บรรณานุกรม

ฝ่ายโปรแกรมระบบ สถาบันบริการคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย. Introduction to IBM Virtual Machine/System Product. 2525.

IBM Virtual Machine Facility/370: CMS Command and Macro Reference. IBM, 1979.

Polivka, Raymond P. and Sandra Pakin. APL: The Language and Its Usage. Englewood Cliffs, N.J.: Prentice-Hall, 1975.

Terminal Guide. IBM, 1981.

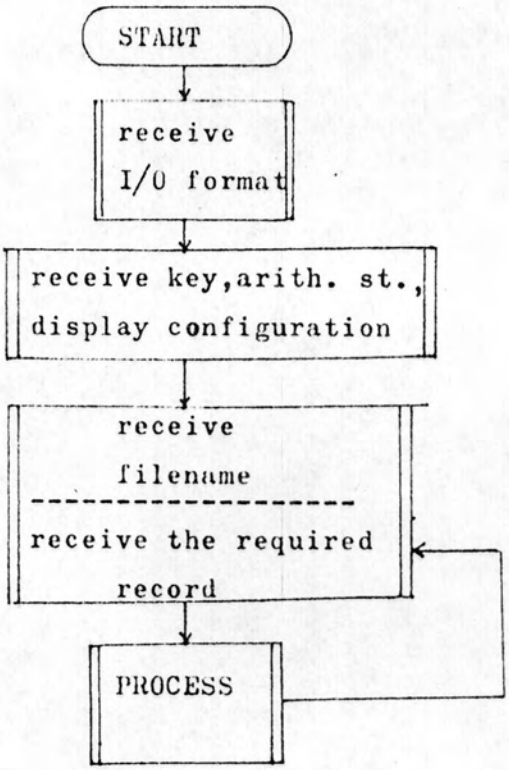
Using CMS Study Guide 1. IBM, 1981.

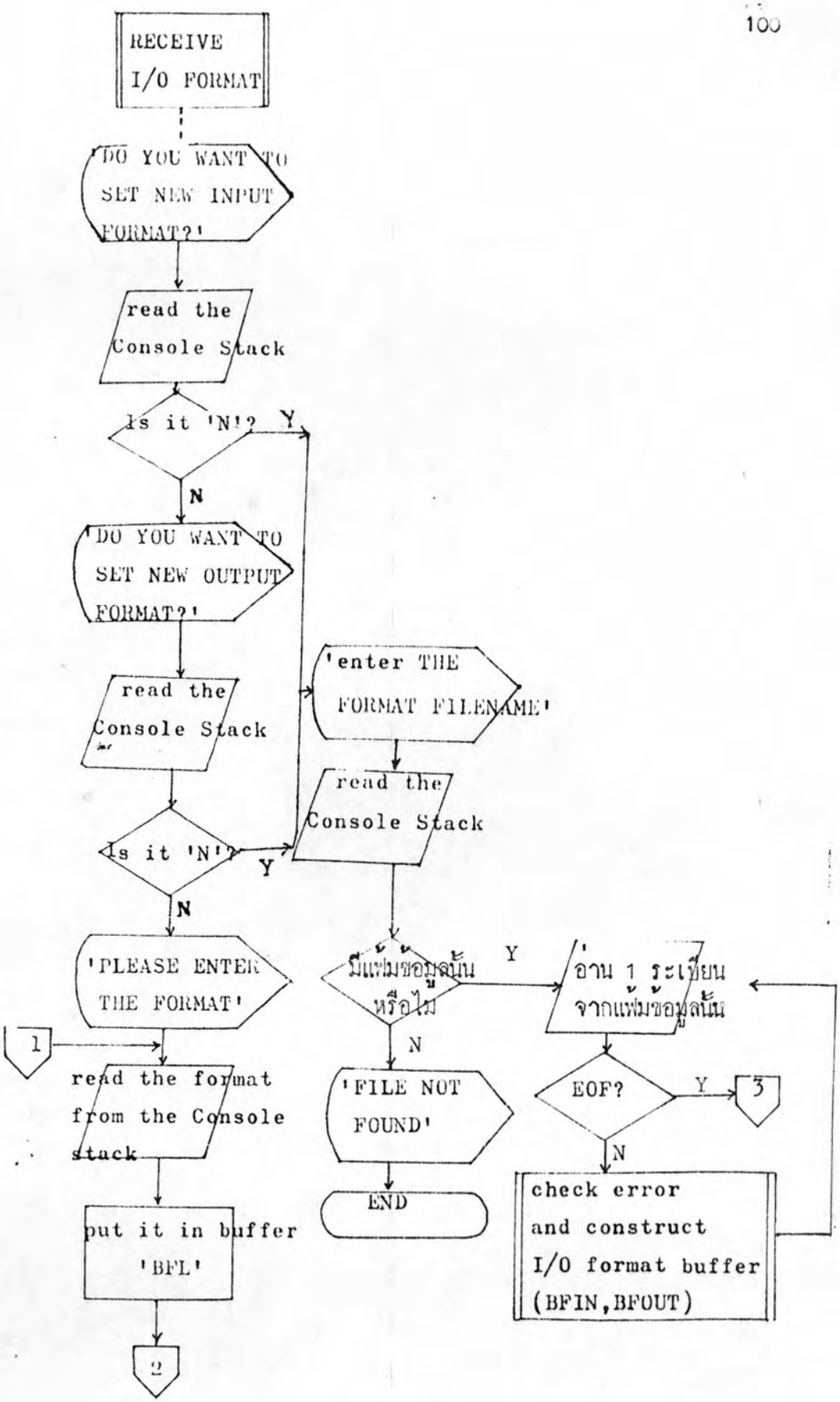
Using CMS Study Guide 2. IBM, 1981.

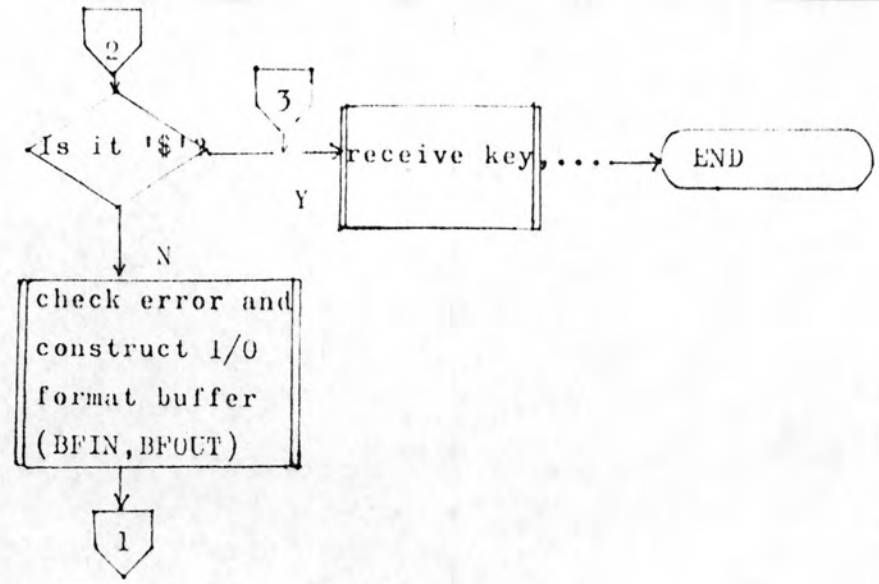
VM/370 Operations CMS for Operators. IBM, 1980.

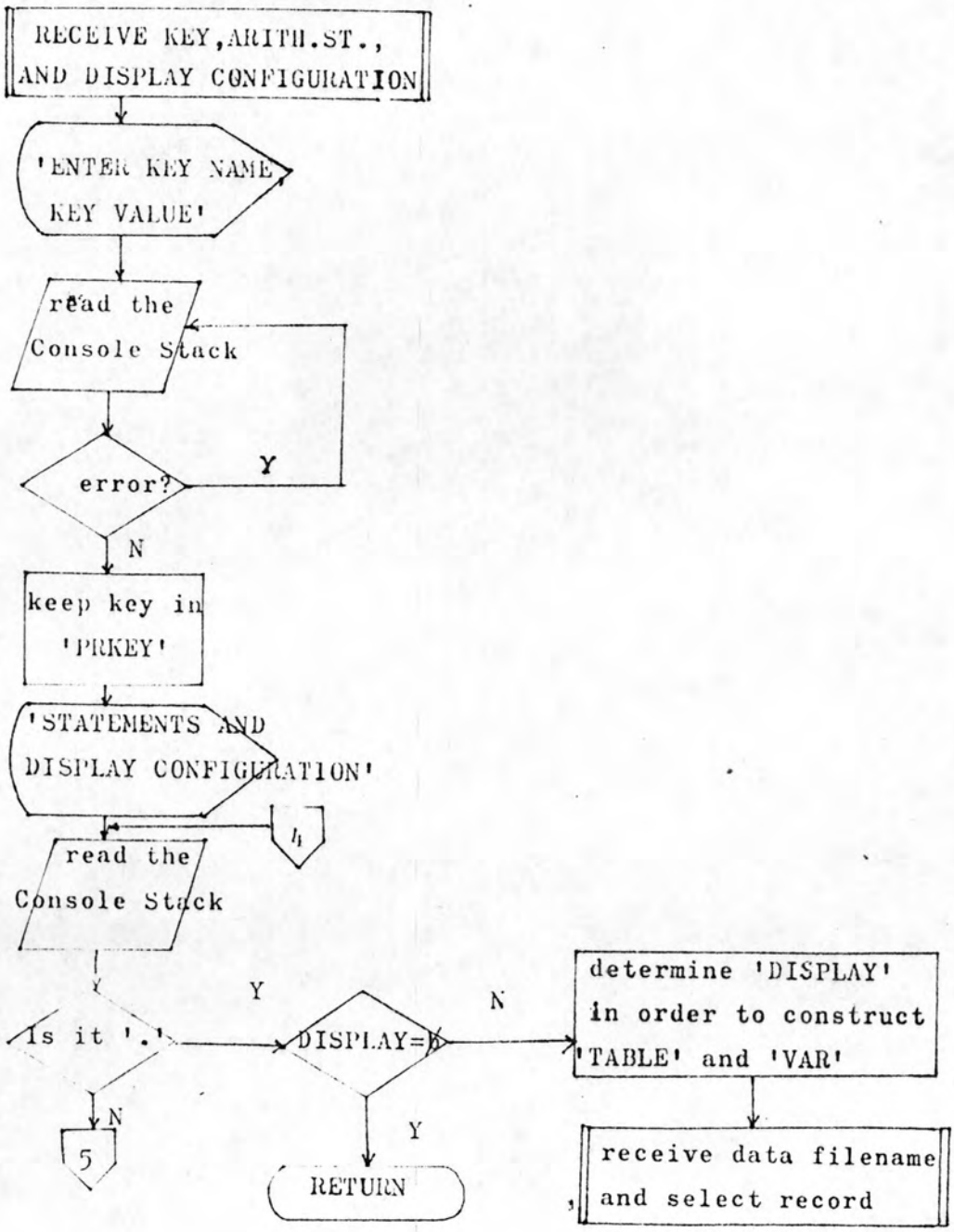
ภาคผนวก ก.

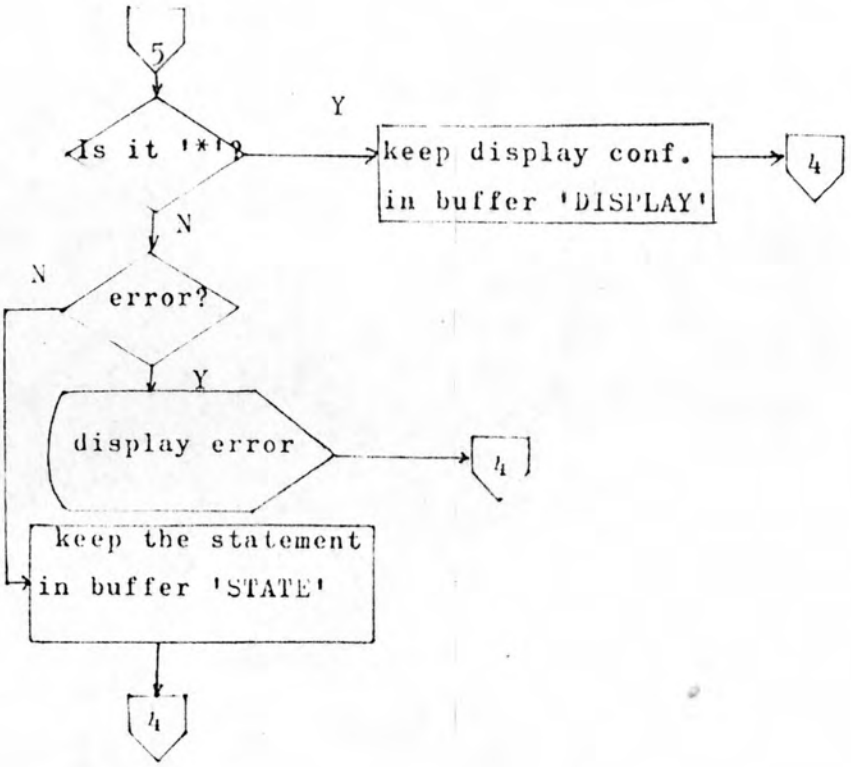
รายละเอียดโปรแกรม

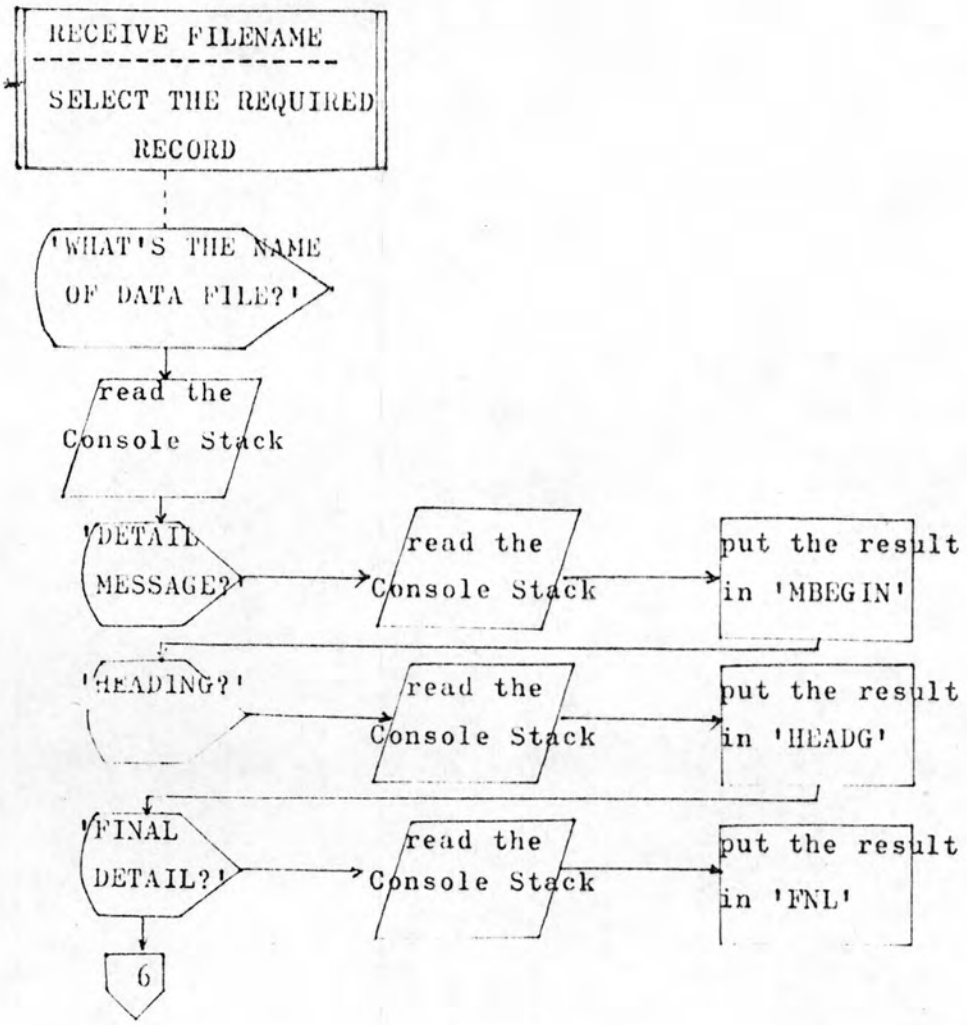

START
receive
I/O format
receive key,arith. st.,
display configuration
receive
filename
receive the required
record
PROCESS

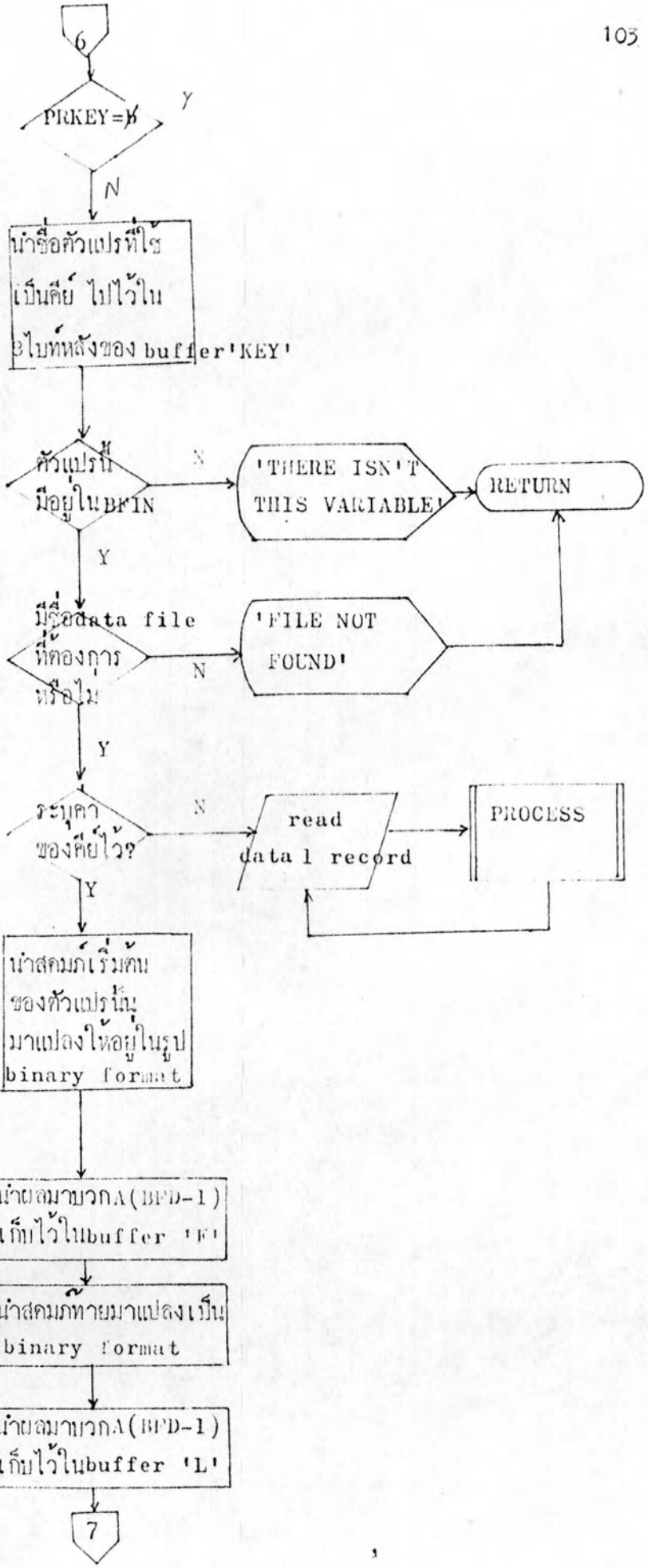

RECEIVE I/O FORMAT
'DO YOU WANT TO SET NEW INPUT FORMAT?'
read the Console Stack
Is it 'N'?
Y
N
'DO YOU WANT TO SET NEW OUTPUT FORMAT?'
read the Console Stack
Is it 'N'?
Y
N
'enter THE FORMAT FILENAME'
read the Console Stack
มีแฟ้มข้อมูลนั้น หรือไม่
Y
N
'FILE NOT FOUND'
END
อ่าน 1 ระเบียน จากแฟ้มข้อมูลนั้น
EOF?
Y
3
N
check error and construct I/O format buffer (BFIN,BFOUT)
'PLEASE ENTER THE FORMAT'
1
read the format from the Console stack
put it in buffer 'BFL'
2

