,25
27 W(K-1)=0.0
    DO 30 K=1,IX
    IK=IOY(K)
    IF(IK-53)28,29,29
28 W(IK)=W(IK)+1.0
    GO TO 30
29 IK=IK-53
    W(IK)=W(IK)-1.0
30 CONTINUE
    IK=IX+1
    JK=IY-1
    IF(JK-1K)34,31,31
31 DO 33 K=1K, JK
    LF=IOY(K)
    KI=LF-53
    IF(KI)32,133,133

```

```

133 IF (FUY(KI,NF)-FUY(IC,NF))33,134,33
134 W(KI)=W(KI)-1.0
    GO TO 33
    32 IF (FLY(LF,NF)-FLY(JC,NF))135,33,135
135 W(LF)=W(LF)+1.0
33 CONTINUE
34 N=0
    W(N)=1.0
    PW=1.0
    IM=0
35 KF=IM+1
    IM=KF
    IF (KF-IDM)36,119,119
36 IW=IOX(KF)
    IF (IW-53)37,39,39
37 IF (W(IW))35,35,38
38 PW=PW+1.0
    GO TO 35
39 IF (W(IW-53))35,35,40
40 PW=PW-1.0
    IF (PW)35,41,35
41 LF=KF
42 LF=LF+1
    IF (LF-IDM)43,43,119
43 IW=IOX(LF)
    IF (IW-53)44,42,42
44 IF (W(IW))42,42,45
45 KF=IOX(KF)-53
    LF=IW
    IF (FUX(KF,NF)-BMG)46,119,119
46 IF (FLX(LF,NF)-AMG)35,35,47
47 FX=FLX(LF,NF)-FUX(KF,NF)
    IF (FX-ACL(1))48,51,51
48 IF (FY-ACW(2))35,49,49
49 IF (FX-ACL(2))35,51,51
51 IF (NC)59,59,50
50 DO 56 I=1,NC
    IF (TLX(I)-FUX(KF,NF))56,53,56
53 IF (TUX(I)-FLX(LF,NF))56,54,56
54 IF (TLY(I)-FUY(IC,NF))56,55,56
55 IF (TUY(I)-FLY(JC,NF))56,35,56
56 CONTINUE
59 IF (MNEB)65,65,60
60 DO 64 I=1,MNEB
    IF (SLX(I)-FUX(KF,NF))64,61,64
61 IF (SUX(I)-FLX(LF,NF))64,62,64
62 IF (SLY(I)-FUY(IC,NF))64,63,64
63 IF (SUY(I)-FLY(JC,NF))64,35,64
64 CONTINUE
65 NC=NC+1
    IF (NC-26)66,67,67
67 NC=NC-1
    GO TO 994
66 TLX(NC)=FUX(KF,NF)

```

```
TUX(NC)=FLX(LF,NF)
TLY(NC)=FUY(IC,NF)
TUY(NC)=FLY(JC,NF)
CL(NC)=FX
CW(NC)=FY
WGT=FX/48.*FY/4.
CA(NC)=WGT
IF(WGT-COMP)35,141,141
141 IF(WGT-BC1)140,140,142
142 BC1=WGT
140 IF(WGT-COMP)35,52,52
52 MNEB=MNEB+1
SLX(MNEB)=FUX(KF,NF)
SUX(MNEB)=FLX(LF,NF)
SLY(MNEB)=FUY(IC,NF)
SUY(MNEB)=FLY(JC,NF)
NC=NC-1
IND=1
994 RETURN
END
```

```

SUBROUTINE BCC
  DIMENSION FLX(26,2),FUX(26,2),FLY(26,2),FUY(26,2),TLY(25),TUY(25),
  1TLX(25),TUX(25),CL(25),CW(25),ACW(2),ACL(2),NFD(2),IOX(52),
  2IOY(52),W(25),CA(25),CU(7),IA(675),AI(150),IB(7),OL(25,25),
  3KOUNT(7),SLX(25),SUX(25),SLY(25),SUY(25),SAVE(7,7),GCW(6,2),
  4GCL(6,2),FG(2),WLX(25),WUX(25),WLY(25),WUY(25),TA(2),WX(2),WL(25),
  5WW(25),WA(25),KID(25),WID(25),SID(25),PID(25),IAA(100),IDA(50),
  6IAB(100),IDB(25),XLX(4),XUX(4),XLY(4),XUY(4),XCL(4),XCW(4),XCA(4)
  COMMON BLX,FLX,BUX,FUX,BLY,FLY,BUY,FUY,TLY,TUY,TLX,TUX,CL,CW,ACW,
  1ACL,NFD,MNEB,NC,IND,IOX,IOY,DDW,W,NF,CA,CU,COMP,IA,AI,IB,OL,KOUNT,
  2MNC,SLX,SUX,SLY,SUY,SAVE,NTI,FG,WLX,WUX,WLY,WUY,WL,WW,WA,NS1,IG,
  3ND,CB,CAO,CAN,NPG,NBN,BC1,BC2,BC3,BC4,CCOMP,NDT
  NS1=0
  IF(NTI)106,106,107
107 IF(MNC-2)106,106,108
106 DO 101 I=1,625
101 OL(I)=-1.
108 DO 102 I=1,7
102 KOUNT(I)=0
  I=0
  1 I=I+1
  IF(I-NC)2,45,45
  2 IB(1)=I
  J=I
  20 J=J+1
  IF(J-NC)4,4,1
  4 IB(2)=J
  CU(2)=CA(I)+CA(J)
  CU(1)=CU(2)
  IJ=2
  IX=2
  IF(CU(2)-COMP)29,64,64
  64 IF(NTI)3,3,65
  65 IF(IJ-MNC)29,3,3
  3 IK=IJ-1
  DO 19 IM=1,IK
  IN=IM+1
  DO 19 JK=IN,IJ
  JI=IB(IM)
  JL=IB(JK)
  JM=0
  T=48.
  IF(OL(JI,JL))5,18,18
  5 IL=JI-JM*2
  JN=JL-JM*2
  XL=TLX(IL)-TLX(JN)
  XU=TUX(IL)-TUX(JN)
  IF(XL*XU)6,6,10
  6 IF(XL)9,7,8
  7 IF(XU)8,8,9
  8 OLL=CL(JI+JM)/T
  GO TO 15
  9 OLL=CL(JL+JM)/T
  GO TO 15

```

```

10 IF(XL)11,11,12
11 XZ=TUX(IL)-TLX(JN)
   IF(XZ)14,14,13
12 XZ=TUX(JN)-TLX(IL)
   IF(XZ)14,14,13
13 OLL=XZ/T
   GO TO 15
14 OL(JI,JL)=0.
   GO TO 18
15 IF(JM-25)16,17,17
16 X=OLL
   JM=25
   T=4.
   GO TO 5
17 OL(JI,JL)=X*OLL
18 CU(1)=CU(1)-OL(JI,JL)
   IF(CU(1)-COMP)114,19,19
114 IF(IJ-3)29,19,29
19 CONTINUE
   NI=IJ-1
   DO 60 NJ=1,NI
   NK=NJ+1
   NM=IB(NJ)
   DO 60 NL=NK,IJ
   NN=IB(NL)
   IF(OL(NM,NN))60,60,734
734 IF(IJ-3)63,735,63
735 O1L=0.
   DO 736 N1J=1,NI
   NK=N1J+1
   NM=IB(N1J)
   DO 736 N1L=NK,IJ
   NN=IB(N1L)
   IF(OL(NM,NN))736,736,737
737 O1L=O1L+1.
736 CONTINUE
   IF(O1L-3.)1220,738,738
738 N11=IB(1)
   N12=IB(2)
   N13=IB(3)
   IF(OL(N11,N12)-OL(N11,N13))739,739,740
739 IF(OL(N11,N12)-OL(N12,N13))741,741,742
741 NN=N11
   NM=N12
   GO TO 745
742 NN=N12
   NM=N13
   GO TO 745
740 IF(OL(N11,N13)-OL(N12,N13))743,743,742
743 NN=N11
   NM=N13
745 IF(TLX(NN)-TLX(NM))1073,1073,919
1073 ZLX=TLX(NM)
   GO TO 920

```

