Developed Software (1975-1979)

W.E.Hunt

Introduction

A general description of the computer programs included in this report has been included in the author's PhD thesis (University of Aston, 1979). The purpose of this report is to provide detailed instructions to illustrate the data preparation with some sample input files, and to provide program listings.

Standard Formats

Certain types of input are required for more than one program. Therefore to minimise confusion the formats of these card inputs have been standardised.

(1) Cell Card

Format: 10X,I10,F10.0

Purpose: To input cell parameters.

Input Variables:

NCELL
The type of lattice is distinguished according to the values of NCELL:
Orthorhombic, tetragonal or cubic
Monoclinic, trigonal or hexagonal
Triclinic

A,B,C
The unit cell translations in Angstrom Units

AF,BT,GM
The angles Alpha, Beta and Gammma in degrees

(2) Symmetry Cards

Format: 6F10.0,7X,3I1

Purpose: To input space group symmetry operations. The symmetry operations allowed are sufficient to handle triclinic, monoclinic and orthorhombic and tetragonal space groups. The equations of transformation have been given in the author's thesis (p43) where the meanings of the input variables are given.

Input Variables:

R1,R2,R3 The diagonal elements of the rotation matrix

T1,T2,T3 The translation vector

I,J,K Integers permuting X,Y and Z. Where these columns are left blank, no permutation is carried out (I=1,J=2 and K=3).

(3) Phusis Card

Format: 15I3,24X,F9.0

Purpose: To input the orientation and step values for the Fourier transform calculation. These parameters are input into the contouring and peak search programs in the same manner as for the Fourier transform calculation.

Input Variables:

NCALC Value of NCALC determines the type of calculation:

0. Ordinary Fourier
1. Difference Fourier
2. Sharpened Patterson
3. E-map

NO Sorting integer. Orientation of the Fourier transform (sections, columns and rows) is determined according to the values of NO:

1. Sec Z Col Y Row X
2. Sec Y Col Z Row X
3. Sec Z Col X Row Y
4. Sec Y Col X Row Z
5. Sec X Col Z Row Y
6. Sec X Col Y Row Z

NS Centrosymmetric: 0. YES, 1. NO

K2M	Maximum index (rows)
K2M	Maximum index (columns)
K3M	Maximum index (sections)
M1I,M1F	Row limits (120ths)
M2I,M2F	Column limits (120ths)
M3I,M3F	Section limits (120ths)
M1D	Row step (120ths)
M2D	Column step (120ths)
M3D	Section step (120ths)
WKON	Set scale for density so the the maximum is WKON
Note:	The values of NCALC, K1M,K2M, K3M and WKON are not required for the programs in this report.

| BAKSUB |

Description

BAKSUB is a program for the reduction of four-circle diffractometer data. For details of the calculations carried out by this program, see the author's thesis (pp 38-41).

Macro

The program may be run with use of the macro EBAKEX. There are three compulsory parameters (which must be specified in the order stated):

1. Instruction File Name
2. Reflexion Data File Name
3. Card Punch Output File Name

The structure factors and standard deviations are written to the Card Punch Output File. Because of infrequency of use, a binary version of this program has not been saved. The program must be recompiled with each run. Core requirement: 40K.

Programming Notes

The program has been implemented only on the ICL 1904S computer. The following non-standard Fortran features have been included:

1. Hollerith fields enclosed in quotation marks

2. Free-format data input

The current version of this program is BAKSUB3.

Instruction File

(1) Output Control Card

Format: 10X,2I10

Purpose: To control writing or suppression of output from the program.

Input Variables:

IW1 Output of raw intensity data on the line printer:
0. Suppress
1. Print

IW2 Output of background and Lp-corrected intensities:
0. Suppress
1. Print

(2) Wavelength Card

Format: 10X,F10.0

Purpose: To input wavelength of radiation used.

Input Variables:

WLAM Wavelength in Angstrom Units

(3) Cell Card

Format: 10X,I10,6F10.0

Purpose: To input unit cell parameters.

Input Variables:
See Standard Formats.

(4) Check Reflexion Card

Format: 10X,I10

Purpose: Input of one integer used to identify check reflexions in the Data File.

Input Variables:

JCHECK Check reflexion identifier.

(5) Paralysis Time Card

Format: 10X,F10.0

Purpose: To input the paralysis time of the counter. For scintillation counters a zero paralysis time may be used.

Input Variables:

PARTIM Paralysis time in seconds

(6) Count Time Card

Format: 10X,3F10.0

Purpose: To input counting times for the peak and low and high-angle backgrounds.

Input Variables:

PKTIM Peak counting time in seconds

BG1TIM Low-angle background counting time in seconds

BG2TIM High-angle background counting time in seconds

(7) Intensity Fluctuation Card

Format: 10x,I10,F10.0

Purpose: To input variation in intensity for the purpose of calculating standard deviations. Three options are allowed:
1. Intensity variation calculated from the counting statistics (see W.E.Hunt, PhD Thesis, pp38-40).
2. Intensity variation supplied by the user.
3. Intensity variation supplied by the user to be used where the variation calculated by the program is negative.

Input Variables:

JOVER Calculation option number (see above)

VFPO Fractional intensity variation

(3) Statisitics Output Card

Format: 10X,I2,17F4.0

Purpose: The number of reflexions meeting the criteria R>X where R is the ratio of the structure factor to its standard deviation and X is input by the user.

Input Variables:

NSRAN The number of values of X to be input

SRAN(I) The values of X

Reflexion Data File

(1) Reflexion Cards

Format: 3I3,I2,I6,I8,I6

Purpose: Input of raw counting statistics for each reflexion.

Input Variables:

IH,K,L Reflexion indices hkl

Note: The file is terminated with a reflexion with IH>998.

ICHECK Reflexion identifier. One value of ICHECK must be reserved to identify check reflexions.

NB1 Background count (low angle)

NPK Peak count

NB2 Background count (high angle)

BAKSUB: MACRO LISTING AND SAMPLE INSTRUCTION FILE - P 1

MACRO LISTING:

```
CE !
CY %B,!
UAFORTRAN PROG BAKSUB3,OWNPD,*CR0 %B,*CR2 !,*CR1 %A,LINES 9999,-
*MT0!,*CP0 %C
```

SAMPLE INSTRUCTION FILE:

```
WRITE CONT          0          1
WAVELENGTH   1.5416
CELL  CARD          2  15.023   20.0802    9.7460    90.000    90.350    90.000
CHECKIDENT          3
PARALYSIS
COUNT TIME   48.0       6.00       6.00
JOVER-VFPO          1   0.03
STD RANGES 5 0.5 1.0 2.0 3.0 5.0
```

MOLJOM

Description

MOLJOM is a program for the calculation of various geometrical parameters (bond distances, angles and fitted planes). Coordinates input may be either atomic positions resulting from a structure analysis or peak positions resulting from a search. The program is further described in the author's thesis (pp38-41).

Programming Notes

The program has been implemented only on the ICL 1904S computer. The following non-standard Fortran features have been included:

1. Hollerith fields enclosed in quotation marks

2. Free-format data input

The current version of this program is NEWJOM. The program calls subroutines from the author's library ELIB2. The core requirement is 20K.

Macro

The program may be run with use of the macro EJOMEX. This macro has two parameters, which must be specified in the order stated, and three optional parameters, which may be specified in any order

Compulsory Parameters:
1. Instruction File Name
2. Atomic Coordinates File Name

Optional Parameters:

##	A record of the time and parameters are stored in the file ERECORD.
#MON	The lineprinter output is sent to the file CELLMONITOR.
#COMP	The main program is recompiled.

#CP Card punch output file: up to two files may be specified. The name of the card punch output file follows this parameter. Where two card punch output files are required, the parameter must be specified twice.

#LSQPL The data is submitted to the Gottingen Least Squares Program for the calculation of least-squares planes through the planes defined on Cards (6) of the Instruction File.

Instruction File

(1) Control Card

Format: 10X,9I5

Purpose: To input parameters to direct the overall running of the program by specifying: (a) the type of calculation to be carried out, (b) the production or suppression of output, (c) the form in which data in the Atomic Coordinates File is to be read.

Input Variables:

NATOMS The number of atomic positions to be read from the Atomic Coordinates File. Current dimensioning restricts NATOMS to values no greater than 100.
Note: Some MOLJOM calculations can be carried out on fractional coordinates from a peak search (see PEAKS). In this case the value of NATOMS should be set to the maximum number of peaks to be used in the calculation.

NSYMM The number of symmetry cards to follow. (For a note on the method of handling symmetry within this program see the author's thesis, p 43. Current dimensioning restricts NSYMM to values no greater than 16.

MODE Atomic coordinates may be entered either as fractional coordinates or as Cartesian coordinates in Angstrom Units according to the value of MODE:
0. Fractional coordinates
1. Cartesian coordinates

ISHOW This parameter controls the printing of atomic coordinates in the lineprinter and the output of certain data to a card

punch file. Values:
0. No atomic coordinates to be printed
1. Cartesian coordinates to be printed
2. Fractional and cartesian coordinates to be printed
3. Output of fractional coordinates to a file in a format 3F10.5
4. Output of cartesian coordinates in a form suitable for input to the version of CNDO/2 maintained by the Pharmacy group at U.M.R.C.C.
5. Output of bond distances and angles to card-image files in a format suitable for publication. Parameters ICMIN and IBOND should be set to 2 and 1 respectively.

IFORM Atomic coordinates may be input either in the format reauired for the Oak Ridge least-squares program (ORFLS) or in the form of free-formatted input. Details of the formats are inlcluded in the description of the Atomic Coordinates File. Values:
0. Coordinates input in Oak Ridge Format
1. Coordinates input in Free Format
2. Coordinates are input in Free Format and output to a card image file in Oak Ridge Format. In this case atomic species and occupancy factors are set to unity and temperature factors to zero.

IQUIT A non-zero value of IQUIT will stop execution of the program before coordinates have been converted to cartesian form and no cards will be read after the Cell Card. Execution is required to continue beyond this point only when explicitly-stated bonds are to be calculated.

ICONN A non-zero value of ICONN will cause all interatomic distances to be calculated. Symmetry operations are not considered.

ICMIN A non-zero value of ICMIN will initiate calculation of interatomic distances with symmetry operations taken into consideration according to the value of ICMIN:
0. No calculation
1. The minimum distance between each atomic position and the same position operated upon by a non-identity symmetry operation is calculated.

2. The minimum distance Rij between two atoms is calculated, allowing for all possible symmetry operations. For calculations of the type Rii the identity operation is not allowed.
3. Action as for ICMIN=2 above except that the identity operation is not allowed for the calculation of any Rij.

IBOND Nonzero valuess of IBOND dcause the calculation of interatomic distances to be output in the form of bond lengths and angles, where the minimum and maximum values of the bond distances and angles are specified on the next card. This facility is normally used in conjunction with the symmetry-minimised distances (ICMIN=2). Note: only the minimum distances Rij are used in the calculation. The calculation does not, therefore, encompass all contacts of distance less than BMAX.

(2) Bond Card

Format: 10X,4F10.0

Note: This card must not be included unless IBOND is nonzero.

Purpose: To input maximum and minimum bond distances and angles.

Input Variables:

BMIN	Minimum bond length in Angstrom Units
BMAX	Maximum bond length in Angstrom Units
PHIMIN	Minimum bond angle in degrees
PHIMAX	Maximum bond angle in degrees

(3) Symmetry Cards

Format: 6F10.0,7X,3I1

Purpose: To input space group symmetry operations.

Input Variables:
See Standard Formats.

(4) Cell Card

Format: 10X,I10,6F10.0

Purpose: To input cell dimensions.

Input Variables:
See Standard Formats.

(5) Nlines Card

Format: 10X,3I10

Purpose: To input the number of lines of atoms for which bond angles, torsion angles,etc. are to be calculated. Provision is also made for the cartesian coordinate system to be rotated so that the z-axis is perpendicular or parallel to a given plane.

Input Variables:

NLINES The number of lines of atoms or planes for which calculations are to be carried out

NROT The number of rotation calculations

IROTOP A parameter for determining the output medium for the rotated cartesian coordinates. At present only card-image output is allowed (IROTOP=4).

(6) Lines Cards

Format: 5X,3I1,36I2

Purpose: To input parameters for the calculation of bond distances, bond angles and torsion angles along a line of atoms. The atoms are referred to by number in the order in which they appear in the Atoms File. Provision is also allowed for the atom list to be considered a closed ring with the last atom bonded to the first, in which case the equation of a plane fitted to the atomic positions will also be calculated. Provision is made for the calculation of hydrogen positions according to either trigonal or tetrahedral geometry. For ring calculations, provision is made for the calculation of the angle between the normal to the plane and any interatomic vector (bond-plane calculations).

Input Variables:

IEXIT Bond-plane calculations are carried out according to the value of IEXIT:
0. No calculation
1. Bond plane calculations. The bonds to be included in the calculation are defined on the Exit Card.

IHYDRO The positions of hydrogen atoms bonded to ring atoms are calculated according to the value of IHYDRO:
0. No calculation
1. Calculation assumes trigonal hybridisation
2. Calculation assumes tetrahedral hybridisation

ITOR Torsion angles and ring plane parameters are calculated according to the value of ITOR:
0. No calculation
2. The atoms are assumed to form a ring: torsion angles and the equation of a fitted plane are to be calculated.

NATOM The number of atoms jto be specified as forming the line or ring.

LINE(I) The numbers (in order of input) of the atoms forming the line or ring

(7) Exit Card

Format: 8X,36I2

Note: This card must not be included in IEXIT=0

Purpose: To input parameters for bond-plane calculations.

Input Variables:

NEXIT An integer equal to twice the number of bonds for which calculations are required.

LINE(I) The numbers (in order of input) of atoms forming the bonds: these numbers will be interpreted in pairs.

(8) Rotation Card

Format: 6X,4I2

Note: The number of Rotation Cards is equal to NROT on Card (5)

Purpose: To indicate the orientation of the cartesian coordinates with respect to the given plane.

Input Variables:

IROTYP — Orientation of the coordinates is determined by the value of IROTYP:
0. z-axis perpendicular to the plane
1. z-axis parallel to the plane

IROT — The number of the plane in order input on Cards(6) with respect to which the calculation is to be made

K,L — Two atoms which determine the orientation of the axial system according to the following scheme: Where $\underline{N}$ is a vector perpendicular to the plane and $\underline{R}kl$ is a vector between the two atoms specified, then the axes of the coordinate system are unit vectors in the following directions:

(1) $\underline{N}$
(2) $\underline{N}$ X $\underline{R}kl$
(3) $\underline{N}$ X ($\underline{N}$ X $\underline{R}kl$)

Atomic Coordinates File (ORFLS Format)

(1) Scale Factor Card

Format F9.6

Purpose: To input scale factor or scale factors to ORFLS.

Note: The parameter on this card is not used by MOLJOM. The card may, therefore, be left blank.

Input Variables:

SCALE — Overall scale factor

(2) Temperature Factor Card

Format: F9.6

Purpose: To input an overall temperature factor to ORFLS.

Note: The parameter on this card is not used by MOLJOM. The card may, therefore, be left blank.

(3) Atomic Coordinates Cards

Format: A4,F9.6,9X,3F9.6/

Note: (1) There are two cards, the second of which is not read.
(2) The number of pairs of cards is equal to NATOMS input on Card (1) of the Instruction File.
(3) Parameters read by ORFLS but not read by MOLJOM or KONTUR are not included in this description.

Purpose: To input atomic coordinates.

Input Variables:

NAME	Atom name
ATOMID	A real whole number to identify the atomic species. This parameter is read by KONTUR but not by MOLJOM.
X,Y,Z	Atomic fractional coordinates

Atomic Coordinates File (Free Format)

(1) Atomic Coordinate Cards

Format: A4,3F0.0

Note: The number of Atomic Coordinates Cards is equal to NATOMS input on Card (1) of the Instruction File.

Purpose: To input the atomic fractional coordinates.

Input Variables:

NAME	Atom name
X,Y,Z	Atomic coordinates

MACRO LISTING: EJOMEX

```
IF ABS(##),GO 9E61
ERECORDEX EJOMEX,%A,%B,%C,%D,%E,%F,%G,%H
9E61
LF %A,FR 0,LI 20
IF ABS(#LSQPL),GO 9E62
IN ETEMP,T????
I£                                                       11111111111111111111
£
(TC£BOND£,P1)C£LINES£
????
ED %A,%A(+1),ETEMP
UAFORTRAN LOAD ELSQPLBN,*CR1 %A,*CR0 %B,EXIT
ER ETEMP
ER %A
GO 9E4
9E62
IF ABS(#CP),GO 9E1
SP W,(,*CP0 %(#CP))
GO 9E7
9E1
IF PRE(#PUB),SP W,(,*CP0 ETEMP1)
IF PRE(#PUB),DP 0,BOND-ANGLES STORED IN FILE ETEMP1
9E7
IF ABS(#LP), GO 9E2
SP V,(,*LP1 %(#LP))
9E2
IF ABS(#PUB),GO 9F3
SP V,(,*LP1 ETEMP3)
DP 0,
DP 0, ***YOUR LINEPRINTER OUTPUT IS IN FILE ETEMP3
DP 0,
9F3
SP U,(%2(#CP))
IF STR(%U)=(),GO 9F1
SP U,(,*CP1 %2(#CP))
9F1
IF ABS(#PUB),GO 9F2
IF ABS(#PUBFILE),GO 9F11
IN ETEMP,T????
T.19,I£1     0     5     1     1     0     2     1£,P.E,T1
I£BOND   CARD   0.300      1.600      100.00     140.00
    1.00       0.00       1.00       0.00       1.00       0.00
£
TC/BOND/,P1
PC£CELL£
TE,E
????
ED %A,%A(+1),ETEMP
ER ETEMP
9F11
IF STR(%U)=(),GO 9F3
GO 9F2
9F3
SP U,(,*CP1 ETEMP2)
DP 0,BOND DISTANCES HAVE BEEN STORED IN FILE ETEMP2
9F2
SP S,(LOAD)
SP T,(BN)
IF ABS(#COMP),GO 9G1
SP S,(PROG)
SP T,(,OWNPD,SAVE NEWJOMBN)
9G1
UAFORTRAN %S NEWJOM%T,-
*CR0 %A,*CR1 %B,LIB :SHS6107.ELIB2,-
EXIT%U%V%W
IF ABS(#LISTLP),GO 9E4
LF %(#LP),*LP
9E4
IF ABS(#LISTCP),GO 9E5
LF %(#CP)
9E5
IF ABS(#PUB),GO 9E6
IF PRE(#CP),SP U,(%(#CP))
IF ABS(#CP),SP U,(ETEMP1)
IF STR(%2(#CP))=(),GO 9F7
SP W,(%2(#CP))
GO 9F8
9F7
SP W,(ETEMP2)
9F8
EPUBEX %U,#ANG
LF %U
EPUBEX %W
LF %W
IF PRE(#PUBFILE),ER %A
9E6
IF ABS(#MON),EX
EJ NO,RT(JOMMONITOR)
EX
```

EXAMPLE: DIPB: DATA FILE FOR CALCULATION OF BOND DISTANCES AND ANGLES FOR ONE MOLECULE, WITH OUTPUT IN A FORMAT SUITABLE FOR PUBLICATION. (BRACKETS MUST BE EDITED INTO THE OUTPUT FILE, SINCE THE LIMIT ON ATOM NAME LENGTHS IS 4 CHARACTERS.)

```
CONTRLCARD   20    1    0    5    1    1    0    2    1
BOND  CARD  0.500    1.700      90.000     150.00
   1.00      0.00      1.00      0.00       1.00       0.00
CELL  CARD            2  8.9290   20.0802    8.9736     90.000     114.05     90.000
```

EXAMPLE: DAQ: CALCULATION OF FITTED PLANES. A PLANE THROUGH THE ATOMS OF THE HETEROCYCLIC RINGS (ATOMS 1 TO 10) IS CALCULATED. THE ANGLES BETWEEN THE BONDS C2-N2 AND C4-N4 AND THE NORMAL TO THIS PLANE ARE CALCULATED (WHERE N2 AND N4 ARE ATOMS 11 AND 12 RESPECTIVELY). THE PLANES THROUGH EACH OF THE RINGS IS CALCULATED SEPARATELY AND THE INTERPLANAR ANGLES WILL APPEAR ON THE FINAL OUTPUT.

```
CONTRLCARD   13   16    0    2    1    0    0    2    0
  1.0000    0.0000    1.0000    0.0000    1.0000    0.0000          123
 -1.0000    0.0000   -1.000     0.0000   -1.000     0.0000
 -1.0000    0.0000   -1.000     0.5000    1.0000    0.0000
  1.0000    0.0000    1.0000    0.5000   -1.0000    0.0000
   -1.0000    0.7500    1.0000    0.2500    1.0000    0.2500        213
    1.0000    0.2500   -1.000     0.7500   -1.000     0.7500        213
    1.0000    0.2500   -1.000     0.2500    1.0000    0.2500        213
    -1.000    0.7500    1.0000    0.7500   -1.000     0.7500        213
  1.0000    0.5000    1.0000    0.5000    1.0000    0.5000          123
 -1.0000    0.5000   -1.000     0.5000   -1.000     0.5000
 -1.0000    0.5000   -1.000     1.0000    1.0000    0.5000
  1.0000    0.5000    1.0000    1.0000   -1.0000    0.5000
   -1.0000    0.2500    1.0000    0.7500    1.0000    0.7500        213
    1.0000    0.7500   -1.000     0.2500   -1.000     0.2500        213
    1.0000    0.7500   -1.000     0.7500    1.0000    0.7500        213
    -1.000    0.2500    1.0000    0.2500   -1.000     0.2500        213
CELL  CARD            1  21.59300  21.59300   7.55000  90.00000  90.0000  90.00000
LINES CARD            3
PLANE10210 1 2 3 4 5 6 7 8 910
EXITCARD 4 211  412
PLANE002 6 1 2 3 4  510
PLANE002 6 5 6 7 8  910
```

PEAKS

Introduction

PEAKS is peak search program, designed to compute a list of peak heights and fractional coordinates from a magnetic tape produced by FURSUM. The list of peaks can be input to MOLJOM to calculate 'bond' distances and 'bond' angles. Since symmetry-equivalent positions can be computed with use of MOLJOM, only ane asymmetric unit should be searched. Since it may be desirable to print more than one asymmetric unit, provision exists for setting the minimum and maximum values for the numbers of the rows, columns and sections. The present dimensioning permits the search of Fouriers up to 31 rows by 31 columns, with an unlimited number of sections. The maximum number of peaks is 900.

The program is further described in the author's thesis, pp 43-44.

Macro

The program may be run with use of the macro EPEAKEX. This macro has two parameters, which must be specified in the order stated, and three optional parameters, which may be specified in any order

Compulsory Parameters:

1. Instruction File Name
2. Magnetic Tape File Name

Optional Parameters:

```
##          A record of the time and parameters are
            stored in the file ERECORD.
#MON        The lineprinter output is sent to the file
            CELLMONITOR.
#COMP       The main program is recompiled.
```

Programming Notes

The program has been implemented only on the ICL 1904S computer. The following non-standard Fortran features have been included:

1. Hollerith fields enclosed in quotation marks

2. Free-format data input

The current version of this program is EPEAKO. The core requirement is 20K.

Instruction File

(1) Control Card

Format: 7I0,F0.0,2I0

Purpose: To input parameters which indicate the region of the Fourier map to be searched for peaks, minimum peak heigh, maximum number of peaks and the type of search (postitive peaks, negative peaks, or both). The maximum and minimum row numbers, column numbers and section numbers are read. Where these numbers are greater than the map limits, then the entire map is searched, thus to search the complete map N1X, N1Y and N1Z should be set to 1 and N2X, N2Y and N2Z to (say) 100.

Input Variables:

N1X	Minimum row number
N2X	Maximum row number
N1Y	Minimum column number
N2Y	Maximum column number
N1Z	Minumum section number
N2Z	Maximum section number
MINPK	Parameter to set a minimum peak height 0. No minimum 1. Minimum to be set
HMIN	Minimum peak height Note: Where MINPK=0 this parameter will not be used. A value must nevertheless be supplied.
NEGPK	Parameter to indicate the type of search. Values: +1 Positive peaks only 0 Positive and negative peaks

-1 Negative peaks only

KNTMAX The naximum number of peaks to be output. If all peaks are to be output KNTMAX should be set to a value greater than current array dimension (at present 900). Where NEGPK=0 the value of KNTMAX will have no effect.

(2) Phusis Card

Format: 15I3,24X,F9.0

Purpose: To input orientation and step values used in the Fourier transform calcualtion.

Input Variables:
See Standard Formats.

PEAKS: MACRO LISTING - P 1

LISTING OF EPEAKEX:

```
WE COMERR,GO 9F2
IF ABS(##),GO 9E1
ERECORDEX EPEAKEX,%A,%B,%C,%D,%E,%F,%G
9E1
IF ABS(#COMPONLY),LF %A,LI80
SP S,(,*CR1)
SP U,()
SP V,(BN)
SP W,(LOAD)
IF ABS(#COMP),GO 9E2
IF ABS(#COMPONLY),GO 9E3
SP S,()
SP T,(NORUN)
SP A,()
SP B,()
9E3
SP U,(,SAVE ERASTEPABN)
SP V,(,OWNPD)
SP W,(PROG)
SP Q,()
IF PRE(#MON),SP Q,(*LP0 ET0,*LP1 ET1,)
9E2
UAFORTRAN %W EPEAK0%V%S %A,-
 *M TO %B,%QEXIT
IF NOT DELETED(OD),GO 9F2
9F2
IF ABS(#MON),GO 9F1
EJ NO,RT(MAPMONITOR)
9F1
EJ
```

| KONTUR |

Description

KONTUR is a general-purpose contouring program which has been specifically adapted to permit a graph-plotter representation of Fourier syntheses. The program is further described in the author's thesis (pp 44-45).

Macro

The program may be run with use of the macro EKONTUREX. This macro has three parameters, which must be specified in the order stated, and three optional parameters, which may be specified in any order

Compulsory Parameters:

1. Instruction File Name
2. Atomic Coordinates or Matrix Card Input File Name
3. Magnetic Tape File Name

Optional Parameters:

A record of the time and parameters are stored in the file ERECORD.
#MON The lineprinter output is sent to the file CELLMONITOR.
#COMP The main program is recompiled.

Programming Notes

The program has been implemented only on the ICL 1904S computer. The following non-standard Fortran features have been included:

1. Hollerith fields enclosed in quotation marks

2. Free-format data input

The current version of this program is ERASTERA. The program calls subroutines from the Gino-F Subroutine Library and from the author's library ELIB1. The core requirement is 20K.

Instruction File

(1) Control Card

Format: 9I0,2F0.0

Purpose: To input parameters which control the type of contoured map to be drawn.

Input Variables:

MATWRT The matrix of values to be contoured is output to the lineprinter according to the value of MATWRT:
0. Not printed
1. Printed

INTERP Interpolation between matrix elements is carried out according to the value of INTERP:
0. No interpolation
1. Linear interpolation

KURV Contour lines drawn between calculated points on the contour line are drawn according to the value of KURV:
0. Straight lines drawn
1. Curved lines drawn
Note: Drawing of curved lines greatly increases the size of the graphplotter file.

KREL Contouring levels are interpreted according to the value of KREL:
0. Contour levels are interpreted as absolute values.
1. Contouring levels are interpreted as fractions of the maximum peak height.
2. The contouring levels are given by

$$Hi=Fi(Hmax-Hmin)$$

where Fi are the values supplied.

ITAPE The method of reading matrix values is determined by ITAPE:
0. Matrix values are read from a card-image file.
1. Matrix values are read from magnetic tape in a format compatible with output from PHUSIS/FURSUM.

KLINE: Contour lines are drawn as solid or dashed according to the values of KLINE:

0. Solid lines only used.
1. The highest contour is solid progressing to more and more open dashes for lower contour levels.
2. Positive contours are solid. Zero and negative contours are dashed.

KGRID A grid is imposed on contoured Fourier maps according to the value of KGRID:
0. Grid not drawn.
1. Grid lines are drawn at separations of 0.1 unit cell translations.

NX,NY The size of the matrix to be input is NX by NY. Where the matrix is input from cards there must be NX cards, each with NY matrix values. For Fourier input the values of NX and NY will be overwritten according to information on the phusis card; nevertheless, because of free formatting, values (any two integers) must be given.

DX,DY The size of the contoured map will be NX*DX by NY*DY.

(2) Contour Card

Format IO,20F0.0

Purpose: To enter the contouring levels. These values may be either absolute or relative (see above).