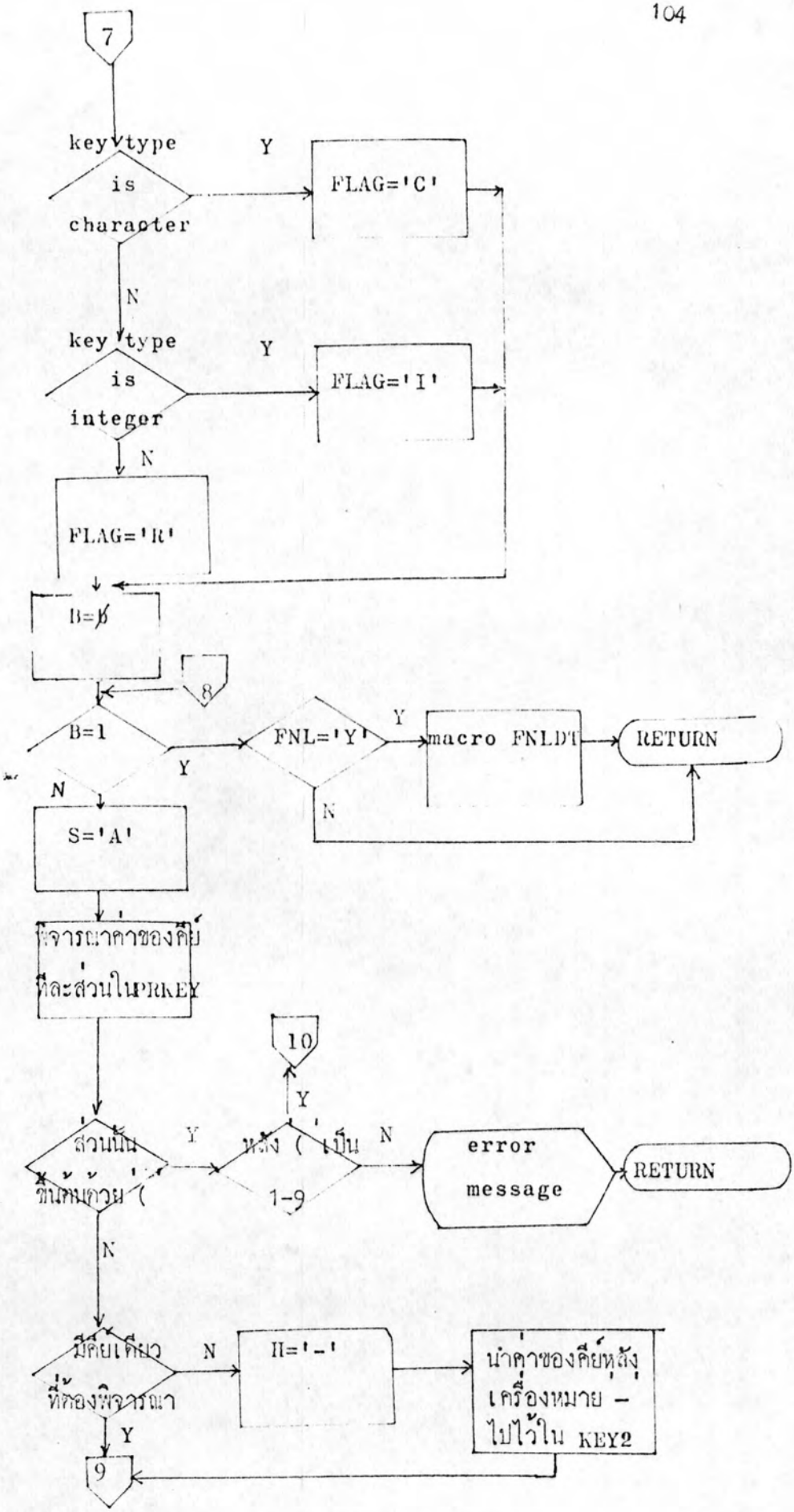

2
Is it '$'?
3
receive key
END
Y
N
check error and
construct I/O
format buffer
(BFIN,BFOUT)
1

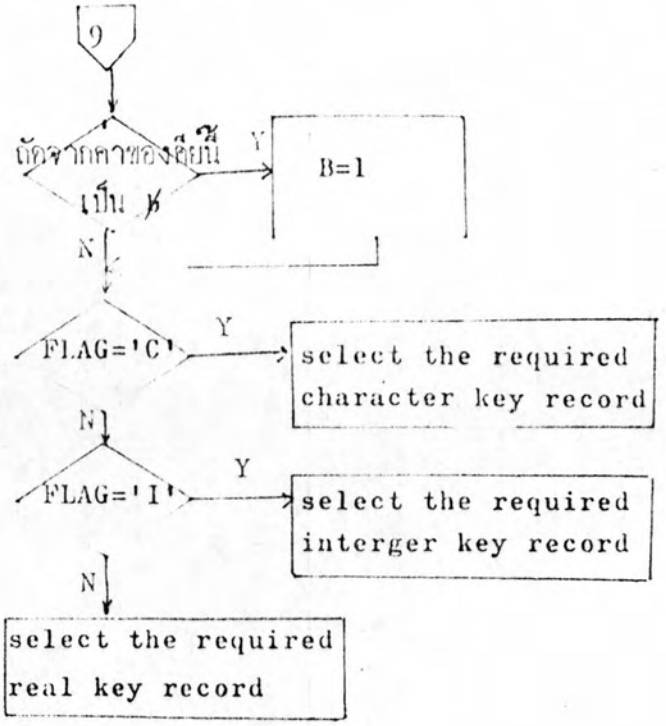

RECEIVE KEY,ARITH.ST.,
AND DISPLAY CONFIGURATION
'ENTER KEY NAME,
KEY VALUE'
read the
Console Stack
error?
Y
N
keep key in
'PRKEY'
'STATEMENTS AND
DISPLAY CONFIGURATION'
4
read the
Console Stack
Is it '.'
Y
DISPLAY=0
N
determine 'DISPLAY'
in order to construct
'TABLE' and 'VAR'
receive data filename
and select record
N
5
Y
RETURN

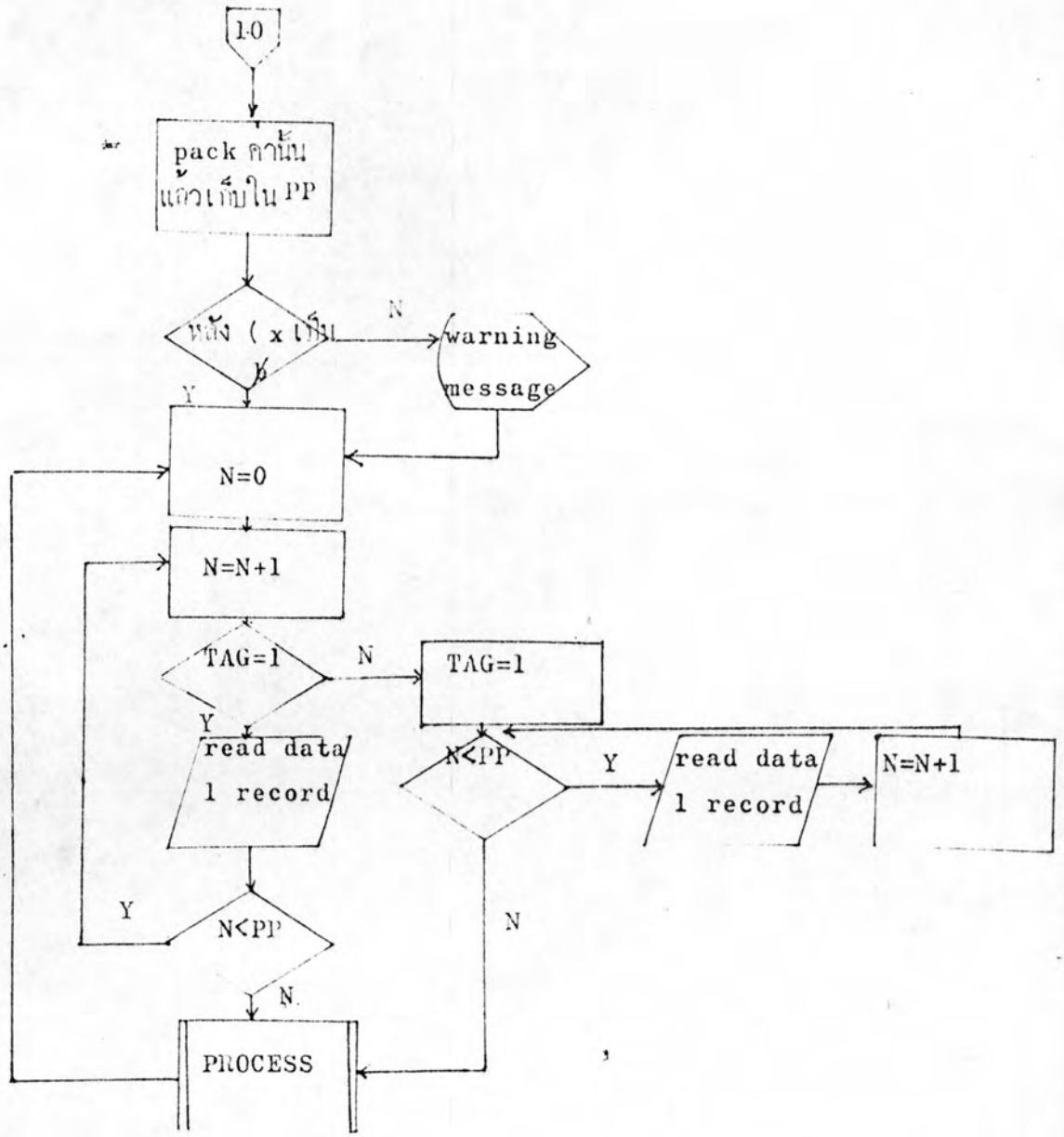

5
Is it '*'?
Y
keep display conf.
in buffer 'DISPLAY'
4
N
N
error?
Y
display error
4
keep the statement
in buffer 'STATE'
4

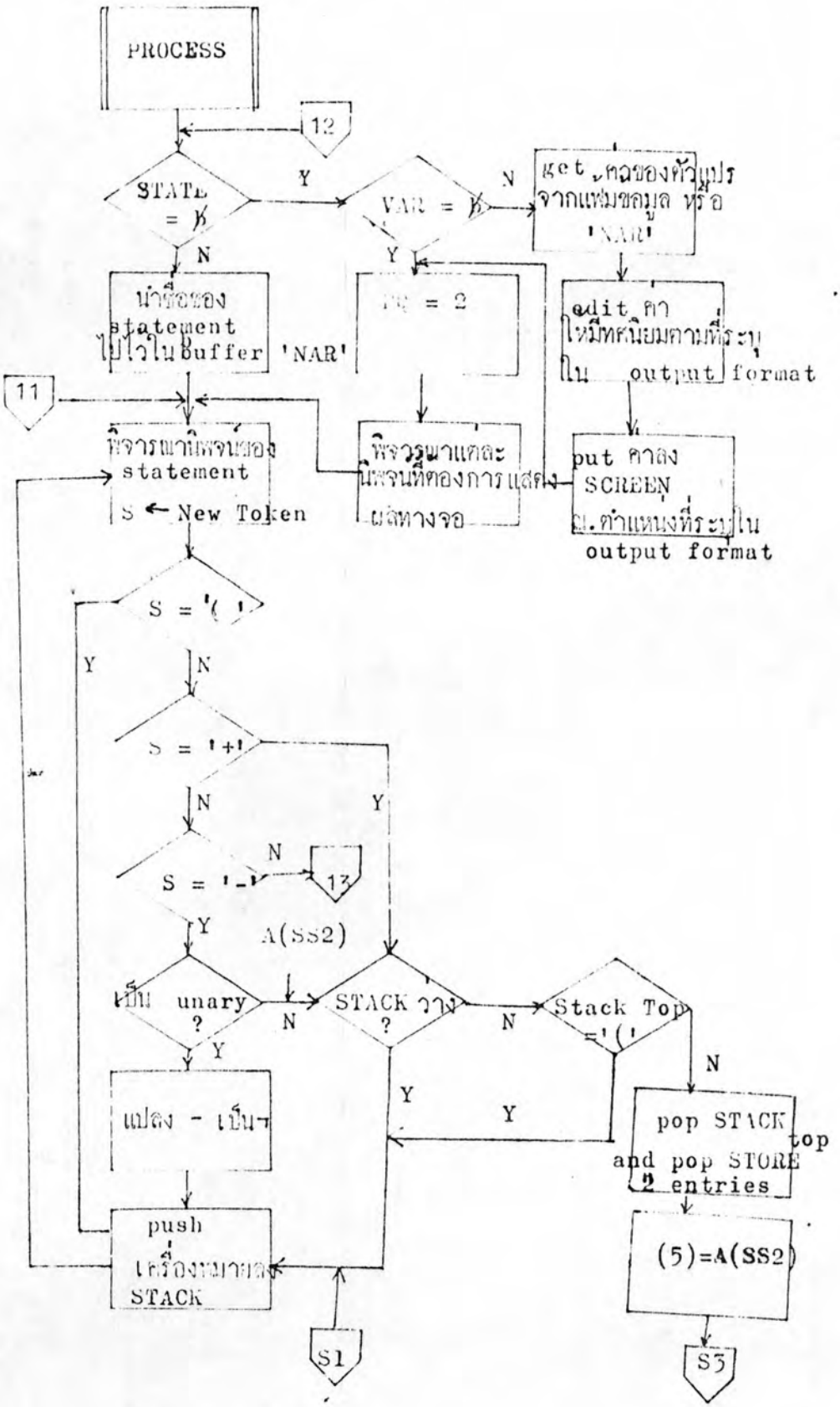

RECEIVE FILENAME
SELECT THE REQUIRED
RECORD
'WHAT'S THE NAME
OF DATA FILE?'
read the
Console Stack
'DETAIL
MESSAGE?
read the
Console Stack
put the result
in 'MBEGIN'
'HEADING?'
read the
Console Stack
put the result
in 'HEADG'
'FINAL
DETAIL?'
read the
Console Stack
put the result
in 'FNL'
6

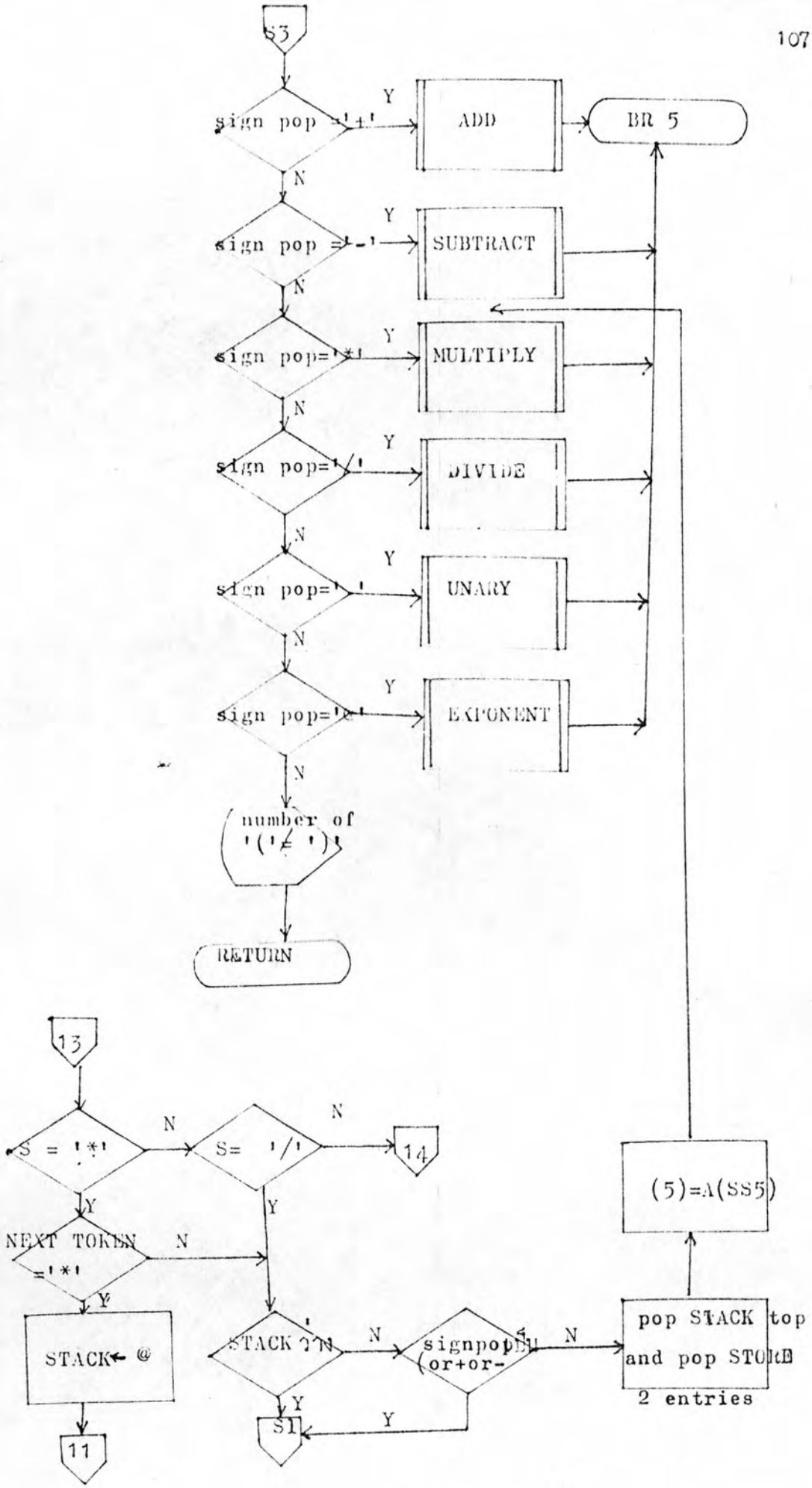

6
PRKEY=K
Y
N
นำชื่อตัวแปรที่ใช้
เป็นคีย์ ไปไว้ใน
3ไบท์หลังของ buffer'KEY'
ตัวแปรนี้
มีอยู่ใน BFIN
N
'THERE ISN'T
THIS VARIABLE'
RETURN
Y
มีชื่อdata file
ที่ต้องการ
หรือไม่
'FILE NOT
FOUND'
N
Y
ระบุคา
ของคีย์ไว้?
N
read
data 1 record
PROCESS
Y
นำสคมภ์เริ่มต้น
ของตัวแปรนั้น
มาแปลงให้อยู่ในรูป
binary format
นำผลมาบวกA(BFD-1)
เก็บไว้ในbuffer 'F'
นำสคมภ์ท้ายมาแปลงเป็น
binary format
นำผลมาบวกA(BFD-1)
เก็บไว้ในbuffer 'L'
7

