BNWL-1095

UC-32

5-
6-69

STATUS OF DATA RETRIEVAL SYSTEM

AS OF APRIL 1, 1969

June 1969

AEC RESEARCH &
DEVELOPMENT REPORT

FILE NO.	LOCATION	FILE ACQUISITION DATE
B-Driggs	313473706	AUG 22 1969

BATTELLE

NORTHWEST

BATTELLE MEMORIAL INSTITUTE

PACIFIC NORTHWEST LABORATORY

BATTELLE BOULEVARD, P. O. BOX 999, RICHLAND, WASHINGTON 99352

BNWL-1095

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

PACIFIC NORTHWEST LABORATORY

RICHLAND, WASHINGTON

operated by

BATTELLE MEMORIAL INSTITUTE

for the

UNITED STATES ATOMIC ENERGY COMMISSION UNDER CONTRACT AT(45-1)-1830

BNWL-1095

UC-32, Mathematics and Computers

~~Special Distribution~~

STATUS OF DATA RETRIEVAL SYSTEM

AS OF APRIL 1, 1969

By

R. L. Reynolds

R. L. Engel

R. T. Toyooka

E. L. Wells

RECEIVED
JUL 1 1969

BATTELLE MEMORIAL INSTITUTE
PACIFIC NORTHWEST LABORATORY
RICHLAND, WASHINGTON

Printed in the United States of America
Available from
Clearinghouse for Federal Scientific and Technical Information
National Bureau of Standards, U.S. Department of Commerce
Springfield, Virginia 22151
Price: Printed Copy \$3.00; Microfiche \$0.65

STATUS OF DATA RETRIEVAL SYSTEM

April 1, 1969

By

R. L. Reynolds, R. L. Engel, R. T. Toyooka, E. L. Wells

INTRODUCTION

Most computer-data retrieval programs now in use have been designed for specific applications and pre-defined data. They are frequently inflexible in use, and even minor changes in output formats or in the type of data being processed require expensive reprogramming. Some general-use programs, though available, are limited in applicability. Moreover, queries in both the specific and general-use programs are generally cumbersome and limited.

Battelle-Northwest is developing a general-purpose data retrieval system which is expected to avoid these objectionable characteristics. The system is planned to provide almost unrestricted flexibility in application and use, with no limitations on the quantity of data, except those imposed by hardware.

SYSTEM COMPONENTS

Conceptually, the system will be composed of a generalized search and retrieval program, interfaced with input, output, and maintenance programs (Figure 1). The interfaces will permit the special-function programs to be replaced with specialized versions for unusual applications, without affecting other parts of the system.

THE QUERY INPUT PROGRAMS are designed for the convenience of the user, and will not require fixed-field formats, redundant information, data translation codes, familiarity with programming languages, or complex instructions. Three forms of input will be available. One will be a check-off list of expected queries for a specific application, and will be used as the source document for simplified keypunching. A second form will accept near-English language, with a keyword approach modified to permit alternate wording and such words as are desired to complete an all-English sentence. A third will permit Boolean-like relationships to be specified for the definition of search instructions, and is expected to be used by persons with some knowledge of mathematical logic.

THE OUTPUT PROGRAM will provide a printout in a standardized, self-explanatory format. The user may, if he wishes, specify his own output format at the time he enters his queries.

THE STARTUP PROGRAM will accept card-image data in a format which is specified during the startup operation.

THE UPDATING PROGRAM will permit additions, deletions, and corrections, with formats and procedures similar to those used for query input.

THE MATHEMATICAL OPERATIONS PROGRAM will provide summations, statistics, and projections when requested as part of the query input.

THE SEARCH AND RETRIEVAL PROGRAM will test appropriate items of data for their relationship to a value specified by the query input, and store the locations of data which meet the required relationship.

Figure 1. Block Diagram of Data Retrieval System

CURRENT STATUS

This data retrieval system is being programmed in Fortran-V, for batch processing on a Univac-1108. The control, search-and-retrieval, query-input, output, and startup programs have been written in preliminary form, and are in the initial debugging and test stages. Current versions of these programs are presented in the appendix.

This work was funded under AEC contract AT-45-1-1830.

APPENDIX

CURRENT VERSIONS OF "DATA RETRIEVAL SYSTEM" PROGRAMS

```

INCLUDE COMN,LIST
INCLUDE COMM,LIST
CALL SETUP
CALL CONTRL
STOP
END

```

```

COMN* FCOPY
COMMON MAXDIC
COMMON DIC(8,500)
COMMON NPOL(700)
COMMON ICBDIC(6,10)
COMMON MQUE(8,200)
COMMON NSAV(3000)
COMMON NTP(450)
COMMON NPUS(450)
COMMON MUNFL(2,200)
COMMON NPOLM
COMMON NBOOL
COMMON ISIZE
COMMON IFIRSD
COMMON IU
COMMON IUF1
COMMON IUF2
COMMON LPCF
COMMON NPLM
COMMON NROWS
COMMON NHEAD(4,500)
COMMON IFASTB(14)
COMMON IFTBL(7)
COMMON MCCCC(20)
COMMON FCCCC(20)
COMMON NER(5)
COMMON IERR
COMMON ICO
COMMON IBLSZ
COMMON IDIC(8,500),FMQUE(8,200)
COMMON CBDIC(6,10)
EQUIVALENCE (MQUE(8,200),FMQUE(8,200)),(DIC(8,500),IDIC(8,500))
EQUIVALENCE (ICBDIC(6,10),CBDIC(6,10))
END
COMM* FCOPY
COMMON NBOI(2000)
COMMON MAXCOR
COMMON COR(20000)
COMMON ICOR(20000)
EQUIVALENCE (COR(20000),ICOR(20000))
END

```

A-2

```

A SEG MAIN-A-B-C-D-E-F-*(SETUP,CONTRL-*(INPUT,LOGIC,REPT))
B SEG NPAUSS$-NOTIN$-SY$SM-TLABL$-TSCRH$-THRU$-TSWAP$-TINTL$-OPENF$
C SEG SETDR$-NINPT$-NFLOT$-NEXP$-NININ$-NDBC$-NEXP1$-NOUT$-NIER$
D SEG NERR$-NFM1$-NIOIN$-NFTV$-NTAB$-NTRAN$-NSTOP$-MAUTOS$
E SEG FLDIC-CBDICT-UNCODE-DICT-DOY-TOD
F SEG NCODE$-NEXP5$-NXPAF$-EXP$-ALOG$
  SEG NXPAX$-EDATE$-ETOD$

```

```

SUBROUTINE CONTRL
INCLUDE COMN,LIST
INCLUDE COMM,LIST
CALL FAKE($110,0)
100 CALL INPUT
110 CALL LOGIC
  WRITE(6,15) (NSAV(I),I=1,100)
  WRITE(6,10) (NPOL(I),I=1,100)
  WRITE(6,10) (MUNFL(1,J),J=1,ICO)
  WRITE(6,10) (MUNFL(2,J),J=1,ICO)
15  FORMAT( 5HONSAV,12I10/(5X,12I10))
10  FORMAT(11I7)
12  FORMAT(2I6,6A6)
  CALL REPT
  GO TO 100
END

```

PROGRAM NO. 217

```

. FAKE 217-0002
. 217-0003
. THIS ROUTINE IS USED TO MAKE A FORTRAN V SUBROUTINE LOOK LIKE A 217-0004
. MAIN PROGRAM FOR WALKBACK PURPOSES. IT IS USED ONCE FOR EACH CALL 217-0005
. STATEMENT (WHICH MUST HAVE A STATEMENT NUMBER) IN THE PSEUDO-MAIN 217-0006
. PROGRAM. 217-0007
. THE CALL TO FAKE IS' 217-0008
. CALL FAKE ($K,IARG) 217-0009
. WHERE 217-0010
. K STATEMENT NUMBER OF THE CALL IN THE PSEUDO-MAIN 217-0011
. SUBROUTINE. 217-0012
. IARG NUMBER OF ARGUMENTS IN THE CALL. 217-0013
. 217-0014
. THE CALL TO FAKE PRECEDES ITS CORRESPONDING CALL. 217-0015
. 217-0016
. EXAMPLE 217-0017
. 217-0018
. SUBROUTINE PSEUDO (I,K,L,M) 217-0019
. I=I+1 217-0020
. CALL FAKE ($10,2) 217-0021

```

```

. 10 CALL SUB1 (K,L) 217-0022
.   M=M-2 217-0023
.   CALL FAKE ($15,3) 217-0024
. 15 CALL SUB2 (K,M,I) 217-0025
.   RETURN 217-0026
. 217-0027
.   NBPR. 217-0028
$(1),FAKE* L,H2 A0,0,B11 217-0029
.   A A0,*1,B11 217-0030
.   SZ,H2 1,B12 217-0031
.   J 3,B11 217-0032
.   END . 217-0033

```

```

SUBROUTINE INPUT
PARAMETER INDEX1=700,INDEX2=200,INDEX3=200,INDEX4=8,INDEX5=200
INCLUDE COMN,LIST
INTEGER OR,AND,SPACE,RP
INTEGER BEFORE,AFTER
INTEGER REPORT,SLASH,SEARCH,COMMA
COMMON/INPT/ IWRD(INDEX2), IPDS(INDEX3), IPOL(INDEX1),
1 IOP(50), IR(50), IRT(50), IFILE(100),INAME(50),
2 NUM(10), ICA(80), NF, LP, RP, NOP, IDS, IQ,
3 IEX, IPER, ISC, ICOL, IC, IK, N, AND, OR, NOT,
4 SPACE, REPORT, COMMA, SLASH, SEARCH, IB6
5 ,NW ,KNDEX, MNDEX
DIMENSION ISRPT(5)
DIMENSION T(2)
DATA IPDS(1),IPOL(1),OR,AND,NOT,KNDEX,MNDEX,SPACE,IB6,IW,KNT,IK,IL
1/100,1,2HOR,3HAND,3HNOT,1,0,1H ,2*6H ,3*0/
DATA (IOP(N),N=1,20)/20*6H /
DATA (IOP(N),N=1,20)/5HEQUAL,2HEQ,2HLT,2HLE,2HGE,2HGT,3HLSS,3HLEQ,
13HEQL,3HGEQ,2HNE,3HGTR,3HNEQ,5HEQUIV,6HEQUALS,4HLESS,6HGREATE ,1H=
2,1H=,1H./NOP/20/
DATA (IRT(N),N=1,20)/3,3,1,5,5,2,1,5,3,6,4,2,4,3,3,1,2,3,2,1/
DATA (ISRPT(I),I=1,5)/6HLAST N,6HPAYROL,6HPLANT ,6HBUILD,6HDEGREE
1/BEFORE,AFTER/6HBEFORE,6HAFTER /
DATA LP,RP/1H(.1H)/
DATA IQ,IEX,IPER,ISC,ICOL/1H0,1H0,1H.,1H$,1H'/
DATA MINUS,IALL,IEVERY/1H-,6HALL ,6HEVERYT /
DATA IEND,IDS/6H ,6H$5$5$5$ /NW/25/
DATA IAS,IAGE/1H*,6HAGE /
DATA (NUM(I),I=1,10)/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H0/
DATA REPORT,COMMA,SLASH,SEARCH,SPACE,IB6/6HREPORT,1H,,1H/,6HSEARCH,2*6H
1,2*6H /N/80/
CALL TOD(T)
WRITE(6,9998)T
1 CONTINUE

```

```

M=0
KNDEX=1
MNDEX=0
IPDS(1)=100
IPOL(1)=1
DO 2 I=1,INDEX2
2 IWRD(I)=IB6
DO 3 I=1,INDEX4
DO 3 J=1,INDEX5
3 MQUE(I,J)=IB6
CALL CHAR
IF(FLD(0,6,IAS).NE.FLD(0,6,ICA(1)))GO TO 4
IK=0
CALL CHEKOF(ICA,I)
GO TO (600,1),I
4 CONTINUE
KNT=1
II=1
5 ISTOP=0
ITST=0
I=1
10 I=I+1
CALL CHAR
20 CONTINUE
IF(M.EQ.30.AND.FLD(0,6,IC).NE.FLD(0,6,SPACE))GO TO 22
M=0
GO TO 25
22 IF(FLD(0,6,IC).NE.FLD(0,6,COMMA))GO TO 80
M=0
25 CONTINUE
IF(KNT.LT.2)GO TO 26
IF(IWRD(KNT-1).EQ.REPORT)ITST=1
26 CONTINUE
IF(FLD(0,6,IC).EQ.FLD(0,6,COMMA))GO TO 60
IF(FLD(0,6,IC).EQ.FLD(0,6,SLASH))GO TO 70
IF(FLD(0,6,IC).EQ.FLD(0,6,IPDS))ISTOP=1
IF(FLD(0,6,IC).EQ.FLD(0,6,SPACE))GO TO 30
28 CONTINUE
ISP=0
M=1ABS(6*(II-1))
FLD(M,6,IWRD(KNT))=FLD(0,6,IC)
II=II+1
IF(II.LE.6)GO TO 50
KNT=KNT+1
II=1
IF(ISTOP.GT.0)GO TO 100
GO TO 50
30 CONTINUE