Input Variables:

NHDIV The number of contouring levels.

HVEC(I) The contouring levels: the number of values must equal NHDIV.

(3) Phusis Card

Format: 3X,I3,12X,9I3

Purpose: To input orientation and step values which have been used in the calculation of the Fourier transform. The values on this card must be identical to those on the Phusis Card used in the calculation of the Fourier transform.

Input Variables:

See Standard Formats.

(4) Atoms Card

Format: 3I0

Purpose: To input the number of symmetry-independent atomic positions to be marked on the map, the number of symmetry cards to be read and the type of representation of the atomic positions.
Note: Where ITAPE=0 this card need not be included.

Input Variables:

NATOMS The number of atomic positions to be read. Where NATOMS=0 no Atomic Coordinates File need be provided. Where NATOMS>0 the atomic positions are read from and ORFLS formatted file (see MOLJOM documentation). To read atomic coordinates from a free formatted file multiply NATOMS by (-1).

NSYMM The number of Symmetry Cards to be read

KHAR The atomic positions are marked with a symbol identifying the atomic species. Where KHAR is nonzero the atom names are additionally marked on the map. (The lettering may go beyond the boundaries of the map.)

(6) Symmetry Cards

Format: 6F10.0

Purpose: To input the space group symmetry operations.

Note: The number of these cards must equal NSYMM above.

Input Variables:
See Standard Formats.

Atomic Coordinates File

This file is required only where atomic positions are to be marked on the map. Where NATOMS>0 the atomic positions are read from an ORFLS formatted file (see MOLJOM documentation.) Where NATOMS<0 the atomic positions must be provided on Atoms Cards (see below). The atomic species identifier determines the type of symbol to be drawn (see Subroutine SYMBOL in the Gino-F Manual: the value used is ID modulo 8).

(1) Atoms Cards (Free Format)

Format: A4,I0,3F0.0

Purpose: To input atomic fractional coordinates and atomic species identifier.

Input Variables:

NAME	Atom name
ID	Atomic species identifier
X,Y,Z	Fractional coordinates

Matrix File (Cards)

Where ITAPE=0 the matrix is fread from a card-image file. The file must contain NX cards, each with NY real values in free format. (The current maximum is ten values per card and a maximum of 31 cards.)

Matrix File (Magnetic Tape)

Where ITAPE=2 the tape output from PHUSIS/FURSUM must be used. Current dimensioning permits the contouring of maps up to 31 rows by 31 columns, with an unrestricted number of sections.

```
KONTUR: MACRO LISTING - P 1

 NAME OF MACRO: EKONTUREX

WE COMERR,GO 9F2
IF ABS(##),GO 9E1
ERE CORDEX EKONTUREX,%A,%B,%C,%D,%E,%F,%G
9E1
IF ABS(#COMPONLY),LF %A,LI80
SP S,(,*CR0)
SP T,(*CR1)
SP U,()
SP V,(BN)
SP W,(LOAD)
IF ABS(#COMP),GO 9E2
IF ABS(#COMPONLY),GO 9E3
SP S,()
SP T,(NORUN)
SP A,()
SP B,()
9E3
SP U,(,SAVE ERASTERABN)
SP V,(,OWNPD)
SP W,(PROG)
SP Q,()
IF PRE(#MON),SP Q,(*LP0 ET0,*LP1 ET1,)
9E2
UAFORTRAN %W ERASTERA%V%S %A,%T %B,LIB :SHS6107.ELIB1,*GP GINO%U,-
*MT0 *MT0 %C,%QEXIT
IF NOT DELETED(00),GO 9F2
9F2
IF ABS(#MON),GO 9F1
EJ NO,RT(MAPMONITOR)
9F1
EJ
```

CELPIC

Description

CELPIC is a program for illustrating in stick diagram form both individual molecules and unit cell contents. The facilities available in the program have been described in the author's thesis (pp 45-46).

Macro

The program may be run with use of the macro ECELLEX. This macro as two parameters, which must be specified in the order stated, and three optional parameters, which may be specified in any order

Compulsory Parameters:
1. Instruction File Name
2. Atomic Coordinates File Name

Optional Parameters:

```
##          A record of the time and parameters are
            stored in the file ERECORD.
#MON        The lineprinter output is sent to the file
            CELLMONITOR.
#COMP       The main program is recompiled.
```

Layout

The drawing field measures 1000 mm in the horizontal (x) direction by 600 mm in the vertical (y) direction. The user spectifies the paper coordinates of the origin of the unit cell and the scale of the drawing in millimetres per Angstrom Unit. The program will probably produce an execution error if any drawing oversteps these margins. Care must therefore be taken to allow sufficient space for each unit cell plot. Since the direction of drawing outward from the origin is determined by the Gino-F software, space must be provided for the drawing to proceed in any direction.

The input of coordinates may be in three different forms. Reference to the different forms will be made as follows:

Paper Coordinates: Millimetre coordinates in the drawing area with the origin in the lower left-hand corner.

Picture Coordinates: Coordinates with respect to the unit cell axes a,b and c*. The units are Angstrom Units multiplied by the current scale factor.

Fractional Coordinates: The crystallographic fractional coordinates.

Programming Notes

The program has been implemented only on the ICL 1904S computer. It contains the following non-standare Fortran codes: (a) Hollerith fields enclosed in quotation marks, and (b) free-formatted input. The program calls subroutines from the Gino-F library.

The current version of the program is EGMOLM. It calls subroutines from the author's libraries ELIB3 and ELIB4.

Instruction File

(1) Control Card

Format: 10X,4I5

Purpose: Input of control parameters.

Input variables:

NATOMS The number of atoms in the Atomic Coordinates File.

NSYMM The number of symmetry operations to be read.

IFORM The format used in the Atomic Coordinates File. The present version allows only free-format (IFORM=1).

NPROJ The number of unit-cell plots to be drawn.

(2) Symmetry Cards

Format: 6F10.0,7X,3I1

Purpose: Input of symmetry operations.

Input variables:
See Standard Formats.

(3) Cell Card

Format: 10X,I10,6F10.0

Purpose: Input of unit cell parameters.

Input Variables:
See Standard Formats.

(4) Axes Cards

Format: 10X,2I0,6F0.0

Purpose: To plot axes corresponding to each projection used. The axes are indicated as follows:
a axis: single arrow
b axis: double arrow
c axis: triple arrow
This plot aids the identifcation of axes in the unit cell plots. At least one Axis Card must be included. More may be included by setting KONTIN=1.

Input Variables:

KONTIN — KONTIN=0 for last Axis Card to be read. Otherwise the next card will be read as an Axis Card.

IPROJ — IPROJ=1...AXON3 projection to be used.
IPROJ=2...PROJ3 projection to be used
IPROJ=3...FROM3 projection to be used.
Note: For further information see Gino-F Manual, subroutines AXON3, PROJ3 and FROM3.

SCALE — Scale of drawing in mm per Angstrom Unit.

X,Y,Z — These parameters define the picture coordinates (X,Y,Z) to be used as the viewpoint. Where IPROJ=1 the result depends only on the ratio X:Y:Z.

XOR,YOR These parameters define the paper coordinates of the origin of the axial system.

(5) LINES CARD

Format: 10X,I10

Purpose: This card defines the way in which the bonds between the atoms are to be calculated. Provision is made for covalent bonds to be drawn automatically where the separation is in the range $0.6 S<r<1.2 S$ where S is the sum of the covalent radii and for hydrogen bonds to be drawn between hydrogen atoms atoms of a suitable species to be hydrogen bond acceptors where the separation is in the range $1.2 S<r<2.15 S$. Covalent bonds may be entered explicitly using the Lines Cards (see below).

Input Variables:

NLINES NLINES defines the number of continuously-bonded lines of atoms to be drawn. Where NLLINES=0 covalent bonds are calculated by the program.

Note: Add 100 to the above if hydrogen bonds are to be drawn.

(6) BOND CARDS

Format: 7X,I1,30I2

Note: (1) If NLINES=0 or NLINES=100 these cards must not be included. Otherwise the number of cards is equal to NLINES or NLINES - 100 in the case where hydrogen bonds have been drawn.
(2) The LINES cards in the MOLJOM input are acceptable for input here.

Purpose: To define explicitly the bonds to be drawn. Bonds are drawn between each of the NL atoms definining the line. Atoms are referred to by number according to their position in the Atomic Coordinates file. Povision is made for the list of atoms to be considered a ring, with the last atom to be drawn bonded to the first.

Input Variables:

ITOR — Where ITOR=2 the list of atoms is considered a ring-- otherwise not.

NL — The number of atoms to be bonded.

LINE(I,J) — The number (by position in the Atomic Coordinates File) of each atom to be bonded.

(7) Projection Cards

Format: 10X,2I0,6F0.0

Purpose: Input of instructions defining the projection and scale of the drawing and the number of symmetry-related molecules to be drawn.

Input Variables:

NSYMOP — The number of symmetry-related molecules to be drawn.

IPROJ — The type of projection: see Cards (4).

X,Y,Z — Picture coordinates defining the viewpoint.

XOR,YOR — Paper coordinates of the origin of the unit cell.

(8) Mark Cards

Format: 3X,I2,3F5.0,6F10.0

Purpose: This card allows the user to input details of special symbols to be drawn or to indicate the Van der Waals spheres of the atoms or to plot a Van der Waals representation of the atoms viewed perpendicular to a slice through the unit cell. Where MODE<3 The card defines the fractional coordinates of the beginning and endpoint of a line to be drawn. Where 3<MODE<6 the card defines the axis perpendicular to which the Van der Waals radii are to be drawn and the fractional coordinate of the slice. Where MODE>9 the card defines which axial symbol (viewed along the axis) is to be drawn at fractional coordinates x,y,z.

Note: All lengths input on this card are specified in Angstrom Units. These distances will be scaled in the same manner as the remainder of the drawing.

Input Variables (MODE<3):

MODE Line to be drawn is defined to be solid (MODE=0), dashed (MODE=1) or chained (MODE=2).

REPEAT The length of a complete repeat sequence.

DASH The length of dashes

DOT The length of dots.

Note: For further information regarding the above parameters see the Gino-F Manual, Subroutine DASHED.

X1,Y1,Z1 Fractional coordinates of the starting point of the line

X2,Y2,Z2 Fractional coordinates of the endpoint of the line.

Input Variables (3<MODE<9):

MODE MODE-3 defines the axis perpendicular to which the section through the unit cell is to be drawn. The present version allows only MODE=5 (y-axis).

XLAYER The fractional coordinates through which the section is to be drawn.

Input Varables (MODE>9):

MODE The value of MODE determines which axial symbol is to be drawn:
10. Centre of Symmetry
20. Twofold axis
21. Two-1 screw axis
40. Fourfold axis
41. Four-1 screw axis
42. Four-2 screw axis
43. Four-3 screw axis

X,Y,Z The fractional coordinates of the symbol.

HT The height of the symbol.

(9) Symmetry Operation Cards

Format: 5I2,3F5.0

Note: The number of these cards must equal NSYMOP above.

Purpose: These cards define the symmetry operation relating the molecule to be drawn to the listed coordinates. The symmetry operations are referred to by number from the order of input.
The translations parallel to the unit cell axes may be either absolute or relative to a molecule whose centre of gravity (with all atoms given equal weight) is within the unit cell.

Input Variables:

ISYMM The symmetry operation number.

ITX The unit cell translation along the a-axis. Add 90 to begin translation at the unit cell centred molecule.

ITY The unit cell translation along the b-axis. Add 90 to begin translation at the unit cell centred molecule.

ITZ The unit cell translation along the c-axis. Add 90 to begin translation at the unit cell centred molecule.

Atomic Coordinates File

(1) Atom Cards

Format: A4,I0,3F0.0

Note: The number of Atom Cards must equal NATOMS in the Instruction File.

Purpose: Input of atomic numbers and fractional coordinates. Covalent radii, stored in the program, are used for calculating covalent bonds and hydrogen bonds. Covalent radii are stored for the following atomic species: H,B,C,N,O,F,Si,P,Cl,As,Se,Br,Sb,Te,I. Van der Waals radii have been included for all of the above atoms except F and S. To handle unusual bonding situations the atomic numbers 80 to

89 have been used to store a range of covalent radii for atoms which are not hydrogen bonded, and the atomic numbers 90 to 100 for atoms which may be hydrogen bond acceptors.

Input Variables:

NAME	Atom name.
IAT	Atomic number.
X,Y,Z	Fractional coordinates of the atom.

CELPIC: SAMPLE DATA FILES - P 1

UNUSUAL HYDROGEN BONDING IN 2,4,7-TRIAMINO-6-CHLOROQUINAZOLINE (ROGAN AND WILLIAMS,1979). SPACE GROUP C2/C,A=29.955,B=3.853,C=17.742, BETA=109.44.

THE STRUCTURE IS TO BE VIEWED ALONG THE B-AXIS, WITH BONDS AND HYDROGEN-BONDS CALCULATED USING COVALANT RADII STORED BY THE PROGRAMME. ONE UNIT CELL WITH ALL EIGHT SYMMETRY-RELATED MOLECULES IS TO BE DISPLAYED. CELL TRANSLATIONS ARE TO BE APPLIED TO ALL MOLECULES SO THAT THE 'CENTRE OF GRAVITY' OF THE MOLECULE LIES WITHIN THE UNIT CELL. ALL MOLECULES ARE TO BE DISPLAYED AS SINGLE CONTINUOUS LINES AND THE UNIT CELL IS TO BE DRAWN (DEFAULT OPTION). THE SCALE OF THE DRAWING IS 8MM REPRESENTING 1 ANGSTROM. BEFORE DRAWING THE UNIT CELL CONTENTS, ARROWS INDICATING THE AXIAL DIRECTIONS ARE TO BE DRAWN.

IN ODER TO DRAW THE UNUSUALLY LONG N-H...N HYDROGEN BONDS AND TO PREVENT THE BONDED CHLORINE FROM ACCEPTING HYDROGEN BONDS, THIS ATOM AND THE AMINO NITROGENS HAVE BEEN ENTERED AS PSEUDO ATOMIC SPECIES

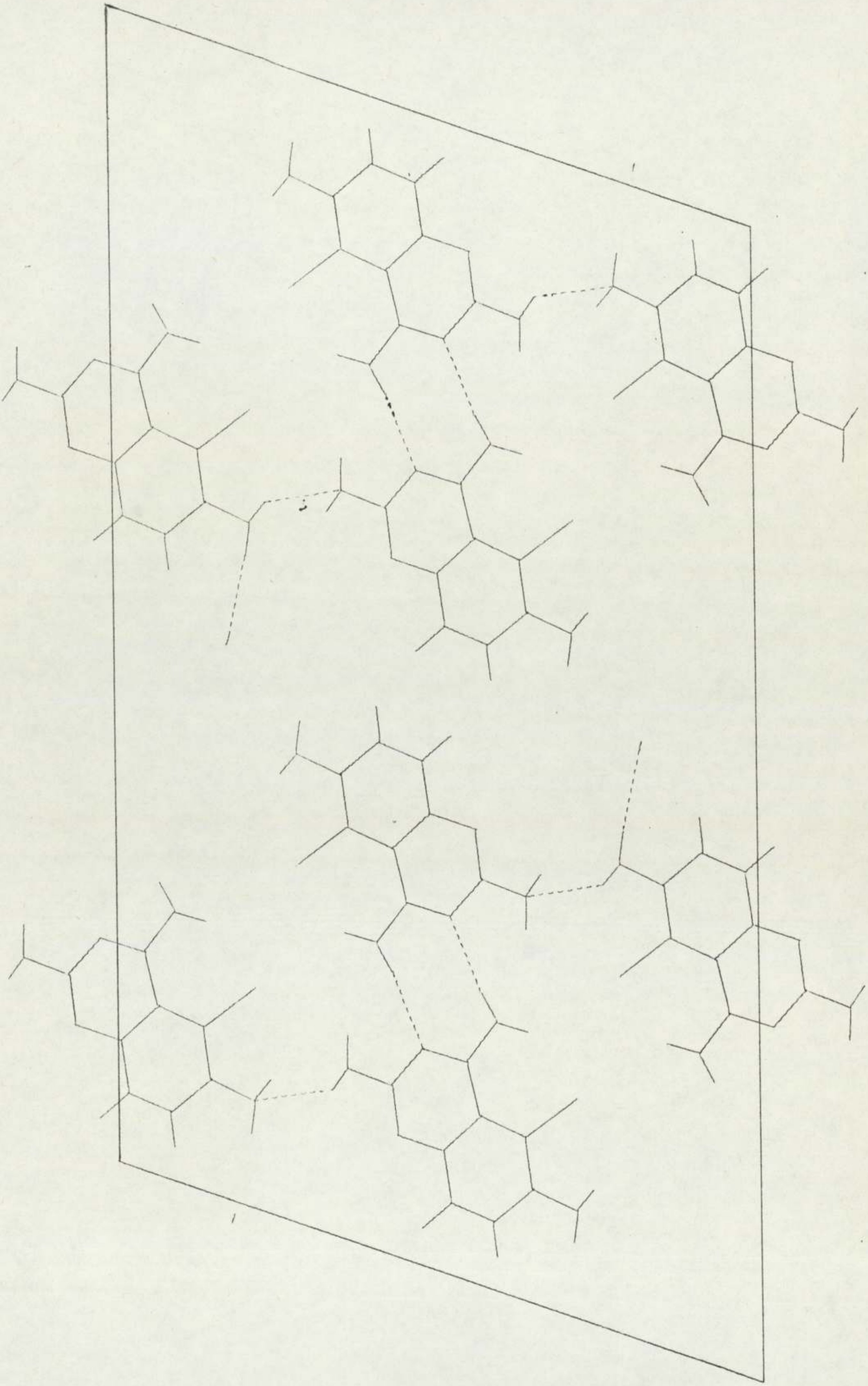
<ATOMS FILE>:

```
CL      87   .69102   .59527   .71650
N1      7    .60536   .03409   .43650
C2      6    .64664   -.00544   .42784
N2      94    .64859   -.15464   .35868
N3       7   .68889  .10306   .47915
C4      6  .63993    .25436   .54664
N4      94    .73325    .35353   .59425
C5      6    .64719    .30337   .56631
C6      6   .64208    .45384   .63625
C7      6    .59876    .49637   .64681
N7      94   .59342    .63309   .71524
C8      6   .55830  .37589   .58492
C9       6   .56150    .22316   .51791
C10     6   .60540   .18735   .50631
O1      8   .4875   .1210   .1801
O2      8   .4795   -.1669   .1760
H2N2    1   .6774  -.2385   .3609
H1N2    1   .6255  -.2778   .3351
H2N4    1   .7540   .3509   .5636
H1N4    1   .7346   .4931   .6438
H1N7     1   .5633   .7409   .7075
H2N7     1   .6160   .7731   .7418
HC8     1   .5289   .3905   .5909
HC9     1   .5314   .1664   .4721
////
```

<INSTRUCTION FILE>:

```
CONTRLCARD    24    8    1    1
   1.000     0.000     1.000     0.000      1.000     0.000
   -1.00     0.000     -1.00     0.000     -1.00     0.000
  -1.00     0.000      1.000     0.000     -1.00     0.500
   1.000     0.000     -1.00     0.000     1.000     0.500
   1.000     0.500     1.000     0.500      1.000     0.000
   -1.00     0.500     -1.00     0.500     -1.00     0.000
  -1.00     0.500      1.000     0.500     -1.00     0.500
   1.000     0.500     -1.00     0.500     1.000     0.500
CELL  CARD          2  29.955     3.853     17.742     90.000     109.44     90.00
PROJECTION     0   1    0.0     1.0    0.0    3.0    50.    200.
LINES CARD        100
PROJECTION     8   1   0.0        1.0   0.0   8.0    800.    500.
 1909090 0
 2909090
 3909090
 4909090
 5909090 0
 6909090
 7909090
 8909090
////

====
```



| A B S O R B |

Description

ABSORB is a program for the calculation of x-ray linear absorption coefficients and crystal density from unit cell contents. The methods of calculating these parameters are described in the author's thesis (pp 46-47).

Programming Notes

The program has been implemented only on the ICL 1904S computer. It contains the following non-standard Fortran code: Enclosure of Hollerith fields within quotation marks. The current versions of the program are:

```
Source: EABSORB
Binary  : EABSORBBN
```

There is only one data file. The program may be run with use of the system UAFORTRAN macro:

```
UAFORTRAN LOAD EABSORBBN, *CR1 <Data File Name>
```

The core requirement is 6 K.

Data are stored within the program for Copper K-alpha and Molybdenum K-alpha radiation with atomic masses and mass absorption coefficients for the following elements: H, C, N, O, F, P, S, Cl, Br, and I.

Data File

(1) Control Card

Format: 10X,2I5,15A4

Purpose: To input the radiation type and the number of formula weights and title of the crystal structure. A number of independent calculations may be carried out in a single run in which case the sequence Card (1) to Card (3) should be repeated. A blank card may be used as a terminator for the Data File.

Input Variables:

IRAD — Type of radiation:
1. Copper K-alpha
2. Molybenum K-alpha
Note: Input of IRAD=0 ends program execution.

NZCELL — The number of formula units per unit cell, i.e. the number of times the asymmetric unit is repeated within the unit cell.

ITLE(I) — The title of the crystal structure. Up to 60 alphanumeric characters.

(2) Cell Card

Format: 10X,I10,6F10.0

Purpose: To input unit cell parameters.

Input Variables:
See Standard Formats.

(3) Formula Card

Format 16I5

Purpose: To input chemical formula.

Input Variables:

IZ(I) — Atomic Number

NZ(I) — Number of times the element of atomic number IZ(I) occurs in the formula unit

Note: — Only one Formula Card is allowed. Up to eight elements may be entered with the order of input IZ(1), NZ(1), IZ(2), NZ(2),etc.