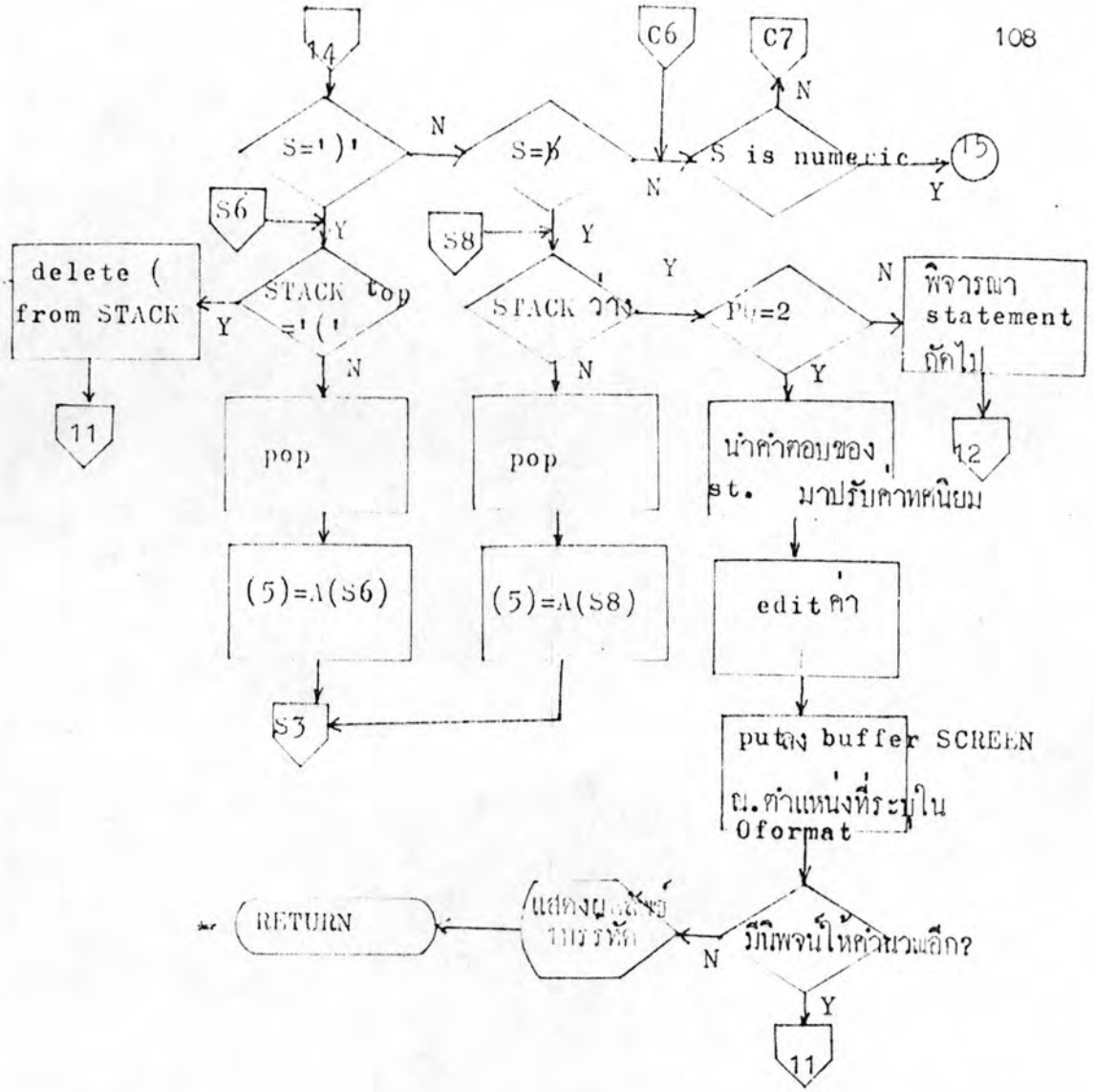

7
key type is character
Y
FLAG='C'
N
key type is integer
Y
FLAG='I'
N
FLAG='R'
B=0
8
B=1
Y
FNL='Y'
Y
macro FNLDT
RETURN
N
S='A'
พิจารณาค่าของคีย์ทีละส่วนในPRKEY
10
Y
ส่วนนั้นมีค่าเป็นตัวเลขหรือไม่
Y
หลัง (เป็น 1-9
N
error message
RETURN
N
มีคีย์เดียวที่ต้องพิจารณา
N
H='-'
นำค่าของคีย์หลังเครื่องหมาย - ไปไว้ใน KEY2
Y
9

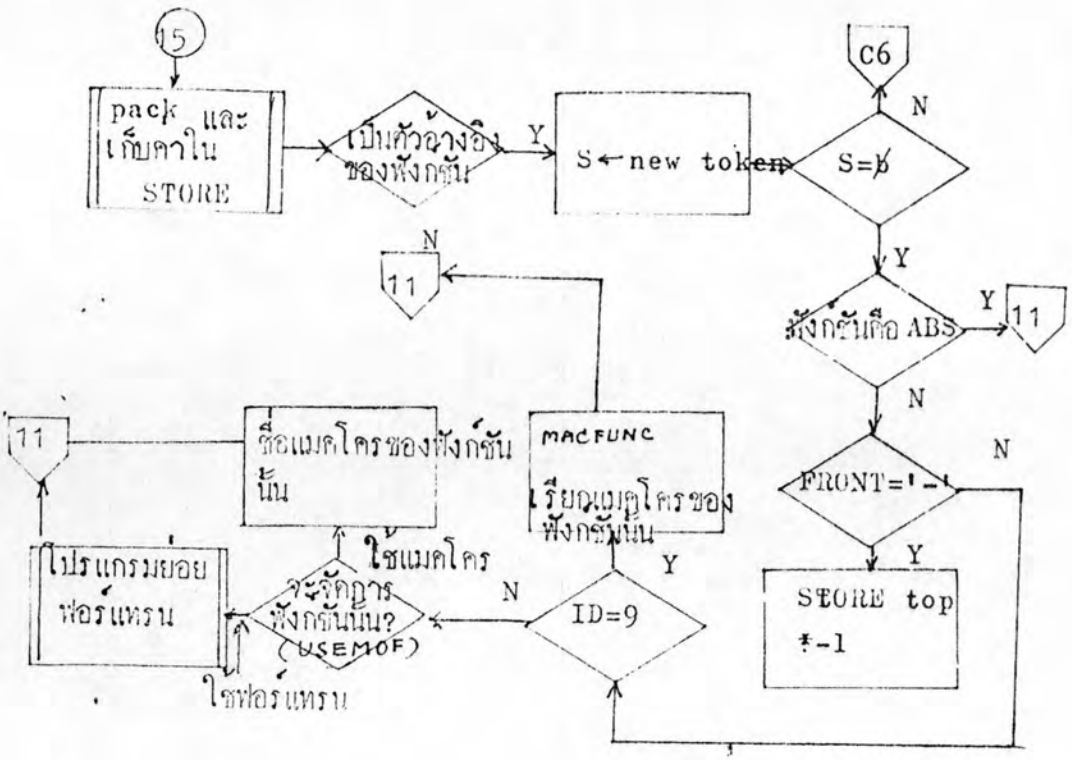

9
ตัวจากตาของมัน
เป็น ' '
Y
B=1
N
FLAG='C'
Y
select the required
character key record
N
FLAG='I'
Y
select the required
interger key record
N
select the required
real key record

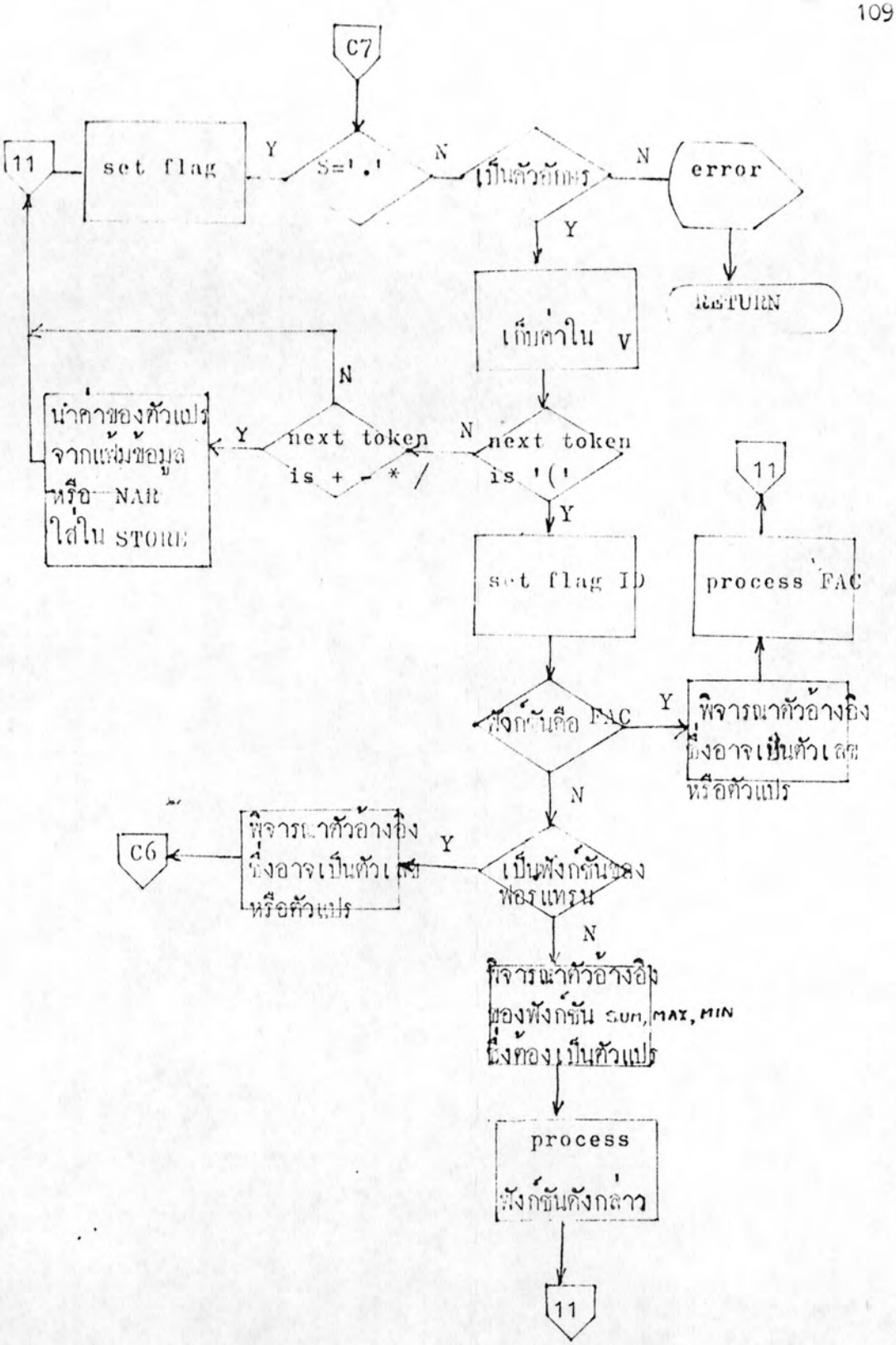

10
pack ตัวมัน
แถวเข้าใน PP
พิมพ์ (x เป็น
' ')
N
warning
message
Y
N=0
N=N+1
TAG=1
N
TAG=1
Y
read data
1 record
N<PP
Y
read data
1 record
N=N+1
Y
N<PP
N
N
PROCESS


PROCESS
12
STATE = K
Y
VAR = K
N
get ค่าของตัวแปร จากแฟ้มข้อมูล หรือ 'NAR'
N
Y
นำชื่อของ statement ไปไว้ใน buffer 'NAR'
PR = 2
edit ค่า ให้มีทศนิยมตามที่ระบุ ใน output format
11
พิจารณาที่พจน์ของ statement
S ← New Token
พิจารณาแต่ละพจน์ที่ต้องการแสดงผลทางจอ
put ค่าลง SCREEN ณ.ตำแหน่งที่ระบุใน output format
S = '('
Y
N
S = '+'
N
Y
S = '-'
N
13
Y
A(SS2)
เป็น unary ?
N
STACK ว่าง ?
N
Stack Top ='('
N
Y
Y
Y
แปลง - เป็น +
pop STACK top and pop STORE 2 entries
push เครื่องหมายลง STACK
(5)=A(SS2)
S1
S3


S3
sign pop = '+'
Y
ADD
BR 5
N
sign pop = '-'
Y
SUBTRACT
N
sign pop='*'
Y
MULTIPLY
N
sign pop='/'
Y
DIVIDE
N
sign pop=' '
Y
UNARY
N
sign pop='↑'
Y
EXPONENT
N
number of '(' ≠ ')'
RETURN
13
S = '*'
N
S= '/'
N
14
Y
NEXT TOKEN ='*'
N
Y
STACK← @
11
STACK ?'N'
N
signpop(1) (or+or-
N
Y
S1
Y
pop STACK top and pop STORE 2 entries
(5)=A(SS5)


14
S=')'
N
S=
C6
C7
N
S is numeric
N
Y
15
S6
Y
S8
Y
delete (from STACK
STACK top =='('
Y
N
STACK ว่าง
Y
N
Pr=2
Y
N
พิจารณา statement ถัดไป
11
pop
pop
นำคำตอบของ st. มาปรับค่าทศนิยม
12
(5)=A(S6)
(5)=A(S8)
edit ค่า
S3
put ลง buffer SCREEN ณ.ตำแหน่งที่ระบุใน Oformat
มีพจน์ให้คำนวณอีก?
N
แสดงผลลัพธ์ทางจอภาพ
RETURN
Y
11


15
pack และ เก็บค่าใน STORE
เป็นตัวอ้างอิงของฟังก์ชัน
Y
S←new token
C6
N
S=
Y
N
11
ฟังก์ชันคือ ABS
Y
11
N
11
ชื่อแมคโครของฟังก์ชันนั้น
MACFUNC เรียกแมคโครของฟังก์ชันนั้น
FRONT='-'
N
Y
โปรแกรมย่อยฟอร์แทรน
ใช้แมคโคร
จะจัดการฟังก์ชันนั้น? (USEMOF)
N
ID=9
Y
STORE top *-1
ใช้ฟอร์แทรน