```

A-5

```
IF(M.LT.30.AND.ITST.GT.0)GO TO 28
II=1
ISP=ISP+1
IF(ISP.GT.6)GO TO 50
KNT=KNT+1
IF(ISTOP.GT.0)GO TO 100
50 IF(I.LT.N)GO TO 10
GO TO 5
60 CONTINUE
IF(ITST.GT.0.AND.M.EQ.30)M=0
KNT=KNT+1
II=1
FLD(0,6,IWRD(KNT))=FLD(0,6,COMMA)
KNT=KNT+1
IF(ITST.GT.0)GO TO 65
GO TO 50
65 CONTINUE
66 IF(I.GE.N)I=0
I=I+1
CALL CHAR
IF(FLD(0,6,IC).EQ.FLD(0,6,SPACE))GO TO 66
GO TO 26
70 CONTINUE
II=1
KNT=KNT+1
FLD(0,6,IWRD(KNT))=FLD(0,6,SLASH)
KNT=KNT+1
75 IF(I.GE.N)I=0
I=I+1
CALL CHAR
IF(FLD(0,6,IC).EQ.FLD(0,6,SPACE))GO TO 75
GO TO 20
80 CONTINUE
IF(ITST.GT.0)GO TO 50
J=2
85 M=IABS(6*(II-1))
FLD(M,6,IWRD(KNT))=FLD(0,6,IC)
II=II+1
IF(II.LE.6)GO TO 90
II=1
KNT=KNT+1
90 CONTINUE
IF(I.GE.N)I=0
I=I+1
CALL CHAR
IF(FLD(0,6,IC).EQ.FLD(0,6,IDS))ISTOP=1
IF(FLD(0,6,IC).EQ.FLD(0,6,SPACE))GO TO 95
IF(FLD(0,6,IC).EQ.FLD(0,6,COMMA))GO TO 95
```

```

      GO TO 85
95  CONTINUE
      IF(II.NE.1)KNT=KNT+1
      DO 97 K=1,J
      KK=IABS(6*(K-1))
      FLD(KK,6,IWRD(KNT))=FLD(0,6,ICOL)
97  CONTINUE
      KNT=KNT+1
      II=1
      M=0
      GO TO 50
100 CONTINUE
      DO 150 I=1,KNT
      IF(FLD(0,6,IWRD(I)).EQ.FLD(0,6,ICOL))CALL MOVE(I)
150 CONTINUE
      NF=0
      DO 200 I=1,KNT
      IF(IWRD(I).EQ.IAGE)CALL AGE(I)
      IF(IWRD(I).EQ.BEFORE)IWRD(I)=IOP(3)
      IF(IWRD(I).EQ.AFTER)IWRD(I)=IOP(6)
      IF(IWRD(I).EQ.REPORT)GO TO 300
200 CONTINUE
      KNT=KNT+1
      IWRD(KNT)=REPORT
      I=I+1
      DO 250 II=1,5
      KNT=KNT+1
      IWRD(KNT)=ISRPT(II)
250 CONTINUE
300 CONTINUE
      NN=I+1
      DO 400 I=NN,KNT
      IW=IWRD(I)
      IF(FLD(0,6,IW).EQ.FLD(0,6,ICOL))GO TO 380
      IF(FLD(0,6,IW).EQ.FLD(0,6,MINUS))GO TO 330
      IF(IWRD(I).EQ.IALL.OR.IWRD(I).EQ.IEVERY)GO TO 340
      IF(FLD(0,6,IW).EQ.FLD(0,6,SLASH))GO TO 400
      IF(FLD(0,6,IW).EQ.FLD(0,6,IDS))GO TO 500
      IF(FLD(0,6,IW).EQ.FLD(0,6,COMMA).OR.FLD(0,6,IW).EQ.FLD(0,6,
1IB6))GO TO 400
      DO 325 K=1,10
      IF(FLD(0,6,IW).EQ.FLD(0,6,NUM(K)))GO TO 350
325 CONTINUE
      I2=1
      GO TO 360
330 CONTINUE
      CALL BCD2I(IW)
      I3=IW

```

```

      GO TO 375
340  CONTINUE
      DO 345 J=1,MAXDIC
      IZ=FLD(12,24,IDIC(1,J))
      IRR=MOD(IZ,10000)
      IFN=(IZ-IRR)/10000
      IFILE(J)=IFN
345  CONTINUE
      NF=J
      GO TO 500
350  CONTINUE
      I2=2
      CALL BCD2I(IW)
360  CALL DICT(IW,I2,I3,I4,I5,I6,I7,I8,I9)
      IF(I3.EQ.0)GO TO 400
375  CONTINUE
      NF=NF+1
      IFILE(NF)=I3
      IF(I3.EQ.-99)GO TO 500
      IF(I2.EQ.2)IFILE(NF)=IW
      GO TO 400
380  CONTINUE
      K=0
      DO 385 J=1,6
      M=IABS(6*(J-1))
      IF(FLD(M,6,IW).EQ.FLD(0,6,SPACE))GO TO 390
      K=K+1
385  CONTINUE
390  CONTINUE
      DO 395 J=1,K
      IWRD(I+1+J)=IB6
395  CONTINUE
400  CONTINUE
500  CONTINUE
      NW=KNT
      WRITE(6,6010)(IWRD(I),I=1,KNT)
      WRITE(6,6015)(IFILE(I),I=1,NF)
      CALL QIFACE
      CALL TOD(T)
      WRITE(6,9999)T
600  RETURN
6005 FORMAT(27H NO REPORT PARAMETERS INPUT )
6010 FORMAT(1H 10HIWRD ARRAY /(1X,10A6))
6015 FORMAT(1H 11HIFILE ARRAY /(1X,10I5))
9998 FORMAT(51H -----ENTERED ** SUBROUTINE INPUT ** AT 2A6)
9999 FORMAT(51H ----- LEFT ** SUBROUTINE INPUT ** AT 2A6)
      END

```


A-8

```

SUBROUTINE CHEKOF(ICA,IBACK)
INCLUDE COMM
DIMENSION DUMMY(1500)
DIMENSION ICA(80)
DATA N1/2222222/,N2/7777777/,MIN/1H-/
DEFINE NO(I)=FLD(2,4,ICA(I))*10+FLD(2,4,ICA(I+1))
DEFINE NOM(I)=-1*FLD(2,4,ICA(I+1))
IBACK=1
I1=4
I2=0
I3=2
N=0
DO 20 J=1,5
M=0
I2=I2+2
I2=NO(I2)*I3+I1-2
DO 10 I=I1,I2,I3
GO TO (4,4,6,7,8),J
4 N=N+1
IF (ICA(I) .EQ. MIN) GO TO 5
NPOL(N)=NO(I)
GO TO 10
5 NPOL(N)=NOM(I)
GO TO 10
6 M=M+1
MQUE(1,M)=NO(I)
GO TO 10
7 M=M+1
MQUE(2,M)=NO(I)
GO TO 10
8 M=M+1
FLD( 0,6,MQUE(3,M))=FLD(0,6,ICA(I ))
FLD( 6,6,MQUE(3,M))=FLD(0,6,ICA(I+1))
FLD(12,6,MQUE(3,M))=FLD(0,6,ICA(I+2))
FLD(18,6,MQUE(3,M))=FLD(0,6,ICA(I+3))
FLD(24,6,MQUE(3,M))=FLD(0,6,ICA(I+4))
FLD(30,6,MQUE(3,M))=FLD(0,6,ICA(I+5))
GO TO 10
10 CONTINUE
I1=I2+4
GO TO (11,12,20,19,20),J
11 N=N+1
NPOL(N)=N1
GO TO 20
12 N=N+1
NPOL(N)=N2
GO TO 20

```

```

SUBROUTINE REPT
INCLUDE COMN,LIST
DIMENSION Ibuff(100), Jbuff(3000), IFMT(10,500), IFM(7)
DIMENSION IMESS(3)
DATA KBL / 1H / , ICW,ISV,ISC,ICB,IDF / 30HCW SV SC CB
1 DF / , Icolon / 1H' / , ISL /1H/ /
DATA (IFM(I),I=1,7)/ 42H(26H ,2X, (7A6/)) 7A6/(23X7A6 )
1 /
DATA (IMESS(I),I=1,3) / 12HMISSING CODE, 07777777777777 /
LPR = 6
IF (NSAV(1) .NE. 7777777) GO TO 100 ' ANY DETAILS SELECTED.

C NO DETAILS SATISFY QUERY -
C WRITE (LPR,9000)
C RETURN

C SEARCH NPOL TABLE FOR LIST OF REPORT FILES - BUILD FORMATS
C
100 DO 120 I=1,NPOLM
IF (NPOL(I) .EQ. 2222222) GO TO 130 '2222222 IS SPACER BETWEEN
120 CONTINUE 'SEARCH FILE NO. AND REPORT
WRITE (LPR,9001) 'FILE NO.
RETURN

130 IBEG= I+1
NFILE = 0
DO 180 I=IBEG,NPOLM
IF (NPOL(I).EQ. 7777777) GO TO 200 '7777777 SIGNALS END OF REPORT
IF (NPOL(I).NE. -99) GO TO 135 'FILE NUMBERS.
K = 0 ' -99 REPORT NO SIGNALS LIST
DO 133 J=1,MAXDIC 'ALL DATA FOR THIS INDIVIDUAL.
L = FLD (12,24,IDIC(1,J))
L = L / 10000
CALL DICT (L,2,M,ITYP,M,M,M,M,IR) 'PLACE ALL FILE NUMBERS EXCEPT
'CB FILES IN ARRAY FOLLOWING

```

A-10

```

      IF (ITYP.EQ.ICB .OR. IR.EQ.-1)GOTO 133'THE -99.
      K = K + 1
      NPOL (I+K) = L
133  CONTINUE
      NPOL (I+K +1) = 7777777
      GO TO 180
C      BUILD OBJECT TIME FORMAT FOR EACH REPORT FILE
135  NFIL = I-IBEG + 1
      IF (NPOL(I).EQ. -98) GO TO 180          '-98 SIGNALS PUNCH MAILING LABELS
      IFMT (1,NFIL) = IFM(1)
      DO 140 J=5,10
140  IFMT(J,NFIL) = KBL
      CALL DICT (NPOL(I), 2, NAM, ITYP, IDA, NFORM, NWT, NCW, IR)
      IF (IR.NE.-1 .AND. ITYP.NE.ICB) GO TO 145
      WRITE (LPR,9002) NPOL(I)              'UNMATCHED OR IMPROPER REPORT
      NPOL(I) = 0                            'FILE NUMBER.
      GO TO 180
145  CALL DICFND(NPOL(I), IND)              'RETURNS IND = INDEX IN IDIC ARRAY.
      DO 150 J=1,4                          'PLACE FILE NAMES IN FORMAT.
      L = NHEAD(J,NFIL)                    'NHEAD = FILE NAMES FROM QUERY.
      IF (NHEAD(1,NFIL).EQ.KBL)L=IDIC(1+J,IND)  'IF BLANK, USE NAME
      FLD (30,6,IFMT(J,NFIL))= FLD(0,6,L)    'FROM DICTIONARY.
      FLD (0,30,IFMT(J+1,NFIL)) = FLD(6,30,L)
150  CONTINUE
      DO 170 J=1,4                          'PLACE ' FOLLOWING LAST NON-BLANK
      DO 170 K=1,31,6                      'CHARACTER IN FILE NAME.
      IF (FLD(31-K,6,IFMT(6-J,NFIL)) .EQ.
1   FLD(0,6,KBL)) GO TO 170
      IF (K.EQ.1) GO TO 160
      FLD (37-K,6,IFMT(6-J,NFIL)) = FLD(0,6,ICOLON)
      GO TO 175
160  IFMT (7-J,NFIL) = ICOLON
      GO TO 175
170  CONTINUE
175  IFMT(6,NFIL) = IFM(2)
      CALL UNCODE (NFORM, IFMT(7,NFIL), IFMT(8,NFIL), NWW, NWD)
      IFMT (9,NFIL) = IFM(4)                'UNCODE RETURNS TWO WORD OBJ
      IF((ITYP.EQ.ICW .OR. ITYP.EQ.ISV)      'TIME FORMAT.
1   .AND. NWW.LE.7) GO TO 177
      IFMT (7,NFIL) = IFM(5)                'USE STANDARD 7A6 FORMAT FOR
      IFMT (8,NFIL) = IFM(6)                'SV,FILES OVER 7 WORDS LONG
      IFMT (9,NFIL) = IFM(7)                'AND ALL DF AND CB FILES.
177  WRITE (LPR,9012) NPOL(1), (IFMT(J,NFIL ), J=1,10)
180  CONTINUE
      WRITE(LPR,9003)                      'NO END SIGNAL IN NPOL ARRAY.
      NPOL(NPOLM) = 7777777                'SET LAST WORD TO SIGNAL.
C
C      READ REPORT FILE FOR INDIVIDUALS SPECIFIED BY NSAV  ARRAY