ABSORB: SAMPLE DATA FILE - P 1

```
ABSORBSION     1     4 CRYSTAL: DIPB
CELL  CARD           2  8.9290   20.0802    8.9786    90.000    114.05    90.000
   1   16    6   16    7    4
ABSORBSION     2     4 CRYSTAL: CGT
CELL  CARD           2  8.8730    10.391    17.140    90.000    115.24    90.00
   1   15    6   11    7    5   17    2
ABSORBSION     2     4 CRYSTAL: DETT
CELL  CARD           2  11.329     9.333    16.094    90.000    104.48    90.
   1   17    6   18    7    5    8    1
ABSORBSION     2    16 CRYSTAL: DAQ
CELL  CARD           1  21.59300  21.59300   7.55000 90.00000  90.0000  90.00000
   1   10    6    8    7    4    8    1
ABSORBSION     2     4 CRYSTAL: HOB
CELL  CARD           2  15.04500   4.28800  10.95100 90.00000 103.3200  90.00000
   1    5    6    7    7    3    8    2
```

```
FORTRAN COMPILATION BY #XFIV MK 3B    DATE  30/10/79  TIME  10/47/05


      LIST(LP)
      PROGRAM(FXXX)
      INPUT 4=CR0
      INPUT 5=CR1
      OUTPUT 6=LP1
      CREATE 1=MT0(SCRATCH)
      INPUT 3=CR2
      OUTPUT 2=CP0
      OUTPUT 7=TP0
      COMPRESS INTEGER AND LOGICAL
      COMPACT
      TRACE 0
      END



      MASTER BAKSUB
C ******************************************************************************
C
C                                PROGRAMME BAKSUB
C
C ******************************************************************************
C
C CORRECTS FOR
C       1  PARALYSIS TIME OF COURNTER
C       2  LORENTZPOLARISATION FACTOR
C        3  TRAPEZOIDAL BACKGROUND SUBTRACTION
C
C OUTPUTS
C       1   CORRECTED PEAK INTENSITY
C       2  STANDARD DEVIATION OF PEAK INTESITY DUE TO
C            A) COUNTING STATISTICS
C           B) LONG TERM BEAM INTENSITY FLUCTUATIONS
C       3  BACKGROUNDS 1 AND 2 COMPARED TO LOCAL AVERAGE
C       4  PEAK INTENSITY AS A MULTIPLE OF ITS STANDARD DEVIATION
C      5  INVERSE LORENTZPOLARISATION FACTOR
C
      DIMENSION NRC(10),NPKSUM(10),PKSQ(10),PKCHAV(10),FSV(10),FSD(10),
     1NTIMES(10),NREC(10),BG1AV(10),BG2AV(10),NREX(20),SUMBG1(20),
     2SUMBG2(20),NB1SUM(10),NB2SUM(10)
      DIMENSION MSRAN(17),SRAN(17)
C
      NTAPOT=2
      NOUT=6
      NCOPY =3
      NIN=4
      ISCRAT=1
      NCARD=5
      NFILE=7
C
C *******FORMAT STATEMENT DEPENDS ON INPUT********
C
 5000 FORMAT(3I3,I2,I6,I8,I6)
C
C *********************************************
C SPECIFY WRITE PARAMETERS
C     IW1=0 TO SUPPRESS COPY OF DATA
C     IW2=0 TO SUPPRESS OUTPUT OF CORRECTED PEAKS
      READ(NCARD,5802) IW1,IW2
 5802 FORMAT(10X,2I10)
C READ WAVELENGTH
      READ(NCARD,5800) WLAM
 5800 FORMAT(10X,F10.0)
C READ CELL DATA
C        NCELL=1 FOR ORTHORHOMBIC &  HIGHER SYMMETRY
C        NCELL=2 FOR MONOCLINIC
C        NCELL=3  FOR TRICLINIC
C           !!!ALL PARAMETERS MUST BE SUPPLIED!!
      READ(NCARD,5805) NCELL,A,B,C,ALFD,BETD,GAMD
 5805 FORMAT(10X,I10,6F10.0)
      DTOR=ATAN(1.)/45.
      ALPHA=DTOR*ALFD
      BETA=DTOR*BETD
      GAMMA=DTOR*GAMD
C READ CHECK REFLEXION IDENTIFIER
      READ(NCARD,5810) JCHECK
 5810 FORMAT(10X,I10)
C READ PARALYSIS TIME OF COUNTER
      READ(NCARD,5815) PARTIM
 5815 FORMAT(10X,F10.0)
C READ PEAK AND BACKGROUND COUNT TIMES
      READ(NCARD,5820) PKTIM,BG1TIM,BG2TIM
 5820 FORMAT(10X,3F10.0)
C        INTENSITY FLUCTUATION PARAMETER DEFAULT
C          JOVER=0:  PARAMETER SUPPLIED BY PROG
C            JOVER=1: PARAMETER SUPPLIED BY USER
C               JOVER=2:  PARAMETER SUPPLIED BY PROG ONLY IF
C                          INCONSISTANCIES FOUND IN CHECK REFS.
      READ(NCARD,5825) JOVER,VFP0
 5825 FORMAT(10X,I10,F10.0)
      READ(NCARD,5920) NSRAN,(SRAN(J),J=1,NSRAN)
 5920 FORMAT(10X,I2,17F4.0)
C
C GENERATE RECIPROCAL CELL
C
      CALL RECIP(NCELL,WLAM,A,B,C,ALPHA,BETA,GAMMA,AST,BST,CST,COSAST,
     1COSBST,COSGST)
C
C SCAN ROUTINE
      NCHECK=0
      IF(IW1.NE.0) WRITE(NOUT,5710)
 5710 FORMAT(1X,'REFLEXION DATA')
 5711 FORMAT(5X,3I3,I2,I6,I8,I6)
C BEGIN SCAN FOR CHECK REFLEXIONS
 2000 READ(NIN,5000) IH,K,L,ICHECK,NB1,NPK,NB2
      IF(IH.GT.998) GO TO 2100
C
      CALL MODIFY(IH,K,L,IFAIL)
      IF(IFAIL.NE.0)WRITE(NOUT,5713) IH,K,L
 5713 FORMAT(5X,3I3,'****REFLEXION UNEXPECTED ***')
      IF(IFAIL.NE.0) GO TO 2000
C
      IF(IW1.NE.0)WRITE(NOUT,5711) IH,K,L,ICHECK,NB1,NPK,NB2
      IF (ICHECK.NE.JCHECK) GO TO 2000
      MREC=10000*(IH+50)+100*(K+50)+L+50
      IF (NCHECK.EQ.0) GO TO 2010
      DO 1001 J=1,NCHECK
      IF (MREC.NE.NREC(J)) GO TO 1001
      NPKSUM(J)=NPKSUM(J)+NPK
      NB1SUM(J)=NB1SUM(J)+NB1
      NB2SUM(J)=NB2SUM(J)+NB2
      FLTNPK=FLOAT(NPK)
      PKSQ(J)=PKSQ(J)+FLTNPK*FLTNPK
      NTIMES(J)=NTIMES(J)+1
      GO TO 2000
 1001 CONTINUE
C ARRIVAL AT THIS POINT MEANS THAT REFLEXION HAS BEEN ENCOUNTERED FOR
C                 FIRST TIME
 2010 NCHECK=NCHECK+1
      NREC(NCHECK)=MREC
      NB1SUM(NCHECK)=NB1
      NB2SUM(NCHECK)=NB2
      NPKSUM(NCHECK)=NPK
      FLTNPK=FLOAT(NPK)
      PKSQ(NCHECK)=FLTNPK*FLTNPK
      NTIMES(NCHECK)=1
      GO TO 2000
 2100 NDIAG=0
C ALL CHECK REFLEXIONS HAVE NOW BEEN READ IN
C CALCULATE STANDARD DEVIATIONS AND SIGMA DUE TO COUNTING STATISTICS
C NCHECK IS NOW THE NUMBER OF DISTINCT CHECK REFLEXIONS
      DO 1101 J=1,NCHECK
      MTIMES=NTIMES(J)
      FLOATM=FLOAT(MTIMES)
      PKMEAN=FLOAT(NPKSUM(J))/FLOATM
      B1MEAN=FLOAT(NB1SUM(J))/FLOATM
      B2MEAN=FLOAT(NB2SUM(J))/FLOATM
      DENOM=FLOAT(MTIMES-1)
      IF (DENOM.LT.0.5) DENOM=1.
C SETS VAR=0. FAULT WILL BE DETECTED 6 LINES LAYER
      VAR=(PKSQ(J)-FLOAT(MTIMES)*PKMEAN*PKMEAN)/DENOM
      IF(VAR.LT.0.)VAR=0.
      SD=SQRT(VAR)
      FSD(J)=SD/PKMEAN
      PKCHAV(J)=PKMEAN
      BG1AV(J)=B1MEAN
      BG2AV(J)=B2MEAN
      VARDIF=VAR-PKMEAN/DENOM
      IF(VARDIF.LT.0.) NDIAG=NDIAG+NREC(J)
C IF NCHECK NE ZERO SD IS LESS THAN PREDICTED
C NDIAG IS THE SUM OF RECOGNITION NUMBERS OF THE REFLEXIONS AFFECTED
C THIS IS A WARNING....
      IF (VARDIF.LT.0.) VARDIF=0.
      SV=SQRT(VARDIF)
      FSV(J)=SV/PKMEAN
 1101 CONTINUE
C VOLTAGE FLUCTUATION PARAMETER IS WEIGTED BY
C                  NUMBER OF TIMES ENCOUNTERED  X  MEAN PEAK INTENSITY
      SUMNUM=0.0
      SUMDEN=0.0
      DO 1201 J=1,NCHECK
      SUMNUM=FSV(J)*PKCHAV(J)*FLOAT(NTIMES(J))+SUMNUM
      KTIMES=NTIMES(J)
      SUMDEN=PKCHAV(J)*FLOAT(KTIMES)+SUMDEN
 1201 CONTINUE
      VFP=SUMNUM/SUMDEN
      IF (JOVER.EQ.1) VFP=VFPO
      IF(JOVER.EQ.2.AND.NDIAG.NE.0) VFP=VFPO
      WRITE(NOUT,5610)VFP
 5610 FORMAT(/5X,'INTENSITY FLUCTUATION PARAMETER=',F10.5)
C
C   INITIAL OUTPUT
C
      WRITE(NOUT,5701) NCELL, WLAM
 5701 FORMAT(//1X,'CELL TYPE=',I3, 10X, 'WAVELENGTH=',
     1F10.5,'  ANGSTROMS'//)
      WRITE(NOUT,5702)
 5702 FORMAT(1X,'DIRECT CELL')
      WRITE(NOUT,5703) A,B,C,ALFD,BETD,GAMD
 5703 FORMAT(10X,'A=',F10.5,'  ANG.         B=', F10.5,'ANG.          C=',
     1F10.5,'  ANG.'/10X,'ALPHA=',F10.5,'  DEG.   BETA=',
     2F10.5,'  DEG.  GAMMA=',F10.5,'  DEG.')
      WRITE(NOUT,5704)
 5704 FORMAT(//1X,'RECIPROCAL CELL')
      WRITE(NOUT,5705)AST,BST,CST,COSAST,COSBST,COSGST
 5705 FORMAT(10X,'ASTAR=',F10.5,'                  BSTAR=',F10.5,
     1'            CSTAR=',F10.5/10X,'COS(ALPHASTAR)=',F10.5,
     2'    COS(BETASTAR)=',F10.5,'  COS(GAMMASTAR)=',F10.5)
C
C
C
      PARPK=PARTIM/PKTIM
      PARBG1=PARTIM/BG1TIM
      PARBG2=PARTIM/BG2TIM
      DO 3000 J=1,20
      NREX(J)=0
      SUMBG1(J)=0.0
      SUMBG2(J)=0.0
 3000 CONTINUE
      DO  3001 J=1,NCHECK
      PEAK=PKCHAV(J)/((1.-PKCHAV(J)*PARPK)*PKTIM)
      BG1=BG1AV(J)/((1.-BG1AV(J)*PARBG1)*BG1TIM)
      BG2=BG2AV(J)/((1.-BG2AV(J)*PARBG2)*BG2TIM)
      FRSIG=FSD(J)
C
C REGENERATE H,K,L
C
      MREC=NREC(J)
      IH=MREC/10000
      MREC=MREC-10000*IH
      K=MREC/100
      L=MREC-100*K
       L=L-50
       K=K-50
       IH=IH-50
C ANGLE ROUTINE
      Q=FLOAT(IH*IH)*AST*AST + FLOAT(K*K)*BST*BST + FLOAT(L*L)*CST*CST
      IF(NCELL.EQ.1) GO TO 2899
      Q=Q+FLOAT(2*IH*L)*AST*CST*COSBST
      IF(NCELL.EQ.2) GO TO 2899
      Q=Q+FLOAT(2*K*L)*BST*CST*COSAST + FLOAT(2*IH*K)*AST*BST*COSGST
 2899 DSTAR=SQRT(Q)
C THERE ARE TWENTY RANGES FROM SIN0 TO SIN90 (DSTAR=0 TO DSTAR=2)
      NRANGE=IFIX(10.*DSTAR+1.)
      NREX(NRANGE)=NREX(NRANGE)+1
      SUMBG1(NRANGE)=SUMBG1(NRANGE)+BG1
      SUMBG2(NRANGE)=SUMBG2(NRANGE)+BG2
C LORENTZPOLARISATION ROUTINE
      TEMP=DSTAR*SQRT(1.-0.25*DSTAR*DSTAR)
      FACLPI=2.*TEMP/(2.-TEMP*TEMP)
      WRITE(ISCRAT) IH,K,L,JCHECK,BG1,PEAK,BG2,FRSIG,FACLPI,NRANGE
 3001 CONTINUE
      NREF=NCHECK
C SCAN THROUGH TAPE REJECTING CHECK REFLEXIONS AND APPLYING CONDITIONS
C
C
 2200 READ(NCOPY,5000) IH,K,L,ICHECK,NB1,NPK,NB2
      IF(IH.GT.998)GO TO 2999
C
      CALL MODIFY(IH,K,L,IFAIL)
      IF(IFAIL.NE.0) GO TO 2200
C
      IF(IH.GT.998) GO TO 2999
      IF(ICHECK.EQ.JCHECK)GO TO 2200
      NREF=NREF+1
      PEAK=FLOAT(NPK)/((1.-PARPK*FLOAT(NPK))*PKTIM)
      BG1=FLOAT(NB1)/((1.-PARBG1*FLOAT(NB1))*BG1TIM)
      BG2=FLOAT(NB2)/((1.-PARBG2*FLOAT(NB2))*BG2TIM)
C STANDARD DEVIATION * SEE RESEARCH NOTES P4
      VAR=PEAK/PKTIM + BG1/BG1TIM + BG2/BG2TIM + VFP*(PEAK*PEAK+BG1*BG1
     1+BG2*BG2)
      IF(VAR.LT.0.)VAR=0.
      FRSIG=SQRT(VAR)/PEAK
C ANGLE ROUTINE
      Q=FLOAT(IH*IH)*AST*AST + FLOAT(K*K)*BST*BST + FLOAT(L*L)*CST*CST
      IF(NCELL.EQ.1) GO TO 2898
      Q=Q+FLOAT(2*IH*L)*AST*CST*COSBST
      IF(NCELL.EQ.2) GO TO 2898
      Q=Q+FLOAT(2*K*L)*BST*CST*COSAST + FLOAT(2*IH*K)*AST*BST*COSGST
 2898 DSTAR=SQRT(Q)
C LORENTZPOLARISATION ROUTINE
      TEMP=DSTAR*SQRT(1.-0.25*DSTAR*DSTAR)
      FACLPI=2.*TEMP/(2.-TEMP*TEMP)
      NRANGE=IFIX(10.*DSTAR+1.)
      NREX(NRANGE)=NREX(NRANGE)+1
      SUMBG1(NRANGE)=SUMBG1(NRANGE)+BG1
      SUMBG2(NRANGE)=SUMBG2(NRANGE)+BG2
      WRITE(ISCRAT)IH,K,L,ICHECK,BG1,PEAK,BG2,FRSIG,FACLPI,NRANGE
      GO TO 2200
 2999 CONTINUE
C
C
      WRITE(NOUT,5606)
 5606 FORMAT(////' RANGE     NO.REF           AV.BG1    AV.BG2  '//)
      DO 3002 J=1,20
      NREXJ=NREX(J)
      IF(NREXJ.EQ.0) GO TO 3002
      SUMBG1(J)=SUMBG1(J)/FLOAT(NREX(J))
      SUMBG2(J)=SUMBG2(J)/FLOAT(NREX(J))
C SUMBG1 AND SUMBG2 ARE NOW THE AVERAGE FOR THE RANGE
      WRITE(NOUT,5605) J,NREX(J),SUMBG1(J),SUMBG2(J)
 5605 FORMAT(2I10,5X,2F10.5)
 3002 CONTINUE
C
      REWIND ISCRAT
C
      IF(IW2.NE.0) WRITE(NOUT,5621)
 5621 FORMAT(////' CHK     H  K  L          REL.BG1   REL.BG2             LP-1
     1         CORR.PEAK.      SD    PK.WRT.SD  RGE'//)
C
C
 5620 FORMAT(I3,5X,3I3,5X,2F10.5,5X,F10.8,F12.2,F10.2,F10.2,5X,I3)
C
      DO 5922 LL=1,NSRAN
 5922 MSRAN(LL)=0
C
C
        NSHEL=1
C
      WRITE(NTAPOT,5945)
 5945 FORMAT('HKLF  -3')
C
      DO 3100 J=1,NREF
      READ(ISCRAT) IH,K,L,ICHECK,BG1,PEAK,BG2,FRSIG,FACLPI,NRANGE
      SIG=FRSIG*PEAK*FACLPI
      PEAK=(PEAK-BG1-BG2)*FACLPI
      ABSPK=PEAK/SIG
C
      DO 5925 LL=1,NSRAN
      IF(ABSPK.LT.SRAN(LL)) GO TO 5930
      MSRAN(LL)=MSRAN(LL) + 1
 5925 CONTINUE
 5930 CONTINUE
      ABSB1=BG1/SUMBG1(NRANGE)
      ABSB2=BG2/SUMBG2(NRANGE)
C
C
      WRITE(NFILE,5620) ICHECK,IH,K,L,ABSB1,ABSB2,FACLPI,PEAK,SIG,ABSPK,
     1NRANGE
      IF(IW2.NE.0) WRITE(NOUT,5620) ICHECK,IH,K,L,ABSB1,
     1ABSB2,FACLPI,PEAK,SIG,ABSPK,NRANGE
C
      IF(PEAK.LE.0.0) GO TO 5900
      STRFAC=SQRT(PEAK)
      STRSIG=0.5*SIG/STRFAC
      GO TO 5905
 5900 CONTINUE
      STRFAC=0.0
      STRSIG=1000.0
 5905 CONTINUE
 C      CHANGE TO CONVENTIONAL CELL  IH=(H+L)/2,L=(-H+L)/2
CONVERT B21/A TO P21/N
C
      IHT=(IH+L)/2
      L=(L-IH)/2
      IH=IHT
C
C
      WRITE(NTAPOT,5627) IH,K,L,STRFAC,STRSIG,NSHEL
 5627 FORMAT(3I4,2F8.2,I4)
C
 3100 CONTINUE
C
      WRITE(NTAPOT,5946)
 5946 FORMAT(/'END')
C
      WRITE(NOUT,5930)
 5930 FORMAT(5X,'RATIO PEAK/SIG(PEAK)',5X,'NO. REFLEXIONS GT. THAN'/)
      DO 5935 J=1,NSRAN
 5935 WRITE(NOUT,5940) SRAN(J),MSRAN(J)
 5940 FORMAT(5X,F10.4,I5)
C
      STOP
      END

END OF SEGMENT, LENGTH  1356, NAME  BAKSUB

C
C
C
C
         SUBROUTINE RECIP(NCELL,WLAM,A,B,C,ALPHA,BETA,GAMMA,AST,BST,CST,
 0366               1COSAST,COSBST,COSGST)
 0367                     IF(NCELL.EQ.3) GO TO 2885
 0368                 AST=WLAM/A
 0369                 BST=WLAM/B
 0370                 CST=WLAM/C
 0371                 COSAST=0.
 0372                 COSBST=0.
 0373                 COSGAM=0.
 0374                 IF(NCELL.EQ.1) GO TO 2887
 0375                 FISIN=1./SIN(BETA)
 0376                 AST=AST*FISIN
 0377                 CST=CST*FISIN
 0378                 COSBST=-COS(BETA)
 0379                 GO TO 2887
 0380            2885 COSALF=COS(ALPHA)
 0381                 COSBET=COS(BETA)
 0382                 COSGAM=COS(GAMMA)
 0383                 SINALF=SIN(ALPHA)
 0384                 SINBET=SIN(BETA)
 0385                 SINGAM=SIN(GAMMA)
 0386                 V=A*B*C*(1.+2.*COSALF*COSBET*COSGAM-COSALF*COSALF-COSBET*COSBET-
 0387                1COSGAM*COSGAM)
 0388                 VST=1./V
 0389                 AST=WLAM*B*C*SINALF*VST
 0390                 BST=WLAM*C*A*SINBET*VST
 0391                 CST=WLAM*A*B*COSGAM*VST
 0392                 COSAST=(COSBET*COSGAM-COSALF)/(SINBET*SINGAM)
 0393                 COSBST=(COSGAM*COSALF-COSBET)/(SINGAM*SINALF)
 0394                 COSGST=(COSALF*COSBET-COSGAM)/(SINALF*SINBET)
 0395            2887 CONTINUE
 0396                 RETURN
 0397                 END

END OF SEGMENT, LENGTH   215, NAME  RECIP

 0398            C
 0399                 SUBROUTINE MODIFY(IH,K,L,IFAIL)
 0400            C REDUCE DATA TO PRIMITIVE CELL
 0401                 RETURN
 0402                 END

END OF SEGMENT, LENGTH    28, NAME  MODIFY

 0403                 FINISH

END OF COMPILATION -  NO ERRORS

     S/C SUBFILE  :     34 BUCKETS USED

  FIRST WORKFILE  :     44 BUCKETS USED

 SECOND WORKFILE  :     33 BUCKETS USED

CONSOLIDATED BY XPCK 12H    DATE    30/10/79   TIME 10/54/29

        *SHORTLIST

        *IN ED (FORTSEMICOMP)

        *LIB ED (SUBGROUPSRF4.SUBROUTINES)

        *WORK ED (FORTWORKFILE)

        ****

PROGRAM FXXX
COMPACT DATA (15AM)
COMPACT PROGRAM (DBM)

SEGMENTS MISSING
%FDP

CORE             6912


*AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
*AAAAAAA                                 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
*AAAAAAA       NUMBER OF PAGES    11     AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
*AAAAAAA                                 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
*AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

FORTRAN COMPILATION BY #XFIV MK 3B   DATE 30/10/79 TIME 10/43/41

```
    LIST(LP)
    PROGRAM (FXXX)
    INPUT 1= CR0
    INPUT 3 = TR0
    USE 4=MT0
    INPUT 5 =CR1
    OUTPUT 2 = CP0
    OUTPUT 6 =LP1
    OUTPUT 8 = CP1
    COMPRESS INTEGER AND LOCICAL
    EXTENDED DATA
    TRACE 0
    END



     MASTER MOLJOM
CAUTION:
CAUTION: FREE FORMAT STATEMENTS USED
CAUTION:
     COMMON/XYZ/X(100),Y(100),Z(100),NAME(100)
     COMMON/LMN/ALA(10),AMA(10),ANA(10)
     COMMON/PARAM/NATOMS,NIN,NOUT,NCARD
     COMMON/LIN/LINE(35),NATOM,ICYCL(10)
     COMMON/CELL/NCELL,A11,A21,A22,A31,A32,A33
     COMMON/MATRIX/B11,B12,B13,B22,B23,B33
     COMMON/PROJOM/NFILE,ISHOW,ICMIN,JOIN,IVEC
     COMMON/SYMMER/MSYMEL(16)
     COMMON/SYMMEL/NSYMM,RX(16),TX(16),RY(16),TY(16),RZ(16),TZ(16)
     COMMON/PARAM2/MODE,IBOND,BMIN,BMAX,PHIMIN,PHIMAX
     COMMON/ROTOP/IROTOP,MAGTAP
     NFILE=2
      MAGTAP=4
     NIN=5
     NOUT=6
     NCARD=1
     READ(NCARD,5000)NATOMS,NSYMM,MODE,ISHOW,IFORM,IQUIT,ICONN,ICMIN,
    1IBOND
     IF(IBOND.NE.0) READ(NCARD,5012) BMIN,BMAX,PHIMIN,PHIMAX
5012 FORMAT(10X,4F10.0)
     IF(NSYMM.EQ.0) GO TO 5060
     WRITE(NOUT,5065)
5065 FORMAT(25X,'SYMMETRY OPERATIONS'/
    110X,'ROTATION',20X,'TRANSLATION')
     DO 5061 J=1,NSYMM
     READ(NCARD,5062) RX(J),TX(J),RY(J),TY(J),RZ(J),TZ(J),MSYMEL(J)
5062 FORMAT(6F10.0,I10)
     WRITE(NOUT,5066) RX(J),RY(J),RZ(J),TX(J),TY(J),TZ(J),MSYMEL(J)
5066 FORMAT(5X,3F8.3,5X,3F8.3,I10)
5061 CONTINUE
     IF(MODE.NE.0) WRITE(NOUT,5063)
5063 FORMAT(5X,'SYMMETRY OPERATIONS NOT ALLOWED WHEN ANGSTROM CORRDINATES
    1ES SUPPLIED'/5X,'SYMMETRY CARDS WILL BE IGNORED')
5060 CONTINUE
     IF(MODE.NE.0.OR.NSYMM.EQ.0) NSYMM=1
     IF(ISHOW.GE.2)WRITE(NOUT,5010)
5010 FORMAT(/5X,'FRACTIONAL COORDINATES'/)
5000 FORMAT(10X,9I5)
     IF(BMIN.LT..001.OR.BMAX.LT..001) IBOND=0
     IF(IFORM.NE.0) GO TO 5055
     READ(NIN,5001) SCALE,BFAC
5001 FORMAT(F10.7/F10.7)
     DO 5050 J=1,NATOMS
     READ(NIN,5002) NAME(J),X(J),Y(J),Z(J)
5002 FORMAT(A4,23X,3F9.7/)
     IF(ISHOW.GE.2)WRITE(NOUT,5004) NAME(J),X(J),Y(J),Z(J)
     IF(ISHOW.EQ.3)WRITE(NFILE,5007)X(J),Y(J),Z(J)
5007 FORMAT(3F10.5)
5050 CONTINUE
     GO TO 5057
5055 CONTINUE
     KOUNT=0
     DO 5058 J=1,NATOMS
     READ(NIN,5006)NAME(J),X(J),Y(J),Z(J)
     IF(X(J).LT.-999.0) GO TO 5056
     IF(ISHOW.GE.2) WRITE(NOUT,5004) NAME(J),X(J),Y(J),Z(J)
     KOUNT=KOUNT+1
5006 FORMAT(A4,3F0.0)
     ONE=1.0
5003 FORMAT(A4,5X,5F9.6/)
     IF(IFORM.EQ.2)WRITE(NFILE,5192)NAME(J),ONE,ONE,X(J),Y(J),Z(J)
5192 FORMAT(A4,5X,5F9.6/)
5058 CONTINUE
5056 NATOMS=KOUNT
5057 CONTINUE
     IF (MODE.EQ.0) CALL CARTES
     IF(ICMIN.GE.2) CALL CONMIN
     IF(ICMIN.EQ.1) CALL SELFIM
     IF(MODE.EQ.0) CALL CARTEX
```

```
      IF(ISHOW.GE.1)WRITE(NOUT,5110)
 5110 FORMAT(/5X,'ANGSTROM COORDINATES'/)
      DO 5051 J=1,NATOMS
      IF(ISHOW.GE.1)WRITE(NOUT,5004) NAME(J),X(J),Y(J),Z(J)
      IF(ISHOW.EQ.4)WRITE(NFILE,5004)NAME(J),X(J),Y(J),Z(J)
 5004 FORMAT(5X,A4,5X,3F7.3)
 5051 CONTINUE
      IF(IQUIT.NE.0) GO TO 5175
      READ(NCARD,5100) NLINES,NROT,IROTOP
C*** IROTOP=1 OUTPUT PROJOECTION COORDS TO MAG TAPE ONLY
C***           2 TO LINEPTINTER ONLY
C***             3 OUTPUT TO BOTH
C**       =4 TO LINEPRINTER AND CARDPUNCH FILE
      NTLINE=NLINES
C STORE WHICH LINES ARE CYCLED IN ARRAY ICYCL
      IF(NLINES.LE.0) GO TO 5150
      DO 5053 J=1,NLINES
      ICYCL(J)=0
 5053 CONTINUE
C LINES WHICH ARE CYCLED WILL BE CHANGED TO ICYCL=1
C IN SUBROUTINE LINES
      MLINES=NLINES+1
 5100 FORMAT(10X,3I10)
 5140 IF(NLINES.LE.0) GO TO 5150
      MLINE=MLINES-NLINES
      CALL LINES(MLINE)
C
      NLINES=NLINES-1
      WRITE(NOUT,5201)
 5201 FORMAT(/)
      GO TO 5140
 5150 CONTINUE
      DO 5166 J=1,NTLINE
      IF(ICYCL(J).EQ.0) GO TO 5166
      JP=J+1
      DO 5167 K=JP,NTLINE
      IF(ICYCL(K).EQ.0) GO TO 5169
      X1=ALA(J)
      Y1=AMA(J)
      Z1=ANA(J)
      X2=ALA(K)
      Y2=AMA(K)
      Z2=ANA(K)
      CALL ANGLE(X1,Y1,Z1,X2,Y2,Z2,PHI)
      IF(PHI.GT.90.0)PHI=180.0-PHI
      WRITE(NOUT,5168)J,K,PHI
 5168 FORMAT(5X,'ANGLE BETWEEN PLANE',I3,'AND PLANE',I3,
     1'  IS',F6.2,'  DEGREES')
 5169 CONTINUE
 5167 CONTINUE
 5166 CONTINUE
      IF(ICONN.EQ.1) CALL CONN
      IF(NROT.EQ.0) GO TO 5249
C**
C**  ROTATION-PROJECTION SECTION
C**
      DO 5250  J=1,NROT
      READ(NCARD,5252)IROTYP,IROT,K,L
 5252 FORMAT(6X,4I2)
      IF(IROTYP.EQ.1) GO TO 5180
      IF(ICYCL(IROT).EQ.0.OR.IROT.GT.NTLINE) GO TO 5248
      XX=ALA(IROT)
      YY=AMA(IROT)
      ZZ=ANA(IROT)
      X1=X(L)-X(K)
      Y1=Y(L)-Y(K)
      Z1=Z(L)-Z(K)
      WRITE(NOUT,5257) X1,Y1,Z1,XX,YY,ZZ
 5257 FORMAT(5X,'ORTHOGANALISED VIEWING AXES--AXES SUPPLIED:'/
     15X,3F10.5,35X,3F10.5)
      CALL ORAXIS(X1,Y1,Z1,X2,Y2,Z2,XX,YY,ZZ)
      WRITE(NOUT,5258) X1,Y1,Z1,X2,Y2,Z2,XX,YY,ZZ
 5258 FORMAT(5X,'AXES CHOSEN:'/3(5X,3F10.5))
      IF(X1.LT.-1.0) GO TO 5247
      WRITE(NOUT,5254) IROT,NAME(K),NAME(L)
 5254 FORMAT(5X,'ROTATION FOR VIEWING PEPENDICULAR TO PLANE NO.',I4/
     15X,A4,'  AND  ',A4,'WILL BE PROJECTED ONTO THE X-AXIS')
      CALL ROTATE(X1,Y1,Z1,X2,Y2,Z2,XX,YY,ZZ)
      GO TO 5250
 5247 WRITE(NOUT,5255) IROT
 5255 FORMAT(5X,'AN X PROJECTION-AXIS PARALELL TO THE Z AXIS HAS BEEN
     1CHOSEN--PLANE NO.',I4/5X,'PROJECTION CALCULATIONS ON PLANE ABANDON
     2ED')
      GO TO 5250
 5248 WRITE(NOUT,5256) IROT
 5256 FORMAT(5X,'PLANE NO.',I4,'IS NOT LISTED--RECHECK CARDS')
      GO TO 5250
 5180 CONTINUE
      WRITE(NOUT,5182) IROT,NAME(K),NAME(L)
 5182 FORMAT(5X,'ROTATION FOR VIEWING ALONG PLANE',I3/
     15X,A4,'AND',A4,'WILL BE PARALLEL THE X-AXIS')
      X1=ALA(IROT)
      Y1=AMA(IROT)
      Z1=ANA(IROT)
      XX=X(L)-X(K)
```

```
MOLJOM: COMPILATION LISTING - P 3

 0181                 YY=Y(L)-Y(K)
 0182                 ZZ=Z(L)-Z(K)
 0183                 CALL NORM(XX,YY,ZZ,RAD)
 0184                 IF(RAD.LT.1.E-7) GO TO 5189
 0185                 WRITE(NOUT,5257) X1,Y1,Z1,XX,YY,ZZ
 0186                 CALL ORAXIS(X1,Y1,Z1,X2,Y2,Z2,XX,YY,ZZ)
 0187                 WRITE(NOUT,5258) X1,Y1,Z1,X2,Y2,Z2,XX,YY,ZZ
 0188                 CALL ROTATE(X1,Y1,Z1,X2,Y2,Z2,XX,YY,ZZ)
 0189                 GO TO 5250
 0190            5189 WRITE(NOUT,5183)
 0191            5183 FORMAT(5X,'INPUT VECTOR ZERO--RECHECK CARDS')
 0192            5250 CONTINUE
 0193            5249 CONTINUE
 0194            5175 CONTINUE
 0195                 STOP
 0196                 END

END OF SEGMENT, LENGTH   978, NAME  MOLJOM


 0197                  FINISH

END OF COMPILATION -  NO ERRORS


     S/C SUBFILE  :      28 BUCKETS USED

  FIRST WORKFILE  :      34 BUCKETS USED

 SECOND WORKFILE  :      25 BUCKETS USED

CONSOLIDATED BY XPCK 12H    DATE    30/10/79   TIME 10/47/58

        *SHORTLIST

        *IN ED (FORTSEMICOMP)

        *LIB ED (SUBGROUPSRF4.SUBROUTINES)

        *WORK ED (FORTWORKFILE)

        ****

PROGRAM FXXX
EXTENDED DATA (22AM)
COMPACT PROGRAM (DBM)

SEGMENTS MISSING
NORM
ROTATE
ORAXIS
CONN
ANGLE
LINES
CARTEX
SELFIM
CONMIN
CARTES

CORE             6336


+BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
+BBBBBBB                                 BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
+BBBBBBB       NUMBER OF PAGES    7      BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
+BBBBBBB                                 BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
+BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
```

```
FORTRAN COMPILATION BY #XFIV MK 3B    DATE  30/10/79  TIME  10/53/19


      LIST(LP)
      PROGRAM (FXXX)
      INPUT 1= CR0
      INPUT 3 = TR0
      INPUT 4=MT0 / (FURMAP)
      INPUT 5 =CR1
      OUTPUT 2 = CP0
      OUTPUT 6 =LP1
      COMPRESS INTEGER AND LOGICAL
      EXTENDED DATA
      TRACE 0
      END