```

919 ZLX=TLX(NN)
920 IF(TUX(NN)-TUX(NM));921,921,922
921 ZUX=TUX(NN)
    GO TO 923
922 ZUX=TUX(NM)
923 IF(TLY(NN)-TLY(NM));924,924,925
924 ZLY=TLY(NM)
    GO TO 926
925 ZLY=TLY(NN)
926 IF(TUY(NN)-TUY(NM));927,927,928
927 ZUY=TUY(NN)
    GO TO 929
928 ZUY=TUY(NM)
929 IOL=IB(1)+IB(2)+IB(3)-NN-NM
    IF(TLX(IOL)-ZLX)931,931,930
930 ZLX=TLX(IOL)
931 IF(TUX(IOL)-ZUX)932,933,933
932 ZUX=TUX(IOL)
933 IF(TLY(IOL)-ZLY)935,935,934
934 ZLY=TLY(IOL)
935 IF(TUY(IOL)-ZUY)936,937,937
936 ZUY=TUY(IOL)
937 ZCA=(ZUX-ZLX)/48.*(ZUY-ZLY)/4.
    CU(1)=CU(1)+ZCA
1220 IF(CU(1)-COMP)29,63,63
    60 CONTINUE
        IND=1
        GO TO 401
    63 KOUNT(IJ)=KOUNT(IJ)+1
        IF(KOUNT(IJ)-25)57,57,56
    56 KOUNT(IJ)=25
        II=1
        GO TO 58
    57 II=0
    58 LN=(IX-IJ)*25
        LM=(IJ-2)*25
        LC=KOUNT(IJ)-1
        IF(LC)23,23,21
    21 DO 22 LJ=1,LC
        KI=LJ+LM
        JJ=LJ
        KJ=LJ+LN
        IF(CU(1)-AI(LJ+LM))22,24,24
    22 CONTINUE
        IF(II)23,23,29
    23 KI=KOUNT(IJ)+LM
        KJ=KOUNT(IJ)+LN
        GO TO 27
    24 KL=LC-(JJ-1)
        DO 26 KM=1,KL
        KN=KOUNT(IJ)-KM+LN
        KNN=KOUNT(IJ)-KM+LM
        DO 25 LI=1,IJ
        LK=(LI-1)*25

```

```

25 IA(KN+LK+1)=IA(KN+LK)
26 AI(KNN+1)=AI(KNN)
27 DO 28 MI=1,IJ
28 IA((MI-1)*25+KJ)=IB(MI)
   AI(KI)=CU(1)
29 IF(MNC-2)20,20,30
30 GO TO(31,31,32,34,36,38,39),IJ
31 K=J
46 K=K+1
   IF(K-NC)40,40,20
40 IB(3)=K
   CU(3)=CU(2)+CA(K)
   CU(1)=CU(3)
   IJ=3
   IX=5
   IF(CU(3)-COMP)29,64,64
32 IF(MNC-3)46,46,33
33 L=K
47 L=L+1
   IF(L-NC)41,41,46
41 IB(4)=L
   CU(4)=CU(3)+CA(L)
   CU(1)=CU(4)
   IJ=4
   IX=9
   IF(CU(4)-COMP)29,64,64
34 IF(MNC-4)47,47,35
35 M=L
48 M=M+1
   IF(M-NC)42,42,47
42 IB(5)=M
   CU(5)=CU(4)+CA(M)
   CU(1)=CU(5)
   IJ=5
   IX=14
   IF(CU(5)-COMP)29,64,64
36 IF(MNC-5)48,48,37
37 N=M
49 N=N+1
   IF(N-NC)43,43,48
43 IB(6)=N
   CU(6)=CU(5)+CA(N)
   CU(1)=CU(6)
   IJ=6
   IX=20
   IF(CU(6)-COMP)29,64,64
38 IF(MNC-6)49,49,50
50 MJ=N
39 MJ=MJ+1
   IF(MJ-NC)44,44,49
44 IB(7)=MJ
   CU(7)=CU(6)+CA(MJ)
   CU(1)=CU(7)
   IJ=7

```

```

IX=27
IF(CU(7)-COMP)29,3,3
45 J=KOUNT(2)
C4=0.
C3=0.
CB=0.
RCO=0.
IF(J-3)1203,1202,1202
1202 INI=3
GO TO 200
1203 INI=J
1206 IF(J)410,410,200
200 DO 400 I=1,INI
TA(1)=0.
TA(2)=0.
DO 99 K=1,49
99 SAVE(K)=-1.
IF(C4)124,124,600
124 IF(C3)1078,601,1078
601 IJ=IA(I)
IK=IA(I+25)
600 CONTINUE
1078 XL=TLX(IJ)-TLX(IK)
XU=TUX(IJ)-TUX(IK)
YL=TLY(IJ)-TLY(IK)
YU=TUY(IJ)-TUY(IK)
IF(CL(IJ)-CL(IK))201,202,202
201 NLC=IK
NSC=IJ
GO TO 203
202 NLC=IJ
NSC=IK
203 IF(CW(IJ)-CW(IK))204,205,205
204 NWC=IK
NNC=IJ
GO TO 206
205 NWC=IJ
NNC=IK
206 IF(XL*XU)220,215,207
207 NTYPE=1
IF(XL)210,209,209
209 TL=TUX(IJ)-TLX(IK)
GO TO 211
210 TL=TUX(IK)-TLX(IJ)
211 IF(YL)213,212,214
212 IF(YU)213,214,214
213 TW=TUY(IK)-TLY(IJ)
GO TO 223
214 TW=TUY(IJ)-TLY(IK)
GO TO 223
215 IF(YL*YU)220,216,220
216 NTYPE=1
GO TO 222
220 NTYPE=2

```

```

222 TL=CL(NLC)
    TW=CW(NWC)
223 IF(CW(IJ)-ACW(2)+1.)224,224,225
224 K=1
    GO TO 226
225 K=2
226 IF(CW(IK)-ACW(2)+1.)227,227,228
227 L=1
    GO TO 229
228 L=2
229 IF(K*L-2)230,241,256
230 IF(TL-ACL(1)*2.)231,234,234
231 IF(TW-ACW(1)*2.)232,233,233
232 NST=3
    GO TO 270
233 NST=5
    GO TO 270
234 IF(TW-ACW(1)*2.)238,235,235
235 IF(NLC-NWC)236,237,236
236 NST=10
    GO TO 270
237 NST=11
    GO TO 270
238 IF(NTYPE-1)239,239,240
239 NST=1
    GO TO 270
240 NST=13
    GO TO 270
241 IF(TL-ACL(1)-ACL(2))242,251,251
242 IF(TW-ACW(1)-ACW(2))246,243,243
243 IF(NTYPE-1)245,245,244
244 NST=12
    GO TO 270
245 IF(NSC-NNC)245,250,1245
1245 NST=9
    GO TO 270
246 IF(TW-ACW(1)*2.)247,248,248
247 NST=3
    GO TO 270
248 IF(CL(NWC)-ACL(1))247,249,249
249 IF(NTYPE-1)250,250,244
250 NST=5
    GO TO 270
251 IF(TW-ACW(1)-ACW(2))252,254,254
252 IF(TW-ACW(1)*2.)238,253,253
253 IF(CL(NWC)-ACL(1))238,254,254
254 IF(NTYPE-1)235,235,255
255 NST=14
    GO TO 270
256 IF(TL-ACL(2)*2.)262,257,257
257 IF(TW-ACW(2)*2.)258,254,254
258 IF(TW-ACW(1)*2.)259,254,254
259 IF(NTYPE-1)260,260,261
260 NST=7

```

```

GO TO 270
261 NST=13
GO TO 270
262 IF(TW-ACW(2)*2.)263,1262,1262
1262 IF(NTYPE-1)1263,1263,244
1263 IF(CL(NSC)-ACL(1))1245,250,250
263 IF(TW-ACW(1)*2.)264,265,265
264 NST=3
GO TO 270
265 IF(CL(IJ)-ACL(1))266,267,267
266 IF(CL(IK)-ACL(1))264,268,268
267 IF(CL(IK)-ACL(1))268,249,249
268 IF(TW-ACW(1)-ACW(2))269,1262,1262
269 NST=3
270 CONTINUE
GO TO(271,271,277,277,278,278,297,297,298,299,299,299,299),NST
271 CHECK=ACL(1)
N=2
272 IF(XL)274,273,275
273 IF(XU)274,274,275
274 NCB=IJ+50
NCA=IK+50
NN=1
MM=2
GO TO 276
275 NCB=IK+50
NCA=IJ+50
NN=2
MM=1
276 IF(NNC-(NCB-50))291,284,291
277 NS1=1
IF(C4)1385,1385,125
125 IXYZ=9
GO TO 86
278 CHECK=ACW(1)
N=1
279 IF(YL)281,280,282
280 IF(YU)281,282,282
281 NCB=IJ
NCA=IK
NN=1
MM=2
GO TO 283
282 NCB=IK
NCA=IJ
NN=2
MM=1
283 IF(NSC-NCB)291,284,291
284 WX(N)=TLY(NCB)+CHECK
IF(WX(N)-TLY(NCA))285,285,288
285 IF(NST-10)286,287,287
286 SAVE(NN,MM)=TUY(NCB)
TUY(NCB)=TLY(NCA)
GO TO 379

```