```

```

C      AND REPORT USING OBJECT TIME FORMATS BUILT IN THE PREVIOUS
C      SECTION
C
200 NDET = 0                                'NDET IS INDIVIDUAL ITEM
210 IF (NSAV(NDET+1).EQ.7777777.OR.NDET.GE.'COUNTER. INDEX IS DETAILS
1 3000) GO TO 410                            'RELATIVE LOCATION IN FILE.
    NDET = NDET + 1
    INDEX = NSAV(NDET)
    IF (NFIL.LE.0) GO TO 210
    DO 400 I=1,NFIL
    ISUB = IBEG - 1 + I
    IF (NPOL(IBEG-1+I).EQ.-99) GO TO 300      '-99 SIGNALS LIST ALL INFO.
    IF (NPOL(IBEG-1+I).EQ.-98) GO TO 302      '-98 SIGNALS PUNCH MAIL LABELS
    IF (NPOL(IBEG-1+I).LE. 0) GO TO 400
    CALL DICT (NPOL(IBEG-1+I),2,NAM, ITYP, 'SUBR. DICT RETURNS FILE
1 IDA, NFORM, NWT, NCW, IR)                 'DESCR. AND LOCATION.
    IFTBL(3) = IDA +(INDEX-1)* NCW           'SET DRUM POINTER.
    CALL NTRAN (IU,2,NCW,IBUFF,ITST)         'READ DRUM.
215 IF (ITST +1) 220,215,230                'DELAY UNTIL READ IS COMPLETE.
220 WRITE (LPR,9004) NPOL(ISUB ),INDEX      'READ ERROR.
    CALL NTRAN (IU,22)                       'RE-ENABLE UNIT.
    GO TO 400
C
230 IF (ITYP.NE.ICW .AND. ITYP.NE.ISV) GOTO 250
    CALL LINE (IFMT(1,I), IBUFF, NCW)        'CW OR SV FILE, REPORT IT.
    GO TO 400
C
250 IF (ITYP.NE. ISC) GO TO 280
    IF (IBUFF(1).EQ. 0) GO TO 295
    CALL CBDICT (NPOL(IBEG-1+I),IF2)         'ISC FILE, FIND AND READ
    CALL DICT (IF2,2, NAM, JTYP, JDA,        'ASSOC. CB FILE.
1 JFORM, JTW, JCW, JR )                     'CBDICT RETURNS ASSOC FILE NO.
    IF (JR .EQ.-1 .OR. JTYP.NE.ICB) GOTO 270 'DICT RETURNS FILE DESC AND
    JBUFF(1) = IBUFF(1)                     'DRUM LOCATION.
    JBUFF(2) = IMESS(1)                     'WRITE CODE WE WILL SEARCH
    JBUFF(3) = IMESS(2)                     'FOR AT END OF CB FILE.
    JBUFF(4) = IMESS(3)                     'THIS WILL STOP THE DRUM
    IFTBL(3) = JDA + JTW                     'SEARCH.
    CALL NTRAN (IU,1,4,JBUFF,L)
251 IF (L+1 .EQ. 0) GO TO 251
    IFTBL(3) = JDA
    CALL NTRAN (IU, 5, IBUFF)                'SEARCH FOR CODE.
252 CALL NTRAN (IU,3, 3000,JBUFF,L)          'BLOCK READ.
255 IF (L+1) 260,255, 265                    'DELAY UNTIL READ IS COMPLETE.
260 WRITE (LPR,9005) JTYP, NPOL(ISUB ), 'READ ERROR - WRITE MESSAGE
1 INDEX                                     'AND RE-ENABLE UNIT.
    CALL NTRAN (IU,22)

```

A-12

```

GO TO 267
265 CALL LINE (IFMT(1,I), JBUFF(1), L-1) 'REPORT CB DATA.
267 IF (ITYP.NE. ISC .OR. NSAV(NDET+1).NE. 'RESTORE ZEROS AT END OF
1 7777777) GO TO 400 'CB.
DO 268 J=1,4
268 JBUFF(J) = 0
IFTBL(3) = JDA + JTW
CALL NTRAN (IU,1,4,JBUFF,L)
269 IF (L+1.EQ. 0) GO TO 269
GO TO 400

C
270 WRITE (LPR,9006) NPOL(ISUB) 'ERROR - FILE ASSOC WITH AN
NPOL (IBEG-1+I) = 0 'SC FILE IS NOT TYPE CB.
GO TO 400

C
280 IF (ITYP.NE.IDF) GO TO 290

C
IF (IBUFF(1).EQ.0) GO TO 295 'DF FILE - VALUE IS DRUM
IFTBL(3) = IBUFF(1) 'ADDRESS. SET POINTER.
JTY = ITY
GO TO 252

C
290 WRITE (LPR,9007) NPOL(ISUB ),ITYP 'NPOL CONTAINS INVALID TYPE
NPOL (IBEG-1+I) = 0 'OF FILE.
GO TO 400

C
295 CALL LINE (IFMT(1,I), 16H NO DATA AVAIL , 3) 'NO DATA IN DF FILE.
GO TO 400

C
C THIS BLOCK IS PRINTING A MAILING ADDRESS FOR REVIEW OF AN
C INDIVIDUALS TOTAL DESCRIPTION. IT ASSUMES THAT SOME FILES
C HAVE A FIXED USAGE AS FOLLOWS
300 IBUFF(1)= 3 'FIRST NAME IS FILE NO. 3.
IBUFF(2)= 4 'FIRST INIT IS FILE NO. 4.
IBUFF(3)= 5 'SECOND INIT IS FILE NO. 5.
IBUFF(4)= 6 'LAST NAME IS FILE NO. 6 AND
IBUFF(5)= 0 'IS 3 WORDS LONG.
IBUFF(6)= 0
IBUFF(7)= 26 'BLDG NO. IS FILE NO. 26
IBUFF(8)= 25 'AREA NO. IS FILE NO. 25
NWDS = 8
GO TO 304
302 IBUFF(1) = 7 'ORGANIZATION
IBUFF(2) = 1 'PAYROL =
IBUFF(3) = 3 'FIRST NAME
IBUFF(4) = 4 'FIRST MIDDLE INITIAL
IBUFF(5) = 5 'SECOND MIDDLE INITIAL
IBUFF(6) = 6 'LAST NAME (3 WORDS LONG)

```

```

IBUFF(7) = 0
IBUFF(8) = 0
IBUFF(9) = 26
IBUFF(10) = 25
NWDS = 10
304 DO 310 K=1,NWDS
    IF (IBUFF(K).EQ.0) GO TO 310
    CALL DICT (IBUFF(K),2, NAME, ITYP,
1      IDA, NFORM, NWT,NCW, IR)
    IFTBL(3) = IDA +(INDEX-1)* NCW
    CALL NTRAN (IU,2,NCW,JBUFF(K),L)
305 IF (L+1) 310,305,310
310 CONTINUE
    IF (NPOL(IBEG-1+I) .EQ. -98) GO TO 320
    WRITE (LPR,9008) (JBUFF(K),K=1,NWDS)
    GO TO 400
320 PUNCH 9011 , (JBUFF(K),K=1,NWDS)
400 CONTINUE
    WRITE(LPR,9009)
    GO TO 210

C
410 WRITE (LPR,9010) NDET
    RETURN
9000 FORMAT ( 35X1H* / 35X1H* / 31H NO DETAILS SATISFY THIS QUERY )
9001 FORMAT ( 2(35X1H*/) , 54H ERROR - NO END OF SELECT FILE INDICATOR
1 IN NPOL ARRAY )
9002 FORMAT ( 2(35X1H*/) , 8H ERROR - , I6,30H IS AN INVALID REPORT FI
1LE NO. )
9003 FORMAT ( 2(35X1H*/) , 55H ERROR - NO END OF REPORT FILE INDICATOR
1IN NPOL ARRAY )
9004 FORMAT ( 2(35X1H*/) , 26H DRUM READ ERROR. FILE NO. , I6,
1 10H INDEX NO. , I6 )
9005 FORMAT ( 2(35X1H*/) , 37H MISSING CODE OR READ ERROR FOR TYPE ,
1 A2, 24HFILE ASSOC WITH FILE NO.,I6 / 8H INDEX = , I6 ,
2 7H CODE = , A6 )
9006 FORMAT ( 2(35X1H*/) , 17H ERROR - FILE NO. ,I6, 40H IS AN SC FILE
1BUT ASSOC FILE IS NOT CB. )
9007 FORMAT ( 2(35X1H*/) ,24H ERROR - REPORT FILE NO. , I6, 17H IS INVA
1LID TYPE , A2 )
9008 FORMAT ( 1H1/ 2(35X1H*/) , 10H MAIL TO' ,3A1,1X3A6/10XA6,5H BLDG/
1 10XA6, 5H AREA / 16X 5H BNW / 12 (35X1H*/)
2 60H THE FOLLOWING INFORMATION IS YOUR CURRENT DESCRIPTION ON /
3 60H THE BNW PERSONNEL DATA BANK. THIS INFORMATION MUST BE KEPT/
4 60H CURRENT AND ACCURATE FOR THE DATA BANK TO BE VALUABLE. /
5 60H PLEASE FORWARD ANY CORRECTIONS AND/OR ADDITIONS TO' /
6 20X 11HRL REYNOLDS / 20X 8H703 BLDG / 20X 8H700 AREA /
7 20X 3HBNW / 2(35X1H*/) )
9009 FORMAT ( 35X1H*)

```

'BUILDING
'AREA

'DICT RETURNS FILE DESC
'AND DRUM ADDRESS.
'SET POINTER.
'READ
'DELAY UNTIL READ IS COMPLETE.

'WRITE MAIL ADDRESS

'WRITE SPACE LINE.

'WRITE SUMMARY FOR QUERY

A-14

```

9010 FORMAT ( 4( 35X1H*//) 20XI6, 40H DETAILS SATISFIED THE PRECEDING
      IOUERY. )
9011 FORMAT (1XA6, 12XA6/// 1X3A1, 5X3A6 / 1XA6, 5H BLDG / 1XA6,5H AREA
      1 // 6X18HBATTELLE-NORTHWEST //)
9012 FORMAT (I10,10A6)
      END

```

```

      SUBROUTINE UNIQUE(ICO,ID,MAXP)
C
C      ICO      =      NUMBER OF UNIQUE FILES IN THIS QUERY
C
C      ID       =      BASIC BUFFER SIZE
C
C      MAXP     =      TOTAL NUMBER OF WORDS OF CW FILE
      INCLUDE COMN,LIST
      INCLUDE COMM,LIST
      DATA ICW /2HCW/
      IC = 1
      ISUM = 0
      ICWT = 0
      IS = 0
      IW = 0
      ITW = 0
      NWDS = 0
      DO 150 I = 1,NPOLM
C
C      CHECK FOR BOOLEAN OPERATORS
C
      IF(NPOL(I).LE.0) GO TO 150
C
C      CHECK FOR END OF STRING
C
      IF(NPOL(I).EQ.2222222) GO TO 155
C
C      NE IS INDEX INTO MOVE WHICH CONTAINS FILE NUMBER
C
      NE = NPOL(I)
      NFL = MQUE(1,NE)
C
C      CHECK FOR UNIQUE FILE NUMBERS
C
      DO 110 I1= 1,IC
      IF(NFL.EQ.MUNFL(1,I1)) GO TO 150
110 CONTINUE
C
C      SAVE  UNIQUE FILE NUMBERS
C