C7
S='.'
Y
N
เป็นตัวอักษร
N
error
Y
RETURN
เก็บค่าใน V
next token is '('
N
Y
next token is + - * /
N
Y
นำค่าของตัวแปร จากแม่มข้อมูล หรือ NAR ใส่ใน STORE
11
set flag
11
set flag ID
process FAC
ฟังก์ชันคือ FAC
Y
พิจารณาตัวอ้างอิง ซึ่งอาจเป็นตัวเลข หรือตัวแปร
N
เป็นฟังก์ชันของ ฟอร์แทรน
Y
พิจารณาตัวอ้างอิง ซึ่งอาจเป็นตัวเลข หรือตัวแปร
C6
N
พิจารณาตัวอ้างอิง ของฟังก์ชัน SIN, MAX, MIN ซึ่งต้องเป็นตัวแปร
process ฟังก์ชันดังกล่าว
11

```
FILE: MAIN     ASSEMBLE V1                    VM/SP CONVERSATIONAL MONITOR

*********************************************************************
*THIS PROGRAM ACCEPTS INPUT FORMAT AND OUTPUT FORMAT FROM TERMINAL,*
*THEN STORES THEM IN INPUT OR OUTPUT FORMAT BUFFER.*
*********************************************************************
MAIN     CSECT
         BALR  9,0                      ESTABLISH ADDRESSABILITY
         USING FIRST,9,10
FIRST    L     10,=A(FIRST+4096)
         ST    14,SAVRET                SAVE RETURN ADDRESS IN REG.14
         LA    6,BFIN
         LA    7,BFOUT
         L     13,=F'1'
         MVC   CNT,=CL4'0001'
         ZAP   K,=P'0'
* PROMPTING MESSAGE
         WRTERM 'DO YOU WANT TO SET NEW INPUT FORMAT? (Y/N)'
* ACCEPT A LINE FROM THE TERMINAL AND PLACES IT IN THE BUFFER MMX
         RDTERM MMX
         MVC   MM(1),MMX
         LA    11,M2
         CLI   MM,C'N'
         BE    X1
M2       MVC   MMX,MMX-1
         WRTERM 'DO YOU WANT TO SET NEW OUTPUT FORMAT? (Y/N)'
         RDTERM MMX
         MVC   MM(1),MMX
         LA    11,ER
         CLI   MM,C'N'
         BE    X1
ER       WRTERM 'PLEASE KEY THE FORMAT YOU WANT.  AFTER ENTERING EACH
               LINE, WAIT UNTIL ''C.K'' OR ERROR APPEAR IN ORDER TO EN
               DIF THAT YOUR FORMAT IS CORRECT OR NOT.',EDIT=LONG
FFORMAT  MVC   BFL(80),=C' '
         MVC   MMX,MMX-1
         RDTERM MMX
         MVC   BFL(80),MMX
         BAL   8,SCR
         B     FFORMAT
** THE INTERNAL SUBROUTINE NAMED 'SCR' CHECKS ALL ERRORS THAT MAY OCCU
** IN EACH LINE.
SCR      MVI   AST,C'*'
         MVI   SPEC,C' '
         L     4,=F'1'
         LA    2,BFL
         L     12,=F'1'
         LA    5,NAME
         MVC   NAME(8),NAME-1
*
* CHECK THE FIRST PART OF RECORD FORMAT
*
         CLI   0(2),C'$'   IF THE USER ENTERS FORMAT THROUGH TERMINAL,
         BE    EV          "$" MUST BE USE TO STOP TO EXECUTE THE
*                          EXTERNAL SUBROUTINE 'ENDVAR'.
         CLI   0(2),X'C1'
         BL    ERROR1
```

```
FILE: MAIN     ASSEMBLE V1                    VM/SP CONVERSATI     MONITOR

           CLI   0(2),X'CA'
           BL    COM
           CLI   0(2),X'D1'
           BL    ERRCR1
           CLI   0(2),X'DA'
           BL    COM
           CLI   0(2),X'E2'
           BL    ERRCR1
           CLI   0(2),X'E9'
           BH    ERRCR1
COM        MVC   0(1,5),0(2)
           A     3,=F'1'
           A     5,=F'1'
           A     4,=F'1'
           C     4,=F'8'
           BH    COLUMN10
           CLI   0(2),C' '
           BE    CN10
           CLI   0(2),X'C1'
           BL    ERR
           CLI   0(2),X'CA'
           BL    COM
           CLI   0(2),X'D1'
           BL    ERR
           CLI   0(2),X'DA'
           BL    COM
           CLI   0(2),X'E2'
           BL    ERR
           CLI   0(2),X'EA'
           BL    COM
           CLI   0(2),X'F0'
           BL    ERR
           CLI   0(2),X'F9'
           BH    ERR
           B     COM
COLUMN10   CLI   0(2),C' '
           BNE   ERRR
*
* CHECK THE SECOND PART OF RECORD FORMAT
*
CN10       A     2,=F'1'
           CLI   0(2),C' '
           BE    CN10
           CLI   0(2),C'*'  IF THE 2ND PART ARE LEADED BY '*',THE VARIABL
           BE    PBFOUT     NAME WILL BE PUT IN THE OUTPUT FORMAT BUFFER
           MVC   0(3,6),NAME
           S     2,=F'1'
           B     MF
ERRR       WRTERM 'LENGTH OF VARIABLE NAME IS TOO LONG. KEY AGAIN.'
           BR    8
EV         LA    1,ADDR
           L     15,=V(ENDVAR)
           BALR  14,15
           B     EXIT
PBFOUT     MVC   0(8,7),NAME
```

```
FILE: MAIN     ASSEMBLE V1                    VM/SP CONVERSATIONAL MONITOR

               MVI   AST,C'*'
MF             MVC   FCOL(3),=3C' '
               MVC   LCOL(3),=3C' '
               LA    4,FCOL
ZERO           A     2,=F'1'
               CLI   0(2),C'0'
               BE    ZERO
COL10          CLI   0(2),C'1'
               BL    E2
               CLI   0(2),C'9'
               BH    E2
               L     5,=F'0'
CHECK          MVC   0(1,4),0(2)
               A     4,=F'1'
               A     2,=F'1'
               CLI   SPEC,C' '
               BNE   CBLANK
               CLI   0(2),C'-'
               BNE   CBLANK
               LA    4,LCOL
               MVI   SPEC,C'*'
               B     ZERO
CBLANK         CLI   0(2),C' '
               BE    CCOL
               A     5,=F'1'
               C     5,=F'3'
               BH    ERRCR3
               CLI   0(2),C'0'
               BL    E2
               CLI   0(2),C'9'
               BH    E2
               B     CHECK
CCOL           CLI   AST,C'*'
               BNE   LA3
               LA    4,8(0,7)
               B     DO
LA3            LA    4,8(0,6)
DO             LA    3,FCOL+1
               LA    5,FCOL
               CLC   0(3,5),=3C' '
               BE    ERRXX
               BAL   12,A
               CLI   AST,C'*'
               BNE   ASK
               CP    0(2,4),LPS LPS CONTAINS THE LAST CCL. OF THE FORMER VAR
               BNH   ERROR6
ASK            CLI   LCOL,C' '
               BNE   B
               ZAP   2(2,4),0(2,4)
               B     KA
B              A     4,=F'2'
               LA    3,LCOL+1
               LA    5,LCOL
               BAL   12,A
               S     4,=F'2'
```

```
         CP    0(2,4),2(2,4)
         BH    ERROR4
KA       CLI   AST,C'*'
         BNE   COLLB
         ZAP   PREVTS(2),LPS(2)
         ZAP   LPS,2(2,4)
COLLB    A     4,=F'4'
         LA    5,TEM
CLB      A     2,=F'1'
         CLI   0(2),C' '
         BNE   AAA
*
* CHECK THE THIRD PART OF RECORD FORMAT
*
         C     2,=A(BFL+71)
         BE    BLK
         B     CLB
AAA      CLI   0(2),C'0'
         BE    PACKIN
         CLI   0(2),C'1'
         BL    ERROR7
CLB      CLI   0(2),C'9'
         BH    ERROR7
         MVC   0(1,5),0(2)
         A     2,=F'1'
         CLI   0(2),C' '
         BE    KEPT
         CLI   0(2),C'0'
         BL    ERROR7
         CLI   0(2),C'9'
         BH    ERROR7
         A     5,=F'1'
         MVC   0(1,5),0(2)
         B     JUMP
PACKIN   ZAP   0(2,4),=PL2'0'
         B     BLK
KEPT     MVC   TEM+1(1),TEM
         MVI   TEM,C'0'
JUMP     PACK  P(2),TEM(2)
         CP    P(2),=PL2'10'
         BH    ERROR8
         ZAP   0(2,4),P(2)
BLK      CLI   AST,C'*'
         BNE   BIN
         A     7,=F'14'
         L     4,CNT
         C     4,=F'20'
         BE    WARN1
         A     4,=F'1'
         ST    4,CNT
         B     OK
BIN      A     6,=F'14'
         C     13,=F'20'
         BE    WARN
         A     13,=F'1'
```

```
CK      WRTERM 'O.K'
        BR    8        LOOP TO READ THE NEXT FORMAT LINE.
* THE INTERNAL SUBROUTINE ,'A',ARRANGES NUMBERS IN FOOL OR LOOL
* IN ORDER TO PACK IT LATER.
A       CLI   0(3),C' '
        BE    S1
        A     3,=F'1'
        CLI   0(3),C' '
        BE    S2
C       PACK  0(2,4),0(3,5)
        BR    12
S1      MVC   2(1,5),0(5)
        MVC   0(2,5),=2C'C'
        B     C
S2      MVC   2(1,5),1(5)
        MVC   1(1,5),0(5)
        MVI   0(5),C'0'
        B     C
* READ FORMAT FROM DISK
X1      AP    K+1),ONE(1)
        ST    11,NINE
        MVC   MMX,MMX-1
        WRTERM 'WHAT IS THE NAME OF THE FORMAT FILE ?'
        RDTERM MMX
        MVC   FORMATN(8),MMX
* USE THE FSCBD MACRO INSTRUCTION TO GENERATE A DSECT AND REFER TO THE
* LABELS IN THE DSECT TO MODIFY THE FSCB.
        LA    5,FORMAT
        USING FSCBD,5
* THE MVC INSTRUCTION PLACES THE FILENAME CONTAINED IN FORMATN INTO
* THE FSCBFN(FILENAME) FIELD OF THE FSCB.
        MVC   FSCBFN,FORMATN
        FSSTATE FSCB=FORMAT,ERROR=ERR1  DETERMINE IF INPUT FILE EXISTS
READFORM MVC  BFL(80),BFL-1
        FSREAD FSCB=FORMAT,ERROR=EOF  READ A RECORD FROM INPUT FILE
        BAL   8,SCR
        B     READFORM     LOOP BACK FOR NEXT RECORD
ERR1    WRTERM 'FILE NOT FOUND'  IF INPUT FILE DOES NOT EXIST
        B     EXIT
* COME HERE IF ERROR READING INPUT FILE
EOF     C     15,=F'12'  END OF FILE?
        BNE   ERR2       ERROR IF NOT
        LA    15,0       ALL O.K. - ZERO OUT REG.15
* IF THE FILE HAVING ALREADY READ IS OUTPUT FORMAT FILE, GO TO THE
* SUBROUTINE 'ENDVAR'. OTHERWISE,BRANCH TO CONTENT IN REG.11
        CP    K,=P'2'
        BE    EV
        L     11,NINE
        BR    11
* IF READING ERROR WAS NOT NORMAL END OF FILE.
ERR2    LR    12,15
        LINEDIT TEXT='ERROR CODE .... IN READING FILE',SUB=(DEC,(12))
EXIT    L     14,SAVRET  LOAD RETURN ADDRESS
        LA    15,0       SET RETURN CODE IN REG.15
        BR    14         RETURN TO CALLER
```

```
WARN     WRTERM '**WARNING - THIS IS THE LAST VARIABLE NAME OF DATA FILE
               THAT YOU CAN NAME**'
         BR    8
WARN1    WRTERM '**WARNING--YOU CANNOT NAME VARIABLES OF OUTPUT FORMAT
               ANY MORE**'
         BR    8
E2       WRTERM 'SECOND PART MUST CONSIST OF 1-9 . KEY AGAIN.'
         BR    8
ERROR1   WRTERM '** ERROR--COLI MUST BE A-Z.  KEY AGAIN,**'
         BR    8
ERROR3   WRTERM '** ERROR--THE LARGEST RECORD LENGTH OF A DATA FILE IS
               130 CHARACTERS.**'
         BR    8
ERROR4   WRTERM '**ERROR--STARTING COL. MUST BE LESS THAN ENDING COL. K
               EY AGAIN.'
         BR    8
ERROR5   WRTERM '**ERROR--RECHECK COL. OF OUTPUT FORMAT YOU KEYED.**'
         B     EXIT
ERROR7   WRTERM '**ERROR--RECHECK THE THIRD PART. KEY AGAIN**'
         B     EXIT
ERROR8   WRTERM '**ERROR--NUMBERS BEHIND DECIMAL POINT CANNOT EXCEED 10 D
               IGITS.**'
BBR3     CLI   AST,C'*'
         BNE   PLPS
         ZAP   LPS(2),PREVTS(2)
PLPS     BR    8
ERR      WRTERM '**ERROR--VARIABLE NAME MUST BE COMPOSED OF A-Z OR 0-9.
               KEY AGAIN.'
         BR    8
ERRXX    WRTERM 'YOU MISSED THE SECOND PART. KEY AGAIN.'
         BR    8
FORMAT   FSLD  'F FORMAT A1',BUFFER=BFL,RSIZE=80
MINE     DS    F
SDEC     DS    CL1
NM       DS    CL1
K        DS    PL1
FORMATN  DC    2C' '
LFIN     DC    280C' '
BFOUT    DC    280C' '
FCOL     DC    3C' '
LCOL     DC    3C' '
F        DS    PL4
ONE      DC    PL1'1'
LPS      DC    PL2'0'
PREVTS   DC    PL2'0'
PREVT    DC    PL2'0'
TEM      DC    2C' '
SAVRET   DS    F
         DC    C' '
BFL      DC    80C' '
AST      DS    CL1
         DC    C' '
NAME     DS    CL8
         DC    C' '
MMX      DS    CL130
```

```
CNT       DS    F
ADDR      DC    A(BFIN)
          DC    A(BFOUT)
          FSCBC
          END
```

```
FILE: ENDVAR    ASSEMBLE V1                    VM/SP CONVERSATIONAL MONITOR

*****************************************************************************
* ACCEPT KEY NAME,KEY VALUE,ARITHMETIC STATEMENTS AND DISPLAY*
* CONFIGURATION FROM TERMINAL,MOREOVER IT KEEPS THEM IN TABLES*
*****************************************************************************
ENDVAR   CSECT
         BALR  11,0
         USING FIRST,11,12
FIRST    L     12,BASE
         STM   13,14,SAVE
         STM   15,10,RELEVN
         L     10,0(1)
         MVC   BFIN(256),0(10)
         MVC   BFIN+256(24),256(10)
         MVC   BFOUT(256),280(10)
         MVC   BFOUT+256(24),536(10)
         LA    10,PRKEY+236
EN       LA    9,PRKEY
E        WRTERM 'ENTER KEY NAME OR KEY VALUES'
         MVC   BFL,BFL-1
         MVC   MMX,MMX-1
         RDTERM MMX
         MVC   BFL(80),MMX
         L     4,=F'1'
         L     2,=A(BFL)
         CLI   0(2),C' '
         BE    ENDKEY
* CHECK SYNTAX OF KEY NAME
         CLI   0(2),C'A'
         BL    ERROR1
         CLI   0(2),X'CA'
         BL    COM
         CLI   0(2),X'D1'
         BL    ERROR1
         CLI   0(2),X'DA'
         BL    COM
         CLI   0(2),X'E2'
         BL    ERROR1
         CLI   0(2),C'Z'
         BH    ERROR1
COM      A     2,=F'1'
         A     4,=F'1'
         C     4,=F'8'
         BH    ATM
         CLI   0(2),C' '
         BE    E1
         CLI   0(2),C':'
         BE    SAV
         CLI   0(2),X'C1'
         BL    ERR
         CLI   0(2),X'EA'
         BL    COM
         CLI   0(2),X'F0'
         BL    ERR
         CLI   0(2),X'F9'
         BH    ERR
         B     COM
```

```
E1      WRTERM 'YOU MUST ENTER PRIMARY KEY .KEY AGAIN'      *
        B     E
AT1     CLI   0(2),C':'
        BE    SAV
        B     E1
ERROR1  WRTERM 'COL.2 MUST BE A-Z.KEY AGAIN'
        B     E
ERR     WRTERM 'PART 2 MUST BE COMPOSED OF A-Z OR 0-9.KEY AGAIN.'
        B     E
* MOVE KEY STATEMENT TO BUFFER 'PFKEY'
SAV     MVC   0(80,9),BFL
* IF COL.80 ISN'T BLANK,THE NEXT KEY LINE WILL BE READ.
        CLI   79(9),C' '
        BE    ENDKEY
        A     9,=F'79'
        CR    9,10
        BH    ERR2
        MVC   BFL(80),BFL-1
        RDTERM MMX
        MVC   BFL(80),MMX
*       RDTERM BFL,EDIT=PHYS,LENGTH=80
        B     SAV
ENDKEY  LA    9,STATE
        LA    6,VAR
        LA    10,XPSION
        WRTERM 'ENTER STATEMENTS OF DISPLAY CONFIGURATION'
BBFL    MVC   BFL(80),BFL-1
        MVC   MMX(130),MMX-1
        RDTERM MMX
        MVC   BFL(80),MMX
*       RDTERM BFL,EDIT=PHYS,LENGTH=80
        LA    2,BFL
* IF THE 1ST COL. HAS '.',STOP ENTERING ARITHMETIC STATEMENTS AND
* DISPLAY CONFIGURATION.
        CLI   0(2),C'.'
        BE    FINISH
* IF THE 1ST COL. HAS '*', 3 TABLES WILL BE ESTABLISHED.
* THEY ARE 'TABLE','NAM' AND 'XPSION'
        CLI   0(2),C'*'
        BE    SCAN
* KEEP ALL STATEMENTS IN BUFFER 'STATE'
        CLI   0(2),C'A'
        BL    ERR0
        CLI   0(2),C'Z'
        BH    ERR0
        C     9,=A(STATE+1999)
        BH    ERR77
        MVC   0(80,9),0(2)
        A     9,=F'80'
        B     BBFL
ERR0    WRTERM 'THE ABOVE LINE MUST BE ARITHMETIC STATEMENT OR DISPLAY
              OUTPUT CONFIGURATION.'
        B     BBFL
ERR77   WRTERM '20 ARITHMETIC STATEMENTS ARE THE MOST.KEY AGAIN.'
        B     BBFL
```

```
SCAN     LA    4,DISPLAY
CL4      MVC   0(80,4),0(2)
         CLI   79(4),C' '
         BE    BBFL
         A     4,=F'79'
         C     4,=A(DISPLAY+236)
         BH    XC
         MVC   BFL,BFL-1
         MVC   MMX,MMX-1
         RDTERM MMX
         MVC   BFL(80),MMX
*        RDTERM BFL,EDIT=PHYS,LENGTH=80
         B     CL4
XC       WRTERM 'FULL MESSAGE IN DISPLAY CONFIGURATION CAN''T BE CONTAIN
               ED IN THE BUFFER.KEY /CAIN.'
         B     BBFL
BFINISH  CLI   DISPLAY,C' '
         BE    EXIT2
         LA    4,TABLE
         LA    3,DISPLAY
INCR1    A     2,=F'1'
         CLI   0(2),C' '
         BE    INCR1
         ZAP   SEQ(2),=P'0'
LR72     LR    7,2
         AP    SEQ(2),=P'1'
ESSO     CLI   0(2),C','      **
         BE    NXX             *
         CLI   0(2),C' '       * CONCERN WITH TABLE "VAR" AND "TABLE"
         BE    NXX             *
         C     2,=A(DISPLAY+236) **
         BH    NXX
         CLI   0(2),C'A'     **
         BL    EXPRESS        *
         CLI   0(2),X'EA'     *
         BL    V              ** CONCERN WITH "XPSION" AND "TABLE"
         CLI   0(2),C'0'      *
         BL    EXPRESS        *
         CLI   0(2),C'9'      *
         BH    EXPRESS       **
V        A     2,=F'1'
         B     ESSO
* RUNNING NO. WILL BE PUT IN BOTH 'TABLE' AND 'VAR'
NXX      ZAP   0(2,6),SEQ(2)
         ZAP   0(2,4),SEQ(2)
***      AP    SEQ(2),=PL2'1'
         A     6,=F'2'
         A     4,=F'2'
         ST    6,TMP
* PUT VARIABLE NAME IN 'VAR'
M67      MVC   0(1,6),0(7)
         A     6,=F'1'
         A     7,=F'1'
         CR    7,2
         BNE   M67
```

```
FILE: ENDVAR   ASSEMBLE V1                    VM/SP CONVERSATIONAL MONITO

          L     6,TMP
* IS THE VARIABLE NAME MENTIONED IN THE OUTPUT FORMAT BUFFER?
          LA    8,BFOUT
C68       CLC   0(8,6),0(8)
          BE    GCL
          A     8,=F'14'
          C     8,=A(BFOUT+280)
          BL    C68
* IS THE VARIABLE NAME IN THE INPUT FORMAT BUFFER?
          LA    8,BFIN
CCCC      CLC   0(8,6),0(8)
          BE    SLENGTH
          A     8,=F'14'
          C     8,=A(BFIN+280)
          BL    CCCC
          BAL   3,A2X
CB        CLI   0(7),C' '
          BE    ENH
          C     7,=A(DISPLAY+236)
          BH    ENH
          A     2,=F'1'
          A     6,=F'8'
          B     LR72
*ENH       B     EXIT2
ENH       LA    1,TOFINISH
          L     15,=V(CHAR)
          BALR  14,15
          B     EXIT2
*GCL       A     6,=F'8'
GCL       ZAP   0(2,4),8(2,8)  BRING STARTING COL.,ENDING COL.&TYPE OF
          ZAP   2(2,4),10(2,8) THAT VARIABLE TO 'TABLE'
          MVC   4(2,4),12(8)
          ZAP   JACK,2(2,4) JACK CONTAINS LAST COL. OF FORMER VARIABLE
          A     4,=F'6'