```

287 JJ=1
    GO TO 307
288 IF(NST-10)290,289,289
289 JJ=2
    GO TO 307
290 SAVE(NN,MM)=TUY(NCB)
    SAVE(MM,NN)=TLY(NCA)
    TUY(NCB)=WX(N)
    TLY(NCA)=WX(N)
    GO TO 379
291 WX(N)=TUY(NCA)-CHECK
    IF(WX(N)-TUY(NCB))292,294,294
292 IF(NST-10)290,293,293
293 JJ=3
    GO TO 307
294 IF(NST-10)296,295,295
295 JJ=4
    GO TO 307
296 SAVE(MM,NN)=TLY(NCA)
    TLY(NCA)=TUY(NCB)
    GO TO 379
297 CHECK=ACL(2)
    N=2
    GO TO 272
298 CHECK=ACW(2)
    N=1
    GO TO 279
299 IF(CW(NNC)-ACW(2))300,301,301
300 XLMIN=ACL(1)
    GO TO 302
301 XLMIN=ACL(2)
302 IF(CL(NSC)-ACL(1))304,303,303
303 WMIN=ACW(1)
    GO TO 305
304 WMIN=ACW(2)
305 IF(NST-11)306,306,334
306 CHECK=XLMIN
    N=2
    KK=0
    GO TO 272
307 KK=KK+1
    IF(KK-2)308,309,309
308 CHECK=WMIN
    II=JJ
    N=1
    NCBS=NCB
    NCAS=NCA
    NNS=NN
    MMS=MM
    GO TO 279
309 GO TO(310,311,311,314),II
310 XLMIN=ABS(XL)
311 IF(NST-10)312,312,313
312 AA=(CL(NNC)-XLMIN)*(TW-CW(NSC))

```

```

      GO TO 315
313 AA=(CL(NNC)-XLMIN)*(TW-CW(NLC))
      GO TO 315
314 XLMIN=ABS(XU)
      GO TO 311
315 GO TO(319,316,316,319),II
316 IF(NST-10)317,317,318
317 AB=(XLMIN-(TL-CL(NWC)))*(TW-CW(NLC))
      GO TO 320
318 AB=(XLMIN-(TL-CL(NWC)))*(TW-CW(NSC))
      GO TO 320
319 AB=0.
320 GO TO(321,322,322,325),JJ
321 WMIN=ABS(YL)
322 IF(NST-10)323,323,324
323 AC=(TL-CL(NNC))*(CW(NSC)-WMIN)
      GO TO 326
324 AC=(TL-CL(NWC))*(CW(NSC)-WMIN)
      GO TO 326
325 WMIN=ABS(YU)
      GO TO 322
326 GO TO(330,327,327,330),JJ
327 IF(NST-10)328,328,329
328 AD=(TL-CL(NWC))*(CW(NLC)-(TW-WMIN))
      GO TO 860
329 AD=(TL-CL(NNC))*(WMIN-(TW-CW(NLC)))
      GO TO 860
330 AD=0.
860 IF(NST-11)331,861,331
861 CONTINUE
      IF(CW(IJ)-CW(IK))978,236,978
978 EXL=XLMIN-TL+CL(NWC)
      IF(EXL-ACL(2))331,979,979
979 IF(XLMIN-ACL(1))331,878,331
878 AF=CW(NLC)*(TL-XLMIN)
      AG=TW*(XLMIN-(TL-CL(NWC)))
      IF(C3)982,982,981
981 IF(L1-2)984,986,982
986 IF(CB)1200,1200,984
1200 IF(NO1-2)983,983,982
984 JI=IK
      JL=IJ
      M1=1
      RCO=1.
      GO TO 752
983 AH=(CA(IJ)+CA(IK)-OL(IK,IJ))*192.-(AF+AG)
      RCO=0.
      GO TO 985
982 AH=(CA(IJ)+CA(IK)-OL(IJ,IK))*192.-(AF+AG)
985 CONTINUE
      IF(AH-(AA+AB))862,331,331
862 IF(AH-(AC+AD))863,332,332
863 NCBS=NCBS-50
      NCAS=NCAS-50

```

```

      IF (II-2) 864, 865, 866
866  IF (II-3) 863, 865, 867
864  SAVE (NNS, MMS) = TUX (NCBS)
      TUX (NCBS) = TLX (NCAS)
      GO TO 868
865  IF (NNC-NCAS) 1061, 1060, 1061
1060 SAVE (1, 4) = TUX (NCAS)
      TUX (NCAS) = TUX (NCBS)
      GO TO 1062
1061 SAVE (4, 1) = TLX (NCBS)
      TLX (NCBS) = TLX (NCAS)
1062 SAVE (MMS, NNS) = TLX (NCAS)
      SAVE (NNS, MMS) = TUX (NCBS)
      TUX (NCBS) = WX (2)
      TLX (NCAS) = WX (2)
      GO TO 868
867  SAVE (MMS, NNS) = TLX (NCAS)
      TLX (NCAS) = TUX (NCBS)
868  IF (JJ-2) 869, 869, 870
869  SAVE (1, 3) = TUY (NCB)
      TUY (NCB) = TUY (NCA)
      GO TO 871
870  SAVE (3, 1) = TLY (NCA)
      TLY (NCA) = TLY (NCB)
871  CONTINUE
      N=2
      GO TO 379
331  IF ((AA+AB) - (AC+AD)) 333, 332, 332
332  N=1
      GO TO (286, 290, 290, 296), JJ
333  NCB=NCBS
      NCA=NCAS
      NN=NNS
      MM=MMS
      N=2
      GO TO (286, 290, 290, 296), II
334  IF (NST-13) 335, 366, 335
335  IJJ=NSC
      IKK=NLC
      CHECK=WMIN
      K=0
      M=0
      N=1
      IF (IJJ-IJ) 337, 336, 337
336  NM=1
      MN=2
      GO TO 338
337  NM=2
      MN=1
338  IF (ABS(YU) - ABS(YL)) 346, 339, 339
339  NN=1
      WX(N) = TUY (IJJ) - CHECK
      IF (WX(N) - TUY (IKK)) 345, 340, 340
340  WX(N) = TUY (IKK)

```

```

        IF(NST-14)344,341,341
341 IF(N-1)343,343,342
342 TA(2)=CW(NNC)*ABS(XU)/192.+CA(NWC)
        GO TO 371
343 TA(1)=CL(NSC)*ABS(YU)/192.+CA(NLC)
        MM=NN
        GO TO 366
344 SAVE(NM,MN)=TLY(IJJ)
        TLY(IJJ)=TUY(IKK)
        GO TO 379
345 WR=WX(N)-TLY(IKK)
        GO TO 354
346 NN=2
        WX(N)=TLY(IJJ)+CHECK
        IF(WX(N)-TLY(IKK))348,348,353
348 WX(N)=TLY(IKK)
        IF(NST-14)352,349,349
349 IF(N-1)350,350,351
350 TA(1)=CL(NSC)*ABS(YL)/192.+CA(NLC)
        MM=NN
        GO TO 366
351 TA(2)=CW(NNC)*ABS(XL)/192.+CA(NWC)
        GO TO 371
352 SAVE(NM,MN)=TUY(IJJ)
        TUY(IJJ)=TLY(IKK)
        GO TO 379
353 WR=TUY(IKK)-WX(N)
354 IF(WR-ACW(K+2))355,361,361
355 IF(WR-ACW(3*K+1))357,356,356
356 IF(CL(IKK-M)-ACL(1-K))357,361,361
357 IF(NST-14)277,358,358
358 IF(N-1)359,359,360
359 NST=13
        GO TO 366
360 TA(2)=0.
        NST=12
        GO TO 371
361 TA(N)=(CL(IJJ-M)*CHECK+CL(IKK-M)*WR)/192.
        IF(NST-14)362,347,347
347 IF(N-1)398,398,371
398 MM=NN
        GO TO 366
362 IF(TA(N)-AMAX1(CA(IJ),CA'      )277,277
363 IF(NN-1)364,364,365
364 SAVE(NM,MN)=TLY(IJJ)
        SAVE(MN,NM)=TUY(IKK)
        TLY(IJJ)=WX(N)
        TUY(IKK)=WX(N)
        GO TO 1009
365 SAVE(NM,MN)=TUY(IJJ)
        SAVE(MN,NM)=TLY(IKK)
        TUY(IJJ)=WX(N)
        TLY(IKK)=WX(N)
        GO TO 1009

```