```

```

      MUNFL(1,IC) = NFL
C
C   GET INDEX INTO DIC FOR NFL
C
      CALL DICFND(NFL,INDEX)
      IF(INDEX.EQ.0)CALL ERROR(' *UNIQUE*UNDEFINED FILE NO      ',NFL, 2)
C
C   GET NUMBER WORD/CW FOR NFL
C
      FLD(24,12,IS) = FLD(24,12,DIC(8,INDEX))
C
C   SUM UP IS FOR ALL FILES
C
      ISUM = ISUM + IS
C
C   UPDATE FILE COUNTER
C
      IC = IC + 1
C
C   END OF STRING LOOP
C
150 CONTINUE
      CALL ERROR(' *UNIQUE*NPOL SETUP INCORRECTLY',IC,2)
C
C   NUMBER OF UNIQUE FILES
C
155 ICO = IC - 1
C
C   FIND CW FILE IN DIC
C
      DO 160 J1 = 1,MAXDIC
      IF(FLD(0,12,IDIC(1,J1)).EQ.FLD(0,12,ICW)) GO TO 170
160 CONTINUE
C   NO CW FILE
      N = MUNFL(1,1)
      CALL DICFND(N,INDEX)
      IF(INDEX.EQ.0) CALL ERROR(' *UNIQUE*UNDEFINED FILE NO.      ', N,2)
      FLD(24,12,IW) = FLD(24,12,DIC(8,INDEX))
      FLD(12,24,ITW) = FLD(0,24,DIC(8,INDEX))
      ICWT = ITW/IW
      GO TO 175
170 FLD(12,24,ICWT) = FLD(0,24,DIC(8,J1))
175 MAXP = ICWT
      ID = MAXCOR/ISUM
      ID = MIN0(ID,ICWT)
      ICCO = ICO - 1
      MUNFL(2,1) = 1
C

```


C ALLOCATE COR AREA

```

C
  DO 190 I =1,ICCO
    CALL DICFND(MUNFL(1,I),INDEX)
    IF(INDEX.EQ.0) CALL ERROR('*UNIQUE*UNDEFINED FILE NO. ',
1 MUNFL(1,I),2)
    FLD(24,12,NWDS) = FLD(24,12,DIC(8,INDEX))
    MUNFL(2,I+1) = NWDS*ID + MUNFL(2,I)
190 CONTINUE
C
C RETURN
C
200 CONTINUE
  RETURN
  END

```

```

  SUBROUTINE BOOL(M,IERR)
C
C   M      = 1 OR 0 (TRUE OR FALSE)
C
C   IERR   = 0 OR -1(NO ERROR OR ERROR)
C
C   THIS ROUTINE CHECKS POLISH STRING TO BE TRUE OR FALSE FOR EACH
C   COMMAND WORD
C
  INCLUDE COMN,LIST
  INCLUDE COMM,LIST
  M = 0
  NP = 0
  IERR = 0
C
C ARRAY NTP CONTAINS POLISH STRING
C
  DO 1001 I=1,NBOOL
C
C   NEGATIVE NUMBERS ARE BOOLEAN OPERATORS
C
  IF(NTP(I).LE.0) GO TO 901
C
C   CHECK FOR END OF POLISH STRING
C
  IF(NTP(I).EQ.2222222) GO TO 1010
C
C   PLACE TRUE OR FALSE VALUES(1 OR 2) IN STACK NPUS
C
  NP = NP + 1
  NPUS(NP) = NTP(I)

```

```

      GO TO 1001
C
C   .OR. OPERATOR CHECK
C
901 IF(NTP(I).NE.-1) GO TO 910
C
C   .OR. OPERATOR
C
      IF((NPUS(NP).EQ.1).OR.(NPUS(NP-1).EQ.1)) GO TO 906
C
C   .OR./.AND. FALSE
C
903 NP = NP - 1
      NPUS(NP+1) = 0
      NPUS(NP) = 2
      GO TO 1001
C
C   .OR./.AND. TRUE
C
906 NP = NP - 1
      NPUS(NP+1) = 0
      NPUS(NP) = 1
      GO TO 1001
C
C   .AND. OPERATOR CHECK
C
910 IF(NTP(I).NE.-2)GO TO 915
C
C   .AND. OPERATOR
C
      IF((NPUS(NP).EQ.1).AND.(NPUS(NP-1).EQ.1)) GO TO 906
      GO TO 903
C
C   CHECK FOR .NOT.
C
915 IF(NTP(I).NE.-3) GO TO 1005
C
C   .NOT. OPERATOR
C
      NPUS(NP) = IABS(NPUS(NP) - 3)
C
C   END OF STRING LOOP
C
1001 CONTINUE
C
C   ERROR
C
1005 IERR = -1

```

A-18

```

      GO TO 1030
C
C   CHECK STRING TRUE OR FALSE
C
1010 IF(NPUS(1).NE.1 ) GO TO 1030
      M = 1
1030 CONTINUE
      RETURN
      END

      SUBROUTINE COMPAR(IWDS,ITYPE,KK,N1,IND)
C
C   IWDS      =   NUMBER WORDS/CW
C
C   ITYPE     =   TYPE OF DATA(I,A,F,ORE)
C
C   KK        =   INDEX INTO ARRAY COR FOR A GIVEN FILE
C
C   N1        =   COLUMN INDEX INTO MQUE
C
C   IND       =   1 OR 2 IS RETURNED BY COMPAR FOR MATCH OR NO MATCH
C
      INCLUDE COMN,LIST
      INCLUDE COMM,LIST
      IRT = 0
      ICK = 0
C
C   PLACE RELATIONAL OPERATOR IN ICK
C
      FLD(18,18,ICK) = FLD(18,18,MQUE(2,N1))
C
C   BRANCH ON PROPER DATA TYPE
C
      GO TO (100,110,130,130),ITYPE
C
C   I-TYPE DATA
C
100 LC = 1
      GO TO 125
110 LC = 2
C
C   MAKE ARRAY TO ARRAY COMPARISON
C
125 CALL CMPAA(IWDS,IRT,LC,ICOR(KK),MQUE(3,N1))
      WRITE(6,10) IWDS,IRT,LC,ITYPE,ICK
10  FORMAT(1H0,5HCMPAA,5I5)
      GO TO 140

```

A-19

```

C
C F AND E TYPES
C
130 LC = 1
    CALL CMPAA(IWDS,IRT,LC,COR(KK),FMQUE(3,N1))
C
C BRANCH ON RELATIONAL OPERATOR
C
140 GO TO (150,160,170,180,190,195),ICK
C
C .LT. OPERATOR
C
150 IF( IRT.EQ.1) GO TO 200
    GO TO 202
C
C .GT.
C
160 IF(IRT.EQ.3) GO TO 200
    GO TO 202
C
C .EQ.
C
170 IF(IRT.EQ.2) GO TO 200
    GO TO 202
C
C .NE.
C
180 IF((IRT.EQ.1).OR.(IRT.EQ.3)) GO TO 200
    GO TO 202
C
C .LE.
C
190 IF((IRT.EQ.2) .OR.(IRT.EQ.1)) GO TO 200
    GO TO 202
C
C .GE.
C
195 IF((IRT.EQ.2).OR.(IRT.EQ.3)) GO TO 200
C
C NO MATCH
C
202 IND = 2
    RETURN
C
C MATCH
C
200 IND = 1
    RETURN

```

END

```

      SUBROUTINE DICT(I1,I2,I3,ITYPE,IDA,NFORM,NWT,NCW,IR)
C  *INPUT*
C   I1      =      FILE NAME(6-CHARACTERS) WHEN  I2 = 1   *INPUT*
C           =      FILE NUMBER                WHEN  I2 = 2   *INPUT*
C
C   I2      =      1 OR 2 ,INPUT INDICATOR                *INPUT*
C *OUTPUT*
C   I3      =      FILE NUMBER                WHEN  I2 = 1
C           =      FILE NAME                  WHEN  I2 = 2
C
C   ITYPE   =      FILE CODE (CW,SC,CB,SV,DF) -LEFT ADJUSTED WITH BLANKS
C
C   IDA     =      STARTING DRUM ADDRESS OF FILE
C
C   NFORM   =      CODED FORMAT FOR OUTPUTTING FILE CONTENTS
C
C   NWT     =      TOTAL NUMBER OF WORDS IN FILE
C
C   NCW     =      NUMBER OF WORDS PER CW
C
C   IR      =      REPORT CODE
C
      INCLUDE COMN,LIST
      INCLUDE COMM,LIST
      DATA ITY/6H      /
      ITYPE = ITY
      IRF = 0
      NWT = 0
      NCW = 0
      DO 100 I =1,MAXDIC
      GO TO (70,75),I2
      70 IF(IDIC(2,I).NE.I1) GO TO 100
      75 IRF = FLD(12,24,DIC(1,I))
C
C   REPORT NUMBER
C
      IR = MOD(IRF,10000)
C
C   FILE NUMBER
C
      I3 = (IRF- IR)/10000
      GO TO (85,80),I2
      80 IF(I3.NE.I1) GO TO 100
C
C   FILE NAME

```

```

      I3 = IDIC(2,I)
C
C   DRUM ADDRESS
C
      85 IDA = IDIC(7,I)
C
C   FILE CODE
C
      FLD(0,12,ITYPE) = FLD(0,12,IDIC(1,I))
C
C   FORMAT CODE
C
      NFORM = IDIC(6,I)
C
C   TOTAL WORDS IN FILE
C
      FLD(12,24,NWT) = FLD(0,24,IDIC(8,I))
C
C   WORDS PER CW
C
      FLD(24,12,NCW) = FLD(24,12,IDIC(8,I))
      RETURN
100 CONTINUE
C
C   ERROR
C
      IR = -1
      RETURN
      END

      SUBROUTINE UNCODE(NFOR,FORT1,FORT2,NWW,NWD)
C
C   NFOR   =   CODED FORMAT - CALLED NOUT IN ROUTINE DICT
C
C   FORT1   =   FIRST WORD OF OBJECT TIME FORMAT
C   FORT2   =   SECOND WORD OF OBJECT TIME FORMAT-INCLUDES COMMA
C
C   NWW     =   NUMBER OF WORDS (EQUALS -1,IF ERROR IN CODE )
C
C   NWD     =   WIDTH OF FIELD
C
      DATA AL1,AL2,AL3,BLK/6H   A6 ,6H,A   ,6HA   ,6H,   /,
1      FF1,FF2,FF3,XI/6HF   . ,6H F   . ,6H F   . ,6HI   /,
2      EE1,EE2,EE3 /6HE   . ,6H E   . ,6H E   . /
      NTYPE = 0
      NPD = 0
      NWW = 0

```

A-22

```

C
C  DECODE FORMAT
C
      FLD(24,12,NTYPE) = FLD(0,12,NFOR)      ' FORMAT TYPE CODE
      FLD(24,12,NWD)   = FLD(12,12,NFOR)     ' FIELD WIDTH
      FLD(24,12,NPD)   = FLD(24,12,NFOR)     ' DIGITS AFTER DECIMAL
C
C  CHECK FOR LEGAL FORMAT TYPE
C
      IF((NTYPE.LE.0).OR.(NTYPE.GT.4)) GO TO 400 ' ERROR RETURN
      GO TO (100,200,300,303),NTYPE
C
C  DECODE I-TYPE FORMAT
C
100  FORT1 = XI
C
C  CONVERT INTEGER TO FIELD DATA
C
      XT = FLDIC(NWD)
C
C  CHECK RANGE OF INTEGER
C
      IF(NWD.LT.10) GO TO 130
      IF(NWD.LT.100) GO TO 120
C
C  I - RANGE (100-999)
C
      FLD(6,18,FORT1) = FLD(18,18,XT)
      GO TO 140
C  I - RANGE (10-99)
C
120  FLD(6,12,FORT1) = FLD(24,12,XT)
      GO TO 140
C  I -RANGE (1-9)
C
130  FLD(6,6,FORT1) = FLD(30,6,XT)
C
140  CONTINUE
      NWW = 1
      FORT2 = BLK
      RETURN
C
C  DECODE A-TYPE FORMAT
C
200  CONTINUE
      IF( NWD .GT. 6) GO TO 210
C
C  6-CHARACTERS OR LESS

```

•

—

1

10


```

C
C  DECODE  F - TYPE FORMAT
C
  300 T1 = FF1
      T2 = FF2
      T3 = FF3
      GO TO 310
C
C  DECODE  E- TYPE FORMAT
C
  303 T1 = EE1
      T2 = EE2
      T3 = EE3
C
C  CONVERT  I  TO FIELD DATA
C
  310 TC1 = FLDIC(NWD)
      TC2 = FLDIC(NPD)
C
      IF(NWD.LT.10) GO TO 370
      IF(NWD.LT.100) GO TO 340
C
C  FIELD BTWN  100- 999
C
      FORT1 = T1
C
C  GXXX.X
C
      FLD(6,18,FORT1) = FLD(18,18,TC1)
      GO TO 380
C
C
      FLD(12,12,FORT1) = FLD(24,12,TC1)
      GO TO 380
  370 FORT1 = T3
C
C  GX.X
C
      FLD(18,6,FORT1) = FLD(30,6,TC1)
C
  380 FLD(30,6,FORT1) = FLD(30,6,TC2)
      FORT2 = BLK
      NWW = 1
      RETURN
C
C  FORMAT TYPE CODE NOT DEFINED
C
  400 NWW = -1

```

' FIELD WIDTH
' DIGITS AFTER DEC.