      MASTER PEAKS
      COMMON/LIMITS/N1X,N1Y,N1Z,N2X,N2Y,N2Z
      COMMON/FUSIS/NO,M1I,M1F,M2I,M2F,M3I,M3F,M1D,M2D,M3D
      COMMON/PEAK/IJPK(900),LPK(900),HPK(900)
      COMMON/PARAM/NIN,NOUT,NTAPE,NCP
      COMMON/MINMUM/MINPK,HMINPK,KOUNT,LEVEL,NEGPK,KNTMAX
       COMMON/SIZE/NX,NY,NZ,NDIM
      COMMON/MATRIX/AMAT(31,31,3)
      KOUNT=0
      NTAPE=4
      NCP=2
      NIN=5
      REWIND NTAPE
      READ(5,1010) N1X,N2X,N1Y,N2Y,N1Z,N2Z,MINPK,HMINPK,NEGPK,KNTMAX
 1010 FORMAT(7I0,F0.0,2I0)
      WRITE(6,1008) NX,NY,NZ,MINPK,HMINPK,NEGPK
 1008 FORMAT(2X,'NX,NY,NZ,MINPK,HMINPK,NEGPK',3X,3I4,I3,F6.3,I3)
C
C  NEGPK = -1(NEG ONLY)  =0 (EITHER POS OR NEG) = +1 (POS ONLY)
C
C INTERPRET PHUSIS CARD
      READ(5,4075) NO,M1I,M1F,M2I,M2F,M3I,M3F,M1D,M2D,M3D
 4075 FORMAT(3X,I3,12X,9I3)
      NX=(M1F-M1I)/M1D + 1
      NY=(M2F-M2I)/M2D + 1
      NZ=(M3F-M3I)/M3D + 1
      N2X=MIN0(N2X,NX)
      N2Y=MIN0(N2Y,NY)
      N2Z=MIN0(N2Z,NZ)
      N1X=MAX0(N1X,1)
      N1Y=MAX0(N1Y,1)
      N1Z=MAX0(N1Z,1)
      WRITE(6,1048) NX,NY,NZ,N1X,N1Y,N1Z,N2X,N2Y,N2Z
 1048 FORMAT(15X,'ROWX COLS SECS'/5X,'MAP LIMITS',3I5/
     15X,'START AT',3I5/5X,'FINISH AT',3I5)
      READ(NTAPE) ((AMAT(I,J,1),J=1,NY),I=1,NX)
      READ(NTAPE)((AMAT(I,J,2),J=1,NY),I=1,NX)
      LEVEL=1
      IF(N1Z.EQ.1)CALL SCAN(1)
      N2ZP=N2Z + 1
      N2ZP=MIN0(N2ZP,NZ)
      DO 1050 LEV=3,N2ZP
      READ(NTAPE)((AMAT(I,J,3),J=1,NY),I=1,NX)
      LEVEL=LEV - 1
      CALL SCAN(2)
 1050 CALL RESTAK
      LEVEL=NZ
      IF(N2Z.EQ.NZ)CALL SCAN(3)
      CALL ORDER
      CALL COORDS
      STOP
      END

END OF SEGMENT, LENGTH   337, NAME  PEAKS

      SUBROUTINE SCAN(ISTAK)
      COMMON/PEAK/IJPK(900),LPK(900),HPK(900)
      COMMON/PARAM/NIN,NOUT,NTAPE,NCP
      COMMON/SIZE/NX,NY,NZ,NDIM
      COMMON/MINMUM/MINPK,HMINPK,KOUNT,LEVEL,NEGPK,KNTMAX
      COMMON/MATRIX/AMAT(31,31,3)
      COMMON/LIMITS/N1X,N1Y,N1Z,N2X,N2Y,N2Z
C  SET NDIM=CURRENT DIMENSIONING IN COMMON BLOCK /PEAK/
      NDIM=900
      DO 1100 II=N1X,N2X
      DO 1100 JJ=N1Y,N2Y
C
C
      CALL KONTAK(II,JJ,ISTAK)
      IF(KOUNT.GT.NDIM) RETURN
 1100 CONTINUE
      RETURN
      END

END OF SEGMENT, LENGTH    48, NAME  SCAN

      SUBROUTINE KONTAK(II,JJ,ISTAK)
      COMMON/PEAK/IJPK(900),LPK(900),HPK(900)
      COMMON/PARAM/NIN,NOUT,NTAPE,NCP
      COMMON/SIZE/NX,NY,NZ,NDIM
      COMMON/MINMUM/MINPK,HMINPK,KOUNT,LEVEL,NEGPK,KNTMAX
      COMMON/MATRIX/AMAT(31,31,3)
      HH=AMAT(II,JJ,ISTAK) + 1.E-10
      IF(MINPK.EQ.0) GO TO 2090
      IF(HH.GT.HMINPK.AND.NEGPK.GE.0) GO TO 2090
      FMINPK=-HMINPK
      IF(HH.LT.FMINPK.AND.NEGPK.LE.0) GO TO 2090
      RETURN
 2090 CONTINUE
      K1=1
      K2=3
      IF(ISTAK.EQ.1) K2=2
      IF (ISTAK.EQ.3) K1=2
      I1=II-1
      IF(I1.EQ.0) I1=1
      J1=JJ-1
      IF(J1.EQ.0) J1=1
      I2=II+1
      IF(I2.GT.NX)I2=NX
      J2=JJ + 1
      IF(J2.GT.NY) J2=NY
      IF(NEGPK.LT.0) GO TO 2101
      DO 2100 K=K1,K2
      DO 2100 J=J1,J2
      DO 2100 I=I1,I2
      IF(AMAT(I,J,K).GT.HH)GO TO 2101
 2100 CONTINUE
      GO TO 2120
 2101 CONTINUE
      IF(NEGPK.GT.0) RETURN
      HH=HH - 2.E-10
      DO 2110 K=K1,K2
      DO 2110 J=J1,J2
      DO 2110 I=I1,I2
      IF(AMAT(I,J,K).LT.HH) RETURN
 2110 CONTINUE
 2120 CONTINUE
      KOUNT=KOUNT + 1
C
C     TEST FOR OVERFLOW
      IF(KOUNT.LE.NDIM) GO TO 2105
      WRITE(6,2103) II,JJ,LEVEL
 2103 FORMAT(2X,'ARRAYS CONTAINING PEAKS HAVE OVERFLOWED AT'/
     15X,'ROW=',I5,' COL=',I5,' SECTION=',I5/
     25X,'YOU MAY SUPPRESS WEAK PEAKS BY SETTING MINPK=1 AND SPECIFYING'
     3,1X,'HMINPK')
      CALL DUMP
      RETURN
 2105 CONTINUE
      IJPK(KOUNT)=KODE(II,JJ)
      HPK(KOUNT)=HH
      CALL INTERP(II,JJ,KK,ISTAK,HH,ISET)
      LPK(KOUNT)=ISET
      RETURN
      END

END OF SEGMENT, LENGTH   260, NAME  KONTAK

      SUBROUTINE DUMP(NDIM)
      COMMON/PEAK/IJPK(900),LPK(900),HPK(900)
      COMMON/PARAM/NIN,NOUT,NTAPE,NCP
      DO 1070 I=1,NDIM
      WRITE(NCP,1072) IJPK(I),LPK(I),HPK(I)
 1072 FORMAT(5X,2I10,F10.3)
 1070 CONTINUE
      RETURN
      END

END OF SEGMENT, LENGTH    37, NAME  DUMP

      SUBROUTINE ORDER
      COMMON/PEAK/IJPK(900),LPK(900),HPK(900)
      COMMON/PARAM/NIN,NOUT,NTAPE,NCP
      COMMON/MINMUM/MINPK,HMINPK,KOUNT,LEVEL,NEGPK,KNTMAX
      COMMON/MATRIX/AMAT(31,31,3)
      KM=KOUNT -1
      DO 2300 K=1,KM
      KP=K + 1
      DO 2310 KK=KP,KOUNT
      IF(HPK(K).GE.HPK(KK)) GO TO 2310
      ITEMP=IJPK(K)
      IJPK(K)=IJPK(KK)
      IJPK(KK)=ITEMP
      ITEMP=LPK(K)
      LPK(K)=LPK(KK)
      LPK(KK)=ITEMP
      TEMP=HPK(K)
      HPK(K)=HPK(KK)
      HPK(KK)=TEMP
 2310 CONTINUE
 2300 CONTINUE
      RETURN
      END

END OF SEGMENT, LENGTH    83, NAME  ORDER

      FUNCTION KODE(II,JJ)
      COMMON/MINMUM/MINPK,HMINPK,KOUNT,LEVEL,NEGPK,KNTMAX
      KODE=10000*II + 100*JJ + LEVEL
      RETURN
      END

END OF SEGMENT, LENGTH    30, NAME  KODE

      SUBROUTINE RESTAK
      COMMON/MATRIX/AMAT(31,31,3)
      COMMON/SIZE/NX,NY,NZ,NDIM
      DO 2200 I=1,NX
      DO 2200 J=1,NY
      AMAT(I,J,1)=AMAT(I,J,2)
 2200 AMAT(I,J,2)=AMAT(I,J,3)
      RETURN
      END

END OF SEGMENT, LENGTH    35, NAME  RESTAK

      SUBROUTINE INTERP(II,JJ,KK,ISTAK,HH,ISET)
      COMMON/MATRIX/AMAT(31,31,3)
      COMMON/SIZE/NX,NY,NZ,NDIM
C 3-DIRECTION INTERPOLATION VERSION
      IF(ISTAK.EQ.2) GO TO 1860
      DZ=0.0
      GO TO 1865
 1860 CALL PARAB(AMAT(II,JJ,1),HH,AMAT(II,JJ,3),WT,DZ)
 1865 CONTINUE
      IF(II.EQ.1.OR.II.EQ.NZ) GO TO 1870
      IM=II - 1
      IP=II + 1
      CALL PARAB(AMAT(IM,JJ,ISTAK),HH,AMAT(IP,JJ,ISTAK),WT,DX)
      GO TO 1875
 1870 CONTINUE
      DX=0.0
 1875 CONTINUE
      IF(JJ.EQ.1.OR.JJ.EQ.NY) GO TO 1880
      JM=JJ-1
      JP=JJ + 1
      CALL PARAB(AMAT(II,JM,ISTAK),HH,AMAT(II,JP,ISTAK),WT,DY)
      GO TO 1885
 1880 DY=0.0
 1885 CONTINUE
      NEG1=IFIX(100.*DX + 50.5)
      NEG2=IFIX(100.*DY + 50.5)
      NEG3=IFIX(100.*DZ + 50.5)
      ISET=10201*NEG1 + 101*NEG2 + NEG3
      RETURN
      END

END OF SEGMENT, LENGTH   192, NAME  INTERP

      SUBROUTINE DECODE(ISET,DX,DY,DZ)
      IX=ISET/10201
      ISET=ISET - 10201*IX
      IY=ISET/101
      IZ=ISET - 101*IY
      DX=0.01*FLOAT(IX) - 0.5
      DY=0.01*FLOAT(IY) - 0.5
      DZ=0.01*FLOAT(IZ) - 0.5
      RETURN
      END

END OF SEGMENT, LENGTH    68, NAME  DECODE

      SUBROUTINE PARAB(YM,YO,YP,WT,DIST)
      WT=YP + YM - 2.0*YO
      DIST=0.5*(YM - YP)/WT
      RETURN
```

```
            END

END OF SEGMENT, LENGTH    45, NAME  PARAB

            SUBROUTINE COORDS
            COMMON/FUSIS/NO,M1I,M1F,M2I,M2F,M3I,M3F,M1D,M2D,M3D
            COMMON/PEAK/IJPK(900),LPK(900),HPK(900)
            COMMON/MINMUM/MINPK,HMINPK,KOUNT,LEVEL,NEGPK,KNTMAX
            NCP=2
            DD=1.0/120.0
            DDX=DD*FLOAT(M1D)
            DDY=DD*FLOAT(M2D)
            DDZ=DD*FLOAT(M3D)
            DIX=DD*FLOAT(M1I)
            DIY=DD*FLOAT(M2I)
            DIZ=DD*FLOAT(M3I)
            IF(KOUNT.GE.1) GO TO 1604
            WRITE(NCP,1606) HMINPK
       1606 FORMAT('NO PEAKS OVER',E11.4,' FOUND')
            RETURN
       1604 CONTINUE
      C
            KOUNT=MIN0(KOUNT,KNTMAX)
            DO 1600 K=1,KOUNT
            IJ=IJPK(K)
            LL=LPK(K)
            IX=IJ/10000
            IJ=IJ - 10000*IX
            IY=IJ/100
            IZ=IJ - 100*IY
            CALL DECODE(LL,DX,DY,DZ)
            DEX=DX + FLOAT(IX - 1)
            DEY=DY + FLOAT(IY - 1)
            DEZ=DZ + FLOAT(IZ - 1)
            XX=DIX + DDX*DEX
            YY=DIY + DDY*DEY
            ZZ=DIZ + DDZ*DEZ
      C
      C
            GO TO (1620,1625,1630,1635,1640,1645),NO
       1620 WRITE(NCP,1660) K,XX,YY,ZZ,HPK(K)
            GO TO 1650
       1625 WRITE(NCP,1660) K,XX,ZZ,YY,HPK(K)
            GO TO 1650
       1630 WRITE(NCP,1660) K,YY,XX,ZZ,HPK(K)
            GO TO 1650
       1635 WRITE(NCP,1660) K,YY,ZZ,XX,HPK(K)
            GO TO 1650
       1640 WRITE(NCP,1660) K,ZZ,XX,YY,HPK(K)
            GO TO 1650
       1645 WRITE(NCP,1660) K,ZZ,YY,XX,HPK(K)
       1650 CONTINUE
       1660 FORMAT('P',I3,3X,3(2X,F10.5),5X,F10.5)
       1600 CONTINUE
      C   PUT ON TERMINATOR
      C
            TFIN=-999.99
            WRITE(NCP, 1603) TFIN,TFIN,TFIN,TFIN
       1603 FORMAT('TERM',3X,3(2X,F10.5),5X,F10.5)
      C
            RETURN
            END

END OF SEGMENT, LENGTH   268, NAME  COORDS

            FINISH

END OF COMPILATION -  NO ERRORS

     S/C SUBFILE  :     43 BUCKETS USED

CONSOLIDATED BY XPCK 12H    DATE    30/10/79   TIME 10/57/37

         *SHORTLIST

         *IN ED (FORTSEMICOMP)

         *LIB ED (SUBGROUPSRF4.SUBROUTINES)

         *WORK ED (FORTWORKFILE)

         ****

PROGRAM FXXX
EXTENDED DATA (22AM)
COMPACT PROGRAM (DBM)

SEGMENTS MISSING
%FDP
CORE        15360
```

FORTRAN COMPILATION BY #XFIV MK 3B   DATE 30/10/79 TIME 11/27/36

```
      LIST(LP)
      PROGRAM(FXXX)
      INPUT 4=MT0 / (FURMAP)
      INPUT 5=CR1
      OUTPUT 6=LP1
      INPUT 1=CR0
      COMPRESS INTEGER AND LOGICAL
      COMPACT PROGRAM
      EXTENDED DATA
      TRACE 0
      END



      MASTER KONTUR
      COMMON/SYMMER/MSYMEL(16)
      COMMON/SYMMEL/NSYMM,RX(16),TX(16),RY(16),TY(16),RZ(16),TZ(16)
      COMMON/ATOMS/NATOMS,NSECT(100),ROW(100),COL(100),LSEC(100)
      COMMON/ATOMID/KOUNT,NAME(100),IATOM(100)
      COMMON/FUSIS/NO,M1I,M1F,M2I,M2F,M3I,M3F,M1D,M2D,M3D
      COMMON/MATRIX/AMAT(31,31)
      COMMON/ORDER/MAXPTS,LINK(1800)
      COMMON/SIZE/NCOL,NX,NY,DX,DY,HDIV
      COMMON/POINTS/NPTS
      COMMON/CONTRL/MATWRT,INTERP,KURV,KREL,ITAPE,NTAPE
      COMMON/RANGEH/HMAXA(30),HMINA(30)
       DIMENSION HVEC(20)
      READ(1,4990) MATWRT,INTERP,KURV,KREL,ITAPE,KLINE,KGRID,NX,NY,DX,DY
 4990 FORMAT(9I0,2F0.0)
C
C  ITAPE=1  FURSUM INPUT
C  ITAPE=2 CORDSTEP INPUT
C      CONTOUR OPTIONS
C  KLINE=0  ALL SOLID LINE
C  KLINE=1  FIRST CONTOUR SOLID PROGRESSING TO MORE OPEN DASHES
C  KLINE=2  POSITIVE CONTOURS SOLID,NEGATIVE OR ZERO CONTOURS DASHED
C       SCALING OPTIONS
C   KREL=0  CONTOURS ON ABSOLUTE VALUES
C  KREL=1 CONTOURS ON HDIV*HMAX
C  KREL=2 CONTOURS ON HDIV*(HMAX - HMIN)
C KGRID = 0 NO ACTION
C  KGRID=1 ... GRID TO BE DRAWN AT INTERVALS OF 0.1 CELL TRANSLATION
C
C
      NTAPE=4
      NZ=1
      READ(1,4995) NHDIV,(HVEC(I),I=1,NHDIV)
 4995 FORMAT(I0,20F0.0)
C
      IF(MATWRT.EQ.0) WRITE(6,4940)
 4940 FORMAT(5X,'MATRIX NOT PRINTED')
      IF(KURV.EQ.0) WRITE(6,4941)
 4941 FORMAT(5X,'CONTOURS NOT SMOOTHED')
      IF(INTERP.EQ.0) WRITE(6,4942)
 4942 FORMAT(5X,'NO INTERPOLATION BETWEEN CELLS')
      WRITE(6,4943) DX,DY
 4943 FORMAT(5X,'CELL SIZE',F10.3,5X,'BY',F10.3,5X,'MM.')
      WRITE(6,5050)NX,NY,DX,DY,NHDIV,(HVEC(I),I=1,NHDIV)
 5050 FORMAT(5X,2I5,2F10.3,I3,20F5.1)
      IF(ITAPE.EQ.0) GO TO 4070
      IF(ITAPE.EQ.2) GO TO 4072
C INTERPRET PHUISIS CARD
      READ(1,4075) NO,M1I,M1F,M2I,M2F,M3I,M3F,M1D,M2D,M3D
 4075 FORMAT(3X,I3,12X,9I3)
C
C  NATOMS.LT.0=FREE FORMAT ATOMS FILE
C  KHAR=0  SYMBOLS ONLY MARKED
C  KHAR=1  SYMBOLS AND ATOM NAMES MARKED
C
      READ(1,4990) NATOMS,NSYMM,KHAR
      IF(NSYMM.GT.0) CALL SYMRED
C -- PUT IN CONDITIONAL JUMP HERE
C
      GO TO (2255,2260,2265,2270,2275,2280),NO
 2255 CALL RANGE(XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX)
      RMIN=XMIN
      RMAX=XMAX
      CMIN=YMIN
      CMAX=YMAX
      GO TO 2285
 2260 CALL RANGE(XMIN,XMAX,ZMIN,ZMAX,YMIN,YMAX)
      RMIN=XMIN
      RMAX=XMAX
      CMIN=ZMIN
      CMAX=XMAX
      GO TO 2285
 2265 CALL RANGE(YMIN,YMAX,XMIN,XMAX,ZMIN,ZMAX)
      RMIN=YMIN
      RMAX=YMAX
      CMIN=XMIN
      CMAX=XMAX
      GO TO 2285
 2270 CALL RANGE(ZMIN,ZMAX,XMIN,XMAX,YMIN,YMAX)
      RMIN=ZMIN
      RMAX=ZMAX
      CMIN=XMIN
      CMAX=XMAX
      GO TO 2285
 2275 CALL RANGE(YMIN,YMAX,ZMIN,ZMAX,XMIN,XMAX)
       RMIN=YMIN
      RMAX=YMAX
      CMIN=ZMIN
      CMAX=ZMAX
      GO TO 2285
 2280 CALL RANGE(ZMIN,ZMAX,YMIN,YMAX,XMIN,XMAX)
      RMIN=ZMIN
      RMAX=ZMAX
      CMIN=YMIN
      CMAX=YMAX
 2285 CONTINUE
      WRITE(6,2286) XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX
 2286 FORMAT(5X,'RANGES'/10X,3(5X,2F6.3))
C
      IF(NATOMS.NE.0) CALL ATREAD(XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX)
      WRITE(6,4078) NHDIV,NX,NY,NZ
 4078 FORMAT(2X,'NHDIV=',I4,2X,'NX=',I4,'NY=',I4,'NZ=',I4)
      NX=(M1F-M1I)/M1D + 1
      NY=(M2F-M2I)/M2D + 1
      NZ=(M3F-M3I)/M3D + 1
      IF(NHDIV.EQ.0) GO TO 4070
      IF(KREL.EQ.0) GO TO 4070
      CALL TSCAN(NZ,HMIN,HMAX)
 4072 CONTINUE
      HMULT=HMAX
      IF(KREL.EQ.2) HMULT=HMULT - HMIN
      DO 4076 II=1,NHDIV
 4076 HVEC(II)=HMULT*HVEC(II)
 4070 CONTINUE
      CALL OPENGINOGP
      SPACX=10.
      SPACY=10.
      HT=DY*FLOAT(NY-1)
      WD=DX*FLOAT(NX-1)
      IXMAX=900.0/(WD + SPACX)
      DO 5110 IJK=1,NZ
      CALL ORIGIN(WD,HT,SPACX,SPACY,IJK,IXMAX)
      IF(NHDIV.EQ.0)GO TO 4967
      IF(ITAPE.EQ.0) GO TO 5014
      READ(NTAPE) ((AMAT(I,J),J=1,NY),I=1,NX)
      IF(MATWRT.EQ.0) GO TO 5013
 5014 CONTINUE
       DO 5010 I=1,NX
      IF(ITAPE.EQ.0) READ(5,5020) (AMAT(I,J),J=1,NY)
      IF(MATWRT.NE.0)WRITE(6,5055) (AMAT(I,J),J=1,NY)
 5010 CONTINUE
 5013 CONTINUE
 5020 FORMAT(10F0.0)
 5055 FORMAT(5X,10F10.5)
C  SCAN TROUGH MATRIX SETTING MAX AND MIN
C     TAPE HAS ALREADY BEEN PRESCANNED
      IF(ITAPE.NE.0) GO TO 4961
      HMAX=-1.E5
      HMIN=1.E5
      DO 4962 II=1,NX
      DO 4962 JJ=1,NY
      H=AMAT(II,JJ)
      IF(H.GT.HMAX)HMAX=H
       IF(H.LT.HMIN)HMIN=H
 4962 CONTINUE
      WRITE(6,4802) HMAX,HMIN
 4802 FORMAT(5X,'MAX VALUE=',E10.3,'MIN VALUE=',E10.3)
      IF(KREL.EQ.0) GO TO 4961
      HMULT=HMAX
      IF(KREL.EQ.2) HMULT=HMULT - HMIN
      DO 4964 IJ=1,NHDIV
 4964 HVEC(IJ)=HMULT*HVEC(IJ)
 4960 CONTINUE
      WRITE(6,4804)(HVEC(II),II=1,NHDIV)
 4804 FORMAT(5X,'CONTOUR LEVELS ADJUSTED TO'/10X,10E10.3/10X,10E10.3)
 4961 CONTINUE
 4967 CONTINUE
      CALL ATMARK(KHAR,IJK)
      CALL RECT(WD,HT,RMIN,RMAX,CMIN,CMAX,KGRID)
      IF(NHDIV.EQ.0) GO TO 5110
       DO 5101 I=1,NHDIV
       HDIV=HVEC(I)
C
C ELIMINATE CONTOURS WHICH ARE TOO HIGH OR TOO LOW
      IF(KREL.NE.0.AND.(HDIV.GT.HMAX.OR.HDIV.LT.HMIN)) GO TO 5101
C
      IF (KLINE.NE.0) CALL LLINE(KLINE,I,NHDIV,HDIV)
      IF(ITAPE.EQ.1.AND.(HDIV.GT.HMAXA(IJK).OR.
     1HDIV.LT.HMINA(IJK))) GO TO 5101
       CALL RASTER
      CALL LSEEK
 5101 CONTINUE
 5110 CONTINUE
      CALL DEVEND
      STOP
      END

END OF SEGMENT, LENGTH   732, NAME  KONTUR

      SUBROUTINE ORIGIN(WD,HT,SPACX,SPACY,II,IXMAX)
      IY=II/IXMAX
      IF(MOD(II,IXMAX).EQ.0) IY=IY-1
      IX=II - IY*IXMAX
      XOR=50.0 + SPACX + (WD + SPACX)*FLOAT(IX-1)
      YOR=SPACY + (HT+SPACY)*FLOAT(IY)
      WRITE(6,1990) XOR,YOR
 1990 FORMAT(5X,'ORIGIN',2F10.3)
      CALL MOVTO2(0.,0.)
      CALL TRANSF(-1)
      CALL TRANSF(2)
      CALL SHIFT2(XOR,YOR)
      RETURN
      END

END OF SEGMENT, LENGTH   103, NAME  ORIGIN

      SUBROUTINE ATREAD(XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX)
      COMMON/ATOMS/NATOMS,NSECT(100),ROW(100),COL(100),LSEC(100)
      COMMON/ATOMID/KOUNT,NAME(100),IATOM(100)
      COMMON/SYMMEL/NSYMM,RX(16),TX(16),RY(16),TY(16),RZ(16),TZ(16)
      COMMON/FUSIS/NO,M1I,M1F,M2I,M2F,M3I,M3F,M1D,M2D,M3D
      COMMON/SIZE/NCOL,NX,NY,DX,DY,HDIV
      LOGICAL FREE,INCELL
      FREE=NATOMS.LT.0
      IF(FREE)NATOMS=-NATOMS
      IF(.NOT.FREE)READ(5,2105) XX,YY
 2105 FORMAT(F9.6/F9.6)
      KOUNT=0
      DO 2100 I=1,NATOMS
      IF(.NOT.FREE)READ(5,2110) NAME(I),Z,XX,YY,ZZ
      IF(FREE) READ(5,2198) NAME(I),NZZ,XX,YY,ZZ
 2198 FORMAT(A4,I0,3F9.0)
      WRITE(6,2111) NAME(I),Z,XX,YY,ZZ
 2111 FORMAT(1X,'ATREAD',A4,4F10.3)
      IF(.NOT.FREE) NZZ=IFIX(Z + .01)
       DO 2220 J=1,NSYMM
      XXX=XX
      YYY=YY
      ZZZ=ZZ
      CALL SYMTRY(XXX,YYY,ZZZ,J)
      CALL MODCEL(XXX)
      CALL MODCEL(YYY)
      CALL MODCEL(ZZZ)
      WRITE(6,2197) I,J,XXX,YYY,ZZZ
 2197 FORMAT(5X,'SYMM',2I5,3F10.3)
      IF(.NOT.INCELL(XXX,YYY,ZZZ,XMIN,XMAX,YMIN,YMAX,
     1ZMIN,ZMAX)) GO TO 2220
      KOUNT=KOUNT + 1
      IATOM(KOUNT)=I
      NSECT(KOUNT)=MIN0(NZZ,8)
 2110 FORMAT(A4,5X,F9.6,9X,3F9.6/)
      GO TO (2115,2120,2125,2130,2135,2140),NO
 2115 CALL MAP(XXX,ROW(KOUNT),YYY,COL(KOUNT),ZZZ,LSEC(KOUNT))
      GO TO 2150
 2120 CALL MAP(XXX,ROW(KOUNT),ZZZ,COL(KOUNT),YYY,LSEC(KOUNT))
      GO TO 2150
 2125 CALL MAP(YYY,ROW(KOUNT),XXX,COL(KOUNT),ZZZ,LSEC(KOUNT))
      GO TO 2150
 2130 CALL MAP(ZZZ,ROW(KOUNT),XXX,COL(KOUNT),YYY,LSEC(KOUNT))
      GO TO 2150
 2135 CALL MAP(YYY,ROW(KOUNT),ZZZ,COL(KOUNT),XXX,LSEC(KOUNT))
      GO TO 2150
 2140 CALL MAP(ZZZ,ROW(KOUNT),YYY,COL(KOUNT),XXX,LSEC(KOUNT))
 2150 CONTINUE
      WRITE(6,2155) KOUNT,NAME(I),XXX,YYY,ZZZ,ROW(KOUNT),COL(KOUNT),
     1LSEC(KOUNT)
 2155 FORMAT(1X,'ATREAD',I5,A4,5F10.3,I10)
 2220 CONTINUE
 2100 CONTINUE
      RETURN
      END