```

366 IJJ=NNC+50
    IKK=NWC+50
    CHECK=XLMIN
    N=2
    K=1
    M=25
    IF(NNC-1J)368,367,368
367 NM=1
    MN=2
    GO TO 369
368 NM=2
    MN=1
369 IF(ABS(XU)-ABS(XL))346,339,339
371 IF(AMAX1(TA(1),TA(2))-AMAX1(CA(IJ),CA(IK)))277,277,372
372 IF(TA(1)-TA(2))376,373,373
373 N=1
    NN=MM
    IF(NN-1)375,375,374
374 SAVE(MN,NM)=TUY(NSC)
    SAVE(NM,MN)=TLY(NLC)
    TUY(NSC)=WX(1)
    TLY(NLC)=WX(1)
    GO TO 1009
375 SAVE(MN,NM)=TLY(NSC)
    SAVE(NM,MN)=TUY(NLC)
    TLY(NSC)=WX(1)
    TUY(NLC)=WX(1)
    GO TO 1009
376 N=2
    IF(NN-1)377,377,378
377 SAVE(NM,MN)=TLX(NNC)
    SAVE(MN,NM)=TUX(NWC)
    TLX(NNC)=WX(2)
    TUX(NWC)=WX(2)
    GO TO 1009
378 SAVE(NM,MN)=TUX(NNC)
    SAVE(MN,NM)=TLX(NWC)
    TUX(NNC)=WX(2)
    TLX(NWC)=WX(2)
1009 CU(1)=TA(N)
    GO TO 1000
379 CU(1)=(TUY(IJ)-TLY(IJ))*(TUX(IJ)-TLX(IJ))/192.
    CU(1)=CU(1)+(TUY(IK)-TLY(IK))*(TUX(IK)-TLX(IK))/192.
    IF(C3)1000,1000,400
1000 IF(CU(1)-COMP)404,380,380
404 CONTINUE
    IF(C4)116,116,129
129 IF(C3)117,117,403
117 IXYZ=7
    GO TO 86
116 IF(C3)1362,1362,403
1362 NS1=1
    IF(CU(1)-CCOMP)403,1360,1360
1360 IF(CU(1)-BC2)403,403,1361

```

```

1361 BC2=CU(1)
      GO TO 403
380 CONTINUE
403 CONTINUE
      IF(C4)118,118,130
130 IF(C3)119,119,400
119 IXYZ=8
      GO TO 86
118 IF(C3)877,877,400
877 IF(NST-11)381,873,381
873 IF(SAVE(1,3))875,874,874
874 TUY(NCB)=SAVE(1,3)
875 IF(SAVE(3,1))1063,876,876
876 TLY(NCA)=SAVE(3,1)
1063 IF(SAVE(1,4))1065,1064,1064
1064 TUX(NCAS)=SAVE(1,4)
1065 IF(SAVE(4,1))381,1066,1066
1066 TLX(NCBS)=SAVE(4,1)
381 IF(N-1)382,382,390
382 IF(SAVE(1,2))386,383,383
383 IF(TLY(IJ)-SAVE(1,2))385,384,384
384 TLY(IJ)=SAVE(1,2)
      GO TO 386
385 TUY(IJ)=SAVE(1,2)
386 IF(SAVE(2,1))402,387,387
387 IF(TLY(IK)-SAVE(2,1))389,388,388
388 TLY(IK)=SAVE(2,1)
      GO TO 402
389 TUY(IK)=SAVE(2,1)
      GO TO 402
390 IF(SAVE(1,2))394,391,391
391 IF(TLX(IJ)-SAVE(1,2))392,393,393
392 TUX(IJ)=SAVE(1,2)
      GO TO 394
393 TLX(IJ)=SAVE(1,2)
394 IF(SAVE(2,1))402,395,395
395 IF(TLX(IK)-SAVE(2,1))397,396,396
396 TLX(IK)=SAVE(2,1)
      GO TO 402
397 TUX(IK)=SAVE(2,1)
402 DO 399 K=1,49
399 SAVE(K)=-1.
1385 IF(NS1)1353,1353,1355
1353 CONTINUE
      IND=1
      GO TO 401
1355 IF(I-IND)1356,400,400
1356 NS1=0
400 CONTINUE
      IF(CB)410,410,844
410 IF(MNC-3)401,411,411
411 IF(C3)654,412,654
412 IF(KOUNT(3))716,716,718
716 CONTINUE

```

```

      GO TO 3000
718  IF(KOUNT(3)-3)1350,1350,1381
1350 K3C=KOUNT(3)
      GO TO 1352
1381 K3C=3
1352 DO 1032 I3=1,K3C
      IF(C4)122,122,123
122  IJ=IA(I3+50)
      IK=IA(I3+75)
      IKK=IA(I3+100)
123  K1=0
      K2=0
      K3=0
      N01=0
      CN=0.
      CB=0.
      N=0
      NL1=0
      NS1=0
      NLIS=0
      DO 715 J=1,25
      WLX(J)=0.
      WUX(J)=0.
      WLY(J)=0.
      WUY(J)=0.
      WL(J)=0.
      WW(J)=0.
715  WA(J)=0.
      DO 1074 J=1,49
1074 SAVE(J)=-1.
      IF(OL(IJ,IK))413,413,414
414  K1=1
413  IF(OL(IJ,IKK))417,417,415
415  K2=1
417  IF(OL(IK,IKK))418,418,416
416  K3=1
418  IF(K1+K2+K3-2)419,420,494
419  CONTINUE
      L1=3
      GO TO 496
494  CONTINUE
      L1=2
      D=0.
      GO TO 433
420  CONTINUE
      D=0.
      D1=0.
      D2=0.
      L1=1
496  IF(K1)421,421,422
421  IF(L1-1)424,423,424
423  A1=1.
      GO TO 432
422  IF(L1-3)424,1052,424

```

```

1052 A1=1.
      GO TO 434
424 IF(K2)426,426,429
426 IF(L1-2)427,428,428
427 A1=2.
      GO TO 432
428 A1=3.
      GO TO 434
429 IF(L1-1)431,430,431
431 A1=2.
      GO TO 434
430 A1=3.
432 IF(A1-2.)464,470,465
464 IF(OL(IJ,IKK)-OL(IK,IKK))467,466,466
466 IE=IKK
      IG=IK
      IH=IJ
      IK=IKK
      GO TO 854
467 IE=IKK
      IG=IJ
      IH=IK
      IJ=IK
      IK=IKK
      GO TO 854
465 IF(OL(IJ,IK)-OL(IJ,IKK))468,469,469
468 IE=IJ
      IG=IK
      IH=IKK
      IK=IKK
      GO TO 854
469 IE=IJ
      IG=IKK
      IH=IK
      GO TO 854
470 IF(OL(IJ,IK)-OL(IK,IKK))472,471,471
472 IE=IK
      IG=IJ
      IH=IKK
      IJ=IK
      IK=IKK
      GO TO 854
471 IE=IK
      IG=IKK
      IH=IJ
      GO TO 854
473 FX=TUX(IG)-TLX(IG)
      IF(FX-ACL(1))474,475,475
474 XMIN=ACL(2)
      YMIN=ACW(2)
      GO TO 659
475 XMIN=ACL(1)
      YMIN=ACW(1)
659 IF(L1-1)763,763,764

```

```

763 T1=TUY(IE)-TLY(IG)
    T2=TUY(IG)-TLY(IE)
764 T3=TUX(IE)-TLX(IG)
    T4=TUX(IG)-TLX(IE)
    NO1=0
    H=0.
    E=0.
    IF(D1-D2)476,1300,477
1300 IF(L1-1)762,762,477
762 IF(T1-T2)477,476,476
476 WLY(3)=TLY(IG)
    WUY(3)=TLY(IG)+YMIN
    TLY(IG)=WUY(3)
    IF(WUY(3)-TLY(IE))478,478,479
478 IF(L1-1)653,653,665
665 NO1=1
    NL1=0
    GO TO 691
479 FY1=TUY(IE)-WUY(3)
    FX1=CL(IE)
    AREA1=FX1*FY1
    ILL=1
    NL1=1
    GO TO 603
477 WUY(3)=TUY(IG)
    WLY(3)=TUY(IG)-YMIN
    TUY(IG)=WLY(3)
    IF(WLY(3)-TUY(IE))481,482,482
482 IF(L1-1)653,653,665
481 FY1=WLY(3)-TLY(IE)
    FX1=CL(IE)
    AREA1=FX1*FY1
    ILL=1
    NL1=2
    GO TO 603
490 IF(NL1-1)480,480,483
480 WLY(1)=WUY(3)
    WUY(1)=TUY(IE)
    GO TO 484
483 WLY(1)=TLY(IE)
    WUY(1)=WLY(3)
484 WLX(1)=TLX(IE)
    WUX(1)=TUX(IE)
713 IF(T3-T4)486,485,485
485 WUX(3)=TLX(IG)+XMIN
    WLX(3)=TLX(IG)
    IF(WUX(3)-TLX(IE))482,482,1302
1302 FX1=TUX(IE)-WUX(3)
    FY1=CW(IE)
    AREA2=FX1*FY1
    ILL=2
    NL1=NL1+1
    GO TO 603
486 WLX(3)=TUX(IG)-XMIN

```