          B     CB
* THE LENGTH OF THAT VARIABLE = STARTING COL.-ENDING COL.
SLENGTH   ZAP   LC(2),10(2,8)
          SP    LC(2),8(2,8)
          CLI   12(8),C' '
          BE    AJK
          CLI   12(8),C'0'
          BE    IG
          ZAP   4(2,4),12(2,8)
AJK       AP    JACK,=P'2'
* NOW JACK CONTAINS STARTING COL. OF THE VARIABLE DISPLAYED AT CRT
          ZAP   0(2,4),JACK
          AP    JACK,LC
* NOW JACK CONTAINS ENDING COL. OF THE VARIABLE DISPLAYED AT CRT
          ZAP   2(2,4),JACK
```

```
         A     4,=F'6'
         B     CR
IG       ZAP   4(2,4),=P'C'
         B     AJK
* RUNNING NO. WILL BE PUT IN BOTH 'TABLE'I'XPSION'
EXPRESS  ZAP   0(2,4),SEQ(2)
         ZAP   C(2,1B),SEQ(2)
         A     1C,=F'2'
         A     4,=F'2'
* PUT THE EXPRESSION IN 'XPSION'
MIL      MVC   C(1,1C),0(7)
         A     1C,=F'1'
         A     7,=F'1'
         CLI   0(7),C','
         BE    CCA
         CLI   0(7),C' '
         BE    BBA
         C     7,=A(DISPLAY+236)
         BH    ER2
         B     MIL
CCA      BAL   3,A2X
         A     1C,=F'1'
         A     7,=F'1'
         LR    2,7
         B     LR72
BBA      BAL   3,A2X
         B     ENH
ER2      XRTERM ' YOU WANT TOO MUCH KEYS.'
         B     EN
A2X      AP    JACK,=P'2'
* JACK CONTAINS THE STARTING COL.OF THE RESULT OF THE EXPRESSION THAT
* WILL BE DISPLAYED ON THE SCREEN.
         ZAP   0(2,4),JACK
         AP    JACK,=PL2'19'
* JACK CONTAINS THE ENDING COL. OF THE RESULT OF THE EXPRESSION THAT
* WILL BE DISPLAYED ON THE SCREEN
         ZAP   2(2,4),JACK
         A     4,=F'6'
         BR    3
EXIT2    LM    13,14,SAVE          * * * * *
         LM    15,1C,RELEVEN
         LA    15,0
         BR    14                        *
SAVE     DS    0
RELEVEN  DS    11F
BASE     DC    A(PIPSTR4C96)
         DC    C' '
BFL      DS    CL80
         DC    C' '
MAX      DS    CL130
PFKEY    DC    3370' '                   *
ANS      DS    C11                       *
DOT      DS    PL1                       *
STATE    DC    1600C' '                *
VAR      DC    200C' '                 *
```

```
XPSION   DC    7CCC' '                 *
DISPLAY  DC    237C' '                 *
TABLE    DC    240C' '                 *
SE)      DS    PL2
JACK     DC    PL7'-1'
LC       DS    PL2
TMP      DS    F
BFIN     DS    CL280
BFOUT    DS    CL280
TOFINISH DC    A(BFIN)
         DC    A(BFOUT)
         DC    A(PRKEY)
         DC    A(STATE)
         DC    A(VAR)
         DC    A(XPSION)
         DC    A(TABLE)
         END
```

```
FILE: CHAR     ASSEMBLE V1                    VM/SP CONVERSATIONAL MONITOR

*********************************************************************
*ACCEPT DATA FILENAME FROM THE CONSOLE STACK AND SELECTS RECORDS WHICH
*WHICH USER WANTS FROM IT.
*********************************************************************
CHAR     CSECT
         BALR  9,0
         USING INCBASE,9,13
INCBASE  L     13,BASE
         STM   14,8,FLEV
         STM   10,12,TH3
*         ST    14,SAVE
         L     2,0(1)
         MVC   BFIN(256),0(2)
         MVC   BFIN+256(24),256(2)
         MVC   BFOUT(256),280(2)
         MVC   BFOUT+256(24),536(2)
         L     2,8(1)
         MVC   PRKEY(237),0(2)
         L     2,12(1)
         MVC   STATE(256),0(2)
         MVC   STATE+256(256),256(2)
         MVC   STATE+512(256),512(2)
         MVC   STATE+768(256),768(2)
         MVC   STATE+1024(256),1024(2)
         MVC   STATE+1280(256),1280(2)
         MVC   STATE+1536(64),1536(2)
         L     2,16(1)
         MVC   VAR(200),0(2)
         L     2,20(1)
         MVC   XPSION(256),0(2)
         MVC   XPSION+256(256),256(2)
         MVC   XPSION+512(188),512(2)
         L     2,24(1)
         MVC   TABLE(240),0(2)
         WRTERM 'WHAT'S THE NAME OF DATA FILE TO BE PROCESSED?'
         RDTERM DF
         MVC    DFN(8),DF
         MVC    NF(8),DF
         MVI    TAG,C'1'
         WRTERM 'IS THERE DETAIL MESSAGE BEFORE PROCESSING STATEMENTS?
               (Y/N)'
         RDTERM DF
         MVC   MBEGIN(1),DF
         WRTERM 'ARE THERE HEADINGS?(Y/N)'
         RDTERM DF
         MVC   HEADG(1),DF
         WRTERM 'ARE THERE ANY FINAL DETAILS BEFORE STOPPING PROCESSIN
               ? (Y/N)'
         RDTERM DF
         MVC   FNL(1),DF
         CLI   MBEGIN,C'Y'
         BNE   CPKB
         DETAIL
*IF PRKEY =BLANK,ALL RECORDS IN THE DATA FILE WILL BE USE
CPKB     CLI   PRKEY,C' '
```

```
          BE      TX
          L       10,=A(BFIN)
          LA      4,BFIN+279
          LA      3,KEY+8
          LA      5,PRKEY
XYX       MVC     0(1,3),0(5)
          A       5,=F'1'
          A       3,=F'1'
          CLI     0(5),C':'
          BNE     XYX
          LA      3,KEY+8
* IS KEY NAME ONE OF THE VARIABLES IN INPUT FORMAT BUFFER?
XYZ       CLC     0(8,3),0(10)
          BE      TX
          A       10,=F'14'
          CP      10,4
          BH      ER3
          B       XYZ
ER3       LINEDIT TEXT='THERE IS NOT THE KEY ........',SUB=(CHARA,(3))
          B       EXIT
ERR1      WRTERM 'FILE NOT FOUND'
          B       EXIT
EOF       C       15,=F'12'
          BNE     ERR2
EXIT      CLI     FNL,C'Y'
          BNE     OFF
          FNLDT
OFF       LA      15,0
          LM      14,8,FLEV
          LM      10,12,TH3
          LA      15,0
          BR      14
ERR2      LR      10,15
          LINEDIT TEXT='CODE ..... IN READING FILE',SUB=(DEC,(10))
          B       EXIT
TX        LA      6,RFSCB
          USING FSCBD,6
          MVC     FSCBFN(8),NF
          FSSTATE FSCB=RFSCB,ERROR=ERR1
          ZAP     WW(3),=P'0'
LOAD1     CLI     PRKEY,C' '
          BE      ALL
          CLI     1(5),C' '
          BE      ALL
          ZAP     DOL,8(2,10)
          CVB     2,DOL
          LA      7,BFD       BFD IS THE BUFFER CONTAINING A DATA RECORD
          AR      7,2
          S       7,=F'1'
          ST      7,F         THE 1ST COL. OF KEY FIELD
          ZAP     DOL,10(2,10)
          CVB     2,DOL
          LA      7,BFD
          AR      2,7
          S       2,=F'1'
          ST      2,L         THE LAST COL. OF KEY FIELD
* GIVE A SYMBOL TO FLAG FOLLOWING THE TYPE OF KEY VARIABLE
          CLI     12(10),C' '
          BE      FC
          CP      12(2,10),=P'0'
          BE      FI
          MVI     FLAG,C'R'
* DETERMINE EACH KEY VALUE
LOAD      CLI     B,C'1'
```

```
         BE    EXIT
         MVC   KEY(16),KEY-1
         MVC   KEY2(15),KEY2-1
         MVI   S,C'A'
         A     5,=F'1'
         LA    R,KEY
         CLI   0(5),C'('
         BE    ADKEY
CK       MVC   0(1,R),0(5)
         A     R,=F'1'
         A     5,=F'1'
         CLI   0(5),',
         BE    ASKCIR
         CLI   0(5),C' '
         BE    XY
         CLI   0(5),C'-'
         BNE   CK
         MVI   R,C' '
         LA    R,KEY2
         A     5,=F'1'
MVCC     MVC   0(1,R),0(5)
         A     5,=F'1'
         CLI   0(5),C','
         BE    ASKCIR
         CLI   0(5),C' '
         BE    XY
         A     5,=F'1'
         B     MVCC
ALL      FSREAD FSCB=RFSCB,ERROR=ECF
         AP    PW(3),=PL3'1'
         BAL   11,PROC
         B     ALL
FC       MVI   FLAG,C'C'
         B     LOAD
FI       MVI   FLAG,C'I'
         B     LOAD
XY       MVI   R,C'1'
         B     ASKCIR
ADKEY    A     5,=F'1'
         CLI   0(5),C'L'
         BL    ER4
         CLI   0(5),C'9'
         BH    ER4
         PACK  PP(1),0(1,5)
         CLI   1(5),C' '
         BNL   NZ
SOS      ZAP   N,=P'0'
TUV      AP    N,=P'1'
         CLI   TAG,C'1'
         BE    RDF
         MVI   TAG,C'1'
CNP      CP    N,PP
         BNL   RPC
         BAL   12,READDF
         AP    N(1),=P'1'
```

```
         B     CMP
ADF      BAL   12,READDF
         CP    N,PP
         BL    TUV
IPC      BAL   11,PROC
         B     SCS
LRV      WRITERM 'IT MUST BE NUMERIC I-V NEXT TO ''('''
         B     EXIT
W2       WRITERM '**WARNING--THE MESSAGE FOLLOWING (X WON''T BE DETERMI
               ED'
         B     SCS
ASKCIR   CLI   FLAG,C'C'
         BE    FDDC
         CLI   FLAG,C'I'
         BE    FDDI
         B     FDDF
* THE COMPARISON OF KEY VALUE AND DATA BETWEEN GIVEN COL. IN CASE OF
* THE TYPE OF THAT KEY IS CHARACTER FORMAT
FDDC     CLI   TAG,C'L'
         BE    FDD
         MVI   TAG,C'L'
         B     NFD
FDD      BAL   12,READDF
NFD      L     7,F
         LA    3,KEY
         CLI   H,C'='
         BNE   FIXK
* THE INTERVAL OF KEY VALUE
COMC     CLC   0(1,7),0(3)
         BL    FDD
         CLC   0(1,7),0(3)
         BH    NKY
         A     3,=F'1'
         A     7,=F'1'
         C     7,L
         BH    FRCR
         CLI   0(3),C' '
         BNE   COMC
         BAL   11,PROC
         B     FDD
FROR     BAL   11,PROC
NEXTKEY  BAL   12,READDF
NKY      L     7,F
         LA    3,KEY2
COMC2    CLC   0(1,7),0(3)
         BL    PROR
         BH    CLEAR
         A     3,=F'1'
         A     7,=F'1'
         C     7,L
         BH    PROF
         CLI   0(3),C' '
         BNE   COMC2
         B     PROR
* SINGLE KEY VALUE
```

```
FIXK     CLC   0(2,7),0(3)
         BL    RDD
         BH    CLOAD
         A     3,=F'1'
         C     3,=A(KEY+15)
         BH    PL
         A     7,=F'1'
         C     7,L
         BH    PL
         CLI   0(3),C' '
         BNE   FIXK
PL       BAL   11,PROC
         B     RDD
READDF   FSREAD FSCB=RFSCB,ERROR=ECF
         AP    PW(3),=PL3'1'
         BR    12
* THE COMPARISON OF KEY VALUE AND DATA BETWEEN GIVEN COL. IN CASE OF
* THE TYPE OF THAT KEY IS INTEGER FORMAT.
* BOTH KEY VALUE AND DATA ARE CHANGED TO PACK FORM.
RDDI     ST    5,FIVE
         MVC   KEYI8(15),KEYI8-1
         LA    10,PPI
         MVI   0,C' '
         LA    5,KEY
         LA    6,KEY+15
         LA    3,KEYI8+14
         BAL   7,CGX
         CLI   KEY2,C' '
         BE    F115
         A     10,=F'8'
         MVC   KEYI8(15),KEYI8-1
         LA    6,KEY2+14
         LA    3,KEYI8+14
         LA    5,KEY2
         BAL   7,CGX
F115     CLI   TAG,C'1'
         BNE   BLAPPI
         L     10,=A(PPI+16)
         BAL   8,FI
LAPPI    LA    10,PPI
         CLI   H,C'-'
         BNE   FIXKI
* THE INTERVAL OF INTEGER KEY VALUES
         CP    16(8,10),0(8,10)
         BL    F115
         BH    CX3
CAL      BAL   11,PROC
         L     10,=A(PPI+16)
         BAL   8,FI
         LA    10,PPI
CX3      CP    16(8,10),8(8,10)
         BH    BLOAD
         B     CAL
BLAPPI   MVI   TAG,C'1'
         B     LAPPI
```

```
ERRA    ARTERM 'INTEGER KEY ISN''T NUMERIC.'
        B      EXIT
FI      BAL    12,READDF
        MVC    KEYI8(15),KEYI8-1
        LA     5,KEYI8+14
        L      6,L
        L      5,F
        BAL    7,C6X
        BR     8
* SINGLE INTEGER KEY VALUE
FIXKI   CP     16(8,10),0(8,10)
        BH     BLOAD
        BL     FIIS
        BAL    11,PROC
        B      FIIS
C6X     CLI    0(5),C' '
        BNL    UNY
        A      5,=F'1'
        B      C6X
C56     CR     5,6
        BH     ERRA
ZRO     CLI    0(5),C'0'
        BNE    LOT
        A      5,=F'1'
        MVI    Z,C'*'
        B      ZRO
UNY     CLI    0(5),C'-'
        BNE    C56
        MVI    U,C'-'
        A      5,=F'1'
        B      C56
LOT     CLI    0(6),C' '
        BNE    LLL
        S      6,=F'1'
        CR     6,5
        BNL    LOT
        CLI    Z,C'*'
        BNE    ERRA
        MVI    Z,C' '
        B      PC
LLL     CLI    0(6),C'0'
        BL     ERRA
        CLI    0(6),C'9'
        BH     ERRA
        MVC    0(1,3),0(6)
        S      6,=F'1'
        S      3,=F'1'
        CR     6,5
        BNL    LLL
PC      PACK   0(8,10),KEYI8(15)
        CLI    U,C'-'
        BNE    A10
        MVI    U,C' '
        MP     0(8,10),=P'-1'
A10     BR     7
```

```
FLOAD   MVI   CODE,C'0'
BLOAD   L     5,FIVE
        MVI   U,C' '
CLOAD   MVI   TAG,C'2'
        MVI   H,C' '
        B     LOAD
* THE COMPARISON BETWEEN REAL KEY AND DATA IN GIVEN COL.
ADDR    ST    5,FIVE
        LA    6,KEY
        LA    2,KEY18+15
        ZAP   DOL,12(2,10)
        CVB   5,DOL
        SR    2,5
        LR    4,2
        S     4,=F'1'
        ST    4,FFP
        ST    2,BFP
SCC     MVC   KEY18(15),KEY18-1
        LA    3,KEY18+14
        MVI   U,C' '
        CLI   0(6),C'-'
        BNE   *+KY2
        MVI   U,C'-'
        A     6,=F'1'
MKY2    LA    7,14(0,6)
        L     4,FFP
        L     2,BFP
        LA    5,KEY18
C8      CLI   0(6),C'.'
        BE    D2
        A     6,=F'1'
        CLI   0(6),C' '
        BE    BSEC2
        CR    6,7
        BH    BSEC2
        B     C8
BSEC2   S     6,=F'1'
        B     SEC2
D2      LR    8,6
        S     6,=F'1'
D1      A     8,=F'1'
        CLI   0(8),C' '
        BE    SEC2
        CR    8,7
        BH    SEC2
        MVC   0(1,2),0(8)
        A     2,=F'1'
        CR    2,3
        BNH   D1
SEC2    CLI   CODE,C'0'
        BNE   CR68
        LA    8,KEY
        B     CAB
CR68    LA    8,KEY2
CAB     CR    6,8
```

```
         BL    PDC
         MVC   0(1,4),0(6)
         S     4,=F'1'
         S     6,=F'1'
         B     CAB
PDC      CLI   CODE,C'2'
         BE    TAXI
         CLI   CODE,C'1'
         BE    BUS
         PACK  RK1(8),KEY18(15)
         CLI   U,C'-'
         BNE   TOK
         MP    RK1(8),=P'-1'
TOK      CLI   H,C'-'
         BNE   PRD
         MVI   CODE,C'2'
         LA    6,KEY2
         B     SCC
TAXI     PACK  RK2(8),KEY18(15)
         CLI   U,C'-'
         BNE   PRD
         MP    RK2(8),=P'-1'
PRD      CLI   TAG,C'1'
         BNE   BSSS
         BAL   12,READDF
         MVC   KEY2(15),KEY2-1
         L     8,F
         LA    12,KEY2
CVV      CLI   0(8),C' '
         BNE   CLL
         A     8,=F'1'
         B     CVV
BSSS     MVI   TAG,C'1'
         B     SSS
CLL      CLI   0(8),C'-'
         BE    MRD
CDE      CLI   0(8),C'.'
         BE    MRD
         CLI   0(8),C'0'
         BL    ER16
         CLI   0(8),X'FA'
         BL    MRD
ER16     WRTERM 'KEY BETWEEN GIVEN COLUMN IN THE FILE ISN''T NUMERIC.'
         B     EXIT
MRD      MVC   0(1,12),0(8)
         A     8,=F'1'
         CLI   0(8),C' '
         BE    KU
         C     8,L
         BH    KU
         A     12,=F'1'
         B     CDE
KU       MVI   CODE,C'1'
         LA    6,KEY2
         B     SCC
```

```
BUS       PACK  RK3(3),KEY19(15)
          CLI   U,C'-'
          BNE   SSS
          MP    RK3(8),=P'-1'
SSS       CLI   S,C'B'
          BE    SHELL
          CLI   U,C'-'
          BNE   FIXR
* DETERMINE WHENGIVING THE INTERVAL OF KEY VALUE
          CP    RK3(8),RK1(8)
          BL    PRD
          CP    RK3(8),RK1(8)
          BH    SHELLB
SHL       BAL   11,PROC
          B     PRD
SHELLB    MVI   S,C'B'
SHELL     CP    RK3(8),RK2(8)
          BH    RLOAD
          BAL   11,PROC
          B     PRD
* SINGLE REAL KEY
FIXR      CP    RK3(8),RK1(8)
          BH    RLOAD
          CP    RK3(8),RK1(8)
          BL    PRD
          BAL   11,PROC
          B     PRD
PROC      LA    1,ARGTPC
          L     15,=V(PROCESS)
          BALR  14,15
          BR    11
AFSCB     FSCB  'DATA DATA A1',BUFFER=BFD,BSIZE=130
BASE      DC    A(INCBASE+4096)
ELEV      DS    11F
THB       DS    3F
DFN       DC    8C' '
DF        DC    130C' '
YN        DS    CL1
          DC    C' '
KEY       DC    16C' '
DGL       DS    D
BFD       DS    CL130
F         DS    F
L         DS    F
FLAG      DS    C'R'
B         DC    C' '
H         DC    C' '
          DC    C' '
KEY2      DC    15C' '
PP        DS    PL1
N         DS    PL1
          DC    C'0'
KEY18     DC    15C' '
U         DC    C' '
FIVE      DS    F
```

```
FFP     DS    F
BFP     DS    F
PPI     DC    240' '
RK1     DC    PL8'0'
RK2     DC    PL8'0'
RK3     DC    PL8'0'
CODE    DC    C'0'
S       DS    CL1
TAG     DS    CL1
SAVE    DS    F
NF      DC    8C' '
ANS     DS    CL1
SECOL   DS    CL10
BFIN    DS    CL280
BFOUT   DS    CL280
STATE   DS    CL1600
PRKEY   DS    CL237
VAR     DS    CL200
XPSION  DS    CL700
TABLE   DS    CL240
WW      DS    PL3
REG9    DS    F
REG13   DS    F
MBEGIN  DS    CL1
HEADG   DS    CL1
FNL     DS    CL1
Z       DC    C' '
ARGTPC  DC    A(BFIN)
        DC    A(STATE)
        DC    A(VAR)
        DC    A(XPSION)
        DC    A(TABLE)
        DC    A(NF)
        DC    A(WW)
        DC    A(BFD)
        DC    A(HEADG)
        FSCBD
        END
```

```
FILE: PROCESS  ASSEMBLE V1                    . VM/SP CONVERSATIONAL MONI

*********************************************************************
* PROCESS ALL ARITHMETIC STATEMENTS AND EXPRESSIONS, THEN DISPLAYS
* THE RESULTS AT CRT.
*********************************************************************
PROCESS  CSECT
         BALR  12,0
         USING FIRST,12,2,3
FIRST    LM    2,3,BASE2
         B     STR14
BASE2    DC    A(FIRST+4096,FIRST+8192)
STR14    STM   13,1,RGT
         STM   4,11,RGT2
         L     9,0(1)
         MVC   BFIN(256),0(9)
         MVC   BFIN+256(24),256(9)
         L     9,4(1)
         MVC   STATE(256),0(9)