' UNDEFINED FORMAT TYPE

RETURN
END

```

SUBROUTINE DICFND(IFLNO,INDEX)
INCLUDE COMN,LIST
C**  GIVE THE EXTERNAL FILE NUMBER FIND THE CORRECT SECOND INDEX
C**      IN DIC
      DO 10 I=1,MAXDIC
      IF ( IFLNO.EQ.(FLD(12,24,IDIC(1,I))/10000) ) GO TO 20
10 CONTINUE
C**  NO MATCHING FILE NUMBER
      INDEX = 0
      RETURN
C**  SAVE MATCHING INDEX INTO FILE
20 INDEX = I
      RETURN
      END

```

' ELT FOURSL,1,690318, 13436

```

XRD EQU 10
XRSUB EQU 11
B11 EQU 11
AR1 EQU 12
AR2 EQU 13
AR3 EQU 14
A0 EQU 12
A1 EQU 13
A2 EQU 14
A3 EQU 15
A4 EQU 16
R1 EQU 65
XRA EQU A1
XRB EQU A2
XRC EQU A3

```

```

.....BLOCK TO BLOCK COMPARE....E1.....E1 00
$(1)
CMPAA* LX XRA,(1*/18)          1 TO DELTA XRA,XRB          E1 01
      LXM XRA, 3,XRSUB          A(J)                      E1 02
      LX XRC,*0,XRSUB          N
      LX XRB,XRA                      E1 04
      ANX XRC,1,0,016          N-1                        E1 05
      LXM XRB, 4,XRSUB          B(K)                      E1 06
      LA AR1,0,*XRA                      E1 07
      TE AR1,0,*XRB                      E1 08
      J $+5                      UNEQUAL                   E1 09
      JGD XRC,$-3                LOOP                      E1 10

```

A-26

```

LA  AR1,2,0,016          EQUAL JD=2          E1  11
SA  AR1,*1,XRSUB          E1  12
J   6,XRSUB
    LX      XRA,*2,XRSUB      XRA=1/2
    TLEM    XRA,2,0,016
    J       AR12              XRA=1 ARITH
JN  AR1,ARI2-2            XRA 2 LOGICAL
TN  -0,XRB                A POS IS B POS
J   AR12                  BOTH POSITIVE
J   ARI2+2                . A POS. IS B POSITIVE...
TN -0,XRB                  A NEG
J   ARI2+1
ARI2 TG      AR1,-0,XRB          XXX
    AX      XRA,2,0,016
    L       A0,ARIT3-1,A1.
    SA      AR1,*1,XRSUB        STORE JD
    J       6,XRSUB            RETURN
.....BLOCK TO LIST COMPARE.....E2.....E2  00
A5  EQU 17
CMPAL* . (N,J,LC,A,C1,C2,...CN)
    L       A1,(1,0)
    L,14    B11,0,B11 .
    L       A2,*0,B11 .
    L       A5,A2 . SAVE N FOR LOOP
    A       A2,B11 .
    A       B11,A1 . PUT 1 IN LEFT HALF OF B11
    LXM     A1,3,B11 . SET ARRAY ADDRESS
    L       A3,B11 . SAVE CALL TO SET J
    AN,14   A5,1 .
    L       A4,0,*A1 . LOAD WORD FROM ARRAY
    TE      A4,*4,*B11 . IS IT EQUAL TO CORRESPONDING WD IN LIST
    J       $+5 .
    JGD     A5,$-3 .
    L,14    A0,2 . ALL COMPARE EQUAL.SET J TO 2
    S       A0,*1,A3
    J       5,A2 . RETURN
    L       A1,*2,A3 . A1 =1 OR 2. TYPE OF COMPARISON
    TLEM,14 A1,2 .
    J       ARIT2 . 1 ARITH
    JN      A4,ARIT2-2 . 2 LOGICAL
    TN      *3,B11 . A POS. IS B POS
    J       ARIT2 . BOTH POS
    L       A0,ARIT3-1,A1 .
    S       A0,*1,A3 .
    J       5,A2 . RETURN
    TN      *3,B11 . A NEG. IS B NEG...
    J       $+2 .
    ARIT2   TG      A4,*3,B11 .

```

A-27

```

A      A1,2,0,14
L      A0,ARIT3-1,A1 .
S      A0,*1,A3      .
J      5,A2          . RETURN
.....LIST TO LIST COMPARE.....E3.....E3      00
CMPLL* L      A3,*0,B11 . PICK UP N
      L      A1,A3      . SAVE N FOR LOOP
      L,14    B11,0,B11
      SSC     A3,35      . CALC 2*N
      L      A2,B11      .
      A      A3,B11      . CALC RETURN
      A      B11,(2,0)   .
      AN,14   A1,1       .
      L      A0,*3,B11   .
      TE      A0,*4,*B11 .
      J      $+5         .
      JGD     A1,$-3     .
      L,14    A0,2        .
      S      A0,*1,A2    .
      J      4,A3        .
      L      A1,*2,A2    . GET J
      TLEM,14 A1,2       .
      J      ARITH2      .
      JN      A0,ARITH2-2 .
      TN      *2,B11      .
      J      ARITH2      .
      L      A0,ARIT3-1,A1 .
      S      A0,*1,A2    .
      J      4,A3        .
      TN      *2,B11      .
      J      $+2         .
ARITH2 TG      A0,*2,B11 .
      A,14    A1,2        .
      L      A0,ARIT3-1,A1 .
      S      A0,*1,A2    .
      J      4,A3        .
$(0) .
ARIT3  +1 .
      +1 .
      + 03 .
      + 03 .
.....CONSTANT TO BLOCK COMP....E4.....E4      00
$(1) .
CMPKA* .
      LX      A3,(1*/18)      1 TO DELTA XRA
      LXM     A3,3,XRSUB      A(J)
      LXM     XRA, 3,XRSUB     A(J)
      LR      R1,*0,XRSUB      E4      02

```

A-28

```

LA AR2,1,0,016          FOR JD STORE          E4  04
  LA AR1,*2,XRSUB
  SE AR1,0,*A3
AA AR2,1,0,016          NOT E8DJ3             E   07
  SA AR2,*1,XRSUB
  J   5,XRSUB
. ....CONSTANT TO LIST COMPARE.E5.....E5  00
CMPKL* .
  LX XRSUB,,XRSUB,016
  AX XRSUB,(1,0)
  L   A3,*0,B11 .
  L   R1,A3 .
  A   A3,B11 .
  JGD R1,$+1
LX XRA,XRSUB          SAVE FOR JD          E5  02
  LA,016 A4,1          FOR JD STORE
  LA AR1,*2,XRSUB
  TNE AR1,*3,*XRSUB    LIST RECORD EQUAL CONST
  J   $+3
  JGD R1,$-2          LOOP
  AA,016 A4,1          NOT EQUAL
  SA A4,*1,XRA          JD
  J 4,A3
END

```

```

' ELT FINDKA,1,690318, 13438
FINDKA* LX 15,K .
  LXM 15,3,11
  LR 65,*0,11
  SZ *1,11
  LA 13,65
  LA 12,*2,11
  SE 12,0,*15
  J 5,11
  ANA 13,65
  SA 13,*1,11
  J 5,11
$(1).
K + 1,0 .
  END

```

```

' ELT FINDKL,1,690318, 13440
B11 EQU 11
$(0)
FINDKL* L,14 B11,0,B11 .
  A 11,K .

```

```

      L      15,*0,11      .
      L      13,11        .
      L      65,15        .
      A      15,11        .
      SZ      *1,11        .
      JGD     65,$+1       .
      L      12,*2,11      .
      TNE     12,*3,*11    .
      J      $+3           .
      JGD     65,$-2       .
      J      4,15          .
      AN      11,13        .
      S      11,*1,13      .
      J      4,15          .
$ (1)
K      +      01,0
      END

```

```

SUBROUTINE FILIN
INCLUDE COMN,LIST
INCLUDE COMM,LIST
DIMENSION C(MAXCOR),IC(MAXCOR)
EQUIVALENCE (C,IC),(C,COR)
IFLGTH = 1792*100 - (150*28)
CALL OPENF(IUF1,0,150*28,IFASTB(1),0)
CALL OPENF(IUF2,150*28,IFLGTH,IFASTB(8),0)
NWD = (MAXCOR / 28) * 28
C** READ DICTIONARY, ETC. FROM 1ST LOGICAL
CALL NTRAN(IUF1,2,4064,C,L)
20 IF (L+1) 30,20,40
30 WRITE(6,35) L
35 FORMAT('TROUBLE READING 1ST LOGICAL FASTRAND FILE. STATUS = ',I2)
RETURN 0
40 I=0
DO 50 K=1,500
DO 50 J=1,8
I = I + 1
DIC(J,K) = C(I)
50 CONTINUE
DO 60 K=1,10
DO 60 J=1,6
I = I + 1
CBDIC(J,K) = C(I)
60 CONTINUE
I=I+1
MAXDIC = IC(I )
ISIZE = IC(I+1)

```

A-30

```

      IFIRSD = IC(I+2)
      NROWS = IC(I+3)
      ITSIZE = ISIZE
      IFTBL(3) = IFIRSD
70  IF (NWD.LT.ITSIZE) GO TO 80
      NWD = ITSIZE
      IEOFF = 1
80  CALL NTRAN(IUF2,2,NWD,C,L)
90  IF (L+1) 100,90,110
100 WRITE(6,105) L
105 FORMAT('1TROUBLE READING 2ND LOGICAL FASTRAND. STATUS = ',I2)
      RETURN 0
110 CALL NTRAN(IU,1,NWD,C,L)
120 IF (L+1) 130,120,140
130 WRITE(6,135) L
135 FORMAT('1TROUBLE WRITING 432. STATUS = ',I2)
      RETURN 0
140 ITSIZE = ITSIZE - NWD
      IF (IEOFF.EQ.0) GO TO 70
C**  DIC,CBDIC FILLED. ALL FASTRAND OUT TO 432.
      RETURN
      END

```

```

      SUBROUTINE ERROR(NER,LV,IGO)
      INCLUDE COMN,LIST
10  FORMAT(10H0**ERROR**, 5A6,6X,I10)
      WRITE(6,10)NER,LV
      IF(IGO.GT.1) CALL CONTRL
      RETURN
      END

```

```

      SUBROUTINE CBDICT(IFILE1,IFILE2)
      INCLUDE COMN,LIST
C**
C**  RETURNS FILE NUMBER OF CB FILE (IFILE2) ASSOCIATED WITH GIVE
C**    SV FILE (IFILE1).
C**
      IFILE2 = 0
      DO 10 K=1,10
      DO 10 J=1,5
      IF ( ICBDIC(J,K).EQ.IFILE1 ) GO TO 20
10  CONTINUE
      RETURN
20  IFILE2 = ICBDIC(6,K)
      RETURN
      END

```

```

      FUNCTION FLDIC(I)
      DIMENSION T(2)
      DATA BLK/1H /ISXTY/48/
      CALL TOD(T)
      WRITE(6,9998)T
      DUM=BLK
      WRITE(6,6000)I
      DO 30 J=1,6
      IF(I.LT.10**(J-1))GO TO 50
      K=MOD(I/10**(J-1),10)+ISXTY
      IN=(6-J)*6
      FLD(IN,6,DUM)=FLD(30,6,K)
      WRITE(6,6010)DUM,K
30    CONTINUE
50    FLDIC=DUM
      CALL TOD(T)
      WRITE(6,9999)T
      RETURN
6000  FORMAT(1H I5,45H WILL BE CONVERTED FROM INTEGER TO ALPHAMERIC  )
6010  FORMAT(1H 4HDUM=012,5X,2HK=012  )
9998  FORMAT(51H -----ENTERED ** FUNCTION    FLDIC    ** AT 2A6)
9999  FORMAT(51H -----LEFT ** FUNCTION    FLDIC    ** AT 2A6)
      END

```

```

      SUBROUTINE SETUP
      INCLUDE COMN,LIST
      INCLUDE COMM,LIST
      DATA NBLK/6H      /
      DO 101 LI=1,7
101    IFTBL(LI) = 0
      DO 200 LL=1,4
      DO 200 LLL=1,500
      NHEAD(LL,LLL) = NBLK
200    CONTINUE
      IU = 20
      CALL SETDR(IU,131073, 1179547,IFTBL)
      MAXCOR = 20000
      NPOLM  = 700
      NBOOL  = 450
      IBLSZ  = 2000
      IUF1   = 1
      IUF2   = 2
      CALL FILIN
      RETURN
      END