```

      WUX(3)=TUX(IG)
      IF(WLX(3)-TUX(IE))1304,482,482
1304 FX1=WLX(3)-TLX(IE)
      FY1=CW(IE)
      AREA2=FX1*FY1
      ILL=2
      NL1=NL1+3
      GO TO 603
487 IF(NL1-3)488,488,489
488 WLX(2)=WUX(3)
      WUX(2)=TUX(IE)
      GO TO 667
489 WLX(2)=TLX(IE)
      WUX(2)=WLX(3)
667 WLY(2)=TLY(IE)
      WUY(2)=TUY(IE)
668 IF(WUX(1))492,492,491
491 NK1=1
      GO TO 493
492 NK1=0
493 IF(WUX(2))684,684,683
683 NK1=NK1+1
684 IF(NK1-1)690,686,685
685 IF(AREA1-AEA2)689,689,687
687 TLX(IE)=WLX(1)
      TUX(IE)=WUX(1)
      TLY(IE)=WLY(1)
      TUY(IE)=WUY(1)
      CA(IE)=AREA1/192.
      CW(IE)=TUY(IE)-TLY(IE)
728 IF(NL1-3)725,726,727
725 NL1=1
      GO TO 732
726 NL1=2
732 CONTINUE
      NL1S=NL1
      GO TO 688
727 IF(NL1-4)728,725,726
688 IF(L1-2)653,691,653
689 TLX(IE)=WLX(2)
      TUX(IE)=WUX(2)
      TLY(IE)=WLY(2)
      TUY(IE)=WUY(2)
      CA(IE)=AREA2/192.
      CL(IE)=TUX(IE)-TLX(IE)
729 IF(NL1-3)730,730,731
730 NL1=3
      GO TO 733
731 NL1=4
733 CONTINUE
      NL1S=NL1
      GO TO 688
686 IF(WUX(1))689,689,687
690 CONTINUE

```

```

      E=1.
      IF(L1-2)1301,691,1301
1301 NS1=1
      IF(C4)1017,1017,126
      126 IXYZ=10
      GO TO 86
      691 IF(D1-D2)1311,1310,1310
1310 IF(WLY(3)-TUY(IH))1312,780,780
1312 FY1=WLY(3)-TLY(IH)
      NL1=2
1314 FX1=CL(IH)
      AREA3=FX1*FY1
      ILL=3
      MSC=0
      GO TO 603
1311 IF(WUY(3)-TLY(IH))780,780,1313
      780 NO1=NO1+2
      NL1=0
      GO TO 602
1313 FY1=TUY(IH)-WUY(3)
      NL1=1
      GO TO 1314
      702 IF(T3-T4)1315,1316,1316
1315 WLX(3)=TUX(IG)-XMIN
      WUX(3)=TUX(IG)
      IF(WLX(3)-TUX(IH))1341,780,780
1341 FX1=WLX(3)-TLX(IH)
      NL1=NL1+3
1317 FY1=CW(IH)
      AREA4=FX1*FY1
      ILL=4
      GO TO 603
1316 WUX(3)=TLX(IG)+XMIN
      WLX(3)=TLX(IG)
      IF(WUX(3)-TLX(IH))780,780,1342
1342 FX1=TUX(IH)-WUX(3)
      NL1=NL1+1
      GO TO 1317
1318 IF(MSC-1)655,1320,1319
      655 CONTINUE
      H=1.
      GO TO 602
1319 IF(MSC-2)1318,1321,1322
1320 IF(NL1-3)1323,1324,1325
1323 TLY(IH)=WUY(3)
1326 CW(IH)=TUY(IH)-TLY(IH)
1329 CA(IH)=CW(IH)/4.*CL(IH)/48.
      GO TO 602
1324 TUY(IH)=WLY(3)
      GO TO 1326
1325 IF(NL1-4)1320,1323,1324
1321 IF(NL1-3)1327,1327,1328
1327 TLX(IH)=WUX(3)
1330 CL(IH)=TUX(IH)-TLX(IH)

```

```

GO TO 1329
1328 TUX(IH)=WLX(3)
GO TO 1330
1322 IF(AREA3-AREA4)1321,1320,1320
434 IF(A1-2.)435,436,437
435 IG=IKK
GO TO 1058
436 IG=IK
IK=IKK
GO TO 1058
437 IG=IJ
IJ=IK
IK=IKK
1058 DO 1053 J=4,6
IF(J-5)1054,1055,1056
1054 JNC=IJ
GO TO 1057
1055 JNC=IK
GO TO 1057
1056 JNC=IG
1057 WLX(J)=TLX(JNC)
WUX(J)=TUX(JNC)
WLY(J)=TLY(JNC)
WUY(J)=TUY(JNC)
WL(J)=CL(JNC)
WW(J)=CW(JNC)
1053 WA(J)=CA(JNC)
GO TO 653
433 IF(TUY(IJ)-TUY(IK))438,440,439
438 IF(TUY(IJ)-TUY(IKK))442,441,441
441 D1=TUY(IK)-TUY(IJ)
IE1=IJ
IG1=IK
IH1=IKK
IJ1=IJ
IK1=IKK
GO TO 449
442 IF(TUY(IK)-TUY(IKK))444,765,443
765 D=1.
IF(CW(IK)-CW(IKK))441,441,448
443 D1=TUY(IK)-TUY(IKK)
IE1=IKK
IG1=IK
IH1=IJ
IJ1=IJ
IK1=IKK
GO TO 449
444 D1=TUY(IKK)-TUY(IK)
IE1=IK
IG1=IKK
IH1=IJ
IJ1=IJ
IK1=IK
GO TO 449

```

```

440 IF(TUY(IJ)-TUY(IKK))448,770,771
770 D1=0.
      D=1.
      GO TO 449
771 D=1.
      IF(CW(IJ)-CW(IK))447,447,443
439 IF(TUY(IK)-TUY(IKK))446,445,445
445 D1=TUY(IJ)-TUY(IK)
      IE1=IK
      IG1=IJ
      IH1=IKK
      IJ1=IK
      IK1=IKK
      GO TO 449
446 IF(TUY(IJ)-TUY(IKK))448,772,447
772 D=1.
      IF(CW(IJ)-CW(IKK))445,445,444
447 D1=TUY(IJ)-TUY(IKK)
      IE1=IKK
      IG1=IJ
      IH1=IK
      IJ1=IK
      IK1=IKK
      GO TO 449
448 D1=TUY(IKK)-TUY(IJ)
      IE1=IJ
      IG1=IKK
      IH1=IK
      IJ1=IJ
      IK1=IK
449 IF(TLY(IJ)-TLY(IK))450,452,451
450 IF(TLY(IJ)-TLY(IKK))454,773,453
773 D=D+2.
      IF(CW(IJ)-CW(IKK))456,456,457
453 D2=TLY(IJ)-TLY(IKK)
      IE2=IJ
      IG2=IKK
      IH2=IK
      IJ2=IJ
      IK2=IK
      GO TO 461
452 IF(TLY(IJ)-TLY(IKK))775,774,453
774 D2=0.
      D=D+2.
      GO TO 461
775 D=D+2.
      IF(CW(IJ)-CW(IK))455,455,459
454 IF(TLY(IK)-TLY(IKK))456,456,455
455 D2=TLY(IKK)-TLY(IJ)
      IE2=IKK
      IG2=IJ
      IH2=IK
      IJ2=IK
      IK2=IKK

```

```

      GO TO 461
456 D2=TLY(IK)-TLY(IJ)
      IE2=IK
      IG2=IJ
      IH2=IKK
      IJ2=IK
      IK2=IKK
      GO TO 461
451 IF(TLY(IK)-TLY(IKK))458,776,457
776 D=D+2.
      IF(CW(IK)-CW(IKK))460,460,453
457 D2=TLY(IK)-TLY(IKK)
      IE2=IK
      IG2=IKK
      IH2=IJ
      IJ2=IJ
      IK2=IK
      GO TO 461
458 IF(TLY(IJ)-TLY(IKK))460,460,459
459 D2=TLY(IKK)-TLY(IK)
      IE2=IKK
      IG2=IK
      IH2=IJ
      IJ2=IJ
      IK2=IKK
      GO TO 461
460 D2=TLY(IJ)-TLY(IK)
      IE2=IJ
      IG2=IK
      IH2=IKK
      IJ2=IJ
      IK2=IKK
461 IF(D1-D2)462,766,463
766 IF(D1)767,767,463
767 CONTINUE
      NS1=1
      IF(C4)1357,1357,127
127 IXYZ=11
      GO TO 86
462 IG=IG2
      IE=IE2
      IJ=IJ2
      IK=IK2
      IH=IH2
854 DO 849 J=4,6
      IF(J-5)850,851,852
850 KI=IG
      GO TO 853
851 KI=IE
      GO TO 853
852 KI=IH
853 WLX(J)=TLX(KI)
      WUX(J)=TUX(KI)
      WLY(J)=TLY(KI)

```