END OF SEGMENT, LENGTH   392, NAME  ATREAD

      SUBROUTINE ATMARK(KHAR,ISEC)
      COMMON/ATOMS/NATOMS,NSECT(100),ROW(100),COL(100),LSEC(100)
      COMMON/ATOMID/KOUNT,NAME(100),IATOM(100)
      COMMON/FUSIS/NO,M1I,M1F,M2I,M2F,M3I,M3F,M1D,M2D,M3D
      COMMON/SIZE/NCOL,NX,NY,DX,DY,HDIV
```

```
KONTUR: COMPILATION LISTING - P 4

                     DIMENSION NAMTMP(1)
                     DO 2200 I=1,KOUNT
                     IF(LSEC(I).NE.ISEC) GO TO 2200
                     IF(ROW(I).GT.NX) GO TO 2200
                     IF(COL(I).GT.NY) GO TO 2200
                     XMAP=ROW(I)*DX
                     YMAP=COL(I)*DY
                     CALL MOVTO2(XMAP,YMAP)
                     IATI=IATOM(I)
                     WRITE(6,2197) I,IATI,NAME(IATI),LSEC(I),NSECT(I),XMAP,YMAP
                2197 FORMAT(1X,'ATMARK',2I5,2X,A4,2I5,2F10.3)
                     CALL SYMBOL(NSECT(I))
                     IF(KHAR.EQ.0) GO TO 2200
                     XMAP=XMAP + 10.0
                     CALL MOVTO2(XMAP,YMAP)
                     NAMTMP(1)=NAME(IATI)
                     CALL CHAARR(NAMTMP,1,4)
                2200 CONTINUE
                     RETURN
                     END

END OF SEGMENT, LENGTH   126, NAME  ATMARK


                     FINISH

END OF COMPILATION -  NO ERRORS


     S/C SUBFILE  :       36 BUCKETS USED

CONSOLIDATED BY XPCK 12H     DATE     30/10/79    TIME 11/31/45

        *SHORTLIST

        *IN ED (FORTSEMICOMP)

        *LIB ED (SUBGROUPSRF4.SUBROUTINES)

        *WORK ED (FORTWORKFILE)

        ****

PROGRAM FXXX
EXTENDED DATA (22AM)
COMPACT PROGRAM (DBM)

SEGMENTS MISSING
DEVEND
LSEEK
RASTER
LLINE
RECT
OPENGINOGP
TSCAN
RANGE
SYMRED
SHIFT2
TRANSF
MOVTO2
INCELL
MAP
MODCEL
SYMTRY
CHAARR
SYMBOL
%FDP

CORE             10560


*DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
*DDDDDDD                                    DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
*DDDDDDD        NUMBER OF PAGES    9        DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
*DDDDDDD                                    DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
*DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
```

CELPIC: COMPILATION LISTING - P 1

FORTRAN COMPILATION BY #XFIV MK 3B   DATE 30/10/79 TIME 11/28/38

```
      LIST(LP)
      PROGRAM (FXXX)
      INPUT 1= CR0
      INPUT 3 = TR0
      INPUT 5 =CR1
      OUTPUT 6 =LP1
      COMPRESS INTEGER AND LOGICAL
      EXTENDED DATA
      TRACE 0
      END
```

```
      MASTER GINMOL
      COMMON/SYMMEL/NSYMM,RX(16),TX(16),RY(16),TY(16),RZ(16),TZ(16)
      COMMON/SYMMER/MSYMEL(16)
        COMMON/ATOMS/NATOMS,X(100),Y(100),Z(100)
        COMMON/RADII/NAME(100),NAT(100),RCOV(100),RVW(100)
        COMMON/CELL/A,B,C,ASINB,ACOSB
      DIMENSION NLINE(30),LVEC(450,2)
      DIMENSION LINE(30,30)
      DIMENSION MBOND(100)
      DIMENSION XX(100),YY(100),ZZ(100)
      DIMENSION MPENX(8)
      DIMENSION MATSYM(20),MATITX(20),MATITY(20),MATITZ(20),MATPER(20)
       EQUIVALENCE (LINE(1,1),LVEC(1,1))
      DIMENSION MAT(100)
      DIMENSION LAXIS(3)
      DATA LAXIS/12H X = Y = Z =/
      DATA MPENX/0,0,1,0,0,-1,0,0/
      LOGICAL BCALC,HCALC,LDASH,KENTRE,KONTAK
      KENTRE=.FALSE.
C
C SET COVALENT RADII -- SOURCE G.I.BROWN: A NEW GUIDE TO MODERN VALENCE
C   THEORY--1972 ED.
C COVALENT RADII GIVEN FOR THE FOLLOWING ATOMCI SPECIES(ATOMIC NOS.)
C H(1)  B(5)  C(6)  N(7)  O(8)  F(9)  SI(14)  P(15)  S(16)  CL(17)
C AS(33)  SE(34)  BR(35)  SB(51)  TE(52)  I(53)
C
      DO 2000 I=1,100
 2000 RCOV(I)=0.
      RCOV(1)=.3
      RCOV(5)=.8
      RCOV(6)=.77
      RCOV(7)=.74
      RCOV(8)=.74
      RCOV(9)=.72
      RCOV(14)=1.117
      RCOV(15)=1.10
      RCOV(16)=1.04
      RCOV(17)=.99
      RCOV(33)=1.21
      RCOV(34)=1.17
      RCOV(35)=1.14
      RCOV(51)=1.41
      RCOV(52)=1.37
      RCOV(53)=1.33
C
C     ENTER PSEUDO-ATOMIC SPECIES FOR UNUSUAL H-BONDING DISTANCES
C    ATOMIC NOS GE 90 FOR H-BOND ACCEPTOS
C
      RCOV(80)=.4
      RCOV(81)=.5
      RCOV(82)=.6
      RCOV(83)=.75
      RCOV(84)=.85
      RCOV(85)=.90
      RCOV(86)=.95
      RCOV(87)=1.00
      RCOV(88)=1.25
      RCOV(89)=1.35
      RCOV(90)=.65
      RCOV(91)=.69
      RCOV(92)=.78
      RCOV(93)=.81
      RCOV(94)=.85
      RCOV(95)=.90
      RCOV(96)=.95
      RCOV(97)=1.05
      RCOV(98)=1.10
      RCOV(99)=1.20
      RCOV(100)=1.25
C
C    SET VANDERVALS RADII -- SOURCE AS ABOVE + CARBON=1.75
C
      RVW(1)=1.0
      RVW(6)=1.75
      RVW(7)=1.50
      RVW(8)=1.40
      RVW(9)=1.35
      RVW(15)=1.90
      RVW(16)=1.85
      RVW(17)=1.80
      RVW(33)=2.00
      RVW(34)=2.00
      RVW(35)=1.95
      RVW(51)=2.20
      RVW(52)=2.20
      RVW(53)=2.15
C
C     PUT IN CHECKING FACILITY FOR ABSENT ELEMENTS
C
C  AUTOMATIC BONDING
C      IF .6(R1+R2)<RIJ<1.2(R1+R2)  ATOMS BONDED
C      IF1.2(R1+R2)<RUJ<2.15(R1+R2) ATOMS H-BONDED
C  WHERE R1 AND R2 ARE THE COVALENT RADII
C
      READ(5,1000) NATOMS,NSYMM,IFORM,NAX
 1000 FORMAT(10X,4I5,9F0.0)
      WRITE(6,1003)NATOMS,NSYMM,IFORM,NAX
 1003 FORMAT(5X,4I5,9F7.3)
      DO 1005 I=1,NSYMM
      READ(5,1010) RX(I),TX(I),RY(I),TY(I),RZ(I),TZ(I),MSYMEL(I)
      WRITE(6,1011)RX(I),TX(I),RY(I),TY(I),RZ(I),TZ(I),MSYMEL(I)
 1011 FORMAT(5X,6F10.4,I10)
 1010 FORMAT(6F10.0,I10)
 1005 CONTINUE
      READ(5,1002)NCELL,A,B,C,ALF,BET,GAM
      WRITE(6,1002)NCELL,A,B,C,ALF,BET,GAM
 1002 FORMAT(10X,I10,6F10.0)
      DTOR=ATAN(1.)/45.
      ALF=DTOR*ALF
      SBET=DTOR*(90. - BET)
      BET=DTOR*BET
      SINBET=SIN(BET)
      COSBET=COS(BET)
      ASINB=A*SINBET
      ACOSB=A*COSBET
      GAM=DTOR*GAM
      TANBET=TAN(SBET)
       CALL OPENGINOGP
 1019 READ(5,1090) KONTIN,IPROJ,AXONX,AXONY,AXONZ,SCAL,XOR,YOR
      IF(IPROJ.EQ.0) GO TO 1019
      CALL TRFORM(IPROJ,AXONX,AXONY,AXONZ,SCAL,XOR,YOR,A,B,C,
     1SINBET,TANBET)
      CALL AXES(0.2,0.2,0.2)
      IF(KONTIN.NE.0) GO TO 1019
      READ(5,1001) NLINES
       WRITE(6,1001)NLINES
 1001 FORMAT(10X,I10,2F10.0)
      HCALC=NLINES.GE.100
       IF(HCALC)WRITE(6,9001)
 9001 FORMAT(2X,'HYDROGEN BONDS TO BE CALCULATED')
      IF(HCALC) NLINES=NLINES-100
      BCALC=NLINES.EQ.0
      IF(BCALC) GO TO 1030
      DO 1015 I=1,NLINES
      READ(5,1020) ITOR,NL,(LINE(I,J),J=1,NL)
       WRITE(6,1020) ITOR,NL,(LINE(I,J),J=1,NL)
 1020 FORMAT(7X,I1,30I2)
      IF(ITOR.NE.2) GO TO 1025
      NL=NL+1
      LINE(I,NL)=LINE(I,1)
 1025 CONTINUE
      NLINE(I)=NL
 1015 CONTINUE
 1030 CONTINUE
C
      IF(NATOMS.EQ.0) GO TO 1350
      IF(IFORM.EQ.0) GO TO 1058
      DO 1055 J=1,NATOMS
      READ(1,1057) NAME(J),NAT(J),X(J),Y(J),Z(J)
 1057 FORMAT(A4,I0,3F0.0)
      MAT(J)=MOD(NAT(J),4)
      IF(MAT(J).EQ.0) MAT(J)=4
      MBOND(J)=0
 1055 CONTINUE
 1058 CONTINUE
C
      IF(.NOT.BCALC) GO TO 1040
      NM=NATOMS-1
      DO 1031 J=1,NM
      NJ=NAT(J)
      RCOVJ=RCOV(NJ)
      JP=J+1
      CALL CHECK(RCOVJ)
      DO 1033 JJ=JP,NATOMS
      NJJ=NAT(JJ)
      RCOVJJ=RCOV(NJJ)
      CALL CHECK(RCOVJJ)
      BMED=RCOVJ+RCOVJJ
      BMIN=.6*BMED
      BMAX=1.2*BMED
      DX=X(J)-X(JJ)
      DY=Y(J)-Y(JJ)
      DZ=Z(J)-Z(JJ)
      DD=DISTM(DX,DY,DZ,B,C,ASINB,ACCSB)
C
      IF(DD.LT.BMIN.OR.DD.GT.BMAX) GO TO 1033
      NLINES=NLINES + 1
      LVEC(NLINES,1)=J
      LVEC(NLINES,2)=JJ
      WRITE(6,1036) NAME(J),NAME(JJ),DD
 1036 FORMAT(5X,A4,'  IS BONDED TO',A4,' BOND DIST=',F9.3)
      MBOND(J)=1
      MBOND(JJ)=1
 1033 CONTINUE
 1031 CONTINUE
      WRITE(6,1039) NLINES
 1039 FORMAT(5X,'BONDS CALCULATED--NO.FOUND=',I5)
 1040 CONTINUE
      NPOINT=0
      DO 1205 J=1,NATOMS
      IF(MBOND(J).NE.0) GO TO 1205
      NPOINT=NPOINT + 1
      NTEMP=NLINES + NPOINT
      LVEC(NTEMP,1)=J
       WRITE(6,1210) NPOINT,NAME(J)
 1210 FORMAT(5X,'ISOLATED ATOM NO.',I5,2X,'NAME',A4)
 1205 CONTINUE
 1350 CONTINUE
       DO 1032 KK=1,NAX
      PENA=0.1/(SCAL*A)
      PENB=0.1/(SCAL*B)
      PENC=0.1/(SCAL*C*SINBET)
       READ(5,1090) NSYMOP,IPROJ,AXONX,AXONY,AXONZ,SCAL,XOR,YOR
      WRITE(6,1091)NSYMOP,IPROJ,AXONX,AXONY,AXONZ,SCAL,XOR,YOR
 1091 FORMAT(2X,'PROJ CARD',2I6,3F8.3,F8.2,2F8.1)
      NMARK=IPROJ/1000
      IPROJ=IPROJ - 1000*NMARK
      NREPT=IPROJ/100
      IPROJ=IPROJ-NREPT
      IBOX=IPROJ/10
      IPROJ=IPROJ - 10*IBOX
      KREPT=NREPT
 1090 FORMAT(10X,2I0,6F0.0)
      CALL TRFORM(IPROJ,AXONX,AXONY,AXONZ,SCAL,XOR,YOR,A,B,C,
     1SINBET,TANBET)
      IBOX=IBOX + 1
      CALL DASHED(0,4.0,4.0,0.0)
      GO TO(1125,1130,1135,1130),IBOX
 1135 CALL DASHED(1,2.0,1.0,0.0)
 1125 CONTINUE
 1120 CONTINUE
      CALL BOX(1.,1.,1.)
 1130 CONTINUE
      IF(NMARK.EQ.0.OR.IBOX.EQ.4) GO TO 1160
      DO 1150 I=1,NMARK
      READ(5,1152) MODE,REPEAT,DASH,DIT,X1,Y1,Z1,X2,Y2,Z2
 1152 FORMAT(3X,I2,3F5.0,6F10.0)
      WRITE(6,1153)MODE,REPEAT,DASH,DIT,X1,Y1,Z1,X2,Y2,Z2
 1153 FORMAT(2X,'MARK CARD',I3,3F6.2,2(2X,3F7.3))
      IF(MODE.LT.10)GO TO 1156
      JAX=MODE/10
      IAX=MODE-10*JAX
      X1=SCAL*X1
      IF(JAX.EQ.1)CALL BAR1(REPEAT,DASH,DIT,X1)
      IF(JAX.EQ.2) CALL TWO(IAX,REPEAT,DASH,DIT,X1)
       IF(JAX.EQ.4) CALL FOUR(IAX,REPEAT,DASH,DIT,X1)
      GO TO 1150
 1156 CONTINUE
C
C  CODE FOR VOID ILLUSTRATION
C
      IF(MODE.LE.3) GO TO 1800
      LAYER=MODE-3
      IF(LAYER.LE.3) GO TO 1720
      WRITE(6,1718) MODE
1718  FORMAT(2X,'MODE=',I4,' NOT ALLOWED')
      GO TO 1150
 1720 CONTINUE
      XLAYER=REPEAT
      WRITE(6,1722) LAXIS(LAYER),XLAYER,PPA
1722  FORMAT(2X,A4,2F7.3)
C LAYER=1,2 OR 3 -- X,Y,OR Z-AXES
      IF(LAYER.EQ.2) GO TO 9210
      WRITE(6,9215)
9215  FORMAT(2X,'PROJECTION ALONG Y-AXIS ONLY PROGRAMMED...')
      GO TO 1150
9210  CONTINUE
      DO 9250 I=1,NATOMS
      IATI=NAT(I)
      RVWI=RVW(IATI)
      R2=RVWI*RVWI
      DO 9260 K=1,NSYMM
      XA=X(I)
      YA=Y(I)
      ZA=Z(I)
      CALL SYMTRY(XA,YA,ZA,K)
      DELTAY=XYZMIN(YA-XLAYER)
      D2=B*B*DELTAY*DELTAY
      IF(D2.GE.R2) GO TO 9660
       RCIRC=2.0*SCAL*SQRT(R2-D2)
      CALL DASET(MAT(I))
       CALL BAR1(XA,YA,ZA,RCIRC)
      WRITE(6,9665) I,K,RVWI,DELTAY,R2,D2,RCIRC
 9665 FORMAT(2X,'CIRCLE',2I5,5F10.3)
      XA=XA + 1.0
      CALL BAR1(XA,YA,ZA,RCIRC)
      ZA=ZA + 1.0
      CALL BAR1(XA,YA,ZA,RCIRC)
      XA=XA - 1.0
      CALL BAR1(XA,YA,ZA,RCIRC)
      CALL DASHED(0,4.0,4.0,0.0)
 9660 CONTINUE
 9260 CONTINUE
 9250 CONTINUE
C
C   END OF VOID ILLUSTRATION
C
 1800 CONTINUE
      CALL DASHED(MODE,REPEAT,DASH,DIT)
        CALL MOVTO3(X1,Y1,Z1)
      CALL LINTO3(X2,Y2,Z2)
      CALL DASHED(0,1.0,0.0,0.0)
 1150 CONTINUE
 1160 CONTINUE
      KREPT=KREPT-1
      IF(KREPT.GE.0) GO TO 1120
      IF(IBOX.EQ.4)CALL STAND(A,B,C,SCAL,NMARK)
C
 2648 CONTINUE
C
      IF(NSYMOP.EQ.0) GO TO 1032
      CALL DASHED(0,4.0,4.0,0.0)
      RPOINT=SQRT(SCAL)
      RPOINT=0.5*SQRT(RPOINT)
      LDASH=.FALSE.
      DO 1035 I=1,NSYMOP
      READ(5,1050)ISYM,ITX,ITY,ITZ,MODE,REPEAT,DASH,DIT,IPER
      WRITE(6,1051) ISYM,ITX,ITY,ITZ,MODE,REPEAT,DASH,DIT,IPER
      IF(I.EQ.1) IPER=0
      ITT=ITX+ITY+ITZ
      IF(ITT.LT.60) GO TO 1432
      IF(KENTRE) GO TO 1434
      XC=0.
      YC=0.
      ZC=0.
      DO 1436 J=1,NATOMS
      XC=XC+X(J)
      YC=YC+Y(J)
      ZC=ZC+Z(J)
 1436 CONTINUE
      TEMP=1.0/FLOAT(NATOMS)
      XC=XC*TEMP
      YC=YC*TEMP
      ZC=ZC*TEMP
      WRITE(6,1435) XC,YC,ZC
 1435 FORMAT(2X,'MOL. CENTRE AT',3F8.3)
      KENTRE=.TRUE.
 1434 CONTINUE
      XCC=XC
      YCC=YC
      ZCC=ZC
      CALL SYMTRY(XCC,YCC,ZCC,ISYM)
      IF(ITX.GE.80)ITX=ITX-90+KSHIFT(XCC)
        IF(ITY.GE.80)ITY=ITY-90+KSHIFT(YCC)
        IF(ITZ.GE.80)ITZ=ITZ-90+KSHIFT(ZCC)
      IPER=0
      WRITE(6,1438) KK,ISYM,ITX,ITY,ITZ
 1438 FORMAT(2X,'MOL. CENTRE SHIFTED INTO UNIT CELL - DRAWING',I3,
     1'  SYMMETRY OPERATION NO.',I3,'  SHIFTS',3I3)
 1432 CONTINUE
      IF(.NOT.HCALC) GO TO 1290
      MATSYM(I)=ISYM
      MATITX(I)=ITX
      MATITY(I)=ITY
      MATITZ(I)=ITZ
      MATPER(I)=IPER
 1290 CONTINUE
 1051 FORMAT(2X,5I2,3F10.3,I5)
 1050 FORMAT(5I2,3F10.0,I5)
      IF(LDASH.AND.(MODE.EQ.0.OR.MODE.EQ.3))CALL DASHED(0,1.,0.,0.)
      LDASH=MODE.NE.0.AND.MODE.NE.3
      IF(LDASH) CALL DASHED(MODE,REPEAT,DASH,DIT)
       DO 1060 J=1,NATOMS
      IF(IPER.NE.0)GO TO 1602
      XXX=X(J)
      YYY=Y(J)
      ZZZ=Z(J)
      GO TO 1604
 1602 XXX=XX(J)
      YYY=YY(J)
      ZZZ=ZZ(J)
 1604 CONTINUE
      CALL SYMTRY(XXX,YYY,ZZZ,ISYM)
      XX(J)=XXX + FLOAT(ITX)
      YY(J)=YYY + FLOAT(ITY)
      ZZ(J)=ZZZ + FLOAT(ITZ)
 1060 CONTINUE
      IF(.NOT.BCALC) GO TO 1079
      DO 1070 J=1,NLINES
      I1=LVEC(J,1)
      I2=LVEC(J,2)
      KREPT=NREPT
 1230 CONTINUE
      CALL MOVTO3(XX(I1),YY(I1),ZZ(I1))
      CALL LINTO3(XX(I2),YY(I2),ZZ(I2))
      IF(MODE.NE.3) GO TO 1320
      DO 1310 K=1,6
      KP1=K+1
      KP2=K+2
      XXI1=XX(I1) + FLOAT(MPENX(K))*PENA
      XXI2=XX(I2) + FLOAT(MPENX(K))*PENA
      YYI1=YY(I1) + FLOAT(MPENX(KP1))*PENB
      YYI2=YY(I2) + FLOAT(MPENX(KP1))*PENB
      ZZI1=ZZ(I1) + FLOAT(MPENX(KP2))*PENC
      ZZI2=ZZ(I2) + FLOAT(MPENX(KP2))*PENC
      CALL MOVTO3(XXI1,YYI1,ZZI1)
      CALL LINTO3(XXI2,YYI2,ZZI2)
 1310 CONTINUE
 1320 CONTINUE
C
      KREPT=KREPT-1
      IF(KREPT.GE.0) GO TO 1230
 1070 CONTINUE
      IF(NPOINT.EQ.0) GO TO 1035
      DO 1222 J=1,NPOINT
      LTEMP=NLINES + J
       NTEMP=LVEC(LTEMP,1)
      CALL MOVTO3(XX(NTEMP),YY(NTEMP),ZZ(NTEMP))
      CALL DOT(RPOINT)
 1222 CONTINUE
       GO TO 1035
 1079 CONTINUE
      DO 1080 J=1,NLINES
       NL=NLINE(J)
      I1=LINE(J,1)
       CALL MOVTO3(XX(I1),YY(I1),ZZ(I1))
      IF(NL.NE.1) GO TO 1062
      CALL SYMBOL(8)
      GO TO 1080
 1062 CONTINUE
      DO 1085 K=2,NL
       I2=LINE(J,K)
      CALL LINTO3(XX(I2),YY(I2),ZZ(I2))
      I1=I2
 1085 CONTINUE
 1080 CONTINUE
      IF(LDASH) CALL DASHED(0,1.0,1.0,0.0)
 1035 CONTINUE
      IF(.NOT.HCALC) GO TO 1032
      NHCALC=0
       CALL DASHED(1,2.0,1.0,0.)
      RCOV1=RCOV(1)
      DO 1260 J=1,NSYMOP
      DO 1270 L=1,NATOMS
      NJ=NAT(L)
      IF(NJ.NE.1) GO TO 1270
      IF(MATPER(J).NE.0) GO TO 1640
      XXX=X(L)
      YYY=Y(L)
      ZZZ=Z(L)
      GO TO 1650
 1640 CONTINUE
      XXX=XX(L)
      YYY=YY(L)
      ZZZ=ZZ(L)
 1650 CONTINUE
       CALL SYMTRY(XXX,YYY,ZZZ,MATSYM(J))
      XX(L)=XXX+FLOAT(MATITX(J))
      YY(L)=YYY+FLOAT(MATITY(J))
      ZZ(L)=ZZZ+FLOAT(MATITZ(J))
      DO 1262 K=1,NSYMOP
      DO 1272 LL=1,NATOMS
      NJJ=NAT(LL)
      IF(NJJ.LE.6) GO TO 1272
      IF(NJJ.LE.9) GO TO 1273
      IF (NJJ.EQ.17) GO TO 1273
      IF(NJJ.EQ.35) GO TO 1273
      IF(NJJ.EQ.53) GO TO 1273
C     TEST FOR PSEUDO ATOMIC SPECIES
      IF(NJJ.GE.90) GO TO 1273
      GO TO 1272
 1273 CONTINUE
      IF(MATPER(K).NE.0) GO TO 1670
      XXXX=X(LL)
      YYYY=Y(LL)
      ZZZZ=Z(LL)
      GO TO 1660
 1670 XXXX=XX(LL)
      YYYY=YY(LL)
      ZZZZ=ZZ(LL)
 1660 CONTINUE
      CALL SYMTRY(XXXX,YYYY,ZZZZ,MATSYM(K))
      XX(LL)=XXXX+FLOAT(MATITX(K))
      YY(LL)=YYYY+FLOAT(MATITY(K))
      ZZ(LL)=ZZZZ+FLOAT(MATITZ(K))
      DX=XX(LL)-XX(L)
      DY=YY(LL)-YY(L)
      DZ=ZZ(LL)-ZZ(L)
      DD=DISTM(DX,DY,DZ,B,C,ASINB,ACOSB)
      BMED=RCOV1+RCOV(NJJ)
      BMAX=2.15*BMED
      IF(DD.GT.BMAX) GO TO 1272
      BMIN=1.2*BMED
      IF(DD.LT.BMIN) GO TO 1272
      CALL MOVTO3(XX(L),YY(L),ZZ(L))
      CALL LINTO3(XX(LL),YY(LL),ZZ(LL))
      WRITE(6,1275) NAME(L),MATSYM(J),MATITX(J),MATITY(J),
     1MATITZ(J),NAME(LL),MATSYM(K),MATITX(K),MATITY(K),MATITZ(K),DD
 1275 FORMAT(2X,'HBOND FROM',2X,A4,2X,4I2,2X,'TO',2X,A4,2X,4I2,F6.3)
      NHCALC=NHCALC+1
 1272 CONTINUE
 1262 CONTINUE
 1270 CONTINUE
 1260 CONTINUE
      WRITE(6,9003)NHCALC
 9003 FORMAT(2X,'HBONDS CALCULATED--NO. FOUND=',I5)
 1032 CONTINUE
      CALL DEVEND
      STOP
      END
```

```
END OF SEGMENT, LENGTH  2211, NAME  GINMOL
```

```
      FUNCTION KSHIFT(XYZ)
      KSHIFT=0
      IF(XYZ.LT.0.0) GO TO 1440
       IF(XYZ.LE.1.0) RETURN
      KSHIFT=-IFIX(XYZ)
      RETURN
 1440 KSHIFT=IFIX(-XYZ + 1.0)
      RETURN
      END
```

```
END OF SEGMENT, LENGTH    45, NAME  KSHIFT
```

```
      SUBROUTINE STAND(A,B,C,SCAL,NO)
      WRITE(6,2098) NO
 2098 FORMAT(2X,'STANDARD CELL NO.',I4)
      GO TO (2100,2200,2300),NO
 2100 CONTINUE
C P21/C VIEWED ALONG B
      YY=0.
      HT=1.
      DZ=1.2*HT/C
      DX=1.2*HT/A
      DO 1510 I=1,2
      X1=FLOAT(I-1)
      DO 1520 J=1,2
      TEMP=0.5*FLOAT(J-1)
      Z1=0.5*DZ + TEMP
      Z2=0.25 - DZ + TEMP
      Z3=0.25 + DZ + TEMP
      Z4=0.5 - 0.5*DZ + TEMP
      WRITE(6,3000) X1,Z1,Z2,Z3,Z4
 3000 FORMAT(2X,'A',5F10.3)
      CALL MOVTO3(X1,YY,Z1)
      CALL LINTO3(X1,YY,Z2)
      CALL MOVTO3(X1,YY,Z3)
      CALL LINTO3(X1,YY,Z4)
 1520 CONTINUE
 1510 CONTINUE
      DO 1530 I=1,2
      X1=FLOAT(I-1)
      DO 1540 J=1,2
      TEMP=0.5*FLOAT(J-1)
      Z1=DX + TEMP
      Z2=0.5 - DX + TEMP
      CALL MOVTO3(Z1,YY,X1)
      CALL LINTO3(Z2,YY,X1)
      WRITE(6,3002) Z1,Z2,X1
 3002 FORMAT(2X,'B',3F10.3)
 1540 CONTINUE
 1530 CONTINUE
      HTT=0.5*HT
      DO 1550 I=1,3
      X1=0.5*FLOAT(I-1)
      DO 1560 J=1,3
```