```



```

SUBROUTINE QIFACE
PARAMETER INDEX1=700,INDEX2=200,INDEX3=200,INDEX4=8,INDEX5=200
INCLUDE COMN,LIST
INTEGER OR,AND,SPACE,RP
INTEGER REPORT,SLASH,SEARCH,COMMA
INTEGER CW,SV,SC,CB,FMT
COMMON/INPT/ IWRD(INDEX2), IPDS(INDEX3), IPOL(INDEX1),
1 IOP(50), IR(50), IRT(50), IFILE(100),INAME(50),
2 NUM(10), ICA(80), NF, LP, RP, NCP, IDS, IQ,
3 IEX, IPER, ISC, ICOL, IC, IK, N, AND, OR, NOT,
4 SPACE, REPORT, COMMA, SLASH, SEARCH, IB6
5 ,NW ,KINDEX, MINDEX
DIMENSION FMT(22)
DIMENSION T(2)
DATA (FMT(I),I=1,22)/6H( ,20*6H ,6H ) /
DATA NPOLIN/0/
DATA CW,SV,SC,CB/2HCW,2HSV,2HSC,2HCB /
CALL TOD(T)
WRITE(6,9998)T
J2=2
J3=3
I=1
1 CONTINUE
IW=IWRD(I)
IF(IW.EQ.REPORT)GO TO 107
IF(FLD(0,6,IWRD(I)).EQ.FLD(0,6,ICOL))GO TO 100
IF(FLD(0,6,IW).EQ.FLD(0,6,SPACE))GO TO 105
IF(FLD(0,6,IW).EQ.FLD(0,6,LP))GO TO 7
IF(FLD(0,6,IW).EQ.FLD(0,6,RP))GO TO 8
5 CONTINUE
IF(FLD(0,12,IW).EQ.FLD(0,12,OR))GO TO 10
IF(FLD(0,18,IW).EQ.FLD(0,18,AND))GO TO 50
IF(FLD(0,18,IW).EQ.FLD(0,18,NOT))GO TO 90
IF(FLD(0,6,IW).EQ.FLD(0,6,IDS))GO TO 107
IW=IB6
GO TO 105
7 CONTINUE
C A LEFT PARENTHESIS HAS BEEN ENCOUNTERED SO PUT IT ON THE PUSH-DOWN
C STACK. (-4=CODE FOR LEFT PAREN)
KINDEX=KINDEX+1
IPDS(KINDEX)=-4
MINDEX=MINDEX+1
NPOLIN=NPOLIN+1
IPOL(MINDEX)=NPOLIN
IF(IPDS(KINDEX-1).NE.100)GO TO 105
IPDS(KINDEX-1)=-1

```

```

      MNDEX=MNDEX+1
      NPOLIN=NPOLIN+1
      IPOL(MNDEX)=NPOLIN
      GO TO 105
8     CONTINUE
C   A RIGHT PARENTHESIS HAS BEEN ENCOUNTERED SO UNSTACK PUSH-DOWN STACK
C   DOWN TO THE LEFT PARENTHESIS PUTTING THE OPERATORS IN THE POLISH
C   STRING AND THEN ELIMINATE THE LEFT PARENTHESIS IN THE PUSH DOWN
C   STACK.
      IF(IPDS(KNDEX).EQ.-4) GO TO 9
      MNDEX=MNDEX+1
      IPOL(MNDEX)=IPDS(KNDEX)
      IPDS(KNDEX)=0
      KNDEX=KNDEX-1
      GO TO 8
9     CONTINUE
      IPDS(KNDEX)=0
      KNDEX=KNDEX-1
      GO TO 105
10    CONTINUE
      IW=IB6
      IWRD(I)=IB6
C   IS THERE AN AND, NOT, OR OR ON THE PUSH-DOWN STACK0
      IF(IPDS(KNDEX).EQ.-1) GO TO 20
      IF(IPDS(KNDEX).EQ.-2) GO TO 30
      IF(IPDS(KNDEX).EQ.-3) GO TO 40
C   THERE IS NOTHING ON THE PUSH-DOWN STACK OR A LEFT PARENTHESIS ON PUSH
C   -DOWN STACK.
      KNDEX=KNDEX+1
      IPDS(KNDEX)=-1
      MNDEX=MNDEX+1
      NPOLIN=NPOLIN+1
      IPOL(MNDEX)=NPOLIN
      GO TO 105
20    CONTINUE
C   PUSH-DOWN STACK CONTAINS AN OR SO PUT OLD OR IN POLISH STRING AND PUT
C   NEW OR IN PUSH-DOWN STACK.
      MNDEX=MNDEX+1
      IPOL(MNDEX)=-1
      GO TO 105
30    CONTINUE
C   PUSH-DOWN STACK CONTAINS AN AND SO THE AND WILL GET UNSTACKED AND PUT
C   IN THE POLISH STRING AND THE OR WILL BE PUT ON THE PUSH-DOWN STACK.
      MNDEX=MNDEX+1
      NPOLIN=NPOLIN+1
      IPOL(MNDEX)=NPOLIN
      MNDEX=MNDEX+1
      IPOL(MNDEX)=-2

```

```

        IPDS(KNDEX)=-1
        GO TO 105
40    CONTINUE
C    PUSH-DOWN STACK CONTAINS A NOT SO THE NOT WILL BE PUT IN THE POLISH
C    STRING AND THE OR WILL BE PUT ON THE PUSH-DOWN STACK.
        MNDEX=MNDEX+1
        IPOL(MNDEX)=-3
        IPDS(KNDEX)=-1
        GO TO 105
50    CONTINUE
        IW=IB6
        IWRD(I)=IB6
C    IS THERE AN AND, NOT, OR OR ON THE PUSH-DOWN STACK NOW?
        IF(IPDS(KNDEX).EQ.-3) GO TO 80
        IF(IPDS(KNDEX).EQ.-1) GO TO 60
        IF(IPDS(KNDEX).EQ.-2) GO TO 70
C    NOTHING IS ON PUSH DOWN STACK SO WILL PUT AND ON STACK.
        KNDEX=KNDEX+1
        IPDS(KNDEX)=-2
        MNDEX=MNDEX+1
        NPOLIN=NPOLIN+1
        IPOL(MNDEX)=NPOLIN
        GO TO 105
60    CONTINUE
C    PUSH-DOWN STACK CONTAINS AN OR SO WILL STACK AND ON P.DS.
        KNDEX=KNDEX+1
        IPDS(KNDEX)=-2
        MNDEX=MNDEX+1
        NPOLIN=NPOLIN+1
        IPOL(MNDEX)=NPOLIN
        GO TO 105
70    CONTINUE
C    PUSH-DOWN STACK CONTAINS AND AND SO WILL PUT OLD AND IN POLISH STRING
C    AND NEW AND ON PUSH-DOWN STACK.
        MNDEX=MNDEX+1
        NPOLIN=NPOLIN+1
        IPOL(MNDEX)=NPOLIN
        MNDEX=MNDEX+1
        IPOL(MNDEX)=-2
        GO TO 105
80    CONTINUE
C    PUSH-DOWN STACK CONTAINS NOT SO UNSTACK NOT AND PUT IN POLISH STRING
C    AND PUT AND IN PUSH DOWN STACK
        MNDEX=MNDEX+1
        IPOL(MNDEX)=-3
        IPDS(KNDEX)=-2
        GO TO 105
90    CONTINUE

```

```

      IW=IB6
      IWRD(I)=IB6
C   THE NOT WILL GET PUT ON THE PUSH-DOWN STACK UNDER ALL CONDITIONS.
      KINDEX=KINDEX+1
      IPDS(KINDEX)=-3
      MNDEX=MNDEX+1
      NPOLIN=NPOLIN+1
      IPOL(MNDEX)=NPOLIN
      GO TO 105
100  CONTINUE
      K=0
      DO 102 J=1,6
      M=IABS(6*(J-1))
      IF(FLD(M,6,IWRD(I)).EQ.FLD(0,6,SPACE))GO TO 103
      K=K+1
102  CONTINUE
103  I=I+K
105  I=I+1
      IF(I.LE.NW)GO TO 1
107  CONTINUE
C   CHECK OUT STATEMENTS FOLLOW
      WRITE(6,6667) (IPDS(I),I=1,KINDEX)
6667  FORMAT(1H ,12HIPDS ARRAY =,12I9)
C   AND CHECK OUT STATEMENTS
      IF(MNDEX.EQ.0)MNDEX=1
      IF(MOD(NPOLIN,2).NE.0.AND.KINDEX.GT.1.AND.IPDS(KINDEX).GT.-3)GOTO108
      GO TO 109
108  CONTINUE
      MNDEX=MNDEX+1
      NPOLIN=NPOLIN+1
      IPOL(MNDEX)=NPOLIN
109  CONTINUE

      DO 110 I=KINDEX,1,-1
      IF(IPDS(I).EQ.100)GO TO 110
      IF(IPDS(I).EQ.-4)GO TO 110
      MNDEX=MNDEX+1
      IPOL(MNDEX)=IPDS(I)
110  CONTINUE
115  CONTINUE
117  DO 120 I=1,MNDEX
      NPOL(I)=IPOL(I)
120  CONTINUE
125  NPOL(MNDEX+1)=2222222

```

```

      MNDEX=MNDEX+2
      MN=MNDEX+NF-1
      KN=0
      DO 130 I=MNDEX,MN
      KN=KN+1
      NPOL(I)=IFILE(KN)
130  CONTINUE
      NPOL(MN+1)=7777777
C   CHECK OUT STATEMENTS FOLLOW
      MN=MN+1
      WRITE(6,6666) (NPOL(IJK),IJK=1,MN)
6666  FORMAT(1H ,12HNPOL ARRAY =,12I9)
C   END CHECKOUT STATEMENTS
      KNT=0
      IP=0
      DO 140 I=1,100
      IFILE(I)=0
140  CONTINUE
      I=1
      IL=0
      IJK=0
150  CONTINUE
      IF(IWRD(I).EQ.REPORT)GO TO 575
      IF(FLD(0,6,IWRD(I)).EQ.FLD(0,6,SLASH))GO TO 550
      WRITE(6,7777)I,IWRD(I)
      ICC=IWRD(I)
      IF(FLD(0,6,ICC).EQ.FLD(0,6,LP).OR.FLD(0,6,ICC).EQ.FLD(0,6,RP))
1ICC=IB6
      IW=ICC

```

```

      IW=IB6
      IWRD(I)=IB6
C   THE NOT WILL GET PUT ON THE PUSH-DOWN STACK UNDER ALL CONDITIONS.
      KNDEX=KNDEX+1
      IPDS(KNDEX)=-3
      MNDEX=MNDEX+1
      NPOLIN=NPOLIN+1
      IPOL(MNDEX)=NPOLIN
      GO TO 105
100  CONTINUE
      K=0
      DO 102 J=1,6
      M=IABS(6*(J-1))
      IF(FLD(M,6,IWRD(I)).EQ.FLD(0,6,SPACE))GO TO 103
      K=K+1
102  CONTINUE
103  I=I+K
105  I=I+1
      IF(I.LE.NW)GO TO 1
107  CONTINUE
C   CHECK OUT STATEMENTS FOLLOW
      WRITE(6,6667) (IPDS(I),I=1,KNDEX)
6667  FORMAT(1H ,12HIPDS ARRAY =,12I9)
C   AND CHECK OUT STATEMENTS
      IF(MNDEX.EQ.0)MNDEX=1
      IF(MOD(NPOLIN,2).NE.0.AND.KNDEX.GT.1.AND.IPDS(KNDEX).GT.-3)GOTO108
      GO TO 109
108  CONTINUE
      MNDEX=MNDEX+1
      NPOLIN=NPOLIN+1
      IPOL(MNDEX)=NPOLIN
109  CONTINUE
      IF(ICC.EQ.IB6)GO TO 550
      IF(FLD(0,6,ICC).EQ.FLD(0,6,ICOL))GO TO 400
      DO 200 M=1,NOP
      IF(IW.EQ.IOP(M)) GO TO 250
200  CONTINUE
      GO TO 300
250  CONTINUE
      IP=1
      IWRD(I)=IB6
      KNT=KNT+1
      IR(KNT)=IRT(M)
275  IW=IB6
      GO TO 550
300  CONTINUE
      IF(IP.GT.0)GO TO 500
      DO 310 J=1,NOP