```

      WUY(J)=TUY(KI)
      WL(J)=CL(KI)
      WW(J)=CW(KI)
849  WA(J)=CA(KI)
      GO TO 473
463  IG=IG1
      IE=IE1
      IJ=IJ1
      IK=IK1
      IH=IH1
      GO TO 854
602  IF(WW(4)-WW(5))786,785,785
785  IF(WW(4)-WW(6))786,797,797
786  CN=1.
      GO TO 799
797  CN=2.
      GO TO 800
799  IF(E+H-1.)802,801,801
801  NS1=1
      IF(C4)1017,1017,128
128  IXYZ=12
      GO TO 86
802  IF(N01-2)746,746,653
800  IF(E+H-1.)805,806,801
805  IF(N01-2)746,746,808
746  JI=IK
      JL=IJ
      JM=0
      T=48.
      M1=1
      N1=1
      GO TO 612
808  N1=0
      GO TO 610
806  IF(E)810,810,811
810  IF(IE-IG)803,803,804
803  IJ=IE
      IK=IG
      JI=IG
      JL=IE
812  M1=1
      GO TO 632
804  IJ=IG
      IK=IE
      JI=IE
      JL=IG
      GO TO 812
811  IF(IH-IG)813,813,814
813  IJ=IH
      IK=IG
      JI=IG
      JL=IH
      GO TO 812
814  IJ=IG

```

```

      IK=IH
      JI=IH
      JL=IG
      GO TO 812
610 CU(3)=CA(IJ)+CA(IK)-OL(IJ,IK)
632 CW(IG)=TUY(IG)-TLY(IG)
      IF(CL(IG)-ACL(1))633,634,634
633 IF(CW(IG)-ACW(2))815,636,636
634 IF(CW(IG)-ACW(1))815,636,636
815 IF(E+H)653,653,820
820 CONTINUE
      GO TO 801
636 CA(IG)=CW(IG)/4.*CL(IG)/48.
816 IF(E+H-1.)750,752,801
750 IF(IG-IJ)676,675,675
675 JI=IG
      JL=IJ
      GO TO 677
676 JI=IJ
      JL=IG
677 M1=2
752 JM=0
      T=48.
612 DO 674 J=1,M1
613 IL=JI-JM*2
      JN=JL-JM*2
      XL=TLX(IL)-TLX(JN)
      XU=TUX(IL)-TUX(JN)
      IF(XL*XU)614,614,615
614 IF(XL)617,616,618
616 IF(XU)618,618,617
617 OLL=CL(JL+JM)/T
      GO TO 625
618 OLL=CL(JI+JM)/T
      GO TO 625
615 IF(XL)620,620,619
620 XZ=TUX(IL)-TLX(JN)
      IF(XZ)622,622,621
619 XZ=TUX(JN)-TLX(IL)
      IF(XZ)622,622,621
621 OLL=XZ/T
      GO TO 625
622 OL(JI,JL)=0.
      GO TO 630
625 IF(JM-25)626,627,627
626 X=OLL
      JM=25
      T=4.
      GO TO 613
627 OL(JI,JL)=X*OLL
630 CU(J)=CA(JI)+CA(JL)-OL(JI,JL)
      IF(RCO)809,809,983
809 IF(CN-1.)983,653,977
977 IF(E+H-1.)670,653,801

```

```

670 IF(IG-IK)679,678,678
678 JI=IG
      JL=IK
      GO TO 680
679 JI=IK
      JL=IG
680 JM=0
      T=48.
674 CONTINUE
756 IF(N1)631,642,631
631 N1=0
      CU(3)=CU(1)
      GO TO 632
603 IF(FX1-ACL(1))637,638,638
637 IF(FX1-ACL(2))708,639,639
639 IF(FY1-ACW(2))708,709,709
709 IF(ILL-2)490,487,1343
1343 IF(ILL-3)709,1344,1345
1344 MSC=1
      GO TO 702
1345 MSC=MSC+2
      GO TO 1318
638 IF(FY1-ACW(1))708,709,709
708 IF(ILL-2)713,668,1346
1346 IF(ILL-3)708,702,1318
642 IF(CU(1)-CU(2))643,645,644
644 IF(CU(1)-CU(3))649,650,650
650 IF(IG-IJ)652,651,651
651 IKK=IK
      IK=IG
      GO TO 653
652 IKK=IK
      IK=IJ
      IJ=IG
      GO TO 653
643 IF(CU(2)-CU(3))649,646,646
646 IF(IG-IK)647,648,648
647 IKK=IJ
      IJ=IG
653 C3=1.
      J=1
      INI=1
      GO TO 200
648 IKK=IJ
      IJ=IK
      IK=IG
      GO TO 653
649 IKK=IG
      GO TO 653
645 IF(CU(3)-CU(1))650,649,649
654 IF(C4)529,529,528
528 ND=ND+3
      NFD(1)=ND
      NFF=3

```

```

      GO TO 530
529 ND=ND+2
      NFD(1)=ND
      NFF=2
530 IF(NS1)1079,1079,1051
1051 CONTINUE
      IF(C4)1017,1017,84
      84 IXYZ=1
      GO TO 86
1079 DO 682 K=1,NFF
      IF(K-1)681,681,531
531 IF(K-2)657,657,532
681 JK1=IJ
      ND2=ND-1
      GO TO 658
657 JK1=IK
      ND2=ND
      GO TO 658
532 JK1=IU
      ND2=ND-2
658 FLX(ND2,NF)=TLX(JK1)
      FUX(ND2,NF)=TUX(JK1)
      FLY(ND2,NF)=TLY(JK1)
      FUY(ND2,NF)=TUY(JK1)
      CL(JK1)=TUX(JK1)-TLX(JK1)
      CW(JK1)=TUY(JK1)-TLY(JK1)
682 CA(JK1)=CL(JK1)/48.*CW(JK1)/4.
909 NA1=IJ
      DO 830 J=7,8
      WLX(J)=TLX(NA1)
      WUX(J)=TUX(NA1)
      WLY(J)=TLY(NA1)
      WUY(J)=TUY(NA1)
      WL(J)=CL(NA1)
      WW(J)=CW(NA1)
      WA(J)=CA(NA1)
      IF(J-7)827,827,830
827 NA1=IK
830 CONTINUE
      IF(L1-2)989,1080,1015
1015 NN1=IG
      GO TO 1006
1080 CAO=CA(IJ)+CA(IK)
      GO TO 989
886 IF(NST-10)1059,825,824
1059 CB=2.
      GO TO 1042
824 IF(NST-13)825,1059,825
825 IF(N-1)822,826,822
822 CONTINUE
      GO TO 1059
826 IF(CN-1.)1072,1072,905
905 IF(IG-IJ)907,906,907
906 CONTINUE

```

```

      GO TO 1059
907 IF (IG-IK) 1072,908,1072
1072 IF (NLIS-3) 940,938,939
938 IF (TLX(IJ)-TLX(IK)) 941,943,942
941 KB=IJ
      IKK=IK
      GO TO 955
942 KB=IK
      IKK=IJ
      GO TO 955
943 IF (WLX(5)-WLX(6)) 944,944,945
944 KB=IE
      IKK=IH
      KN=5
      GO TO 946
945 KB=IH
      IKK=IE
      KN=6
946 TLX(KB)=WLX(KN)
954 CL(KB)=TUX(KB)-TLX(KB)
      CA(KB)=CL(KB)/48.*CW(KB)/4.
      IF (WUX(4)-TLX(IKK)) 948,948,947
947 TUX(IG)=TLX(IKK)
949 TLX(IG)=WLX(4)
      CL(IG)=TUX(IG)-TLX(IG)
      TUY(IG)=WUY(4)
      TLY(IG)=WLY(4)
      CW(IG)=WW(4)
      CA(IG)=CL(IG)/48.*CW(IG)/4.
      GO TO 950
948 TUX(IG)=WUX(4)
      GO TO 949
950 IF (KB-IG) 951,952,952
951 IJ=KB
      IK=IG
      GO TO 832
952 IJ=IG
      IK=KB
      GO TO 832
955 IF (KB-IE) 956,957,956
956 KN=6
      GO TO 946
957 KN=5
      GO TO 946
939 IF (TUX(IJ)-TUX(IK)) 958,960,959
958 KB=IK
      IKK=IJ
      GO TO 967
959 KB=IJ
      IKK=IK
      GO TO 967
960 IF (WUX(5)-WUX(6)) 961,961,962
961 KB=IH
      KN=6

```