```
                Z1=0.5*FLOAT(J-1)
                WRITE(6,3004) X1,YY,Z1
          3004 FORMAT(2X,'BAR1',3F10.3)
                HT=0.25*SCAL
                CALL BAR1(X1,YY,Z1,HT)
                IF(J.EQ.3) GO TO 1560
                Z1=Z1 + 0.25
                WRITE(6,3006) X1,YY,Z1
          3006 FORMAT(2X,'TWO',3F10.3)
                HT=0.5*SCAL
                CALL TWO(1,X1,YY,Z1,HT)
          1560 CONTINUE
          1550 CONTINUE
          2200 CONTINUE
          2300 CONTINUE
                RETURN
                END

END OF SEGMENT, LENGTH   263, NAME  STAND

                FUNCTION XYZMIN(XXX)
                IF(XXX.LT.0.5) GO TO 2550
                XYZMIN=XXX-IFIX(XXX+0.5)
                RETURN
          2550 CONTINUE
                IF(XXX.GT.-0.5) GO TO 2560
                XYZMIN=XXX + IFIX(-XXX + 0.5)
                RETURN
          2560 CONTINUE
                XYZMIN=XXX
                RETURN
                 END

END OF SEGMENT, LENGTH    56, NAME  XYZMIN

                SUBROUTINE DASET(NATJ)
                GO TO (4010,4020,4030,4040),NATJ
          4010 CONTINUE
                CALL DASHED(0,4.0,4.0,0.0)
                RETURN
          4020 CONTINUE
                CALL DASHED(1,4.0,3.0,0.0)
                RETURN
          4030 CONTINUE
                CALL DASHED(2,4.0,2.4,0.2)
                RETURN
          4040 CONTINUE
                CALL DASHED(1,2.0,1.25,0.0)
                RETURN
                END

END OF SEGMENT, LENGTH    59, NAME  DASET

                FINISH

END OF COMPILATION -  NO ERRORS

     S/C SUBFILE  :      57 BUCKETS USED

  FIRST WORKFILE  :      55 BUCKETS USED

 SECOND WORKFILE  :      67 BUCKETS USED

CONSOLIDATED BY XPCK 12H   DATE    30/10/79   TIME 11/33/55

        *SHORTLIST

        *IN ED (FORTSEMICOMP)

        *LIB ED (SUBGROUPSRF4.SUBROUTINES)

        *WORK ED (FORTWORKFILE)

        ****

PROGRAM FXXX
EXTENDED DATA (22AM)
COMPACT PROGRAM (DBM)

SEGMENTS MISSING
DISTM
DEVEND
SYMBOL
DOT
LINTO3
MOVTO3
SYMTRY
FOUR, etc.
CORE        10560
```

FORTRAN COMPILATION BY #XFIV MK 3B   DATE 30/10/79 TIME 10/45/10

```
      LIST (LP)
      PROGRAM (FXXX)
      INPUT  1 = CR0
      INPUT  3 = TR0
      INPUT  5 = CR1
      OUTPUT 2 = LP0/132
      OUTPUT 6 = LP1/132
      COMPRESS INTEGER AND LOGICAL
      COMPACT PROGRAM
      EXTENDED DATA
      TRACE 2
      END
```

```
      TRACE 1
       MASTER ABSORB
      DIMENSION ITLE(15)
      DIMENSION AMUST(100),AM(100),AMF(8),IZ(8),NZ(8)
      DTOR=ATAN(1.)/45.
      DO 1000 I=1,100
      AMUST(I)=0.0
1000  AM(I)=0.0
C
C   ENTER ATOMIC MASSES
C
      AM(1)=1.008
      AM(6)=12.011
      AM(7)=14.007
      AM(8)=15.999
      AM(9)=18.998
      AM(15)=30.974
      AM(16)=32.064
      AM(17)=35.453
      AM(35)=79.904
      AM(53)=126.904
1080  CONTINUE
      READ(5,1100) IRAD,NZCELL,(ITLE(I),I=1,15)
1100  FORMAT(10X,2I5,15A4)
      IF(IRAD.EQ.0) GO TO 1090
      WRITE(6,1004) (ITLE(I),I=1,15)
1004  FORMAT(/10X,15A4)
      GO TO (2100,2110), IRAD
2100  CONTINUE
      WRITE(6,2104) NZCELL
2104  FORMAT(2X,'NO. FORMULA UNITS PER CELL=',I4,'  COPPER K-ALPHA')
      AMUST(1)=.435
      AMUST(6)=4.60
      AMUST(7)=7.52
      AMUST(8)=11.5
      AMUST(15)=74.1
      AMUST(16)=89.1
      AMUST(17)=106.0
      AMUST(35)=99.6
      AMUST(53)=294.0
      GO TO 2180
2110  CONTINUE
C
      WRITE(6,2112) NZCELL
2112  FORMAT(2X,'NO. FORMULA UNITS PER CELL=',I4,'  MOLY K-ALPHA')
      AMUST(1)=0.380
      AMUST(6)=0.625
      AMUST(7)=0.916
      AMUST(8)=1.31
      AMUST(9)=1.80
      AMUST(15)=7.89
      AMUST(16)=9.55
       AMUST(17)=11.4
      AMUST(35)=79.8
      AMUST(53)=37.1
       GO TO 2180
C
C   THIS SPACE RESERVED FOR MASS ABSORBSION COEFFICIENTS
C   FOR OTHER RADIATIONS
C
 2180 CONTINUE
      READ(5,1105) ICELL,A,B,C,ALFD,BETD,GAMD
 1105 FORMAT(10X,I10,6F10.0)
      IF(ICELL.NE.1) GO TO 1120
      VCELL=A*B*C
      GO TO 1180
 1120 CONTINUE
      IF (ICELL.NE.2) GO TO 1130
      BET=DTOR*BETD
      VCELL=A*B*C*SIN(BET)
      GO TO 1180
 1130 CONTINUE
      COSA=COS(DTOR*ALFD)
      COSB=COS(DTOR*BETD)
```

```
ABSORB: COMPILATION LISTING - P 2

                       COSG=COS(DTOR*GAMD)
                       TEMP=1.+2.*COSA*COSB*COSG-COSA*COSA-COSB*COSB-COSG*COSG
                       VCELL=A*B*C*SQRT(TEMP)
                 1180  CONTINUE
                       WRITE(6,1185) A,B,C,ALFD,BETD,GAMD,VCELL
                 1185  FORMAT(2X,'A=',F7.3,' B=',F7.3,' C=',F7.3,' ALF=',F6.1,' BET=',
                     1 F6.1,' GAM=',F6.1,' VCELL=', F8.1)
                       READ(5,1110)(IZ(I),NZ(I),I=1,8)
                 1110  FORMAT(16I5)
                       NEL=0
                       DO 1106 I=1,8
                       IF(IZ(I).EQ.0) GO TO 1107
                       NEL=NEL+1
                 1106  CONTINUE
                 1107  CONTINUE
                       WRITE(6,1108) NEL
                 1108  FORMAT(2X,'NO. OF ELEMENTS=',I4)
                C
                       AMWT=0.0
                       DO 2200 I=1,NEL
                       IZI=IZ(I)
                       AMIZI=AM(IZI)
                       IF(AMIZI.GT.0.00001) GO TO 1210
                       WRITE(6,1212) IZI
                 1212  FORMAT(2X,'ATOMIC MASS DATA FOR ELEMENT',I5,' NOT AVAILABLE')
                 1210  CONTINUE
                       AMF(I)=AMIZI*FLOAT(NZ(I))
                       AMWT=AMWT+AMF(I)
                 2200  CONTINUE
                       DO 2250 I=1,NEL
                       AMF(I)=AMF(I)/AMWT
                       IZI=IZ(I)
                       WRITE(6,2252) IZ(I),NZ(I),AMF(I),AMUST(IZI)
                 2252  FORMAT(2X,'ELEMENT',I4,' NO ATOMS', I4, ' MASS FRACTION',F7.4,
                     1 ' AT MASS ABS COEFF=',F8.4)
                 2250  CONTINUE
                       DX=AMWT*FLOAT(NZCELL)*1.66/VCELL
                       WRITE(6,2204) AMWT,DX
                 2204  FORMAT(2X,'MOL. WT.=',F12.5,' XRAY DENSITY=', F7.3)
                       SMUST=0.0
                       DO 2300 I=1,NEL
                       IZI=IZ(I)
                       AMUSTI=AMUST(IZI)
                       IF(AMUSTI.GT.0.000001) GO TO 2310
                       WRITE(6,2312) IZI
                 2312  FORMAT(2X,'ATOMIC MASS ABSORBSION DATA FOR ELEMENT',I5,
                     1 ' UNAVAILABLE')
                 2310  CONTINUE
                       SMUST=SMUST + AMUSTI*AMF(I)
                 2300  CONTINUE
                       SMU=DX*SMUST
                       WRITE(6,2304) SMUST,SMU
                 2304  FORMAT(2X,'MASS ABSORBSION COEFF.=',F8.3,' LINEAR ABS. COEFF.=',
                     1 F8.3)
                       GO TO 1080
                 1090  CONTINUE
                       STOP
                       END

END OF SEGMENT, LENGTH   468, NAME  ABSORB


                       FINISH

END OF COMPILATION -  NO ERRORS


    S/C SUBFILE  :     15 BUCKETS USED

CONSOLIDATED BY XPCK 12H    DATE    30/10/79   TIME 10/47/54

        *SHORTLIST

        *IN ED (FORTSEMICOMP)

        *LIB ED (SUBGROUPSRF4.SUBROUTINES)

        *WORK ED (FORTWORKFILE)

        ****

PROGRAM FXXX
EXTENDED DATA (22AM)
COMPACT PROGRAM (DBM)
CORE                 5888


*FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
*FFFFFFF                                   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
*FFFFFFF          NUMBER OF PAGES    5     FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
*FFFFFFF                                   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
*FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

ELIB1: COMPILATION LISTING - P 1

FORTRAN COMPILATION BY #XFIV MK 3B   DATE 30/10/79 TIME 12/32/46

```
            LIST (LP)
            PROGRAM (FXXX)
            INPUT  1 = CR0
            INPUT  3 = TR0
            INPUT  5 = CR1
            OUTPUT 2 = LP0/132
            OUTPUT 6 = LP1/132
            COMPRESS INTEGER AND LOGICAL
            COMPACT PROGRAM
            EXTENDED DATA
            TRACE 2
            END



            TRACE 1
            MASTER ELIB1
            END
```

END OF SEGMENT, LENGTH   4, NAME ELIB1

```
            SUBROUTINE RASTER
            COMMON/MATRIX/AMAT(31,31)
            COMMON/ORDER/MAXPTS,LINK(1800)
            COMMON/SIZE/NCOL,NX,NY,DX,DY,HDIV
            HDV=HDIV
            NCOL=2*NY
            MAXPTS=NX*NCOL
            DO 1010 I=1,MAXPTS
       1010 LINK(I)=-100
            NXM=NX-1
            NYM=NY-1
            DO 1022 I=1,NXM
            I1=ISIZE(AMAT(I,1),HDV)
            IP=I+1
            I4=ISIZE(AMAT(IP,1),HDV)
            DO 1022 J=1,NYM
            JP=J+1
            I2=ISIZE(AMAT(I,JP),HDV)
            I3=ISIZE(AMAT(IP,JP),HDV)
            KASE=8*I1 + 4*I2 + 2*I3 + I4
            IF(KASE.EQ.0) GO TO 1020
            K1=(I-1)*NCOL +2*J -1
            K2=K1 + 1
            K3=K1 + 2
            K4=K2 + NCOL
            GO TO(1110,1120,1130,1140,1150,1160,1170,1180,1190,1200,1210,1220,
           11230,1240,1020),KASE
       1110 LINK(K4)=K1
            GO TO 1020
       1120 LINK(K3)=K4
            GO TO 1020
       1130 LINK(K3)=K1
            GO TO 1020
       1140 LINK(K2)=K3
            GO TO 1020
       1150 LINK(K2)=K3
            LINK(K4)=K1
            GO TO 1020
       1160 LINK(K2)=K4
            GO TO 1020
       1170 LINK(K2)=K1
            GO TO 1020
       1180 LINK(K1)=K2
            GO TO 1020
       1190 LINK(K4)=K2
            GO TO 1020
       1200 LINK(K1)=K2
            LINK(K3)=K4
            GO TO 1020
       1210 LINK(K3)=K2
            GO TO 1020
       1220 LINK(K1)=K3
            GO TO 1020
       1230 LINK(K4)=K3
            GO TO 1020
       1240 LINK(K1)=K4
       1020 CONTINUE
            I1=I2
            I4=I3
       1022 CONTINUE
            RETURN
            END
```

END OF SEGMENT, LENGTH   251, NAME RASTER

```
      FUNCTION ISIZE(A,B)
      IF(A-B) 1310,1320,1320
 1310 ISIZE=0
      RETURN
 1320 ISIZE=1
      RETURN
      END

END OF SEGMENT, LENGTH    32, NAME  ISIZE

      SUBROUTINE LSEEK
      COMMON/SIZE/NCOL,NX,NY,DX,DY,HDIV
      COMMON/ORDER/MAXPTS,LINK(1800)
C
C  SEARCH AROUND EDGES
C
      DO 1702 L=2,NCOL,2
      IF(LINK(L).GT.0) CALL KOOP(L)
 1702 CONTINUE
      DO 1704 L=1,MAXPTS,NCOL
      IF(LINK(L).GT.0) CALL KOOP(L)
 1704 CONTINUE
       NM=NCOL-1
      DO 1706 L=NM,MAXPTS,NCOL
      IF(LINK(L).GT.0) CALL KOOP(L)
 1706 CONTINUE
       NM=MAXPTS-NCOL
      DO 1708 L=NM,MAXPTS
      IF(LINK(L).GT.0) CALL KOOP(L)
 1708 CONTINUE
      DO 1710 L=1,MAXPTS
      IF(LINK(L).LE.0) GO TO 1710
       CALL KOOP(L)
 1710 CONTINUE
       RETURN
      END

END OF SEGMENT, LENGTH    99, NAME  LSEEK

      SUBROUTINE KOOP(KK)
      COMMON/ORDER/MAXPTS,LINK(1800)
      COMMON/CONTRL/MATWRT,INTERP,KURV,KREL,ITAPE,NTAPE
      COMMON/GFCURV/COSBEG,SINBEG,COSFIN,SINFIN,XBEGEX,YBEGEX,
     1XFINEX,YFINEX
      DIMENSION LINE(300),XL(300),YL(300)
      IF(KURV.NE.0) GO TO 1828
      CALL LOOP(KK)
      RETURN
 1828 CONTINUE
      LPT=KK
      NPT=0
      KBEG=KK
      L1=LTYP(KK)
 1830 CONTINUE
      NPT=NPT + 1
      IF(NPT.GE.300) CALL MESAGE(1)
      LINE(NPT)=LPT
      NEW=LINK(LPT)
       LINK(LPT)=0
      KFIN=LPT
      LPT=NEW
      IF(LPT.GT.0) GO TO 1830
      IF(L1.NE.0) GO TO 1840
      NPT=NPT + 1
      LINE(NPT)=LINE(1)
 1840 CONTINUE
      DO 1860 K=1,NPT
      LL=LINE(K)
      CALL LOCATE(LL,XX,YY)
      XL(K)=XX
 1860 YL(K)=YY
      IF(L1) 1870,1870,1880
 1870 LST=NPT - 1
      IST=2
      XBEGEX=XL(LST)
      YBEGEX=YL(LST)
      XFINEX=XL(IST)
      YFINEX=YL(IST)
      CALL CURTO2(XL,YL,NPT,-1,-1)
      RETURN
 1880 L2=LTYP(KFIN)
      CALL START(COSBEG,SINBEG,L1,1)
      CALL START(COSFIN,SINFIN,L2,2)
      CALL CURTO2(XL,YL,NPT,1,1)
      RETURN
      END

END OF SEGMENT, LENGTH   161, NAME  KOOP
```

```
             SUBROUTINE LOOP(KK)
            COMMON/ORDER/MAXPTS,LINK(1800)
            LPT=KK
            NEW=-1
       1730 CONTINUE
            CALL LOCATE(LPT,XX,YY)
            IF(NEW)1732,1733,1733
       1732 CONTINUE
             CALL MOVTO2(XX,YY)
            GO TO 1734
       1733 CONTINUE
             CALL LINTO2(XX,YY)
       1734 CONTINUE
            NEW=LINK(LPT)
             LINK(LPT)=0
            LPT=NEW
            IF(NEW.GT.0) GO TO 1730
            RETURN
            END

END OF SEGMENT, LENGTH    43, NAME  LOOP

             SUBROUTINE RECT(WD,HT,RMIN,RMAX,CMIN,CMAX,KGRID)
            CALL MOVTO2(0.,0.)
            CALL LINTO2(WD,0.)
            CALL LINTO2(WD,HT)
            CALL LINTO2(0.,HT)
            CALL LINTO2(0.,0.)
            IF(KGRID.EQ.0)RETURN
            SX=.1*WD/(RMAX-RMIN)
            SY=.1*HT/(CMAX-CMIN)
            REPT=5.0
            DASH=1.0
            DOT=0.5
            CALL DASHED(-2,REPT,DASH,DOT)
            DO 2005 I=1,10
            SSX=SX*FLOAT(I)
            IF(SSX.GE.WD) GO TO 2010
            CALL MOVTO2(SSX,0.)
            CALL LINTO2(SSX,HT)
       2005 CONTINUE
       2010 CONTINUE
            DO 2015 I=1,10
            SSY=SY*FLOAT(I)
            IF(SSY.GE.HT) GO TO 2020
            CALL MOVTO2(0.,SSY)
            CALL LINTO2(WD,SSY)
       2015 CONTINUE
       2020 CONTINUE
            CALL DASHED(0,REPT,DASH,DOT)
            RETURN
            END

END OF SEGMENT, LENGTH   145, NAME  RECT

            SUBROUTINE LOCATE(KK,XX,YY)
            COMMON/SIZE/NCOL,NX,NY,DX,DY,HDIV
            COMMON/MATRIX/AMAT(31,31)
            COMMON/CONTRL/MATWRT,INTERP,KURV,KREL,ITAPE,NTAPE
            KROW=KK/NCOL
            IF(MOD(KK,NCOL).EQ.0) KROW=KROW-1
            KOL=KK-KROW*NCOL
            KTYP=MOD(KOL,2)
            KKROW=KROW + 1
            KKOL=(KOL + 1)/2
            KROW2=KKROW + KTYP
            KOL2=KKOL + 1 - KTYP
            YY=0.5*DY*FLOAT(KOL-1)
            KROW=2*KROW + MOD(KOL,2)
            XX=0.5*DX*FLOAT(KROW)
            IF(INTERP.EQ.0) RETURN
            IF(KKROW.EQ.0.OR.KKOL.EQ.0) RETURN
            H1=AMAT(KKROW,KKOL)
            H2=AMAT(KROW2,KOL2)
            IF(KTYP)1670,1670,1680
       1670 YY=YY + DY*ANTERP(H1,H2)
            RETURN
       1680 XX=XX + DX*ANTERP(H1,H2)
            RETURN
            END

END OF SEGMENT, LENGTH   143, NAME  LOCATE

            FUNCTION ANTERP(H1,H2)
            COMMON/SIZE/NCOL,NX,NY,DX,DY,HDIV
            DIFF=H2 - H1
            IF(ABS(DIFF) - 1.E-5) 1750,1750,1760
       1750 ANTERP=0.0
```

```
      RETURN
 1760 ANTERP=(HDIV-H1)/DIFF - 0.5
      RETURN
      END

END OF SEGMENT, LENGTH    44, NAME  ANTERP

      FUNCTION LTYP(KK)
      COMMON/SIZE/NCOL,NX,NY,DX,DY,HDIV
      LTYP=0
      MK=MOD(KK,NCOL)
      IF(MK.NE.1) GO TO 1920
      LTYP=2
      RETURN
 1920 NM=NCOL-1
      IF(MK.NE.NM) GO TO 1930
      LTYP=3
      RETURN
 1930 CONTINUE
      IF(KK.GE.NCOL) GO TO 1940
      IF(MOD(KK,2).NE.0)RETURN
      LTYP=1
      RETURN
 1940 NM=NCOL*(NX-1)
      IF(KK.LE.NM) RETURN
      LTYP=4
      RETURN
      END

END OF SEGMENT, LENGTH    96, NAME  LTYP

      SUBROUTINE START(CS,SN,LT,IS)
      GO TO (1945,1950,1955,1960),LT
 1945 SN=0.0
      CS=1.0
      IF(IS.EQ.2) CS=-CS
      RETURN
 1950 SN=1.0
      CS=0.0
      IF(IS.EQ.2) SN=-SN
      RETURN
 1955 SN=-1.0
      CS=0.0
      IF(IS.EQ.2) SN=-SN
      RETURN
 1960 SN=0.0
      CS=-1.0
      IF(IS.EQ.2) CS=-CS
      RETURN
      END

END OF SEGMENT, LENGTH    90, NAME  START

         SUBROUTINE SYMRED
      COMMON/SYMMER/MSYMEL(16)
      COMMON/SYMMEL/NSYMM,RX(16),TX(16),RY(16),TY(16),RZ(16),TZ(16)
      DO 7001 J=1,NSYMM
      READ(1,7002) RX(J),TX(J),RY(J),TY(J),RZ(J),TZ(J),MSYMEL(J)
 7002 FORMAT(6F10.0,I10)
      WRITE(6,7002) RX(J),TX(J),RY(J),TY(J),RZ(J),TZ(J),MSYMEL(J)
 7001 CONTINUE
      RETURN
      END

END OF SEGMENT, LENGTH    83, NAME  SYMRED

      SUBROUTINE LLINE(KLINE,IDIV,NHDIV,HDIV)
      REPT=4.0
      GO TO (1980,1990),KLINE
 1980 CONTINUE
      IF(IDIV.GE.1) GO TO 1982
      CALL DASHED(0,0.0,0.0,0.0)
      RETURN
 1982 CONTINUE
      F=FLOAT(NHDIV - IDIV +1)/ FLOAT(NHDIV)
      FR=F*REPT
      DASH=FR*F
      DOT=FR*(1.0 - F)
      CALL DASHED(-2,REPT,DASH,DOT)
      RETURN
 1990 CONTINUE
      IF(HDIV) 1992,1992,1994
 1994 CALL DASHED(0,0.0,0.0,0.0)
      RETURN
 1992 DASH=0.7*REPT
      CALL DASHED(-1,REPT,DASH,0.0)
      RETURN
      END

END OF SEGMENT, LENGTH    93, NAME  LLINE

      SUBROUTINE TSCAN(NZ,HMIN,HMAX)
      COMMON/MATRIX/AMAT(31,31)
      COMMON/SIZE/NCOL,NX,NY,DX,DY,HDIV
      COMMON/CONTRL/MATWRT,INTERP,KURV,KREL,ITAPE,NTAPE
      COMMON/RANGEH/HMAXA(30),HMINA(30)
      HMIN=1.E10
      HMAX=-1.E10
      DO 1350 IZ=1,NZ
      HMAXAT=-1.E10
      HMINAT=1.E10
      READ(NTAPE) ((AMAT(I,J),J=1,NY),I=1,NX)
      DO 1360 I=1,NX
      DO 1360 J=1,NY
      H=AMAT(I,J)
      IF(H.GT.HMAXAT) HMAXAT=H
      IF(H.LT.HMINAT) HMINAT=H
 1360 CONTINUE
      IF(HMINAT.LT.HMIN)HMIN=HMINAT
      IF(HMAXAT.GT.HMAX)HMAX=HMAXAT
      HMAXA(IZ)=HMAXAT
      HMINA(IZ)=HMINAT
 1350 CONTINUE
      REWIND NTAPE
      RETURN
      END

END OF SEGMENT, LENGTH   136, NAME  TSCAN

      SUBROUTINE RANGE(RMIN,RMAX,CMIN,CMAX,SMIN,SMAX)
       COMMON/FUSIS/NO,M1I,M1F,M2I,M2F,M3I,M3F,M1D,M2D,M3D
       R=1.0/120.0
       RMIN=R*FLOAT(M1I)
      RMAX=R*FLOAT(M1F)
      CMIN=R*FLOAT(M2I)
      CMAX=R*FLOAT(M2F)
      SMIN=R*FLOAT(M3I)
      SMAX=R*FLOAT(M3F)
      RETURN
      END

END OF SEGMENT, LENGTH    70, NAME  RANGE

      LOGICAL FUNCTION INCELL(XXX,YYY,ZZZ,XMIN,XMAX,
     1YMIN,YMAX,ZMIN,ZMAX)
      INCELL=.FALSE.
      IF(XXX.GT.XMAX)RETURN
      IF(XXX.LT.XMIN)RETURN
      IF(YYY.GT.YMAX)RETURN
      IF(YYY.LT.YMIN)RETURN
      IF(ZZZ.GT.ZMAX)RETURN
      IF(ZZZ.LT.ZMIN)RETURN
      INCELL=.TRUE.
      RETURN
      END

END OF SEGMENT, LENGTH   105,•NAME  INCELL

      SUBROUTINE MODCEL(XXX)
      IF(XXX.LT.0.0) XXX=XXX + 1.0
      IF(XXX.GT.1.0) XXX=XXX - 1.0
      RETURN
      END

END OF SEGMENT, LENGTH    26, NAME  MODCEL

       SUBROUTINE MAP(XI,RF,YI,CF,ZI,LSF)
      COMMON/ATOMS/NATOMS,NSECT(100),ROW(100),COL(100),LSEC(100)
      COMMON/ATOMID/KOUNT,NAME(100),IATOM(100)
      COMMON/FUSIS/NO,M1I,M1F,M2I,M2F,M3I,M3F,M1D,M2D,M3D
      COMMON/SIZE/NCOL,NX,NY,DX,DY,HDIV
COMMONS
      RF=COORD(XI,M1I,M1D)
      CF=COORD(YI,M2I,M2D)
      LSF=IFIX(COORD(ZI,M3I,M3D) + 1.5)
C
      RETURN
      END

END OF SEGMENT, LENGTH    62, NAME  MAP
```

```
ELIB1: COMPILATION LISTING - P 6

 0364                FUNCTION COORD(XX,MI,MD)
 0365                COORD=(120.0*XX - FLOAT(MI))/FLOAT(MD)
 0366                RETURN
 0367                END

END OF SEGMENT, LENGTH    36, NAME  COORD

 0368                SUBROUTINE MESAGE(N)
 0369                WRITE(6,1900)
 0370           1900 FORMAT(5X,'ARRAYS XL,YL AND LINE WILL PROBABLY OVERFLOW'/
 0371               15X,'THEY SHOULD BE REDIMENSIONED')
 0372                RETURN
 0373                END

END OF SEGMENT, LENGTH    17, NAME  MESAGE

 0374                FINISH

END OF COMPILATION -  NO ERRORS

     S/C SUBFILE  :      49 BUCKETS USED

CONSOLIDATED BY XPCK 12H    DATE    30/10/79   TIME 12/35/28

        *SHORTLIST

        *IN ED (FORTSEMICOMP)

        *LIB ED (SUBGROUPSRF4.SUBROUTINES)

        *WORK ED (FORTWORKFILE)

        ****

PROGRAM FXXX
EXTENDED DATA (22AM)
COMPACT PROGRAM (DBM)

SEGMENTS MISSING
CURTO2
LINTO2
MOVTO2
DASHED
%FDP

CORE              13248


+GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
+GGGGGGG                                                     GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
+GGGGGGG          NUMBER OF PAGES   12                       GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
+GGGGGGG                                                     GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
+GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
```

```
ELIB2: COMPILATION LISTING - P 1

FORTRAN COMPILATION BY #XFIV MK 3B    DATE  30/10/79  TIME  12/37/13


~~LIST (LP)~~
~~PROGRAM (FXXX)~~
~~INPUT  1 = CR0~~
~~INPUT  3 = TR0~~
~~INPUT  5 = CR1~~
~~OUTPUT 2 = LP0/132~~
~~OUTPUT 6 = LP1/132~~
~~COMPRESS INTEGER AND LOGICAL~~
~~COMPACT PROGRAM~~
~~EXTENDED DATA~~
~~TRACE 2~~
~~END~~


~~TRACE 1~~
~~MASTER ELIB2~~
~~END~~

END OF SEGMENT, LENGTH     4, NAME  ELIB2

      SUBROUTINE CARTES
      COMMON/XYZ/X(100),Y(100),Z(100),NAME(100)
      COMMON/PARAM/NATOMS,NIN,NOUT,NCARD
      COMMON/CELL/NCELL,A11,A21,A22,A31,A32,A33
      COMMON/MATRIX/B11,B12,B13,B22,B23,B33
      COMMON/PROJOM/NFILE,ISHOW,ICMIN,JOIN,IVEC
C FRACTIONAL TO ANGSTROM COORDINATE TRANSFORMATION
C BASED ON MATRIX:
C   A*SIN(B)*SIN(GST)       0           0
C  -A*SIN(B)*COS(GST)       B*SIN(A)    0
C   A*COS(B)                B*COS(A)    C
C
C**  I,J,K VECTORS--
C**       K ALONG C AXIS
C**       J IN BC PLANE
C**       I PERPENDICULAR TO J AND K
C**
      READ(NCARD,4000) NCELL,A,B,C,ALFA,BETA,GAMMA
      WRITE(NOUT,4001)NCELL,A,B,C,ALFA,BETA,GAMMA
 4001 FORMAT(/5X,'CELL TYPE AND DIMENSIONS'/5X,I3,5X,3F10.5,
     15X,3F10.5/)
      A21=0.
      A31=0.
      A32=0.
      ZERO=0.
 4000 FORMAT(10X,I10,6F10.0)
      DTOR=ATAN(1.)/45.
      ALFA=DTOR*ALFA
      BETA=DTOR*BETA
      GAMMA=DTOR*GAMMA
      IF(NCELL-2) 4110,4111,4112
 4110 A11=A
      A22=B
      A33=C
      GO TO 4113
 4111 A11=A*SIN(BETA)
      A22=B
      A31=A*COS(BETA)
      A33=C
      GO TO 4113
 4112 COSALF=COS(ALFA)
      SINALF=SIN(ALFA)
      COSBET=COS(BETA)
      SINBET=SIN(BETA)
      COSGST=(COSALF*COSBET-COS(GAMMA))/(SINALF*SINBET)
      SINGST=SQRT(1.-COSGST*COSGST)
      A11=A*SINBET*SINGST
      A21=-A*SINBET*COSGST
      A31=A*COSBET
      A22=B*SINALF
      A32=B*COSALF
      A33=C
 4113 CONTINUE
      WRITE(NOUT,4120) A11,ZERO,ZERO,A21,A22,ZERO,A31,A32,A33
 4120 FORMAT(//5X,'TRANSFORMATION MATRIX'//5X,3F10.5/5X,3F10.5/5X,
     13F10.5)
      IF(ISHOW.EQ.3)WRITE(NFILE,4122)A11,ZERO,ZERO,
     1A21,A22,ZERO,A31,A32,A33
 4122 FORMAT(3F10.5/3F10.5/3F10.5)
      DET=A11*A22*A33
      DETIN=1.0/DET
      B11=A22*A33*DETIN
      B12=-A21*A33*DETIN
      B13=(A21*A32-A22*A31)*DETIN
      B22=A11*A33*DETIN
      B23=-A11*A32*DETIN
      B33=A11*A22*DETIN
      WRITE(NOUT,4003)B11,ZERO,ZERO,B12,B22,ZERO,B13,B23,B33
 4003 FORMAT(/5X,'REVERSE MATRIX'/5X,3F10.5/5X,3F10.5/5X,3F10.5)
      RETURN
      END

END OF SEGMENT, LENGTH   312, NAME  CARTES

      SUBROUTINE CARTEX
      COMMON/XYZ/X(100),Y(100),Z(100),NAME(100)
      COMMON/PARAM/NATOMS,NIN,NOUT,NCARD
      COMMON/CELL/NCELL,A11,A21,A22,A31,A32,A33
      COMMON/MATRIX/B11,B12,B13,B22,B23,B33
      COMMON/PROJOM/NFILE,ISHOW,ICMIN,JOIN,IVEC
      DO 4002 J=1,NATOMS
      X1=X(J)
      Y1=Y(J)
      Z1=Z(J)
      X2=A11*X1
      Y2=A22*Y1
      Z2=A33*Z1
      IF(NCELL.EQ.1) GO TO 4115
      Z2=Z2+A31*X1
      IF(NCELL.EQ.2) GO TO 4115
      Y2=Y2+A21*X1
      Z2=Z2+A32*Y1
 4115 CONTINUE
      X(J)=X2
      Y(J)=Y2
      Z(J)=Z2
 4002 CONTINUE
      RETURN
      END

END OF SEGMENT, LENGTH    82, NAME  CARTEX

      SUBROUTINE LINES(MLINE)
      COMMON/XYZ/X(100),Y(100),Z(100),NAME(100)
      COMMON/PARAM/NATOMS,NIN,NOUT,NCARD
      COMMON/LIN/LINE(35),NATOM,ICYCL(10)
      DIMENSION PSIARR(32)
      LOGICAL TOR
      READ(NCARD,2050) IEXIT,IHYDRO,ITOR,NATOM,(LINE(J),J=1,NATOM)
C     IEXIT=1 EXBOND CALC
C   IF IHYDRO=3 OR 4  C-H DISTANCE SUPPLIED ON NEXT CARD
C     EXECUTION AS IF IHYDRO=1 OR 2 RESPECTIVELY
      RHYDRO=1.00
      IF(IHYDRO.LE.2) GO TO 2041
      IF (IHYDRO.GE.5) WRITE(NOUT,2043)
 2043 FORMAT(5X,'PERMITTED VALUES OF IHYDRO ARE 1 TO 4 - SEE WRITE-UP'/
     15X,'IHYDRO=',I5,2X,'DETECTED')
       READ(NCARD,2042) RHYDRO
 2042 FORMAT(10X,F10.0)
 2041 CONTINUE
 2046 FORMAT(/5X,'BOND DISTANCES AND ANGLES',
     125X,'PREDICTED HYDROGEN FRACIONAL COORDINATES'/)
      IF(ITOR.EQ.2)WRITE(NOUT,2048)MLINE
 2048 FORMAT(///20X,'PLANE NUMBER',I5//)
      IF(ITOR.NE.2) WRITE(NOUT,2047) MLINE
 2047 FORMAT(///20X,'LINE NUMBER',I5//)
      WRITE(NOUT,2046)
      TOR=ITOR.NE.0
      IFAIL=0
C IF ATOMS FORMA RING THEN SET ICYCL(RING NO) TO 1
C     AND ADD INITIAL TWO ATOMS ON AT END SO THAT THE CROSS
C     PRODUCTS AT THE VERTICES CAN BE CALCULATED
      IF(ITOR.NE.2) GO TO 2049
      ICYCL(MLINE)=1
      NP=NATOM+1
      NPP=NATOM+2
      LINE(NP)=LINE(1)
      LINE(NPP)=LINE(2)
      NPPP=NATOM + 3
      LINE(NPPP)=LINE(3)
      NATOM=NATOM + 3
 2049 CONTINUE
      L=LINE(1)
      M=LINE(2)
      IF(L.GT.NATOMS.OR.M.GT.NATOMS) GO TO 2079
 2050 FORMAT(5X,3I1,36I2)
      X1=X(M)-X(L)
      Y1=Y(M)-Y(L)
      Z1=Z(M)-Z(L)
      CALL NORM(X1,Y1,Z1,R1)
      NATOMM=NATOM-1
      DO 2001 I=2,NATOMM
      IM=I-1
      K=LINE(IM)
      L=LINE(I)
      IP=I+1
      M=LINE(IP)
      IF(M.GT.NATOMS) GO TO 2079
      X2=X(L)-X(M)
      Y2=Y(L)-Y(M)
      Z2=Z(L)-Z(M)
      CALL NORM(X2,Y2,Z2,R2)
      IF(R1.NE.0..AND.R2.NE.0.) GO TO 2040
      WRITE (NOUT,2051)
 2051 FORMAT(/5X,'ERROR IN ATOM LINE LIST-CONSECUTIVE ATOMS COINCIDENT')
      GO TO 2059
 2040 CONTINUE
      CALL ANGLE(X1,Y1,Z1,X2,Y2,Z2,PHI)
C HYDRO CONTAINS NO PROTECTION AGAINST STRAIGHT LINE
C   BOND      THEREFORE THIS MUST BE TESTED BEFORE ENTRY
C    CASE IS STEREOCHEMICALLY UNREASONABLE   THEREFORE SUBSEQUENT
C    HYDROGEN POSITIONIONTING FOR THAT LINE WILL BE ABANDONED
      IF(PHI.GT.179.99)IHYDRO=0
      IF(IHYDRO.EQ.0) GO TO 2038
C
C MAKE COPY OF BOND VECTORS FOR SUBROUTINE HYDRO
      XX1=X1
      YY1=Y1
      ZZ1=Z1
      XX2=X2
      YY2=Y2
      ZZ2=Z2
       CALL HYDRO(IHYDRO,L,XX1,YY1,ZZ1,XX2,YY2,ZZ2,RHYDRO)
 2038 CONTINUE
      IF(IHYDRO-1)2053,2054,2055
 2053 WRITE(NOUT,2052)NAME(K),NAME(L),NAME(M),R1,R2,PHI
      GO TO 2056
 2054 WRITE(NOUT,2052)NAME(K),NAME(L),NAME(M),R1,R2,PHI,XX1,
     1YY1,ZZ1
      GO TO 2056
 2055 WRITE(NOUT,2052)NAME(K),NAME(L),NAME(M),R1,R2,PHI,XX1,YY1,
     1ZZ1,XX2,YY2,ZZ2
 2056 CONTINUE
 2052 FORMAT(5X,3A4,2F8.4,F10.2,8X,3F8.4,3X,3F8.4)
      IF(.NOT.TOR) GO TO 2060
      IFAIL=ITOR
      CALL TORANG(I,X1,Y1,Z1,X2,Y2,Z2,PSI,IFAIL)
      TOR=IFAIL.GE.0
      IF(.NOT.TOR.OR.I.LT.3) GO TO 2060
      IM2=I-2
      PSIARR(IM2)=PSI
 2060 CONTINUE
 2059 CONTINUE
      X1=-X2
      Y1=-Y2
      Z1=-Z2
      R1=R2
 2001 CONTINUE
      GO TO 2072
 2079 WRITE(NOUT,2074)
 2074 FORMAT(5X,'ILLEGAL ATOM IN LINE DECLARATION--'/
     115X,'CALCULATION OF BOND ANGLES ABANDONED')
 2072 CONTINUE
      IF(.NOT.TOR) GO TO 2069
      WRITE(NOUT,2071)
 2071 FORMAT(/5X,'TORSION ANGLES'/)
      NATM3=NATOM-3
      DO 2065 J=1,NATM3
      JP1=J+1
      JP2=J+2
      JP3=J+3
      L1=LINE(J)
      L2=LINE(JP1)
      L3=LINE(JP2)
      L4=LINE(JP3)
      WRITE(NOUT,2066) NAME(L1),NAME(L2),NAME(L3),NAME(L4),
     1PSIARR(J)
 2066 FORMAT(5X,4A4,F10.4)
 2065 CONTINUE
 2069 CONTINUE
      IF(IFAIL.EQ.1) WRITE(NOUT,2070)
 2070 FORMAT(5X,'STRAIGHT LINE BOND ANGLE ENCOUNTERED--'/
     115X,'TORSION ANGLE CALCULATION ABANDONED')
C CYCLE IS CALLED WITH A NEGATIVE INTEGER(=-RING NO) TO
C   TRIGGER OUTPUT FROM CYCLE
      L1=-MLINE
      IF(ITOR.EQ.2)CALL CYCLE(L1,X1,Y1,Z1,R)
       IF(IEXIT.NE.0.AND.IEXIT.NE.2) CALL EXBOND(X1,Y1,Z1)
      RETURN
      END
```