         MVC   STATE+256(256),256(9)
         MVC   STATE+512(256),512(9)
         MVC   STATE+768(256),768(9)
         MVC   STATE+1024(256),1024(9)
         MVC   STATE+1280(256),1280(9)
         MVC   STATE+1536(64),1536(9)
         L     9,8(1)
         MVC   VAR(200),0(9)
         L     9,12(1)
         MVC   XPSION(256),0(9)
         MVC   XPSION+256(256),256(9)
         MVC   XPSION+512(188),512(9)
         L     9,16(1)
         MVC   TABLE(240),0(9)
         L     9,20(1)
         MVC   NF(8),0(9)
         L     9,24(1)
         MVC   AA(3),0(9)
         L     9,28(1)
         MVC   BFD(130),0(9)
         L     9,32(1)
         MVC   HEADG(1),0(9)
         CLI   HEADG,C'Y'
         BNE   JMP
         MVI   0(9),C'N'
         MVI   HEADG,C'*'
JMP      MVC   SCREEN(80),SCREEN-1
         LA    13,VAR
* DETERMINE BUFFER 'STATE'
         LA    9,STATE
         L     5,=A(STATE+1600)
         ST    5,SMOXPN
* 'SMOXPN' CONTAINS (THE LAST ADDRESS OF STATE +1)   OR
*                   (THE LAST ADDRESS OF XPSION +1)
         CLI   0(9),C' '
         BE    EXPS
TT       MVC   STORE(200),STORE-1
         MVC   STACK(20),STACK-1
         LA    8,STORE-10
         LA    4,STACK-1
         ST    9,ECHO
         A     9,=F'79'
         ST    9,FAR
```

```
         L     9,ECHO
         ST    13,COUNT
MB2      MVC   0(1,13),0(9)
         A     9,=F'1'
         A     13,=F'1'
         CLI   0(9),C'='
         BNE   MB2
         L     13,COUNT
         A     13,=F'8'
         L     7,FAR
LOOP     A     9,=F'1'
         CR    9,7
         BH    TCA
* PUSH OPEN PARENTHESIS
C1       CLI   0(9),C'('
         BNE   C2
S1       A     4,=F'1'
         MVC   0(1,4),0(9)
         B     ALCOP
* IN CASE OF FINDING '+' AND '-'
C2       CLI   0(9),C'+'
         BE    S2
         CLI   0(9),C'-'
         BNE   C3
         S     9,=F'1'
         CLI   0(9),C'('
         BE    UNARY
         CLI   0(9),C'*'
         BE    UNARY
         CLI   0(9),C'+'
         BE    Z1
         CLI   0(9),C'-'
         BE    Z2
         CLI   0(9),C'/'
         BE    UNARY
         CLI   0(9),C'='
         BE    UNARY
         CLI   PQ,C'2'
         BNE   ADDRE9
         C     9,A2XPSION
         BL    UNARY
ADDRE9   A     9,=F'1'
S2       ZAP   FLAG,=P'0'
         ZAP   0(2),=P'0'                    ============
         MVI   POINT,C'N'
SS2      C     4,ST2
         BE    S1
         CLI   0(4),C'('
         BE    S1
         MVC   P1,0(4)
         S     4,=F'1'
         LA    5,SS2
S3       CLI   P1,C'+'
         BE    ADD
         CLI   P1,C'-'
```

```
         BE    SUBTRACT
S4       CLI   P1,C'*'
         BE    MULTIPLY
         CLI   P1,C'/'
         BE    DIVIDE
         CLI   P1,C'¬'
         BE    UNA
         CLI   P1,C'@'
         BE    EXPONENT
         MNOTE 'THE NUMBER OF ( ISN''T EQUAL TO ) '
         B     EXIT
Z1       MVI   0(4),C'-'
         A     9,=F'1'
         B     ALOOP
Z2       MVI   0(4),C'+'
         A     9,=F'1'
         B     ALOOP
UNARY    A     4,=F'2'
         MVI   0(4),C'¬'
         A     4,=F'1'
         B     ALOOP
* IN CASE OF FINDING '*'
C3       CLI   2(9),C'*'
         BNE   DIV
* IN CASE OF FINDING '**'
         CLI   1(9),C'*'
         BNE   S5
         A     4,=F'1'
         MVI   0(4),C'@'
         A     9,=F'1'
         ZAP   FLAG,=P'0'
         ZAP   0(2),=P'0'
         MVI   POINT,C'N'
         B     ALOOP
* IN CASE OF FINDING '/'
DIV      CLI   0(9),C'/'
         BNE   C4
S5       ZAP   FLAG,=P'0'
         ZAP   0(2),=P'0'
         MVI   POINT,C'N'
SS5      C     4,ST2
         BE    S1
         CLI   0(4),C'('
         BE    S1
         CLI   0(4),C'+'
         BE    S1
         CLI   0(4),C'-'
         BE    S1
         MVC   P1,0(4)
         S     4,=F'1'
         LA    5,SS5
         B     S4
* IN CASE OF FINDING CLOSE PARENTHESIS
C4       CLI   0(9),C')'
         BNE   C5
```

```
         ZAP   FLAG,=P'0'
         ZAP   D(2),=P'0'
         MVI   POINT,C'N'
S6       CLI   D(4),C'('
         BNE   S7
         S     4,=F'1'
         B     ALCOP
S7       MVC   PI,D(4)
         S     4,=F'1'
         LA    5,S6
         B     S8
* IN CASE OF FINDING BLANK
C5       CLI   D(0),C' '
         BNE   C6
TOA      ZAP   FLAG,=P'0'
         ZAP   D(2),=P'0'
         MVI   POINT,C'N'
S8       C     4,ST2
         BE    NEXT
         MVC   PI,D(4)
         S     4,=F'1'
         LA    5,S8
         B     S8
NEXT     CLI   PT,C'2'
         BNE   NEXSTATE
CMR2     CP    B(2,8),=P'2'
         BNH   NCUT
AGAIN    ZAP   TEMP1(10),C(8,8)
         DP    TEMP1(10),TEN
         ZAP   D(8,8),TEMP1(8)
         SP    B(2,8),=P'1'
         CP    B(2,8),=P'2'
         BNE   AGAIN
         CP    TEMP1+8(2),=PL2'C'
         BNL   CTN
         AP    TEMP1+8(2),=P'-1'
CTN      CP    TEMP1+8(2),=P'5'
         BL    BEDIT
         CP    TEMP1(8),=P'C'
         BL    LSTH
         AP    D(8,8),=P'1'
         B     BEDIT
LSTH     SP    D(8,8),=P'1'
         B     BEDIT
NCUT     CP    B(2,8),=P'C'
         BE    BMSK
BEDIT    MVC   TRUE(20),FALSE
         ZAP   VSTM(8),D(8,8)
         ZAP   PENDR(2),B(2,8)
         L     5,FSCL
         B     ZPSV
BMSK     ZAP   TEMPLTEN(8),C(8,8)
         L     5,FSCL
CHMKI    MVC   MASK(20),FALSE2
         ST    7,0
```

```
         B     ASK2
* IN CASE OF FINDING NUMERIC
C6       CLI   0(9),X'F0'
         BL    C7
         CLI   0(9),X'F9'
         BH    EE1
* CHANGE CHARACTER FORM TO PACK FORM
         CP    FLAG,=P'0'
         BNE   MULSUB
         LA    8,10(8)
         ZAP   0(8,8),=P'0'
         ZAP   8(2,8),=P'0'
         ZAP   FLAG,=P'1'
MULSUB   PACK  KEEP,0(1,9)
         MP    0(8,8),TEN
         AP    0(8,8),KEEP
         CLI   POINT,C'N'
         BE    CARITH
         AP    8(2,8),=P'1'
CARITH   CLI   ARITHFC,C'Y'
         BE    FCC
         B     ALOOP
* IN CASE OF FINDING '.'
C7       CLI   0(9),C'.'
         BNE   CC7
         MVI   POINT,C'Y'
         CLI   ARITHFC,C'Y'
         BE    FCC
         B     ALOOP
CC7      CLI   ARITHFC,C'Y'
         BE    EW
         MVC   V(8),=8C' '
         LA    6,V
* IN CASE OF FINDING CHARACTER A-Z
STP      CLI   0(9),X'C1'
         BL    EE1
         CLI   0(9),C'Z'
         BH    EE1
M62      MVC   0(1,6),0(9)
         A     6,=F'1'
         A     9,=F'1'
* IF THERE IS '(' BEHIND A CHARACTER A-Z,FUNCTION MUST BE DETERMINED
         CLI   0(9),C'('
         BE    FUNC
* IF THERE IS +,-,),*,/OR BLANK BEHIND CHARACTER OR NUMERIC,THE GROUP
* OF THOSE CHARACTER AND NUMERIC IS VARIABLE NAME.
         CLI   0(9),C'+'
         BE    VALUE
         CLI   0(9),C')'
         BE    VALUE
         CLI   0(9),C'-'
         BE    VALUE
         CLI   0(9),C'*'
         BE    VALUE
         CLI   0(9),C'/'
```

```
         BE    VALUE
         CLI   0(9),C' '
         BE    VALUE
         C     9,SMLXPN
         BE    VALUE
         CLI   0(9),C')'
         BL    STP
         CLI   0(9),C'9'
         BH    EEIR
         B     M62
EEIR     WRTERM '**EXPRESSION IS INVALID**'
         B     EXIT
* STEP OF FINDING THE RESULT OF THE FUNCTION
FUNC     CLC   V(8),=CL8'SUM'
         BE    SRTSUM
         CLC   V(8),=CL8'MAX'
         BE    SRTMAX
         CLC   V(8),=CL8'MIN'
         BE    SRTMIN
         CLC   V(8),=CL8'FAC'
         BE    SRTFAC
         CLC   V(8),=CL8'SIN'
         BNE   N1
         MVI   ID,C'1'
         B     FORT
N1       CLC   V(8),=CL8'COS'
         BNE   N2
         MVI   ID,C'2'
         B     FORT
N2       CLC   V(8),=CL8'TAN'
         BNE   N3
         MVI   ID,C'3'
         B     FORT
N3       CLC   V(8),=CL8'SQRT'
         BNE   N4
         MVI   ID,C'4'
         B     FORT
N4       CLC   V(8),=CL8'ALOG'
         BNE   N5
         MVI   ID,C'5'
         B     FORT
N5       CLC   V(8),=CL8'ALOG10'
         BNE   N6
         MVI   ID,C'6'
         B     FORT
N6       CLC   V(8),=CL8'EXP'
         BNE   N7
         MVI   ID,C'7'
         B     FORT
N7       CLC   V(8),=CL8'ABS'
         BNE   N8
         MVI   ID,C'8'
FORT     MVI   ARITHFC,C'Y'
         B     SAME2
DETERMIN CLI   V,C'='
```

```
         BNL   CVC
         MVI   FRONT,C'-'
         MVC   V(7),V+1
         MVI   V+7,C' '
CVD      CLI   V,C'.'
         BE    SJJ
         CLI   V,C'0'
         BL    VALUE
         CLI   V,C'9'
         BH    VALUE
SJJ      ST    9,JJ
         LA    9,V
         B     CC
FCC      A     9,=F'1'
         CLI   0(9),C' '
         BE    BCAL
         C     9,=A(V+8)
         BNL   BCAL
         B     CC
BCAL     L     9,JJ
REORTGI  MVI   ARITHEC,C'A'
         CLI   ID,C'8'
         BNE   CFT
         MVI   FRONT,C' '
         B     ALCOP
CFT      CLI   FRONT,C'-'
         BNL   UNDER
         MP    0(8,8),=P'-1'
         MVI   FRONT,C' '
UNDER    ST    4,JJ
         ZAP   LEIGHT(8),0(8,8)
         CVB   4,LEIGHT
         ST    4,ISEND
         ZAP   DOUBLEC(8),8(2,8)
         CVB   4,DOUBLEC
         ST    4,MYDEC
         PACK  L1(1),ID(1)
         ZAP   DOUBLEC(8),L1(1)
         CVB   4,DOUBLEC
         ST    4,INDEX
         L     4,JJ
         CLI   ID,C'9'
         BE    FNE
         USEMCF ID,V,LEIGHT,MYDEC
         B     LJJ
FNE      EQU   *
         MACFUNC V,LEIGHT,MYDEC
* CALL FORTRAN SUBPROGRAM .....................
LJJ      ST    13,THIRTEEN
         LA    13,SAVEREG
FORTGI   LA    1,ARGFORT
         L     15,=V(FUNCT)
         BALR  14,15
         L     13,THIRTEEN
         ST    6,DEP
```

```
         L      6,MEMO
         CVD    6,DOUBLEC
         ZAP    0(8,8),DOUBLEC(8)
         ZAP    8(2,8),=PL2'4'
         L      6,DEP
         B      ALOOP
N8       MVI    IC,C'9'
         B      PORT
* FINDING THE MAXIMUM VALUE
SRTMAX   MVI    FMAX,C'Y'
         B      SAME
* FINDING THE MINIMUM VALUE
SRTMIN   MVI    FMIN,C'Y'
         B      SAME
* THE METHOD OF SUMMING VALUES
SRTSUM   MVI    FSUM,C'Y'
SAME     ST     7,SVN
SAME2    MVC    V(8),=8C' '
         ST     6,STSIX
         LA     6,V
         A      9,=F'1'
MVV      MVC    0(1,6),0(9)
         A      6,=F'1'
         A      9,=F'1'
         CLI    0(9),C')'
         BE     EQU
         C      6,=A(V+8)
         BNL    EW
         B      MVV
EQU      L      6,STSIX
         CLI    ARITHEC,C'Y'
         BE     DETERMIN
         CLI    FSUM,C'Y'
         BE     FSY
         CLI    FMIN,C'Y'
         BE     FIY
         LA     7,ARGMAX
CMAX     CLI    0(7),C' '
         BE     FVMAX
         CLC    V(8),0(7)
         BE     VLMAX
         A      7,=F'18'
         C      7,=A(ARGMAX+9C)
         BNL    EMAX
         B      CMAX
FIY      LA     7,ARGMIN
CMIN     CLI    0(7),C' '
         BE     FVMAX
         CLC    V(8),0(7)
         BE     VLMIN
         A      7,=F'18'
         C      7,=A(ARGMIN+9C)
         BNL    EMIN
         B      CMIN
FSY      LA     7,ARGSUM
```

```
CSUM     CLI   0(7),C' '
         BE    FVMAX
         CLC   V(8),0(7)
         BE    VLSUM
         A     7,=F'18'
         C     7,=A(ARGSUM+9C)
         BNL   FSUM
         B     CSUM
VLMAX    MVI   FMAX,C'N'
         B     GETVL
VLMIN    MVI   FMIN,C'N'
         B     GETVL
VLSUM    MVI   FSUM,C'N'
GETVL    LA    8,10(8)
         ZAP   0(8,8),8(8,7)
         ZAP   8(2,8),16(2,7)
         L     7,SVN
         B     ALCCP
FVMAX    ST    6,STC
         LA    6,RFSCB
         USING FSCBD,6
         MVC   FSCBFN(8),NF
         L     6,STC
         FSCLOSE FSCB=RFSCB
         FSSTATE FSCB=RFSCB,ERRCD=ERR1
FRD      FSREAD FSCB=RFSCB,EOFCD=EEE
         B     VALUE
AX       CP    RSM(8),0(8,8)
         BNL     PIM
TRSL     ZAP   RSM(8),0(8,8)
PIM      S     8,=F'10'
         B     FRD
INL      CP    RSM(8),0(8,8)
         BNH     PIM
         B       TRSL
SUMM     AP    RSM(8),0(8,8)
         S     8,=F'10'
         B     FRD
EEE      C     15,=F'12'
         BNE   ERR2
         LA    15,0
         A     8,=F'10'
         ZAP   0(8,8),RSM(8)
         CP    8(2,8),=P'C'
         BNE   ZSV
         ZAP   TEMPLTEN(8),RSM(8)
         B     CFMXI
ZSV      ZAP   SV(8),8(2,8)
         ZAP   TEMPLTEN(8),RSM(8)
         MVC   TRUE(20),FALSE
         B     CVBSV
* PREPARE FOR DISPLAYING THE RESULT OF 'SUM', 'MAX'OR 'MIN'
WMAX     CLI   FMAX,C'Y'
         BE    WMAX
         CLI   FMIN,C'Y'
```

```
         BE    MMIN
         MVC   FNAME(3),=CL3'SUM'
         MVI   FSUM,C'N'
         LA    7,ARGSUM
WLN      WRTERM WLINE,35
         FSCLOSE FSCB=RFSCB
         FSSTATE FSCB=RFSCB,ERRCD=ERR1
         ZAP   CTR(3),=P'1'                    CCCCCCCCCCCC
FD22     FSREAD FSCB=RFSCB,ERRCD=ECF
         CP    CTR(3),NN
         BE    PVX
         AP    CTR(3),=P'1'
         B     FD22
PVX      CLI   C(7),C' '
         BE    PVAX
         A     7,=F'18'
         B     PVX
PVAX     MVC   0(8,7),V
         ZAP   8(8,7),RSM(8)
         ZAP   16(2,7),8(2,8)
         L     7,SVN
         ZAP   RSM(8),=PL8'0'
         MVI   N1,C'1'
         B     ALLOP
MMAX     MVC   FNAME(3),=CL3'MAX'
         MVI   FMAX,C'N'
         LA    7,ARGMAX
         B     WLN
MMIN     MVC   FNAME(3),=CL3'MIN'
         MVI   FMIN,C'1'
         LA    7,ARGMIN
         B     WLN
ESUM     WRTERM 'TOO MANY SUM FUNCTIONS'
         B     EXIT
EMAX     WRTERM 'TOO MANY MAX FUNCTIONS'
         B     EXIT
EMIN     WRTERM 'TOO MANY MIN FUNCTIONS'
         B     EXIT
         CLC   V(8),=CL8'FAC'
         BE    SRTFAC
         B     EXIT
* STEP OF FINDING THE RESULT OF FACTORIAL FUNCTION
SRTFAC   MVI   KY,C'F'
LAV      LA    6,V
         ZAP   ITG(4),=P'0'
         MVC   V(8),=8C' '
         A     9,=F'1'
MOV62    MVC   0(1,6),0(9)
         A     6,=F'1'
         A     9,=F'1'
         CLI   0(9),C')'
         BNE   MOV62
         A     9,=F'1'
         LA    6,V
         CLI   V,C'A'
         BL    W
         CLI   V,X'FA'
```

```
         BL    VALUE
V09      CLI   V,C'0'
         BL    EW
         CLI   V,C'9'
         BH    EW
         PACK  KEEP,C(1,6)
         MP    ITG(4),TEN
         AP    ITG(4),KEEP
         A     5,=F'1'
         CLI   0(6),C' '
         BNE   V09
FACTR    ZAP   IFACT(8),=P'1'
         ZAP   I(4),=P'2'
COMIITG  CP    I(4),ITG(4)
         BH    BFEXIT
         MP    IFACT(8),I(4)
         AP    I(4),=P'1'
         B     COMIITG
BFEXIT   LA    8,10(8)
         ZAP   0(8,8),IFACT(8)
         ZAP   8(2,8),=P'0'
         MVI   0(8),C'0'
         B     ALOOP
MTFAC    ZAP   ITG(4),0(8,8)
         S     8,=F'10'
         B     FACTR
EFALSE   WRTERM 'VALUE OF VARIABLE IN FACTORIAL FUNCTION ISN''T INTEGER'
                '
         B     EXIT
EW       WRTERM 'ARGUMENT IN THE FUNCTION DOESN''T MAKE SENSE.'
         B     EXIT
* FINDING VALUE OF A VARIABLE
VALUE    LA    6,BFIN
         LA    10,NAR
CV       CLC   V(8),0(5)
         BE    FETCH
         A     6,=F'14'
         CLI   0(6),C' '
         BE    NSTATE
         C     5,=A(BFIN+280)
         BL    CV
NSTATE   CLC   0(8,10),=8C' '
         BE    ERX
         CLC   V(8),0(10)
         BE    EARITH
         A     10,=F'18'
         C     10,=A(NAR+360)
```

```
         BL    NSTATE
ERX      LINEDIT TEXT='THERE ISN'T THE VALUE OF ..... ',SUB=(CHARA
               ,V)
         B     EXIT
* USE THIS PART TO FIND THE VALUE OF A VARIABLE EXISTING IN
* INPUT FORMAT BUFFER.
FETCH    CLI   KY,C'F'
         BNE   ZD
         CP    12(2,6),=P'0'
         BNL   EFALSE
ZD       ZAP   DOUBLEC,8(2,6)
         CVB   10,DOUBLEC
         ST    5,XX
         LA    5,BFD
         S     5,=F'1'
         AR    10,5
         ZAP   DOUBLEC,10(2,6)
         CVB   11,DOUBLEC
         AR    11,5
         L     5,XX
         CLI   12(6),C' '
         BE    EE1
XIS      LA    8,10(8)
         ZAP   8(2,8),12(2,6)
         CP    12(2,6),=P'0'
         BNE   REAL
* THE METHOD OF GETTING THE VALUE OF INTEGER VARIABLE FROM DATA RECORD
         ZAP   0(8,8),=P'0'
C10      CLI   0(10),C' '
         BE    XOX
         CLI   0(10),C'-'
         BNE   FIRSTCRT
         MVI   MNSIGN,C'-'
         A     10,=F'1'
FIRSTCRT CLI   0(10),C'0'
         BE    XOX2
         CLI   0(10),C'1'
         BL    EE1
         CLI   0(10),C'9'
         BH    EE1
PK11     PACK  KEEP,0(1,10)
         MP    0(8,8),TEN
         AP    0(8,8),KEEP
         A     10,=F'1'
         CLI   0(10),C' '
         BE    SL2
         CR    10,11
         BH    SL2
         CLI   0(10),C'0'
         BL    EE1
         CLI   0(10),C'9'
         BH    EE1
         B     PK11
XOX      A     10,=F'1'
         CR    10,11
```

```
         BH    EE1
         B     CIC
XOX2     A     10,=F'1'
         CR    10,11
         BH    SL2
         CLI   0(10),C' '
         BNE   DIFSTART
SL2      CLI   MNSIGN,C'-'
         BNE   ASKFLAG
         MP    0(8,3),=P'-1'
         MVI   MNSIGN,C' '
ASKFLAG  CLI   FSUM,C'Y'
         BE    SUMM
         CLI   FMAX,C'Y'
         BE    BFAX
         CLI   FMIN,C'Y'
         BE    BFINL
         CLI   ARITHFC,C'Y'
         BE    BFORTGI
         C     9,SMOXPN
         BNL   TOA
         CLI   KY,C'F'
         BE    MTFAC