```

A-38

```

        IF(IWRD(I+1).EQ.IOP(J))GO TO 330
310  CONTINUE
        ISP=0
        DO 325 J=1,6
        M=IABS(6*(J-1))
        IF(FLD(M,6,IW).NE.FLD(0,6,SPACE))GO TO 325
        ISP=ISP+1
        IF(ISP.LT.2)GO TO 325
        MM=IABS(6*(ISP-2))
        FLD(M,6,IW)=FLD(MM,6,IWRD(I+1))
325  CONTINUE
330  CONTINUE
        CALL DICT(IW,1,I3,ITYPE,IAD,IFOR,IW1,IW2,IRPT)
        WRITE(6,7778)IW,I3,ITYPE,IAD,IFOR,IW1,IW2,IRPT
        IF(IRPT.EQ.-1)GO TO 350
        IT=1
        IF(FLD(0,12,ITYPE).EQ.FLD(0,12,CW))IT=2
        IF(FLD(0,12,ITYPE).EQ.FLD(0,12,SC))IT=3
        IF(FLD(0,12,ITYPE).EQ.FLD(0,12,CB))IT=4
        IF(FLD(0,12,ITYPE).EQ.FLD(0,12,SV))IT=5
        IF=FLD(0,12,IFOR)
        WRITE(6,7779)IT,IF
C..  IF=1 FOR INTEGER, IF=2 FOR A, IF=3 FOR F, IF=4 FOR E
C..  IT=1 FOR DF, IT=2 FOR CW, IT=3 FOR SC, IT=4 FOR CB, IT=5 FOR SV
        IJK=IJK+1
        IFILE(IJK)=I3
        GO TO 275
350  CONTINUE
        IL=IL+1
        INAME(IL)=IW
        GO TO 275
400  CONTINUE
        IF(IP.GT.0)GO TO 550
        K=0
        DO 410 J=1,6
        M=IABS(6*(J-1))
        IF(FLD(M,6,IWRD(I)).EQ.FLD(0,6,SPACE))GO TO 415
        K=K+1
410  CONTINUE
415  I=I+1
        IW=IWRD(I)
        I=I+K-1
        GO TO 300
500  CONTINUE
        IF(IWRD(I).EQ.IB6)GO TO 550
        IP=0
        IF(IT.EQ.0)GO TO 550
        GO TO (510,515,520,530,540),IT

```

C.. DF TYPE

```

510 CONTINUE
    IP5=I+5
    DO 512 JJ=1,IP5
        IF(FLD(0,6,IWRD(JJ)).EQ.FLD(0,6,SPACE))GO TO 513
        IF(FLD(0,6,IWRD(JJ)).EQ.FLD(0,6,ICOL))GO TO 512
        IF(FLD(0,6,IWRD(JJ)).EQ.FLD(0,6,SLASH))GO TO 513
        IJK=IJK+1
        IFILE(IJK)=IWRD(JJ)
    512 CONTINUE
    513 IJK=IJK+1
        FLD(0,6,IFILE(IJK))=FLD(0,6,IDS)
        IJK=IJK+1
        IFILE(IJK)=0
        GO TO 550

```

C.. CW TYPE

```

515 CONTINUE
    GO TO 545

```

C.. SC TYPE

```

520 CONTINUE
    CALL CBDICT(IFILE(IJK),IFILE2)
    521 CALL DICT(IFILE2,2,I3,ITYPE,IAD,IFOR,IW1,IW2,IRPT )
        DO 522 JJ=1,10
            IF(FLD(0,6,IWRD(I)).EQ.FLD(0,6,NUM(JJ)))GO TO 545
    522 CONTINUE
    523 CONTINUE
        IFTBL(3)=IAD
        CALL NTRAN(IU,3,IW2,IPOL,L)
        IF(L.LT.-2)GO TO 527
        JJ=1
        DO 525 J=2,IW2
            K1=1
            L1=1
    524 CONTINUE
            M1=IABS(6*(K1-1))
            N1=IABS(6*(L1-1))
            IF(FLD(N1,6,IPOL(J)).EQ.FLD(0,6,COMMA))GO TO 529
            IF(FLD(M1,6,IWRD(JJ)).EQ.FLD(0,6,SPACE))GO TO 526
            IF(FLD(N1,6,IPOL(J)).EQ.FLD(0,6,SPACE))GO TO 528
            IF(FLD(M1,6,IWRD(JJ)).NE.FLD(N1,6,IPOL(J)))GO TO 523
            K1=K1+1
            L1=L1+1
            IF(K1.GT.6)JJ=JJ+1
            IF(K1.GT.6)K1=1
            IF(L1.GT.6)GO TO 525
        GO TO 524
    526 CONTINUE
        K1=K1+1

```



```

        IF(K1.LE.6)GO TO 524
        K1=1
        JJ=JJ+1
        GO TO 524
528 CONTINUE
        L1=L1+1
        IF(L1.GT.6)GO TO 525
        GO TO 524
525 CONTINUE
529 CONTINUE
        IW=IPOL(1)
        GO TO 545
527 CALL NTRAN(1,22)
        GO TO 550
C.. CB TYPE
530 CONTINUE
        IFILE2=IFILE(IJK)
        GO TO 521
C.. SV TYPE
540 CONTINUE
        IF(IF.EQ.2)GO TO 545
        CALL UNCODE(IFOR,FMT(J2),FMT(J3),NWW,NWD)
        J2=J2+2
        J3=J2+1
        CALL NDCODE(IW,1)
        LV=-4
        READ(LV,FMT)IW
545 IJK=IJK+1
        IFILE(IJK)=IW
        IN=1
        M=6
546 IF(FLD(M,6,IWRD(I-1)).EQ.FLD(0,6,ICOL))GO TO 548
        IJK=IJK+1
        IFILE(IJK)=0
        GO TO 550
548 M=M+6
        IJK=IJK+1
        IFILE(IJK)=IWRD(I+IN)
        IN=IN+1
        GO TO 546
550 CONTINUE
        I=I+1
        IF(I.LF.NW)GO TO 150
575 CONTINUE
        WRITE(6,7780)((IFILE(K),K=1,10),((IR(K),K=1,10)
C... GENERATE MQUE TABLE
        K=0
        KNT=0

```

```

DO 600 I=1,MNDEX
IF(NPOL(I).EQ.2222222)GO TO 700
IF(NPOL(I).LE.0)GO TO 600
KNT=KNT+1
K=K+1
MQUE(1,KNT)=IFILE(K)
MQUE(2,KNT)=IR(KNT)
DO 580 JJ=3,8
K=K+1
IF(IFILE(K).EQ.0)GO TO 600
MQUE(JJ,KNT)=IFILE(K)
580 CONTINUE
K=K+1
MQUE(3,KNT)=IFILE(K)
600 CONTINUE
700 CONTINUE
DO 800 I=1,3
800 WRITE(6,7000)(MQUE(I,J),J=1,KNT)
CALL TOD(T)
WRITE(6,9999)T
RETURN
7000 FORMAT(1H 10HMQUE ARRAY /(1X,10012))
7777 FORMAT(6H IWRD(I2,2H)=012)
7778 FORMAT(4H IW=012/4H I3=012/7H ITYPE=012/5H IAD=012/6H IFOR=012/
15H IW1=012/5H IW2=012/6H IRPT=012 )
7779 FORMAT(4H IT=012,5X,4H IF=012 )
7780 FORMAT(15H IFILE FOR MQUE /1X,10012/12H IR FOR MQUE /1X,10012 )
9998 FORMAT(51H -----ENTERED ** SUBROUTINE QIFACE ** AT 2A6)
9999 FORMAT(51H ----- LEFT ** SUBROUTINE QIFACE ** AT 2A6)
END

```

```

SUBROUTINE CHAR
PARAMETER INDEX1=700,INDEX2=200,INDEX3=200,INDEX4=8,INDEX5=200
INCLUDE COMN,LIST
INTEGER OR,AND,SPACE,RP
INTEGER REPORT,SLASH,SEARCH,COMMA
COMMON/INPT/ IWRD(INDEX2), IPDS(INDEX3), IPOL(INDEX1),
1 IOP(50), IR(50), IRT(50), IFILE(100),INAME(50),
2 NUM(10), ICA(80), NF, LP, RP, NOP, IDS, IQ,
3 IEX, IPER, ISC, ICOL, IC, IK, N, AND, OR, NOT,
4 SPACE, REPORT, COMMA, SLASH, SEARCH, IB6
5 ,NW ,KNDEX, MNDEX
DATA IK,IUNITR/0,5/
IF(IK)5,5,10
5 IK=0
READ(IUNITR,5000)(ICA(I),I=1,N)
WRITE(6,6010)

```

```

        WRITE(6,6000)(ICA(I),I=1,N)
        WRITE(6,6010)
10      IK=IK+1
        IF(IK-N)15,15,5
15      IC=ICA(IK)
        RETURN
5000    FORMAT(80A1)
6000    FORMAT(1X,80A1)
6010    FORMAT(1H040H***** )
        END

SUBROUTINE AGE(I)
    PARAMETER INDEX1=700,INDEX2=200,INDEX3=200,INDEX4=8,INDEX5=200
    INCLUDE COMN,LIST
    INTEGER OR,AND,RP,BEFORE,AFTER
    INTEGER REPORT,SLASH,SEARCH,COMMA
    COMMON/INPT/ IWRD(INDEX2), IPDS(INDEX3), IPOL(INDEX1),
1      IOP(50), IR(50), IRT(50), IFILE(100),INAME(50),
2      NUM(10), ICA(80), NF, LP, RP, NOP, IDS, IQ,
3      IEX, IPER, ISC, ICOL, IC, IK, N, AND, OR, NOT,
4      SPACE, REPORT, COMMA, SLASH, SEARCH, IB6
5      ,NW ,KINDEX, MINDEX
    DIMENSION IY(2),MON(12)
    DATA (MON(M),M=1,12)/3HJAN,3HFEB,3HMAR,3HAPR,3HMAY,3HJUN,
13HJUL,3HAUG,3HSEP,3HOCT,3HNOV,3HDEC /
    DATA MM,BEFORE,AFTER/6HBIRTH ,6HLT ,6HGT /
    CALL TOD(IY)
    WRITE(6,9998)IY
    IWRD(I)=MM
    CALL DOY(IY)
    IP1=I+1
    DO 10 J=IP1,100
    DO 10 K=1,NOP
    IF(IWRD(J).EQ.IOP(K))GO TO 15
10    CONTINUE
    WRITE(6,6000)
    RETURN
15    CONTINUE
    GO TO (20,20,17,17,18,18,17,17,20,18,20,18,20,20,20,17,18,20,18,17
1),K
17    IWRD(J)=AFTER
    GO TO 20
18    IWRD(J)=BEFORE
20    CONTINUE
    IP1=J+1
    DO 25 J=IP1,100
    DO 25 K=1,10

```

```

      IF (FLD(0,6,IWRD(J)).EQ.FLD(0,6,NUM(K)))GO TO 30
25  CONTINUE
      WRITE(6,6005)
      RETURN
30  CONTINUE
      IN=IWRD(J)
      WRITE(6,7030)IWRD(J)
      CALL BCD2I(IN)
      WRITE(6,7000)IN
      IM=IB6
      FLD(0,18,IM)=FLD(6,18,IY(2))
      CALL BCD2I(IM)
      WRITE(6,7010)IM
      IM=IM+100
      WRITE(6,7010)IM
      IM=IM-IN
      IF (IM.GE.100)IM=IM-100
      XM=FLDIC(IM)
      FLD(24,12,IWRD(J))=FLD(0,12,IY(1))
      WRITE(6,7015)XM
      DO 40 K=1,12
      IF (FLD(18,18,IY(1)).EQ.FLD(0,18,MON(K)))GO TO 45
40  CONTINUE
      WRITE(6,6010)
      RETURN
45  CONTINUE
      XK=FLDIC(K)
      WRITE(6,7020)XK
      FLD(12,12,IWRD(J))=FLD(24,12,XK)
      FLD(0,12,IWRD(J))=FLD(24,12,XM)
      WRITE(6,6015)IWRD(J)
      CALL TOD(IY)
      WRITE(6,9999)IY
      RETURN
6000 FORMAT(45H NO MATCH FOUND ON OPERATOR IN SUBROUTINE AGE  )
6005 FORMAT(40H NO MATCH FOUND ON AGE IN SUBROUTINE AGE  )
6010 FORMAT(42H NO MATCH FOUND ON MONTH IN SUBROUTINE AGE  )
6015 FORMAT(12H BIRTH DATE A6,19H GENERATED FROM AGE  )
7000 FORMAT(21H INTEGER VALUE OF AGE I5 )
7010 FORMAT(30H INTEGER VALUE OF PRESENT YEAR I5 )
7015 FORMAT(33H FLD DATA VALUE OF YEAR OF BIRTH A6 )
7020 FORMAT(25H FLD DATA VALUE OF MONTH A6 )
7030 FORMAT(14H AGE INPUT AS A2 )
9998 FORMAT(51H -----ENTERED ** SUBROUTINE AGE ** AT 2A6)
9999 FORMAT(51H ----- LEFT ** SUBROUTINE AGE ** AT 2A6)
      END