```

      IKK=IE
      GO TO 963
962  KB=IE
      KN=5
      IKK=IH
963  TUX(KB)=WUX(KN)
      CL(KB)=TUX(KB)-TLX(KB)
      CA(KB)=CL(KB)/48.*CW(KB)/4.
      IF(WLX(4)-TUX(IKK))964,965,965
964  TLX(IG)=TUX(IKK)
      GO TO 966
965  TLX(IG)=WLX(4)
966  TUX(IG)=WUX(4)
      CL(IG)=TUX(IG)-TLX(IG)
      TUY(IG)=WUY(4)
      TLY(IG)=WLY(4)
      CW(IG)=WW(4)
      CA(IG)=CL(IG)/48.*CW(IG)/4.
      GO TO 950
967  IF(KB-IE)968,969,968
968  KN=6
      GO TO 963
969  KN=5
      GO TO 963
940  IF(D1-D2)828,829,829
829  IF(TUY(IJ)-TUY(IK))831,879,879
879  IKK=IK
      IF(IJ-IG)833,834,834
833  IK=IG
      KB=IJ
835  IF(WLY(4)-TUY(IKK))890,891,891
890  TLY(IG)=TUY(IKK)
      TUY(IG)=WUY(4)
897  CW(IG)=TUY(IG)-TLY(IG)
      TLX(IG)=WLX(4)
      TUX(IG)=WUX(4)
      CL(IG)=WL(4)
      CA(IG)=CL(IG)/48.*CW(IG)/4.
      GO TO 832
891  TLX(IG)=WLX(4)
      TUX(IG)=WUX(4)
      TLY(IG)=WLY(4)
      TUY(IG)=WUY(4)
      CW(IG)=WW(4)
      CL(IG)=WL(4)
      CA(IG)=WA(4)
      GO TO 832
834  IK=IJ
      IJ=IG
      KB=IK
      GO TO 835
831  IKK=IJ
      IF(IK-IG)836,837,837
836  IJ=IK

```

```

      IK=IG
      KB=IJ
      GO TO 835
837 IJ=IG
      KB=IK
      GO TO 835
828 IF(TLY(IJ)-TLY(IK))838,839,839
838 IKK=IK
      IF(IJ-IG)840,840,841
840 IK=IG
      KB=IJ
895 IF(WUY(4)-TLY(IKK))891,891,896
896 TUY(IG)=TLY(IKK)
      TLY(IG)=WLY(4)
      GO TO 897
841 IK=IJ
      IJ=IG
      KB=IK
      GO TO 895
839 IKK=IJ
      IF(IK-IG)842,843,843
842 IJ=IK
      IK=IG
      KB=IJ
      GO TO 895
843 IJ=IG
      KB=IK
      GO TO 895
832 WLX(12)=TLX(IKK)
      WUX(12)=TUX(IKK)
      WLY(12)=TLY(IKK)
      WUY(12)=TUY(IKK)
      WL(12)=CL(IKK)
      WW(12)=CW(IKK)
      WA(12)=CA(IKK)
      CB=1.
      IF(NL1S-3)953,974,974
953 IF(KB-IE)899,898,899
898 KN=5
      GO TO 900
899 KN=6
900 IF(D1-D2)901,902,902
901 TLY(KB)=WLY(KN)
      GO TO 973
914 IF(WL(4)-WL(KN))917,917,916
917 CONTINUE
      GO TO 1059
916 CONTINUE
      GO TO 653
902 TUY(KB)=WUY(KN)
973 CW(KB)=TUY(KB)-TLY(KB)
      CA(KB)=CL(KB)/48.*CW(KB)/4.
974 CONTINUE
      GO TO 914

```

```

844 IF(NS1)912,912,910
910 NS1=0
    GO TO 1059
912 DO 880 J=1,2
    IF(J-1)881,881,882
881 KI=IJ
    GO TO 883
882 KI=IK
883 CL(KI)=TUX(KI)-TLX(KI)
    CW(KI)=TUY(KI)-TLY(KI)
880 CA(KI)=CL(KI)/48.*CW(KI)/4.
    CAN=WA(12)+CA(IJ)+CA(IK)
    DO 848 J=10,11
    IF(J-10)845,845,846
845 KI=IJ
    GO TO 847
846 KI=IK
847 WLX(J)=TLX(KI)
    WUX(J)=TUX(KI)
    WLY(J)=TLY(KI)
    WUY(J)=TUY(KI)
    WL(J)=CL(KI)
    WW(J)=CW(KI)
848 WA(J)=CA(KI)
    GO TO 1034
989 CONTINUE
    KNT=12
    DO 1014 J=1,10
    KNT=KNT+1
    WLX(KNT)=TLX(J)
    WUX(KNT)=TUX(J)
    WLY(KNT)=TLY(J)
    WUY(KNT)=TUY(J)
    WL(KNT)=CL(J)
    WW(KNT)=CW(J)
1014 WA(KNT)=CA(J)
    NCT=NC
    CALL MCA
    IF(NC-1)994,995,996
994 AT=CU(1)
    NS1=1
997 CONTINUE
    NDD=ND-1
    IF(NC-1)110,1002,1002
110 IF(C4)1017,1017,111
111 IXYZ=5
    GO TO 86
1002 CONTINUE
    IF(C4)112,112,113
112 NS1=1
    GO TO 1070
113 IXYZ=6
    GO TO 86
996 NNC=NC-1

```

```

      CN1=CA(1)
      NN1=1
      DO 1003 J=1,NNC
      IF(CN1-CA(J+1))1004,1003,1003
1004 CN1=CA(J+1)
      NN1=J+1
1003 CONTINUE
      GO TO 1005
      995 NN1=1
1005 IF(L1-2)1006,1007,1006
1006 AT=CU(1)+CA(NN1)
1008 IF(COMP-AT)1010,1010,1363
1363 IF(AT-CCOMP)997,1364,1364
1364 IF(AT-BC3)997,997,1365
1365 BC3=AT
      GO TO 997
1010 CONTINUE
1012 CONTINUE
      NDD=ND-1
1048 FG(NF)=NPG
      IF(CB)1070,1070,87
      87 IF(C4)1017,1017,88
      88 IXYZ=2
      GO TO 86
1007 CAO=CAO+CA(NN1)
1070 WLX(9)=TLX(NN1)
      WUX(9)=TUX(NN1)
      WLY(9)=TLY(NN1)
      WUY(9)=TUY(NN1)
      WL(9)=CL(NN1)
      WW(9)=CW(NN1)
      WA(9)=CA(NN1)
      IF(L1-3)1049,91,1049
      91 IF(C4)1017,1017,90
      90 IXYZ=3
      GO TO 86
1034 CONTINUE
      IF(CAO-CAN)1043,1042,1042
1042 I1=7
      I2=9
      SPC=CAO
      IF(COMP-CAO)1044,1044,1045
1044 CONTINUE
      ZW=0.
1046 CONTINUE
      IF(ZW)1048,1048,1380
1380 IF(C4)1132,1132,92
1132 NS1=1
      GO TO 1017
      92 IXYZ=4
      GO TO 86
1045 CONTINUE
      ZW=1.
      IF(SPC-CCOMP)1046,1366,1366

```

```

1366 IF(SPC-BC3)1046,1046,1367
1367 BC3=SPC
      GO TO 1046
1043 I1=10
      I2=12
      SPC=CAN
      IF(COMP-CAN)1044,1044,1045
1049 CHN=0.
      KNI=10
      KNT=12
1030 DO 1016 J=1,KNI
      IF(CHN)1019,1019,1020
1020 IF(L1-2)1022,1022,1021
1021 IF(J-2)1024,1025,1026
1024 JNC=IJ
      GO TO 1023
1025 JNC=IK
      GO TO 1023
1026 JNC=IG
      GO TO 1023
1022 IF(J-2)1027,1028,1029
1027 JNC=IG
      GO TO 1023
1028 JNC=IE
      GO TO 1023
1029 JNC=IH
      GO TO 1023
1019 JNC=J
1023 KNT=KNT+1
      TLX(JNC)=WLX(KNT)
      TUX(JNC)=WUX(KNT)
      TLY(JNC)=WLY(KNT)
      TUY(JNC)=WUY(KNT)
      CL(JNC)=WL(KNT)
      CW(JNC)=WW(KNT)
1016 CA(JNC)=WA(KNT)
      IF(CHN)1031,1031,1068
1031 IF(L1-2)533,886,533
      533 IF(C4)1017,1017,534
      534 IXYZ=13
      GO TO 86
1017 KNI=3
      CHN=1.
      KNT=3
      GO TO 1030
1068 CONTINUE
1039 CONTINUE
      ND=ND-2
      NFD(1)=ND
      C3=0.
      IF(NS1)1351,1351,1357
1351 CONTINUE
      IND=1
      GO TO 401

```