         S     9,=F'1'
         B     ALOOP
BFAX     CLI   11,C'1'
         BNL   AX
MOVETRSM ZAP   RSM(8),0(8,8)
         S     8,=F'10'
         MVI   11,C'2'
         B     FRD
BFINL    CLI   11,C'1'
         BNE   INL
         B     MOVETRSM
EE1      LINEDIT TEXT='THE VALUE OF ..... ISN''T NUMERIC. IT CAN''T
               BE IN THE ARITHMETIC STATEMENT.',SUB=(CHARA,V)
         B     EXIT
* USE THIS PART TO FIND THE VALUE OF THE VARIABLE CONTAINING THE RESUL
* OF ANY ARITHMETIC STATEMENT
FARITH   CLI   KY,C'F'
         BNE   L88
         CP    16(2,10),=P'0'
         BNE   FFALSE
L88      LA    8,10(8)
         ZAP   0(8,3),8(8,10)
         ZAP   8(2,8),16(2,10)
         B     ASKFLAG
* LOOP BACK TO DETERMINE NEXT ARITHMETIC STATEMENT
NEXSTATE ZAP   0(8,13),0(8,8)
         ZAP   8(2,13),8(2,8)
         L     9,ECHO
         A     9,=F'80'
         CLI   0(9),C' '
         BE    EXPS
* IF THERE IS NO ANY STATEMENT LEFT, BRANCH TO DETERMINE TABLE 'VAR'
```

```
         C     9,=A(STATE+1599)
         BH    EXPS
         L     13,COUNT
         A     13,=F'18'
         B     TT
* USE THESE INSTRUCTIONS WHEN ADDITION IS NEEDED
ADD      S     8,=F'10'
         CP    8(2,8),18(2,E)
         BE    AP88
         CP    8(2,8),18(2,E)
         BL    CON
         ZAP   D(2),18(2,E)
         ZAP   VACH(8),1C(8,E)
MPBT     MP    VACH(8),TEN
         AP    D(2),=PL2'1'
         CP    8(2,8),D(2)
         BH    MPBT
         AP    D(8,8),VACH(8)
         BR    5
CON      ZAP   D(2),8(2,8)
         ZAP   VACH(8),0(8,8)
MPP      MP    VACH(8),TEN(2)
         AP    D(2),=PL2'1'
         CP    D(2),18(2,8)
         BL    MPP
         AP    1C(8,8),VACH(E)
         ZAP   C(8,8),10(8,8)
         ZAP   8(2,8),18(2,8)
         BR    5
AP88     AP    D(8,8),10(E,E)
         BR    5
* EXECUTE THIS PART WHEN SUBTRACTION IS NEEDED
SUBTRACT S     8,=F'10'
         CP    8(2,8),18(2,E)
         BE    S88
         BL    CON2
         ZAP   D(2),18(2,E)
         ZAP   VACH(8),1C(8,E)
MP2      MP    VACH(8),TEN
         AP    D(2),=P'1'
         CP    8(2,8),D(2)
         BH    MP2
STR      SP    D(8,8),VACH(E)
         BR    5
CON2     ZAP   VACH(8),0(E,8)
MMP      MP    VACH(8),TEN
         AP    8(2,8),=P'1'
         CP    8(2,8),18(2,E)
         BL    MMP
         SP    VACH(8),1C(8,E)
         ZAP   0(8,8),VACH(8)
         BR    5
S88      SP    0(8,8),10(8,8)
         BR    5
* COME HERE WHEN MULTIPLICATION IS NEEDED
```

```
MULTIPLY S       8,=F'10'
         ZAP     TEMP1(16),0(8,8)
         MP      TEMP1(16),10(8,8)
         ZAP     0(8,8),TEMP1(16)
         AP      8(2,8),18(2,8)
         BR      5
* COME HERE WHEN DIVISION IS NEEDED
DIVIDE   CP      0(8,8),=P'0'
         BE      ER77
         S       8,=F'10'
EXPDIV   ZAP     TEMP1,0(8,8)
         CP      8(2,8),18(2,8)
         BL      XXY
VDP      SP      8(2,8),18(2,8)
COMP5    CP      8(2,8),=P'5'
         BE      DV
         BH      HIGH
MPTT     MP      TEMP1(16),TEN
         AP      8(2,8),=P'1'
         B       COMP5
HIGH     DP      TEMP1(16),TEN
         SP      8(2,8),=P'1'
         CP      8(2,8),=P'5'
         BE      DV
         B       HIGH
XXY      MP      TEMP1(16),TEN
         AP      8(2,8),=P'1'
         CP      8(2,8),18(2,8)
         BL      XXY
         B       VDP
DV       DP      TEMP1(16),10(8,8)
         ZAP     0(8,8),TEMP1(8)
         BR      5
ER77     WRTERM  'DIVISOR IS ZERO'
         B       EXIT
ERR      WRTERM  'TEMPORARY BUFFER IS FULL DURING EXECUTING ARITHMETIC E
                 XPRESSION .(TEMP1)'
         B       EXIT
* THE METHOD OF EXPONENT
EXPONENT S       8,=F'10'
         MVI     C,C'0'
         CP      10(8,8),=PL8'0'
         BH      EX1
         BE      EX0
         MP      10(8,8),=P'-1'
         MVI     C,C'1'
EX1      ZAP     TEMP1(16),0(8,8)
         ZAP     0(2),8(2,8)
         LA      6,1
         LA      10,1
         CVB     11,10(0,8)
EX3      BXLE    6,10,EX2
         ZAP     0(8,8),TEMP1(16)
         CLI     C,C'1'
         BE      BBB
```

```
         BR    5
EX2      MP    TEMP1(16),0(8,8)
         CP    8(2,8),=PL2'0'
         BE    BEX
         AP    8(2,8),0(2)
         CP    8(2,8),=P'5'
         BH    SHFT
BEX      B     EX3
EX0      ZAP   0(8,8),=P'1'
         ZAP   8(2,8),=P'0'
         BR    5
BBR      ZAP   10(8,8),0(8,8)
         ZAP   18(2,8),8(2,8)
         ZAP   0(8,8),=P'1'
         ZAP   8(2,8),=P'0'
         B     EXPDIV
SHFT     DP    TEMP1(16),TEN
**       ZAP   0(8,8),TEMP1(8)
**       ZAP   TEMP1(10),0(8,8)
         ZAP   TEMP1(16),TEMP1(14)
         SP    8(2,8),=P'1'
         CP    8(2,8),=P'5'
         BNH   EX3
         B     SHFT
UNA      MP    0(8,8),=P'-1'
         BR    5
* DETERMINE VALUE OF VARIABLE TO BE DISPLAYED ON THE SCREEN
EXPS     LA    5,VAR
         LA    4,TABLE
VR       CLI   0(6),C' '
         BE    XPN
         ZAP   TWO(8),0(2,6)
         SP    TWO(8),=P'1'
         MP    TWO(8),=P'8'
         CVB   9,TWO
         AR    9,4
         LA    13,SCREEN
         ZAP   SV(8),2(2,9)
         SP    SV(8),=P'1'
         CVB   5,SV
         AR    5,13
         ZAP   SV,4(2,9)
         SP    SV(8),=P'1'
         CVB   7,SV
         AR    7,13
         A     6,=F'2'
```

```
          LA    8,BFIN
* IF THE VARIABLE IS IN THE INPUT FORMAT BUFFER,BRANCH TO LABEL 'C
CXY       CLC   0(8,6),0(8)
          BE    GETD
          A     8,=F'14'
          CLI   0(8),C' '
          BE    YON
          C     8,=A(BFIN+279)
          BNH   CXY
YCN       CLI   VAR,C' '
          BE    ET
          LA    8,VAR
CVX       CLC   0(8,6),0(8)
          BE    BPAKEEP
          A     8,=F'18'
          CLI   0(8),C' '
          BE    ET
          C     8,=A(VAR+359)
          BNH   CVX
* IF THE VARIABLE CONTAINS THE RESULT OF ARITHMETIC STATEMENT,
* THE FOLLOWING INSTRUCTIONS WILL BE USED.
BPAKEEP   ZAP   VSTM(8),8(8,8)
          ZAP   PENAR(2),16(2,8)
PAKEEP    MVC   TRUE(20),FALSE
          CLI   6(9),C' '
          BE    ZPSV
          ZAP   SV(8),6(2,9)
          CP    6(2,9),16(2,8)
          BL    SHFRBD
          BH    FZD
          B     BCVBSV
ZPSV      ZAP   SV(8),PENAR(2)
          ZAP   TEMPLTEN(8),VSTM(8)
          B     CVBSV
BCVBSV    ZAP   TEMPLTEN(8),8(8,8)
CVBSV     CVB   11,SV
          C     11,=F'1'
          BH    ST7Q
          MP    TEMPLTEN(8),TEN
ST7Q      ST    7,Q
          LA    7,TRUE+18
          SR    7,11
          MVC   0(2,7),=XL2'214B'
          A     7,=F'1'
S74       S     7,=F'4'
          C     7,=A(TRUE+2)
          BL    MSK
          MVI   0(7),X'6B'
          B     S74
MSK       MVC   MASK(20),TRUE
MSK2      EDMK  MASK(20),TEMPLTEN
          CP    TEMPLTEN(8),=P'0'
          BNL   MCD
          ST    4,STORER4
          LA    4,MASK+18
```

```
CMP4     CLI   C(4),C' '
         BE    MINUS
         S     4,=F'1'
         B     CMP4
MINUS    MVI   C(4),C'-'
         L     4,STORE4
MCD      L     7,Q
         CLI   FSUM,C'Y'
         BE    WMAX
         CLI   FMIN,C'Y'
         BE    WMAX
         CLI   FMAX,C'Y'
         BE    WMAX
         LA    11,MASK+19
M711     MVC   C(1,7),0(11)
         S     7,=F'1'
         S     11,=F'1'
         CR    7,5
         BNL   M711
         CLI   0(11),C' '
         BNE   WNINGS
CPQ      CLI   PQ,C'2'
         BE    CM23
         B     NEXVAR     LOOP BACK TO FIND THE VALUE OF NEXT VARIABLE
*                          TO BE DISPLAYED ON THE SCREEN.
WNINGS   MVI   C(7),C'*'
         B     CPQ
SHERBD   ZAP   D(2),16(2,8)
         ZAP   TEMPLTEN(10),8(8,8)
DPT      DP    TEMPLTEN(10),TEN
         SP    D(2),=P'1'
         CP    6(2,9),D(2)
         BE    DECCVBSV
         ZAP   TEMPLTEN(10),TEMPLTEN(8)
         B     DPT
DECCVBSV CP    TEMPLTEN+8(2),=PL2'0'
         BNL   CONTINUE
         MP    TEMPLTEN+8(2),=P'-1'
CONTINUE CP    TEMPLTEN+8(2),=PL2'5'
         BL    CVBSV
         CP    TEMPLTEN(8),=P'0'
         BL    LESS
         AP    TEMPLTEN(8),=P'1'
         B     CVBSV
LESS     SP    TEMPLTEN(8),=P'1'
         B     CVBSV
FZD      ZAP   D(2),16(2,8)
         ZAP   TEMP1(8),8(8,8)
MULPY    MP    TEMP1(8),TEN
         AP    D(2),=P'1'
         CP    6(2,9),D(2)
         BNE   MULPY
         ZAP   TEMPLTEN(8),TEMP1(8)
         B     CVBSV
GETD     ST    5,XX
```

```
         LA    5,BFD
         ZAP   TWO(8),8(2,8)
         SP    TWO(8),=P'1'
         CVB   11,TWO
         AR    11,5
         ZAP   TWO(8),10(2,8)
         SP    TWO(8),=P'1'
         CVB   10,TWO
         AR    10,5
         L     5,XX
         CLI   6(9),X'40'
         BE    EACH
         CP    6(2,9),=PL2'0'
         BE    EACH
         ZAP   TWO(8),6(2,9)
         CVB   13,TWO
         ST    7,TH
         SR    7,13
         ST    7,DCMD
         MVI   0(7),C'.'
         A     7,=F'1'
CP11     CLI   0(11),C' '
         BNE   T11
         A     11,=F'1'
         B     CP11
T11      ST    11,T
CDOT     CLI   0(11),C'.'
         BE    PU
         A     11,=F'1'
         CLI   0(11),C' '
         BE    MZ
         CR    11,10
         BNH   CDOT
MZ       MVI   0(7),C'0'
         A     7,=F'1'
         C     7,TH
         BNH   MZ
L7S      L     7,DCMD
MCC      S     7,=F'1'
         S     11,=F'1'
         C     11,T
         BL    NEXVAR
         CR    7,5
         BL    ETRANC
         MVC   0(1,7),0(11)
         B     MCC
PU       ST    11,W
         A     11,=F'1'
MAX      MVC   0(1,7),0(11)
         A     7,=F'1'
         C     7,TH
         BH    BFORMER
         A     11,=F'1'
         CLI   0(11),C' '
         BE    FILLZERO
```

```
         CR    11,10
         BNH   MAX
FILLZERO MVI   0(7),C'0'
         A     7,=F'1'
         C     7,TH
         BNH   FILLZERO
FORMER   L     11,W
         B     L7S
BFORMER  A     11,=F'1'
         CR    11,10
         BH    FORMER
         CLI   0(11),C'5'
         BL    FORMER
         S     7,=F'1'
         CLI   0(7),C'.'
         BE    FORMER
         PACK  BHD(1),0(1,7)
         AP    BHD(1),=P'1'
         UNPK  0(1,7),BHD(1)
         OI    0(7),X'F0'
         A     7,=F'1'
         B     FORMER
TCOAP    CLI   0(11),C' '
         BNE   ASKNUM
         A     11,=F'1'
         B     TCOAP
ASKNUM   ST    11,TV
YYY      CLI   0(11),C'.'
         BE    CHOOSEF
         CLI   0(11),C' '
         BE    CHOOSEF
         A     11,=F'1'
         CR    11,10
         BNH   YYY
CHOOSEF  S     11,=F'1'
         MVC   0(1,7),0(11)
         C     11,TV
         BE    NEXVAR
         S     7,=F'1'
         CR    7,5
         BNL   CHOOSEF
ETRANC   LINEDIT TEXT='TRANCATING THE VALUE OF ..... IN ORDER TO PU
               H SCREEN BUFFER OCCURS',SUB=(CHARA,(8))
         B     EXIT
EACH     MVC   0(1,5),0(11)
         A     5,=F'1'
         A     11,=F'1'
         CR    5,7
         BH    NEXVAR
         CR    11,10
         BNH   EACH
NEXVAR   A     6,=F'8'
         B     VR
* GETTING THE VALUE OF REAL VARIABLE FROM DATA RECORD
FEAL     ZAP   DOUBLEC,12(2,6)
```

```
         ST    9,GG
         ST    4,JJ
         MVC   KEY(15),KEY-1
         CVB   9,DOUBLEC
         LA    4,KEY+15
         SR    4,9
         ST    4,SEVT
CDE      CLI   0(10),C' '
         BE    XOXX
         CLI   0(10),C'-'
         BNL   STTV
         MVI   MNSIGN,C'-'
         A     10,=F'1'
STTV     ST    10,TV
CT       CLI   0(10),C'.'
         BE    PCT
         CLI   0(10),C'0'
         BL    ECI
         CLI   0(10),C'9'
         BH    ECI
         A     10,=F'1'
         CLI   0(10),C' '
         BE    CKY
         CR    10,11
         BNH   CT
CY       L     4,SEVT
CKY      S     4,=F'1'
         S     10,=F'1'
         C     10,TV
         BL    PCK
         C     4,=A(KEY)
         BL    PCK
         MVC   0(1,4),0(10)
         B     CKY
ALOOP    CLI   PQ,C'2'
         BE    LOOPXPN
         B     LOOP
PCK      PACK  0(8,8),KEY(15)
         CLI   MNSIGN,C'-'
         BNL   LGG
         MP    0(8,8),=P'-1'
         MVI   MNSIGN,C' '
LGG      L     9,GG
         L     4,JJ
         CLI   FSUM,C'Y'
         BE    SUMM
         CLI   FMAX,C'Y'
         BE    BFAX
         CLI   FMIN,C'Y'
         BE    BFINL
         CLI   ARITHEC,C'Y'
         BE    BFORTGI
         S     9,=F'1'
         B     ALOOP
XOXX     A     10,=F'1'
```

```
         CR    10,11
         BH    EE1
         B     CDE
POT      ST    10,W
         A     10,=F'1'
MXT      CLI   0(10),C' '
         BE    DDD
         CR    10,11
         BH    DDD
         C     4,=A(KEY+14)
         BH    DDD
         MVC   0(1,4),0(10)
         A     4,=F'1'
         A     10,=F'1'
         B     MXT
DDD      L     10,W
         B     CY
DSPY     CLI   HEADG,C'*'
         BNE   WRITESCR
         HEAD
* DISPLAY A RESULT-LINE
WRITESCR WRTERM SCREEN,80
         MVC   NAR(256),NAR-1
         MVC   NAR+256(104),NAR+255
         ZAP   FLAG(1),=P'0'
         MVC   D(2),=2C' '
         MVI   POINT,C'N'
         MVI   ID,C' '
         MVI   PI,C' '
         MVI   PQ,C'1'
         ZAP   KEEP(1),=P'0'
         MVC   VACH(8),=8C' '
         MVI   C,C'0'
         MVI   KY,C'G'
         MVC   KEY(15),KEY-1
         ZAP   RSM(8),=PL8'0'
         MVI   FRONT,C' '
         MVI   FSJM,C'N'
         MVI   FMAX,C'N'
         MVI   FMIN,C'N'
         MVI   II,C'1'
         MVI   RR,C'1'
         MVI   ARITHEC,C'N'
         B     EXIT
* FIND THE RESULTS OF ALL EXPRESSIONS
XPN      MVI   PQ,C'2'
         LA    9,XPSION
         CLC   0(2,9),=2C' '
         BE    DSPY
         MVC   STORE,STORE-1
         MVC   STACK,STACK-1
         LA    8,STORE-10
         LA    4,STACK-1
         L     5,=A(XPSION+700)
         ST    5,SMOXPN
```

```
         S      9,=F'1'
CM23     A      9,=F'1'
         LA     13,TABLE
         C      9,SMDXPN
         BNL    DSPY
         CLI    0(9),C' '
         BE     DSPY
COMPR23  CP     0(2,9),0(2,13)
         BE     FSC
         A      13,=F'8'
         B      COMPR23
FSC      LA     10,SCREEN
         ZAP    SV(8),2(2,13)
         SP     SV(8),=P'1'
         CVB    5,SV
         AR     5,10
         ST     5,FSCL
         ZAP    SV(8),4(2,13)
         SP     SV(8),=P'1'
         CVB    7,SV
         AR     7,10
         A      9,=F'2'
         B      C1
LOOPXPN  A      9,=F'1'
         C      9,=A(XPSION+699)
         BH     TOA
         B      C1
RFSCB    FSCB 'SP DATA A1',BUFFER=BFD,BSIZE=130
ERR1     WRTERM 'FILE NOT FOUND'
         B      EXIT
ERR2     LR     10,15
         LINEDIT TEXT='CODE .... IN READ ',SUB=(DEC,(10))
         B      EXIT
ET       LINEDIT TEXT='THERE ISN''T THE VARIABLE ........ ',SUB=(CHAR
               (6))
EOF      C      15,=F'12'
         BNE    ERR2
         LA     15,0
EXIT     LM     13,1,RGT
         LM     4,11,RGT2
         LA     15,0
         BR     14
RGT      DS     5F
RGT2     DS     8F
FSCL     DS     F
         DC     C' '
NAR      DC     360C' '
         DC     C' '
STORE    DC     20CL10' '
         DC     C' '
STACK    DC     20C' '
ECHO     DS     F
COUNT    DS     F
FLAG     DC     P'0'
D        DC     2C' '
```

```
POINT    DC    C'N'
ST2      DC    A(STACK-1)
PI       DC    C' '
PQ       DC    C'1'
KEEP     DC    P'0'
TEN      DC    PL2'10'
BFD      DS    CL130
DOUBLEC  DS    D
VACH     DC    80' '
TEMP1    DS    PL16
C        DC    C'0'
TWO      DS    D
ITG      DS    PL4
IFACT    DS    PL8
I        DS    PL4
KY       DC    C'G'
SMOXPN   DS    F
VSTM     DS    PL8
PENAR    DS    PL2
NNG      DS    F
TEMPLTEN DS    PL10
         DC    C' '
SCREEN   DS    CL80
SV       DS    D
TRUE     DC    XL20'402020202020202020202020202020202020'
FALSE    DC    XL20'402020202020202020202020202020202020'
FALSE2   DC    XL20'40202020682020206820202068202020682020'
TH       DS    F
DOMO     DS    F
T        DS    F
W        DS    F
GG       DS    F
JJ       DS    F
TV       DS    F
         DC    C'0'
KEY      DC    150' '
SEVT     DS    F
Q        DS    F
EAP      DS    F
WLINE    DS    DCL35
FNAME    DS    CL3
         DC    C'('
V        DS    CL8
         DC    CL3')= '
MASK     DS    CL20
SVN      DS    F
RSM      DC    PL8'0'
ARGMAX   DC    900' '
ARGMIN   DC    900' '
ARGSUM   DC    900' '
FRONT    DC    C' '
FSUM     DC    C'N'
FMAX     DC    C'N'
FMIN     DC    C'N'
II       DC    C'1'
```

```
STSIX      DS    F
XX         DS    F
CTR        DS    PL3
RR         DC    C'1'
ARITHEC    DC    C'N'
ID         DS    CL1
DEP        DS    F
STO        DS    F
STORER4    DS    F
SAVEREG    DS    18F
THIRTEEN   DS    F
BHD        DS    CL1
MNSIGN     DC    C' '
L1         DS    CL1
ARGFORT    DC    A(INDEX)
           DC    A(ISEND)
           DC    A(MYDEC)
           DC    A(MEMO)
INDEX      DS    F
ISEND      DS    F
MYDEC      DS    F
MEMO       DS    F
BFIN       DS    CL280
STATE      DS    CL1600
VAR        DS    CL200
XPSION     DS    CL700
TABLE      DS    CL240
NF         DS    CL8
WW         DS    CL3
REG9       DS    F
REG13      DS    F
LEIGHT     DS    D
HEADG      DS    CL1
           FSCBD
           END
```