END OF SEGMENT, LENGTH   554, NAME LINES

```
      SUBROUTINE TORANG(I,X1,Y1,Z1,X2,Y2,Z2,PSI,IFAIL)
      CALL CROSS(X1,Y1,Z1,X2,Y2,Z2,XCR2,YCR2,ZCR2)
      CALL NORM(XCR2,YCR2,ZCR2,R)
      IF(IFAIL.EQ.2.AND.I.EQ.2) CALL CYCLE(I,XCR2,YCR2,ZCR2,R)
      IF (R.EQ.0.) GO TO 2607
      IF(IFAIL.EQ.2.AND.I.GT.2) CALL CYCLE(I,XCR2,YCR2,ZCR2,R)
      IF (I.LT.3) GO TO 2601
      CALL ANGLE(XCR1,YCR1,ZCR1,XCR2,YCR2,ZCR2,PSI)
      DOT=XSAVE*XCR2+YSAVE*YCR2+ZSAVE*ZCR2
      IF(DOT.LT.0.)PSI=-PSI
 2601 CONTINUE
      XCR1=XCR2
      YCR1=YCR2
      ZCR1=ZCR2
      XSAVE=X1
      YSAVE=Y1
      ZSAVE=Z1
      GO TO 2609
 2607 CONTINUE
      IFAIL=-1
 2609 CONTINUE
      RETURN
      END
```

END OF SEGMENT, LENGTH   157, NAME TORANG

```
      SUBROUTINE CYCLE(I,XCR2,YCR2,ZCR2,R)
      COMMON/XYZ/X(100),Y(100),Z(100),NAME(100)
      COMMON/LMN/ALA(10),AMA(10),ANA(10)
      COMMON/PARAM/NATOMS,NIN,NOUT,NCARD
      COMMON/LIN/LINE(35),NATOM,ICYCL(10)
      DIMENSION DIST(33),NAMTMP(35)
      DELTA=1.0E-9
C FIRST ENTRY(I=2) INTITIALISE,SUBSEQUENTLY ACCUMULATE
C    FINAL ENTY (I NEGATIVE) OUTPUT
      IF(I-2) 2625,2619,2620
 2619 XTOT=0.
      YTOT=0.
      ZTOT=0.
 2620  CONTINUE
      IF (R.LT.DELTA) RETURN
      DOT=XTOT*XCR2+YTOT*YCR2+ZTOT*ZCR2
      IF(DOT)2621,2622,2622
 2622 XTOT=XTOT+XCR2
      YTOT=YTOT+YCR2
      ZTOT=ZTOT+ZCR2
      GO TO 2623
 2621 XTOT=XTOT-XCR2
      YTOT=YTOT-YCR2
      ZTOT=ZTOT-ZCR2
 2623 CONTINUE
      RETURN
 2625 CONTINUE
      MCYCL=-I
      CALL NORM(XTOT,YTOT,ZTOT,R)
      D=0.0
      NATOML=NATOM-2
      DO 2626 J=1,NATOML
      LTEMP=LINE(J)
      TEMP=XTOT*X(LTEMP)+YTOT*Y(LTEMP)+ZTOT*Z(LTEMP)
      D=D+TEMP
      DIST(J)=TEMP
 2626 CONTINUE
      DMEAN=D/FLOAT(NATOML)
      WRITE(NOUT,2627) XTOT,YTOT,ZTOT,DMEAN
 2627 FORMAT(/5X,'EQUATION OF PLANE:'/10X,F6.4,'*X +',F6.4,'*Y +,',
     1F6.4,'*Z =',F8.4/)
      DO 2628 J=1,NATOML
      DIST(J)=DIST(J)-DMEAN
 2628 CONTINUE
C OUTPUT DISTANCES IN FIELDS OF 8
      WRITE(NOUT,2629)
 2629 FORMAT(/5X,'DISTANCE OF ATOMS FROM PLANE'/)
      IFIELD=NATOML/8
      NFIELD=IFIELD +  1
      DO 2631 J=1,NFIELD
      JM=J -1
      ILHS=8*JM + 1
      IRHS=ILHS + 7
      IRHS=MIN0(IRHS,NATOML)
       DO 2635 JJ=ILHS,IRHS
      LTEMP=LINE(JJ)
      NAMTMP(JJ)=NAME(LTEMP)
 2635 CONTINUE
      WRITE(NOUT,2632)(NAMTMP(JJ),DIST(JJ),JJ=ILHS,IRHS)
 2632 FORMAT(5X,8(A4,F7.3,3X))
 2631 CONTINUE
      ALA(MCYCL)=XTOT
      AMA(MCYCL)=YTOT
      ANA(MCYCL)=ZTOT
C RETURN DIRECTION COSINES TO SUBROUTINE LINES
      XCR2=XTOT
      YCR2=YTOT
      ZCR2=ZTOT
      RETURN
      END
```

END OF SEGMENT, LENGTH   264, NAME CYCLE

```
      SUBROUTINE HYDRO(IHYDRO,L,XX1,YY1,ZZ1,XX2,YY2,ZZ2,RHYDRO)
      COMMON/XYZ/X(100),Y(100),Z(100),NAME(100)
C ON ENTRY XX=BOND VECTOR
C ON EXIT XX=FRACTIONAL COORDINATES OF HYDROGEN POSITIONS
      IF(IHYDRO.NE.1) GO TO 2829
C C-H BOND DISTANCES HAVE BEEN SET TO 1 ANGSTOM
      XTEMP=XX1+XX2
      YTEMP=YY1+YY2
      ZTEMP=ZZ1+ZZ2
      CALL NORM(XTEMP,YTEMP,ZTEMP,R)
      XX2=0.
      YY2=0.
      ZZ2=0.
      XX1=X(L)+XTEMP*RHYDRO
      YY1=Y(L)+YTEMP*RHYDRO
      ZZ1=Z(L)+ZTEMP*RHYDRO
      CALL FRCOR(XX1,YY1,ZZ1)
      RETURN
 2829 CONTINUE
C   IHYDRO=2 RESERVED FOR TETRAHEDRAL SITING
C CALCULATION OF HYDROGEN POSITIONS BY
C     NORM(NORM(XX1 CROSS XX2)-NORM(XX1 + XX2))
C AND
C     NORM(-NORM(XX1 CROSS XX2)-NORM(XX1 + XX2))
C  CROSS PRODUCT IS WEIGHTED BY SQRT(2)
C     I.E. TAN(HALF TETRAHEDRAL ANGLE)
      RT2=1.414214
C  C-H DISTANCES SET EQUAL TO 1.0 ANGSTROM
      CALL CROSS(XX1,YY1,ZZ1,XX2,YY2,ZZ2,XCR,YCR,ZCR)
      CALL NORM(XCR,YCR,ZCR,R)
      XCR=XCR*RT2
      YCR=YCR*RT2
      ZCR=ZCR*RT2
      XTEMP=XX1+XX2
      YTEMP=YY1+YY2
      ZTEMP=ZZ1+ZZ2
      CALL NORM(XTEMP,YTEMP,ZTEMP,R)
      XTEMP=XCR+XTEMP
      YTEMP=YCR+YTEMP
      ZTEMP=ZCR+ZTEMP
      CALL NORM(XTEMP,YTEMP,ZTEMP,R)
      XSAVE=X(L)
      YSAVE=Y(L)
      ZSAVE=Z(L)
      XX1=XSAVE+XTEMP*RHYDRO
      YY1=YSAVE+YTEMP*RHYDRO
      ZZ1=ZSAVE+ZTEMP*RHYDRO
      CALL FRCOR(XX1,YY1,ZZ1)
      XTEMP=-XCR+XTEMP
      YTEMP=-YCR+YTEMP
      ZTEMP=-ZCR+ZTEMP
      CALL NORM(XTEMP,YTEMP,ZTEMP,R)
      XX2=XSAVE+XTEMP*RHYDRO
      YY2=YSAVE+YTEMP*RHYDRO
      ZZ2=ZSAVE+ZTEMP*RHYDRO
      CALL FRCOR(XX2,YY2,ZZ2)
      RETURN
      END

END OF SEGMENT, LENGTH   229, NAME  HYDRO

      SUBROUTINE EXBOND(X1,Y1,Z1)
      COMMON/XYZ/X(100),Y(100),Z(100),NAME(100)
      COMMON/LIN/LINE(35),NATOM,ICYCL(10)
      COMMON/PARAM/NATOMS,NIN,NOUT,NCARD
      READ(NCARD,2901)NEXIT,(LINE(J),J=1,NEXIT)
 2901 FORMAT(8X,36I2)
      MEXIT=NEXIT/2
      WRITE(NOUT,2906)
 2906 FORMAT(/5X,'EXIT BONDS'/)
      DO 2902 J=1,MEXIT
      JT=J*2
      JM=JT-1
      L1=LINE(JT)
      L2=LINE(JM)
      X2=X(L1)-X(L2)
      Y2=Y(L1)-Y(L2)
      Z2=Z(L1)-Z(L2)
      CALL NORM(X2,Y2,Z2,R)
      IF(R.EQ.0.) GO TO 2905
      CALL ANGLE(X1,Y1,Z1,X2,Y2,Z2,PHI)
C PHI GT 90 IF BOND BELOW RING, LT 90 IF ABOVE RING
C   PROVIDED BOND ORDER IS FROM RING ATOM TO EXOCYCLIC ATOM
      WRITE(NOUT,2903)NAME(L1),NAME(L2),R,PHI
 2905 CONTINUE
 2902 CONTINUE
 2903 FORMAT(5X,2A4,2F10.5)
      RETURN
      END

END OF SEGMENT, LENGTH   134, NAME  EXBOND
```

```
      SUBROUTINE CONN
      COMMON/XYZ/X(100),Y(100),Z(100),NAME(100)
      COMMON/PARAM/NATOMS,NIN,NOUT,NCARD
      DIMENSION DISTAN(100)
      WRITE(NOUT,3000)
3000 FORMAT(5X,'INTRAMOLECULAR CONNECTIVITY MATRIX'/5X,'DISTANCES IN
     1ANESTROMS BETWEEN ALL ATOMS IN MOLECULE  GIVEN IN UPPER TRIANGLE'
     2/)
      WRITE(NOUT,3001)(NAME(J),J=1,NATOMS)
3001 FORMAT(10X,18(A4,2X))
      WRITE(NOUT,3002)
3002 FORMAT(/)
      DO 3010 J=1,NATOMS
      DO 3011 K=1,NATOMS
      IF(K.LE.J) GO TO 3012
      DX=X(J)-X(K)
      DY=Y(J)-Y(K)
      DZ=Z(J)-Z(K)
      DISTAN(K)=SQRT(DX*DX + DY*DY + DZ*DZ)
      GO TO 3011
3012 CONTINUE
      DISTAN(K)=0.0
3011 CONTINUE
      MINAT=MIN0(NATOMS,18)
      WRITE(NOUT,3003) NAME(J),(DISTAN(K),K=1,MINAT)
3003 FORMAT(4X,A4,2X,18F6.2)
      IF(NATOMS.LE.20) GO TO 3009
      WRITE(NOUT,3004)(DISTAN(K),K=21,NATOMS)
3004 FORMAT(10X,20F6.2)
3009 CONTINUE
      WRITE(NOUT,3002)
3010 CONTINUE
      RETURN
      END


END OF SEGMENT, LENGTH   160, NAME  CONN


      SUBROUTINE CONMIN
      COMMON/SYMMER/MSYMEL(16)
      COMMON/SYMMEL/NSYMM,RX(16),TX(16),RY(16),TY(16),RZ(16),TZ(16)
      COMMON/XYZ/X(100),Y(100),Z(100),NAME(100)
      COMMON/PARAM/NATOMS,NIN,NOUT,NCARD
      COMMON/PARAM2/MODE,IBOND,BMIN,BMAX,PHIMIN,PHIMAX
      COMMON/PROJOM/NFILE,ISHOW,ICMIN,JOIN,IVEC
      DIMENSION DISTAN(100),ISYMM(100)
      DIMENSION DXM(100),DYM(100),DZM(100)
      DIMENSION MBOND(20),MHIT(20)
C  IF ISHOW=5 PUT BOND ANGLES AND BOND DISTANCES IN SEPARATE
C   FILES IN PUBLICATION FORMAT
 3092 FORMAT(10X,6F10.0)
      NCP1=8
      IF(MODE.EQ.0) GO TO 3094
      WRITE(NOUT,3093)
 3093 FORMAT(5X,'SYMMETRY OPERATIONS CANNOT BE DONE WHEN ANGSTROM COORDI
     1NATES ARE SUPPLIED'/5X,'DISTANCE MINIMISATION PROCEDURE HAS BEEN
     2ABANDONED')
      RETURN
 3094 WRITE(NOUT,3096)
 3096 FORMAT(5X,'CONNECTIVITY MATRIX WITH INTERATOMIC DISTANCES MINIMISE
     1D BY SYMMETRY OPERATIONS')
      LIN=(NATOMS-1)/10 + 1
      DO 3160 JX=1,LIN
      JY=(JX-1)*10 +1
      JZ=JX*10
      JZ=MIN0(NATOMS,JZ)
      WRITE(NOUT,3098) (NAME(I),I=JY,JZ)
 3160 CONTINUE
 3098 FORMAT(10X,10(3X,A4,4X))
      WRITE(NOUT,3102)
C
      IF(ISHOW.EQ.5) GO TO 3085
      IF(IBOND.NE.0)WRITE(NFILE,3125)BMIN,BMAX,PHIMIN,PHIMAX
 3125 FORMAT(5X,'BONDED ATOMS IN RANGE FROM',F8.3,2X,'TO',
     1F8.3,2X,'ANGSTROMS'/5X,'CENTRAL ATOMS LISTED FOLLOWED BY BONDED AT
     2OMS  AND SYMMETRY OPERATIONS'/5X,'BOND IS ATOM1-CENTRAL ATOM-ATOM2
     3'/5X,'ACCEPTED BOND ANGLES ARE',F10.3,'  TO',F10.3/
     45X,'EQN OF PLANE THRU CENTRAL AND BONDED ATOMS: LX+MY+NZ=P'/
     55X,'LIST FOLLOWING CENTRAL ATOMS IS'///
     61X,'ATOM1 SYMOP1  DIST1   ATOM2  SYMOP2  DIST2',2X,
     7'ANGLE      L      M       N          P'//)
 3085 CONTINUE
      DO 3110 J=1,NATOMS
      XJ=X(J)
      YJ=Y(J)
      ZJ=Z(J)
      CALL CART(XJ,YJ,ZJ)
      DO 3111 K=1,NATOMS
      IF(K.LT.J.AND.IBOND.EQ.0) GO TO 3012
      CALL DMIN(J,K,ISYMM(K),DISTAN(K),DXM(K),DYM(K),DZM(K))
 3080 FORMAT(2X,A4,2X,A4,F10.3)
      GO TO 3111
 3012 DISTAN(K)=0.
      ISYMM(K)=0
 3111 CONTINUE
C
      IF(IBOND.EQ.0) GO TO 3149
      IF(ISHOW.NE.5)WRITE(NFILE,3136) NAME(J)
 3136 FORMAT(5X,'CENTRAL ATOM',2X,A4)
      JM=J-1
      NBOND=0
      DO 3133 K=1,NATOMS
      IF(DISTAN(K).LT.BMIN.OR.DISTAN(K).GT.BMAX) GO TO 3133
      NBOND=NBOND + 1
       MBOND(NBOND)=K
      IF(ISHOW.NE.5)GO TO 3133
      IF(K.GE.J) GO TO 3133
      WRITE(NCP1,3080) NAME(J),NAME(K),DISTAN(K)
 3133 CONTINUE
       IF(NBOND.GE.2) GO TO 3141
      IF(NBOND.EQ.1) GO TO 3139
      IF(ISHOW.NE.5)WRITE(NFILE,3138)
 3138 FORMAT(5X,'NO ATOMS BONDED')
      GO TO 3149
 3139 MB=MBOND(1)
      IF(ISHOW.NE.5)WRITE(NFILE,3148) NAME(MB),ISYMM(MB),DISTAN(MB)
 3148 FORMAT(A4,I6,F8.3,2X,A4,I6,F8.3,2X,F8.3,2X,3F7.3,F8.3)
      GO TO 3149
 3141 NM=NBOND-1
      DO 3151 JJ=1,NBOND
 3151 MHIT(JJ)=0
      DO 3412 JJ=1,NM
      JP=JJ+1
      MB=MBOND(JJ)
      D1=DISTAN(MB)
      IB=ISYMM(MB)
      DO 3412 KK=JP,NBOND
      MC=MBOND(KK)
      D2=DISTAN(MC)
      MC=MBOND(KK)
      IC=ISYMM(MC)
      CALL ANGLE(DXM(MB),DYM(MB),DZM(MB),DXM(MC),DYM(MC),DZM(MC),PHI)
      IF(PHI.LT.PHIMIN.OR.PHI.GT.PHIMAX) GO TO 3412
      IF(ISHOW.EQ.5) GO TO 3087
      MHIT(JJ)=1
      MHIT(KK)=1
      CALL CROSS(DXM(MB),DYM(MB),DZM(MB),DXM(MC),DYM(MC),
     1DZM(MC),XCR,YCR,ZCR)
      CALL NORM(XCR,YCR,ZCR,TEMP)
      DORG=XCR*XJ+YCR*YJ+ZCR*ZJ
      WRITE(NFILE,3148)NAME(MB),ISYMM(MB),DISTAN(MB),NAME(MC),ISYMM(MC),
     1DISTAN(MC),PHI,XCR,YCR,ZCR,DORG
      GO TO 3412
 3087 CONTINUE
      WRITE(NFILE,3082) NAME(MB),NAME(J),NAME(MC),PHI
 3082 FORMAT(3(2X,A4),F9.1)
 3412 CONTINUE
      IF(ISHOW.EQ.5) GO TO 3153
      DO 3152 JJ=1,NBOND
      IF(MHIT(JJ).NE.0) GO TO 3152
      MB=MBOND(JJ)
      WRITE(NFILE,3148) NAME(MB),ISYMM(MB),DISTAN(MB)
 3152 CONTINUE
 3153 CONTINUE
 3149 CONTINUE
 3140 CONTINUE
       DO 3170 JX=1,LIN
      JY=(JX-1)*10 + 1
      JZ=JX*10
      JZ=MIN0(JZ,NATOMS)
      WRITE(NOUT,3103) NAME(J),(DISTAN(I),ISYMM(I),I=JY,JZ)
 3103 FORMAT(4X,A4,2X,10(F6.2,I5))
 3170 CONTINUE
      WRITE(NOUT,3102)
 3110 CONTINUE
 3102 FORMAT(/)
      RETURN
      END

END OF SEGMENT, LENGTH   664, NAME  CONMIN -

      SUBROUTINE SELFIM
      COMMON/XYZ/X(100),Y(100),Z(100),NAME(100)
      COMMON/PARAM/NATOMS,NIN,NOUT,NCARD
      DO 3300 J=1,NATOMS
      CALL DMIN(J,J,ISYM,RMIN,DXMIN,DYMIN,DZMIN)
      WRITE(NOUT,3310) NAME(J),ISYM,RMIN
 3310 FORMAT(5X,A4,I10,F10.5)
 3300 CONTINUE
      RETURN
      END

END OF SEGMENT, LENGTH    36, NAME  SELFIM
```

```
      SUBROUTINE DMIN(J,K,ISYM,RMIN,DXMIN,DYMIN,DZMIN)
      COMMON/SYMMER/MSYMEL(16)
      COMMON/SYMMEL/NSYMM,RX(16),TX(16),RY(16),TY(16),RZ(16),TZ(16)
      COMMON/XYZ/X(100),Y(100),Z(100),NAME(100)
      COMMON/PROJOM/NFILE,ISHOW,ICMIN,JOIN,IVEC
      LOGICAL SKIP,IDENTY
      DELTA=0.5
      XJ=X(J)
      YJ=Y(J)
      ZJ=Z(J)
      XK=X(K)
      YK=Y(K)
      ZK=Z(K)
      R2=10000.0
      DO 3120 L=1,NSYMM
      XX=XK
      YY=YK
      ZZ=ZK
      CALL SYMTRY(XX,YY,ZZ,L)
      DX=XX-XJ
      DY=YY-YJ
      DZ=ZZ-ZJ
      IF(ABS(DX).LT.DELTA) GO TO 3113
      DTX=DX + SIGN(0.5,DX)
      ITX=-IFIX(DTX)
      DX=DX + FLOAT(ITX)
      IF(ITX.LT.0) ITX=10+ITX
      GO TO 3114
 3113 ITX=0
 3114 CONTINUE
      IF(ABS(DY).LT.0.5) GO TO 3115
      DTY=DY + SIGN(0.5,DY)
      ITY=-IFIX(DTY)
      DY=DY+FLOAT(ITY)
      IF(ITY.LT.0) ITY=10+ITY
      GO TO 3116
 3115 CONTINUE
      ITY=0
 3116 CONTINUE
       IF(ABS(DZ).LT.DELTA) GO TO 3117
      DTZ=DZ + SIGN(0.5,DZ)
      ITZ=-IFIX(DTZ)
      DZ=DZ + FLOAT(ITZ)
      IF(ITZ.LT.0)ITZ=10+ITZ
      GO TO 3118
 3117 ITZ=0
 3118 CONTINUE
      IDENTY=L.EQ.1.AND.ITX.EQ.0.AND.ITY.EQ.0.AND.ITZ.EQ.0
      SKIP=J.EQ.K.OR.ICMIN.EQ.3
      IF(SKIP.AND.IDENTY) GO TO 3120
      CALL CART(DX,DY,DZ)
      DS2=DX*DX + DY*DY + DZ*DZ
      IF(DS2.GT.R2) GO TO 3120
      R2=DS2
      ISYM=1000*L + 100*ITX + 10*ITY + ITZ
      DXMIN=DX
      DYMIN=DY
      DZMIN=DZ
 3120 CONTINUE
      RMIN=SQRT(R2)
      CALL NORM(DXMIN,DYMIN,DZMIN,RRR)
      RETURN
      END
```

```
END OF SEGMENT, LENGTH   285, NAME  DMIN
```

```
      SUBROUTINE ORAXIS(X1,Y1,Z1,X2,Y2,Z2,XX,YY,ZZ)
C     ON INPUT XX IS ORIENTATION VECTOR  AND  X1 ANY VECTOR
C     ON EXIT  X1,X2,XX  FORM ORTHOGONAL AXES  XX IS INVARIANT
      CALL CROSS(XX,YY,ZZ,X1,Y1,Z1,X2,Y2,Z2)
      CALL NORM(X2,Y2,Z2,R2)
      IF(R2.LT.1.E-7) GO TO 3300
      CALL CROSS(X2,Y2,Z2,XX,YY,ZZ,X1,Y1,Z1)
      CALL NORM(X1,Y1,Z1,R1)
      RETURN
 3300 X1=-2.0
      RETURN
      END
```

```
END OF SEGMENT, LENGTH   100, NAME  ORAXIS
```

```
      SUBROUTINE ROTATE(X1,Y1,Z1,X2,Y2,Z2,XX,YY,ZZ)
      COMMON/XYZ/X(100),Y(100),Z(100),NAME(100)
      COMMON/PARAM/NATOMS,NIN,NOUT,NCARD
      COMMON/SYMMER/MSYMEL(16)
      COMMON/SYMMEL/NSYMM,RX(16),TX(16),RY(16),TY(16),RZ(16),TZ(16)
       COMMON/ROTOP/IROTOP,MAGTAP
      COMMON/PROJOM/NFILE,ISHOW,ICMIN,JOIN,IVEC
      IF(NSYMM.EQ.0) NSYMM=1
      DO 3400 J=1,NSYMM
      DO 3410 K=1,NATOMS
      XK=X(K)
      YK=Y(K)
      ZK=Z(K)
      IF(NSYMM.EQ.1) GO TO 3412
      CALL FRCOR(XK,YK,ZK)
      CALL SYMTRY(XK,YK,ZK,J)
      CALL CART(XK,YK,ZK)
 3412 CONTINUE
      XN=X1*XK + Y1*YK + Z1*ZK
      YN=X2*XK + Y2*YK + Z2*ZK
      ZN=XX*XK + YY*YK + ZZ*ZK
      IF(IROTOP.NE.2)WRITE(NOUT,3414) NAME(K),J,XN,YN,ZN
 3414 FORMAT(5X,A4,4X,I2,4X,3F10.5)
      IF(IROTOP.EQ.1.OR.IROTOP.EQ.3) WRITE(MAGTAP) XN,YN,ZN
      WRITE(NFILE,3416) XN,YN,ZN
 3416 FORMAT(10X,3F15.10)
 3410 CONTINUE
 3400 CONTINUE
      RETURN
      END
```