```

1357 IF(I3-K3C)1358,1032,1032
1358 NS1=0
1032 CONTINUE
3000 IF(MNC-4)401,132,132
132 I4C=0
    I4D=0
    DO 1125 J=1,275
1125 IAA(J)=0
    N4C=KOUNT(4)
    IF(N4C)133,133,134
133 CONTINUE
    GO TO 401
134 DO 66 J=1,N4C
    IQ=IA(J+125)
    IR=IA(J+150)
    IS=IA(J+175)
    IT=IA(J+200)
    IQC=0
    IRC=0
    ISC=0
    ITC=0
    ICC=0
    M4=0
    IDC=0
    IDD=0
    IC4T=0
    IF(OL(IQ,IR))136,136,135
135 IQC=IQC+1
    IRC=IRC+1
    ICC=ICC+1
136 IF(OL(IQ,IS))138,138,137
137 IQC=IQC+2
    ISC=ISC+1
    ICC=ICC+1
138 IF(OL(IQ,IT))140,140,139
139 IQC=IQC+4
    ITC=ITC+1
    ICC=ICC+1
140 IF(OL(IR,IS))142,142,141
141 IRC=IRC+2
    ISC=ISC+2
    ICC=ICC+1
142 IF(OL(IR,IT))144,144,143
143 IRC=IRC+4
    ITC=ITC+2
    ICC=ICC+1
144 IF(OL(IS,IT))146,146,145
145 ISC=ISC+4
    ITC=ITC+4
    ICC=ICC+1
146 IF(ICC-3)148,147,66
147 M4=2
    GO TO 150
148 IF(ICC-2)149,147,146

```

```

149 M4=1
150 CONTINUE
    IF(IQC)152,151,152
151 IDC=IQ
    IF(M4-1)152,152,69
152 IF(IRC)158,153,158
153 IF(M4-1)155,155,154
154 IDC=IR
    GO TO 69
155 IF(IDC)157,157,156
156 IDD=IR
    GO TO 70
157 IDC=IR
    GO TO 158
158 IF(ISC)164,159,164
159 IF(M4-1)161,161,160
160 IDC=IS
    GO TO 69
161 IF(IDC)163,162,163
162 IDC=IS
    GO TO 164
163 IDD=IS
    GO TO 70
164 IF(ITC)170,165,170
165 IF(M4-1)167,167,166
166 IDC=IT
    GO TO 69
167 IF(IDC)169,169,168
168 IDD=IT
    GO TO 70
169 CONTINUE
    GO TO 66
170 CONTINUE
    GO TO 66
70 I4C=I4C+1
    IAA(I4C)=IQ
    IAA(I4C+25)=IR
    IAA(I4C+50)=IS
    IAA(I4C+75)=IT
    ITT=I4C*2
    IDA(ITT-1)=IDC
    IDA(ITT)=IDD
1127 IC4T=IC4T+1
    IF(IC4T-3)66,171,171
69 I4D=I4D+1
    IAB(I4D)=IQ
    IAB(I4D+25)=IR
    IAB(I4D+50)=IS
    IAB(I4D+75)=IT
    IDB(I4D)=IDC
    GO TO 1127
66 CONTINUE
171 CONTINUE
    IF(IC4T)172,172,173

```

```

172 CONTINUE
    GO TO 401
173 JPI=14C
    KPI=14D
    LIC=0
    KIC=0
    MIC=0
174 DO 76 NQ=1,IC4T
    IF(JPI)175,175,178
175 IF(KPI)176,176,177
176 CONTINUE
    GO TO 76
178 LIC=LIC+1
    JPI=JPI-1
    SC=0.
    IQ=IAA(LIC)
    IR=IAA(LIC+25)
    IS=IAA(LIC+50)
    IT=IAA(LIC+75)
    KIC=LIC*2-1
1128 IF(IDA(KIC)-IQ)180,179,180
179 SC=SC+1.
    IF(SC-1.)181,181,80
181 KIC=KIC+1
    GO TO 180
180 IF(IDA(KIC)-IR)184,182,184
182 SC=SC+2.
    IF(SC-2.)183,183,80
183 KIC=KIC+1
    GO TO 1128
184 IF(IDA(KIC)-IS)187,185,187
185 SC=SC+4.
    IF(SC-4.)186,186,80
186 KIC=KIC+1
    GO TO 1128
187 IF(IDA(KIC)-IT)189,188,189
188 SC=SC+7.
    IF(SC-7.)183,183,80
189 CONTINUE
    GO TO 76
    80 IF(SC-5.)190,191,1129
190 IJ=IS
    IK=IT
    IU=IQ
    IV=IR
    GO TO 196
191 IJ=IR
    IK=IT
    IU=IQ
    IV=IS
    GO TO 196
1129 IF(SC-8.)192,193,1130
192 IJ=IQ
    IK=IT

```

```

IU=IR
IV=IS
GO TO 196
193 IJ=IR
    IK=IS
    IU=IQ
    IV=IT
    GO TO 196
1130 IF(SC-9.)80,194,195
194 IJ=IQ
    IK=IS
    IU=IR
    IV=IT
    GO TO 196
195 IJ=IQ
    IK=IR
    IU=IS
    IV=IT
196 C4=1.
    J=1
    INI=1
    C3=0.
    CR=2.
    GO TO 197
177 MIC=MIC+1
    KPI=KPI-1
    IQ=IAB(MIC)
    IR=IAB(MIC+25)
    IS=IAB(MIC+50)
    IT=IAB(MIC+75)
    IF(IDB(MIC)-IQ)199,198,199
198 IJ=IR
    IK=IS
    IKK=IT
    IU=IQ
    GO TO 1087
199 IF(IDB(MIC)-IR)1082,1081,1082
1081 IJ=IQ
    IK=IS
    IKK=IT
    IU=IR
    GO TO 1087
1082 IF(IDB(MIC)-IS)1084,1083,1084
1083 IJ=IQ
    IK=IR
    IKK=IT
    IU=IS
    GO TO 1087
1084 IF(IDB(MIC)-IT)1086,1085,1086
1085 IJ=IQ
    IK=IR
    IKK=IS
    IU=IT
    GO TO 1087

```

```

1086 CONTINUE
      GO TO 76
1087 C3=1.
      K3C=1
      C4=1.
      CR=3.
      197 DO 1131 KZ=1,4
          XLX(KZ)=0.
          XUX(KZ)=0.
          XLY(KZ)=0.
          XUY(KZ)=0.
          XCL(KZ)=0.
          XCW(KZ)=0.
1131 XCA(KZ)=0.
1088 DO 79 MQ=1,4
      IF(MQ-2)1089,1090,1091
1089 MNM=IQ
      GO TO 1094
1090 MNM=IR
      GO TO 1094
1091 IF(MQ-3)1092,1092,1093
1092 MNM=IS
      GO TO 1094
1093 MNM=IT
1094 XLX(MQ)=TLX(MNM)
      XUX(MQ)=TUX(MNM)
      XLY(MQ)=TLY(MNM)
      XUY(MQ)=TUY(MNM)
      XCW(MQ)=CW(MNM)
      XCL(MQ)=CL(MNM)
      79 XCA(MQ)=CA(MNM)
1095 IF(CR-2.)200,200,1352
      86 IF(NS1)1096,1096,121
1096 IF(CR-2.)1097,1097,1098
1097 CU(1)=CU(1)+CA(IU)+CA(IV)
      GO TO 1103
1098 IF(L1-2)1099,1100,1099
1099 CU(1)=AT+CA(IU)
      GO TO 1103
1100 IF(I1-10)1101,1101,1102
1101 CU(1)=CAO+CA(IU)
      GO TO 1103
1102 CU(1)=CAN+CA(IU)
1103 IF(CU(1)-COMP)1104,1105,1105
1104 CONTINUE
      NS1=1
      IF(CU(1)-CCOMP)121,1106,1106
1106 IF(CU(1)-BC4)121,121,1107
1107 BC4=CU(1)
      GO TO 121
1105 CONTINUE
      IND=1
      IF(CR-2.)1108,1108,1110
1108 CONTINUE

```

```

1109 CONTINUE
      GO TO 121
1110 IF(L1-2)1111,1112,1111
1111 CONTINUE
      GO TO 1109
1112 IF(I1-10)525,525,526
      525 I1=7
          I2=9
          GO TO 527
      526 I1=10
          I2=12
      527 CONTINUE
      121 DO 97 MQ=1,4
          IF(MQ-2)1113,1114,1115
1113 MNM=IQ
      GO TO 1118
1114 MNM=IR
      GO TO 1118
1115 IF(MQ-3)1116,1116,1117
1116 MNM=IS
      GO TO 1118
1117 MNM=IT
1118 TLX(MNM)=XLX(MQ)
      TUX(MNM)=XUX(MQ)
      TLY(MNM)=XLY(MQ)
      TUY(MNM)=XUY(MQ)
      CW(MNM)=XCW(MQ)
      CL(MNM)=XCL(MQ)
      97 CA(MNM)=XCA(MQ)
      IF(CR-2.)1121,1121,1119
1119 ND=NDT
      NFD(1)=ND
      NC=NCT
1121 IF(NS1)1122,1122,1123
1122 CONTINUE
      GO TO 401
1123 IF(NQ-IC4T)1124,76,76
1124 NS1=0
      C3=0.
      76 CONTINUE
401 RETURN
      END

```