```

```

SUBROUTINE MOVE(I)
PARAMETER INDEX1=700,INDEX2=200,INDEX3=200,INDEX4=8,INDEX5=200
INCLUDE COMN,LIST
INTEGER OR,AND,SPACE,RP
INTEGER REPORT,SLASH,SEARCH,COMMA
COMMON/INPT/ IWRD(INDEX2), IPDS(INDEX3), IPOL(INDEX1),
1 IOP(50), IR(50), IRT(50), IFILE(100),INAME(50),
2 NUM(10), ICA(80), NF, LP, RP, NOP, IDS, IQ,
3 IEX, IPER, ISC, ICOL, IC, IK, N, AND, OR, NOT,
4 SPACE, REPORT, COMMA, SLASH, SEARCH, IB6 ,NW,KINDEX,MINDEX
DIMENSION T(2)
CALL TOD(T)
WRITE(6,9998)T
J=0
DO 100 K=1,6
M=IABS(6*(K-1))
IF(FLD(M,6,IWRD(1)).EQ.FLD(0,6,SPACE))GO TO 200
J=J+1
100 CONTINUE
200 CONTINUE
J=IABS(I-J)
ISAVE=IWRD(I)
JP=J+1
DO 300 K=I,JP,-1
IF(K.LT.2)GO TO 350
IWRD(K)=IWRD(K-1)
300 CONTINUE
350 CONTINUE
IWRD(J)=ISAVE
CALL TOD(T)
WRITE(6,9999)T
RETURN
9998 FORMAT(51H -----ENTERED ** SUBROUTINE MOVE ** AT 2A6)
9999 FORMAT(51H ----- LEFT ** SUBROUTINE MOVE ** AT 2A6)
END

```

```

SUBROUTINE BCD2I(IW)
PARAMETER INDEX1=700,INDEX2=200,INDEX3=200,INDEX4=8,INDEX5=200
INCLUDE COMN,LIST
INTEGER REPORT,SLASH,SEARCH,COMMA
COMMON/INPT/ IWRD(INDEX2), IPDS(INDEX3), IPOL(INDEX1),
1 IOP(50), IR(50), IRT(50), IFILE(100),INAME(50),
2 NUM(10), ICA(80), NF, LP, RP, NOP, IDS, IQ,
3 IEX, IPER, ISC, ICOL, IC, IK, N, AND, OR, NOT,
4 SPACE, REPORT, COMMA, SLASH, SEARCH, IB6 ,NW,KINDEX,MINDEX
DIMENSION T(2)
DATA MINUS/1H-/

```

```

EQUIVALENCE (IW,X)
CALL TOD(T)
WRITE(6,9998)T
K=1
XMM=0.
XNEG=1.
KK=1
DO 300 KKK=1,6
M=IABS(6*(KKK-1))
IF(FLD(M,6,IW).EQ.FLD(0,6,MINUS))GO TO 250
IF(FLD(M,6,IW).EQ.FLD(0,6,IPER))K=0
IF(K.LT.1)KK=KK-1
IF(FLD(M,6,IW).EQ.FLD(0,6,SPACE))GO TO 400
IF(FLD(M,6,IW).EQ.FLD(0,6,IPER))GO TO 300
DO 100 I=1,10
IF(FLD(M,6,IW).EQ.FLD(0,6,NUM(I)))GO TO 200
100 CONTINUE
200 CONTINUE
IF(I.EQ.10)I=0
XMM=XMM*10.**K+FLOAT(I*10**(KK-1))
GO TO 300
250 XNEG=-1.
300 CONTINUE
400 CONTINUE
IW=XMM*XNEG
IF(K.GT.0)GO TO 600
X=XMM*XNEG
600 CONTINUE
CALL TOD(T)
WRITE(6,9999)T
RETURN
9998 FORMAT(51H -----ENTERED ** SUBROUTINE BCD2I ** AT 2A6)
9999 FORMAT(51H ----- LEFT ** SUBROUTINE BCD2I ** AT 2A6)
END

```

```

SUBROUTINE LOGIC
C
C CALL UNIQUE FOR FILE AND CORE INDICES
C
INCLUDE COMN,LIST
INCLUDE COMM,LIST
DATA IDF/6HDF /
NSAV(1) = 7777777
MAXQ = 200
DO 115 L1=1,2
DO 112 L2=1,MAXQ
MUNFL(L1,L2) = 0

```

```

112 CONTINUE
115 CONTINUE
    CALL UNIQUE(ICO,ID,MAXP)
C
C   BASIC BUFFER O.K.
C
    IF(ID.LT.MAXP) GO TO 200
C
C   NUMBER OF DRUM READS NEEDED TO ALL INFOR INTO CORE
C
    NXXX = 1
    GO TO 210
200 IDC = MAXP/ID
    IF( MAXP.GT.IDC*ID) GO TO 205
    NXXX = IDC
    GO TO 210
205 NXXX = IDC+1
210 CONTINUE
C
C
C
    NIDS = 0
    MS = 0
    DO 999 LL = 1,NXXX
    NWD = 0
C   BASIC BUFFER SIZE
    ID = MIN0(ID,MAXP)
C
C   READ IN ALL FILES OF QUERY
C
    DO 99 II = 1,ICO
C
C   FILE NUMBER
C
    NFNU = MUNFL(1,II)
C
C   GET INDEX INTO DIC FOR NFNU
C
    CALL DICFND(NFNU,INDEX)
    IF(INDEX.EQ.0)CALL ERROR('LOGIC*FILE NO. UNDEFINED',NFNU,2)
C
C   NUMBER WORDS/CW
C
    FLD(24,12,NWD) = FLD(24,12,DIC(8,INDEX))
C
C   INITIAL CORE INDEX INTO ARRAY COR FOR FILE NFNU
C
    NN = MUNFL(2,II)

```

A-47

```
C
C STARTING DRUM ADDRESS FOR FILE NFNU
C
C   NDR = IDIC(7,INDEX)
C
C TOTAL NUMBER OF WORDS TO BE READ FROM DRUM TO FILL BUFFER
C
C   NS = ID * NWD
C
C DRUM ADDRESS AT WHICH READ WILL START FOR FILE NFNU
C
C   NDDR = NDR + (LL-1)*NIDS*NWD
C
C RESET IFTBL(3) FOR NTRAN READ
C
C   IFTBL(3) = NDDR
C
C READ A BLOCK FROM DRUM
C
C   CALL NTRAN(IU,2,NS,COR(NN),MSTAT)
C   40 IF(MSTAT+1)50,40,99
C
C READ ERROR
C
C   50 CALL ERROR('LOGIC*READ ERROR FROM DRUM ',MSTAT,2)
C   99 CONTINUE
C
C PROCESS ALL BUFFERS
C
C   IWDS = 0
C   ITYPE = 0
C   DO 500 JO=1,ID
C
C
C
C   DO 400 KO=1,NPOLM
C
C CHECK FOR BOOLEAN OPERATORS
C
C   IF(NPOL(KO).LE.0) GO TO 390
C
C END OF QUERY STRING CHECK
C
C   IF(NPOL(KO).EQ.2222222) GO TO 420
C
C INDEX INTO MQUE
C
C   N1 = NPOL(KO)
```



```

C
C FILE NUMBER
C
    NFL = MQUE(1,N1)
    DO 350 II =1,ICO
        IF(MUNFL(1,II).EQ.NFL) GO TO 370
    350 CONTINUE
C
C ERROR FILE IN MQUE NOT IN UNIQUE FILE LIST
C
    CALL ERROR('*LOGIC*FILE NO. NOT IN MUNFL ',NFL,2)
C
C MATCH FOUND/GET CORE INDEX
C
    370 ICORD = MUNFL(2,II)
        CALL DICFND(NFL,INDEX)
        IF(INDEX.EQ.0) CALL ERROR('*LOGIC* FILE NO. UNDEFINED ',NFL,2)
C
C NUMBER WORDS/CW
C
    FLD(24,12,IWDS) = FLD(24,12,DIC(8,INDEX))
C
C INDEX INTO ARRAY COR
C
    KK = ICORD + IWDS * (JO - 1)
    IF(FLD(0,12,IDIC(1,INDEX)).NE.FLD(0,12,IDF))GO TO 386
C
C
C
    CALL BOCOM(KK,N1,IND)
C
C
C
    GO TO 389
    386 CONTINUE
C
C FIND TYPE OF DATA (I,A,F,OR E)
C
    FLD(24,12,ITYPE) = FLD( 0,12,DIC(6,INDEX))
C
C COMPARE WORD(S) IN ARRAY COR WITH WORD(S) IN ARRAY MQUE
C
C
    CALL COMPAR(IWDS,ITYPE,KK,N1,IND )
C
C
C
    389 CONTINUE

```

```

      NTP(KO) = IND
      GO TO 400
390 NTP(KO) = NPOL(KO)
400 CONTINUE
C
C   END FOR ONE CW ITEM AND SAVE REPORT MARKER
C
      420 NTP(KO) = NPOL(KO)
C
C   PROCESS REVERSE POLISH FOR TRUE OR FALSE
C
      CALL BOOL(M,IERR)
      IF(IERR.EQ.-1)CALL ERROR(' *LOGIC*NPOL ARRAY INCORRECT ',NPOLM,2)
C
C   DID BOOLEAN EXPRESSION TURN OUT TO BE TRUE
C
      IF(M.NE.1) GO TO 500
C
C   SAVE CW INDEX INTO DRUM FILES
C
      MS = MS + 1
      NSAV(MS) = NIDS*(LL-1) + JO
C
C
      500 CONTINUE
C
C   REDUCE MAXIMUM BASIC FILE BY ONE BUFFER SIZE
C
      MAXP = MAXP - ID
      NIDS = ID
C
C
      999 CONTINUE
      NSAV(MS+1) = 7777777
C
C   END QUERY
C
      GO TO 1020
C
C
      1020 RETURN
      END

```

```

      SUBROUTINE BOCOM(KK,N1,IND)
C
C   BLOCK COMPAR FOR EQUALITY
C

```

A-50

```

      INCLUDE COMN,LIST
      INCLUDE COMM,LIST
      DATA NBLK,NDOL/6H      ,6H$      /
C
      IFTBL(3) = ICOR(KK)
C
C   BLOCK READ
C
      CALL NTRAN(IU,2,IBLSZ,NBOL,MST)
110 IF(MST + 1)100,110,120
100 CALL ERROR('*BOCOM*READ ERROR FROM DRUM      ',MST,2)
120 CONTINUE
C
      IKS = 2
      IK2 = MST-1
      III = 1
      IFS = 0
150 CONTINUE
      I1 = III
      DO 500 I=1,6
      IF = 0
C
C
C
      DO 450 J=1,6
      IK1 = IKS
C
C
C
      IF(FLD(IF,6,MQUE(I+2,N1)).EQ.FLD(0,6,NBLK)) GO TO 450
      IF(FLD(IF,6,MQUE(I+2,N1)).EQ.FLD(0,6,NDOL)) GO TO 550
C
C
C
      DO 350 IK= IK1,IK2
      IFF = IFS
C
C
C
      DO 250 I2 = I1,6
      IF(FLD(IFF,6,NBOL(IK)).EQ.FLD(0,6,NBLK)) GO TO 250
      IF(FLD(IF,6,MQUE(I+2,N1)).EQ.FLD(IFF,6,NBOL(IK))) GO TO 300
      I1 = I1
      IKS=IK
      IFS=IFF
      GO TO 150
250 IFF= IFF + 6
260 I1 = 1

```

A-51

```
      IFS = 0
      GO TO 350
300  IF(I2.EQ.6) GO TO 312
      I1 = I2 + 1
      IFS = IFF + 6
      IKS = IK
      GO TO 450
312  I1 = 1
      IFS = 0
      IKS = IK + 1
      GO TO 450
C
C
C
      350 CONTINUE
C
C NO MATCH
C
      GO TO 520
C
      450 IF = IF + 6
C
C
C
      500 CONTINUE
C
      520 IND = 2
      RETURN
C
C MATCH
C
      550 IND = 1
      RETURN
      END

      SUBROUTINE LINE (FMT, IBUFF, NCW)
      DIMENSION FMT(10), IBUFF(2)
      DATA LPR / 6/
      WRITE (LPR,FMT) (IBUFF(I),I=1,NCW)
      RETURN
      END
```


DISTRIBUTION

AEC, RL

C. L. Robinson

AEC, RDT Site Representative

(2) P. G. Holsted

AEC, Chicago Patent Group

G. H. Lee
R. K. Sharp

AEC, DTIE

(1) For UC-32 Distribution

(3) Battelle Memorial Institute

Battelle-Northwest

(5) Technical Information
(2) Technical Publications
(1) R. L. Engel
(2) R. L. Reynolds
(1) R. T. Toyooka
(1) E. L. Wells