END OF SEGMENT, LENGTH   209, NAME ROTATE

```
      SUBROUTINE SYMTRY(XX,YY,ZZ,L)
      COMMON/SYMMER/MSYMEL(16)
      COMMON/SYMMEL/NSYMM,RX(16),TX(16),RY(16),TY(16),RZ(16),TZ(16)
      IF(MSYMEL(L).EQ.0) GO TO 2730
       ISL=MSYMEL(L)
      IF(ISL.EQ.123) GO TO 2730
      XXX=XX
      YYY=YY
      ZZZ=ZZ
      IF(ISL.NE.132) GO TO 2705
      CALL ROTSYM(XX,YY,ZZ,XXX,ZZZ,YYY,L)
      GO TO 2720
 2705 CONTINUE
      IF(ISL.NE.213) GO TO 2707
      CALL ROTSYM(XX,YY,ZZ,YYY,XXX,ZZZ,L)
      GO TO 2720
 2707 CONTINUE
      IF(ISL.NE.231) GO TO 2709
      CALL ROTSYM(XX,YY,ZZ,YYY,ZZZ,XXX,L)
      GO TO 2720
 2709 CONTINUE
      IF(ISL.NE.312) GO TO 2711
      CALL ROTSYM(XX,YY,ZZ,ZZZ,XXX,YYY,L)
      GO TO 2720
 2711 CONTINUE
      IF(ISL.NE.321) GO TO 2713
      CALL ROTSYM(XX,YY,ZZ,ZZZ,YYY,XXX,L)
      GO TO 2720
 2713 WRITE(NOUT,2714) L,MSYMEL(L)
 2714 FORMAT(5X,'ERROR IN SYMMETRY CARD NO.',I3/
     15X,'PARAMETER MSYMEL=',I8)
      STOP
 2720 CONTINUE
      XX=XX + TX(L)
      YY=YY + TY(L)
       ZZ=ZZ + TZ(L)
      RETURN
 2730 CONTINUE
      XX=XX*RX(L)+TX(L)
      YY=YY*RY(L)+TY(L)
      ZZ=ZZ*RZ(L)+TZ(L)
      RETURN
      END
```

END OF SEGMENT, LENGTH   172, NAME SYMTRY

```
      SUBROUTINE CART(XX,YY,ZZ)
      COMMON/CELL/NCELL,A11,A21,A22,A31,A32,A33
      XTEMP=XX
      YTEMP=YY
      ZTEMP=ZZ
      XX=A11*XTEMP
      YY=A21*XTEMP + A22*YTEMP
      ZZ=A31*XTEMP + A32*YTEMP + A33*ZTEMP
      RETURN
      END
```

END OF SEGMENT, LENGTH   56, NAME CART

```
      SUBROUTINE ROTSYM(XX,YY,ZZ,XXX,YYY,ZZZ,L)
      COMMON/SYMMER/MSYMEL(16)
       COMMON/SYMMEL/NSYMM,RX(16),TX(16),RY(16),TY(16),RZ(16),TZ(16)
      XX=RX(L)*XXX
      YY=RY(L)*YYY
      ZZ=RZ(L)*ZZZ
```

```
ELIB2: COMPILATION LISTING - P 10

 0753                   RETURN
 0754                   END

END OF SEGMENT, LENGTH    65, NAME  ROTSYM

 0755                   SUBROUTINE CROSS(X1,Y1,Z1,X2,Y2,Z2,XCR,YCR,ZCR)
 0756                   XCR=Y1*Z2-Z1*Y2
 0757                   YCR=Z1*X2-X1*Z2
 0758                   ZCR=X1*Y2-Y1*X2
 0759                   RETURN
 0760                   END

END OF SEGMENT, LENGTH    80, NAME  CROSS

 0761                   SUBROUTINE NORM(X,Y,Z,R)
 0762                   R=SQRT(X*X+Y*Y+Z*Z)
 0763                   IF(R.EQ.0.) GO TO 2701
 0764                   R1=1.0/R
 0765                   X=X*R1
 0766                   Y=Y*R1
 0767                   Z=Z*R1
 0768              2701 CONTINUE
 0769                   RETURN
 0770                   END

END OF SEGMENT, LENGTH    61, NAME  NORM

 0771                   SUBROUTINE ANGLE(X1,Y1,Z1,X2,Y2,Z2,PHI)
 0772                   DOT=X1*X2+Y1*Y2+Z1*Z2
 0773                   IF(DOT.GT.1.) DOT=1.
 0774                   IF(DOT.LT.-1.) DOT=-1.
 0775                   PHI=ACOS(DOT)
 0776                   PHI=57.2957795*PHI
 0777                   RETURN
 0778                   END

END OF SEGMENT, LENGTH    82, NAME  ANGLE

 0779                   SUBROUTINE FRCOR(XX,YY,ZZ)
 0780                   COMMON/MATRIX/B11,B12,B13,B22,B23,B33
 0781                   COMMON/PARAM2/MODE
 0782             C  IF ANGSTROM COORDINATES SUPPLIED NO TRANSFORMATION NECESSARY
 0783                    IF(MODE.NE.0) RETURN
 0784             C MULTIPLY XX BY B-TRANSPOSE
 0785                   XTEMP=XX
 0786                   YTEMP=YY
 0787                   ZTEMP=ZZ
 0788                   XX=B11*XTEMP
 0789                   YY=B12*XTEMP + B22*YTEMP
 0790                   ZZ=B13*XTEMP + B23*YTEMP + B33*ZTEMP
 0791                   RETURN
 0792                   END

END OF SEGMENT, LENGTH    61, NAME  FRCOR

 0793                   FINISH

END OF COMPILATION -  NO ERRORS

     S/C SUBFILE  :      96 BUCKETS USED

  S/C FILE          EXTENDED TO      160  BUCKETS

CONSOLIDATED BY XPCK 12H    DATE    30/10/79   TIME 12/44/24

        *SHORTLIST

        *IN ED (FORTSEMICOMP)

        *LIB ED (SUBGROUPSRF4.SUBROUTINES)

        *WORK ED (FORTWORKFILE)

        ****

PROGRAM FXXX
EXTENDED DATA (22AM)
COMPACT PROGRAM (DBM)

SEGMENTS MISSING
%FDP

CORE             12480
```

```
ELIB3: COMPILATION LISTING - P 1

FORTRAN COMPILATION BY #XFIV MK 3B     DATE  30/10/79  TIME  12/42/33


 0000                  LIST (LP)
 0001                  PROGRAM (FXXX)
 0002                  INPUT  1 = CR0
 0003                  INPUT  3 = TR0
 0004                  INPUT  5 = CR1
 0005                  OUTPUT 2 = LP0/132
 0006                  OUTPUT 6 = LP1/132
 0007                  COMPRESS INTEGER AND LOGICAL
 0008                  COMPACT PROGRAM
 0009                  EXTENDED DATA
 0010                  TRACE 2
 0011                  END



 0012                  TRACE 1
 0000                  MASTER ELIB3
 0001                  END

END OF SEGMENT, LENGTH     4, NAME  ELIB3

 0002                     SUBROUTINE SYMRED
 0003                  COMMON/SYMMER/MSYMEL(16)
 0004                  COMMON/SYMMEL/NSYMM,RX(16),TX(16),RY(16),TY(16),RZ(16),TZ(16)
 0005                  DO 7001 J=1,NSYMM
 0006                  READ(NCARD,7002) RX(J),TX(J),RY(J),TY(J),RZ(J),TZ(J),MSYMEL(J)
 0007             7002 FORMAT(6F10.0,I10)
 0008                  WRITE(6,7002) RX(J),TX(J),RY(J),TY(J),RZ(J),TZ(J),MSYMEL(J)
 0009             7001 CONTINUE
 0010                  RETURN
 0011                  END

END OF SEGMENT, LENGTH    83, NAME  SYMRED

 0012                  FINISH

END OF COMPILATION -  NO ERRORS

     S/C SUBFILE  :       8 BUCKETS USED

CONSOLIDATED BY XPCK 12H     DATE     30/10/79    TIME 12/43/23

        *SHORTLIST

        *IN ED (FORTSEMICOMP)

        *LIB ED (SUBGROUPSRF4.SUBROUTINES)

        *WORK ED (FORTWORKFILE)

        ****

PROGRAM FXXX
EXTENDED DATA (22AM)
COMPACT PROGRAM (DBM)
CORE              4800


*IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
*IIIIIII                              IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
*IIIIIII       NUMBER OF PAGES    3   IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
*IIIIIII                              IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
*IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
```

```
ELIB4: COMPILATION LISTING - P 1

FORTRAN COMPILATION BY #XFIV MK 3B    DATE 30/10/79  TIME  12/44/07
```

```
      SUBROUTINE STACKS(NFTWO)
      COMMON/MOVES/MOVE(100),MSYM(100)
      DIMENSION MFWRD(100)
      COMMON/LISTS/NLIST,NMOL,LIST(100),MOLIST(100)
C
      WRITE(6,2002) NLIST,NFTWO
 2002 FORMAT(2X,'STACKS:',2I5)
C
      IF(NFTWO.GE.0) GO TO 2020
      L1=LIST(1)
      MSYM(L1)=1000
      MOLIST(1)=L1
      DO 2005 J=2,NLIST
      JM=J-1
 2005 LIST(JM)=LIST(J)
      NLIST=NLIST - 1
      NMOL=1
      WRITE(6,2212) NLIST,NMOL
      WRITE(6,2214) LIST
      WRITE(6,2214) MOLIST
C -- RESET MOVE AND RETURN
      GO TO 2216
C
C
 2020 CONTINUE
      NFWRD=0
      DO 2040 J=1,NLIST
      MFWRD(J) =NFWRD
      IF(MOVE(J)) 2150,2040,2100
 2150 NFWRD = NFWRD + 1
      GO TO 2040
 2100 NFWRD = NFWRD + 1
      NMOL=NMOL + 1
      MOLIST(NMOL)=LIST(J)
 2040 CONTINUE
C
      WRITE(6,2202) NFWRD
 2202 FORMAT(2X,'STACKS:NFWRD=',I5)
      WRITE(6,2214) MFWRD
C
      DO 2200 J=1,NLIST
      NF=MFWRD(J)
      IF(NF.EQ.0) GO TO 2200
      NEW=J - NF
      LIST(NEW)=LIST(J)
 2200 CONTINUE
      NLIST=NLIST - NFWRD
      WRITE(6,2212) NLIST,NMOL
 2212 FORMAT(5X,'NLIST,NMOL',2I5)
      WRITE(6,2214) LIST
      WRITE(6,2214) MOLIST
      WRITE(6,2214) MOVE
 2214 FORMAT(2X,50I2)
C
C
 2216 CONTINUE
       DO 2217 J=1,NLIST
 2217 MOVE(J)=0
C
C
      RETURN
      END

END OF SEGMENT, LENGTH   202, NAME  STACKS

      SUBROUTINE SYMMEX(IAT,ISYM)
      COMMON/LISTS/NLIST,NMOL,LIST(100),MOLIST(100)
      COMMON/XYZ/X(100),Y(100),Z(100),NAME(100)
      COMMON/SYMMER/MSYMEL(16)
      COMMON/SYMMEL/NSYMM,RX(16),TX(16),RY(16),TY(16),RZ(16),TZ(16)
      X1=X(IAT)
      Y1=Y(IAT)
      Z1=Z(IAT)
      ISN=ISYM/1000
      CALL SYMTRY(X1,Y1,Z1,ISN)
      JX=ISYM - 1000*ISN
      IX=JX/100
      JX=JX - 100*IX
      IY=JX/10
      IZ=JX - 10*IY
      IF(IX.GT.5) IX=10 - IX
      IF(IY.GT.5) IY=10 - IY
      IF(IZ.GT.5) IZ=10 - IZ
      X(IAT)=X1 + FLOAT(IX)
      Y(IAT)=Y1 + FLOAT(IY)
      Z(IAT)=Z1 + FLOAT(IZ)
      RETURN
      END
```

```
ELIB4: COMPILATION LISTING - P 2


END OF SEGMENT, LENGTH   117, NAME  SYMMEX

 0087                  FINISH

END OF COMPILATION -  NO ERRORS

     S/C SUBFILE  :       13 BUCKETS USED

CONSOLIDATED BY XPCK 12H     DATE    30/10/79    TIME 12/45/03

         *SHORTLIST

         *IN ED (FORTSEMICOMP)

         *LIB ED (SUBGROUPSRF4.SUBROUTINES)

         *WORK ED (FORTWORKFILE)

         ****

PROGRAM FXXX
EXTENDED DATA (22AM)
COMPACT PROGRAM (DBM)

SEGMENTS MISSING
SYMTRY

CORE                6464


+JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
+JJJJJJJ                                JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
+JJJJJJJ        NUMBER OF PAGES    4    JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
+JJJJJJJ                                JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
+JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
```

```
ELIB6: COMPILATION LISTING - P 1

FORTRAN COMPILATION BY #XFIV MK 3B    DATE  30/10/79  TIME  12/45/32


      SUBROUTINE BOX(AX,AY,AZ)
      CALL MOVTO3(0.,0.,0.)
      CALL LINTO3(AX,0.,0.)
      CALL LINTO3(AX,AY,0.)
      CALL LINTO3(0.,AY,0.)
      CALL LINTO3(0.,0.,0.)
      CALL LINTO3(0.,0.,AZ)
      CALL LINTO3(AX,0.,AZ)
      CALL LINTO3(AX,AY,AZ)
      CALL LINTO3(0.,AY,AZ)
      CALL LINTO3(0.,0.,AZ)
      CALL MOVTO3(AX,0.,0.)
      CALL LINTO3(AX,0.,AZ)
      CALL MOVTO3(AX,AY,0.)
      CALL LINTO3(AX,AY,AZ)
      CALL MOVTO3(0.,AY,0.)
      CALL LINTO3(0.,AY,AZ)
      CALL MOVTO3(0.,0.,0.)
      RETURN
      END

END OF SEGMENT, LENGTH    93, NAME  BOX

       SUBROUTINE AXES(AX,AY,AZ)
      CALL MOVTO3(0.,0.,0.)
      CALL LINTO3(AX,0.,0.)
      DX=.9*AX
      DY=.1*AY
      DDY=-DY
      CALL MOVTO3(DX,DY,0.)
      CALL LINTO3(AX,0.,0.)
      CALL LINTO3(DX,DDY,0.)
      CALL MOVTO3(0.,0.,0.)
       DY=.9*AY
      DX=.1*AX
      CALL LINTO3(0.,AY,0.)
      DDX=-DX
       CALL MOVTO3(DX,DY,0.)
       CALL LINTO3(0.,AY,0.)
       CALL LINTO3(DDX,DY,0.)
      DDY=.8*AY
      CALL MOVTO3(DX,DDY,0.)
      CALL LINTO3(0.,DY,0.)
      CALL LINTO3(DDX,DDY,0.)
      CALL MOVTO3(0.,0.,0.)
      CALL LINTO3(0.,0.,AZ)
      DZ=.9*AZ
      CALL MOVTO3(DX,0.,DZ)
      CALL LINTO3(0.,0.,AZ)
      CALL LINTO3(DDX,0.,DZ)
      DDZ=.8*AZ
      CALL MOVTO3(DX,0.,DDZ)
      CALL LINTO3(0.,0.,DZ)
      CALL LINTO3(DDX,0.,DDZ)
       DDDZ=.7*AZ
       CALL MOVTO3(DX,0.,DDDZ)
       CALL LINTO3(0.,0.,DDZ)
      CALL LINTO3(DDX,0.,DDDZ)
      RETURN
      END

END OF SEGMENT, LENGTH   151, NAME  AXES

       SUBROUTINE TRFORM(IPROJ,AXONX,AXONY,AXONZ,SCAL,XOR,YOR,A,B,C,
     1SINBET,TANBET)
       AA=SCAL*A
       BB=SCAL*B
      CC=SCAL*C*SINBET
      CALL MOVTO3(0.,0.,0.)
      CALL TRANSF(-1)
      CALL TRANSF(2)
       CALL SHIFT3(XOR,YOR,0.)
C
      GO TO (1100,1110,1120),IPROJ
C   IPROJ=1  AXON  =2  PROJ    =3  FROM
 1100 CALL AXON3(AXONX,AXONY,AXONZ)
      GO TO 1150
 1110 CALL PROJ3(AXONX,AXONY,AXONZ)
      GO TO 1150
 1120 CALL FROM3(AXONX,AXONY,AXONZ)
 1150 CONTINUE
       CALL SHEAR3(1,3,TANBET)
      CALL SCALE3(AA,BB,CC)
      RETURN
      END
```

```
END OF SEGMENT, LENGTH   131, NAME  TRFORM

      FUNCTION DISTM(XX,YY,ZZ,B,C,ASINB,ACOSB)
      DX=ASINB*XX
      DY=B*YY
      DZ=ACOSB*XX + C*ZZ
      DISTM=SQRT(DX*DX + DY*DY +DZ*DZ)
      RETURN
      END

END OF SEGMENT, LENGTH    76, NAME  DISTM

      SUBROUTINE CHECK(RAD)
      IF(RAD.GT..01) RETURN
      WRITE(6,1160)
 1160 FORMAT(1X,'COVALENT RADIUS FOR ATOM TYPE REQUESTED HAS NOT BEEN
     1STORED BY PROGRAMME')
      STOP
      RETURN
      END

END OF SEGMENT, LENGTH    31, NAME  CHECK

       SUBROUTINE TWO(IAX,XC,YC,ZC,HHH)
      CALL MOVTO3(XC,YC,ZC)
      CALL TRABEG
      CALL TRANSF(-1)
      HH=0.5*HHH
      U=.5165*HH
      R=3.5*U
       CALL MOVBY2(0.0,-HH)
      IF(IAX.EQ.1) GO TO 1210
      XF=0.0
      YF=2.*HH
      GO TO 1220
 1210 UH=0.5*U
      YH=2.6458*U
      XF=UH
      YF=YH+HH
 1220 CONTINUE
       CALL ARCBY2(R,HH,XF,YF,0)
      IF(IAX.EQ.0) GO TO 1230
      XD=-XF
      YD=-YH+HH
      CALL MOVBY2(XD,YD)
 1230 CONTINUE
      CALL ARCBY2(-R,-HH,-XF,-YF,0)
      CALL TRAEND
      RETURN
      END

END OF SEGMENT, LENGTH   120, NAME  TWO

      SUBROUTINE BAR1(XC,YC,ZC,RRR)
      CALL MOVTO3(XC,YC,ZC)
      CALL TRABEG
      CALL TRANSF(-1)
      RR=0.5*RRR
      CALL MOVBY2(-RR,0.0)
      CALL ARCBY2(RR,0.0,0.0,0.0,1)
      CALL TRAEND
      RETURN
      END

END OF SEGMENT, LENGTH    54, NAME  BAR1

       SUBROUTINE FOUR(IAX,XC,YC,ZC,HH)
      LOGICAL L42
      CALL MOVTO3(XC,YC,ZC)
      CALL TRABEG
      CALL TRANSF(-1)
      DX=0.5*HH
      DY=-DX
      CALL MOVBY2(0.0,DX)
      DO 1330 J=1,4
      L42=IAX.EQ.2.AND.(J.EQ.2.OR.J.EQ.4)
      SX=DX
      SY=DY
      IF(IAX.EQ.0) GO TO 1350
       IF(IAX.EQ.2.AND..NOT.L42) GO TO 1350
      DDX=-0.5*DX
      DDY=-0.5*DY
      IF(IAX.EQ.1)CALL MOVBY2(DDX,DDY)
 1340 CONTINUE
      SX=1.5*SX
```

```
ELIB6: COMPILATION LISTING - P 3

 0152                   SY=1.5*SY
 0153              1350 CONTINUE
 0154                   CALL LINBY2(SX,SY)
 0155                   IF(IAX.EQ.3.OR.L42)CALL MOVBY2(DDX,DDY)
 0156                   SX=DX
 0157                   DX=DY
 0158                   DY=-SX
 0159              1330 CONTINUE
 0160                   CALL TRAEND
 0161                   RETURN
 0162                   END

END OF SEGMENT, LENGTH   134, NAME  FOUR


 0163                   FINISH

END OF COMPILATION -  NO ERRORS


     S/C SUBFILE  :      25 BUCKETS USED

CONSOLIDATED BY XPCK 12H    DATE    30/10/79   TIME 12/47/38

        *SHORTLIST

        *IN ED (FORTSEMICOMP)

        *LIB ED (SUBGROUPSRF4.SUBROUTINES)

        *WORK ED (FORTWORKFILE)

        ****

PROGRAM FXXX
EXTENDED DATA (22AM)
COMPACT PROGRAM (DBM)

SEGMENTS MISSING
LINTO3
MOVTO3
SCALE3
SHEAR3
FROM3
PROJ3
AXON3
SHIFT3
TRANSF
TRAEND
ARCBY2
MOVBY2
TRABEG
LINBY2

CORE             5440


*KKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKK
```