FILE: FUNCT FORTRAN A1

```
      SUBROUTINE FUNCT(INDEX,ISEND,MYDEC,MEMO)
      SEND=ISEND
      DEC=MYDEC
      SEND=SEND/10.**DEC
      IF (INDEX.EQ.1)GO TO 1
      IF (INDEX.EQ.2)GO TO 2
      IF (INDEX.EQ.3)GO TO 3
      IF (INDEX.EQ.4)GO TO 4
      IF (INDEX.EQ.5)GO TO 5
      IF (INDEX.EQ.6)GO TO 6
      IF (INDEX.EQ.7)GO TO 7
1     RESULT=SIN(SEND)
      GO TO 111
2     RESULT=COS(SEND)
      GO TO 111
3     RESULT=TAN(SEND)
      GO TO 111
4     RESULT=SQRT(SEND)
      GO TO 111
5     RESULT=ALOG(SEND)
      GO TO 111
6     RESULT=ALOG10(SEND)
      GO TO 111
7     RESULT=EXP(SEND)
111   RESULT=10.**4*RESULT
      MEMO=RESULT
      RETURN
      END
```

```
FILE: MYMAC    MACLIB   A1                    VM/SP CONVERSATIONAL MONITO

LIBPDS
         MACRO
         DETAIL
         WRTERM 'THIS PROGRAM USES THE EXAMPLE DATA FILE,KCF DATA A1'
         MEND
/ /
         MACRO
         ENDJOB
         WRTERM '************END OF PROCESSING************'
         MEND
/ /
         MACRO
         HEAD
         WRTERM 'CHAR1.   RESULT1'
         MEND
/ /
         MACRO
         JSLMOD  &ID,&V,&LEIGHT,&MYDEC
*  &ID -- CL1, &V -- CL8, &LEIGHT -- P18, &MYDEC -- F
*  DETAIL MESSAGE OF ARGUMENTS --- SEE MACRO MACFUNC
         MEND
/ /
         MACRO
         MACFUNC &V,&LEIGHT,&MYDEC
*   &V -- CL8          NAME OF FUNCTION
*   &LEIGHT --P18      INTEGER INCLUDED FRACTION CORRESPONDING TO &MYDE
*   &MYDEC -- F        &LEIGHT / 10**&MYDEC IS TRUE NUMBER
         MEND
/ /
         MACRO
         FFF    &D,&P
*   THIS SEGMENT CONVERTS AN INTEGER TO SHORT NORMALIZED FLOATING POI
*   FORM, STORING THE RESULT AT NORMINT.
         STM    4,11,KREG
         ZAP    TENTH(7),=P'1'
         L      10,=F'0'
MORE     C      10,&P
         BE     DIVD
         A      10,=F'1'
         MP     TENTH(7),=P'10'
         B      MORE
DIVD     ZAP    NINE9(11),&D
         DP     NINE9(11),TENTH(7)
         ZAP    DOUBLE(8),NINES(4)
         CVB    9,DOUBLE
         ZAP    DOUBLE(8),NINES+4(7)
         LPR    5,9              TAKE ABSOLUTE VALUE
         L      4,UNZRO          CREATE UNNORM. FLOATED NUMBER
         N      9,MASKSGN        ISOLATE SIGN
         OR     4,9              ATTACH SIGN
         STM    4,5,UNNO         UNNORMALIZED SIGNED NUMBER
         LD     0,UNNO
         AD     0,=D'0'          NORMALIZE
*   THE FOLLOWING SEGMENT CONVERTS DIGITS OF FRACTION INTO
*   FLOATING-POINT FORM, ATTACHING THE NUMBER TO THE PRECEDING RESULT
```

```
CONT     C       1C,=F'7'
         BL      SMALL
         BH      BIG
         CVD     7,DOUBLE          ASSUME OC MINUS SIGN ON DOUBLE
         L       4,UNZRO
         OR      6,5               ATTACH SAME SIGN AS TO INTEGER
         STM     6,7,UNND
         LD      4,UNNL
         DD      4,=D'1.0E7'       MAKE INTL FRACTION , NORMALIZE
         ADR     2,4               COMBINE INTEGER TO FRACTION
         STE     2,NORMINT
         LM      4,11,KREG
         B       OUT
SMALL    AP      DOUBLE(8),=PL2'1C'
         A       1C,=F'1'
         B       CONT
BIG      DP      DOUBLE(8),=PL2'1C'
         ZAP     DOUBLE(8),DOUBLE(6)
         S       1C,=F'1'
         B       CONT
         DS      0F
KREG     DS      8F
SREG     DS      F
UNZRO    DC      X'40000000'       UNNORMALIZED ZERO
MASKSGN  DC      X'80000000'       ANNIHILATES ALL BUT SIGN
UNND     DS      1D
NORMINT  DS      1F
DOUBLE   DS      D
NINE7    DS      PL11
TENTH    DS      PL7
OUT      EQU     *
         MEND
/ /
         MACRO
         FLTX  &FLOAT
*  THIS SEGMENT CONVERTS A SHORT NORMALIZED FLOATING-POINT NUMBER
*  FLT TO A BINARY INTEGER . MULTIPLY IFLOAT BY NORMALIZED NUMBER
*  4BF42400   (10**6 IN DECIMAL). THEN ,THE INTEGER WILL ALSO INCLUD
*  SIX DECIMAL FRACTION DIGITS.  IF THE NUMBER IS TOO LARGE, BRANCH
*  TOOBIG.
         STM     4,11,QQ
         LD      0,=D'0'           CLEAR LOWER HALF OF REGISTER
         LE      0,&FLOAT
         ME      0,TENSIX
         LPER    2,0          IF ABS. VALUE IS NOT LESS THAN 2**31 ,
         CE      2,TWO31                   NUMBER IS TOO LARGE.
         BNL     TOOBIG
         AW      0,TWO32    UNNORMALIZE,COMPLEMENT IF NEGATIVE
         STD     0,DINT       STORE RESULT
         MVC     INTGR(4),DINT+4
         L       5,INTGR
         LINEDIT TEXT='INTEGER IS ........',SUB=(DEC,(5))
         B       SPP
TOOBIG   WRTERM 'ABS. OF FLOATING-POINT NO. IS NOT LESS THAN 2**31'
         B       SPP
```

```
CQ          DS      8F
TENSIX      DC      X'45F4240CCCCCCCC'
INTGR       DS      F
DINT        DS      D
TWO32       DC      X'4E00000100000000'
TWO31       DC      X'48000000'
SPP         EA      4,11,QC
            MENT
/  /
DETAIL              FNLDT               HEAD                LSEMCF              MACFU
FFF                 FLTX
```

ภาคผนวก ช.

การสร้างโปรแกรมในสภาวะซี เอม เอส

การสร้างโปรแกรมในสภาวะซีเอมเอส

การเชื่อมโยงโปรแกรม (Program Linkage)

ในระบบซีเอมเอส การเชื่อมโยงโปรแกรมจะเป็นหน้าที่ของคำสั่ง SVC (supervisor call instruction), SVC 202 โปรแกรมเอสวีซี (SVC handling routine) จะดูแล จัดการเกี่ยวกับการเชื่อมโยงโปรแกรมให้เอง รีจิสเตอร์ต่าง ๆ จะทำหน้าที่ดังนี้

รีจิสเตอร์ 0 : ถ้าคำสั่งถูกเรียกใช้จากเครื่องเทอร์มินอลหรือจากแฟ้มข้อมูลประเภท EXEC รีจิสเตอร์ 0 จะชี้ไปยังพารามิเตอร์ลิสท์ ซึ่งเก็บตำแหน่งที่อ้างถึงคำสั่งนั้น

รีจิสเตอร์ 1 : ชี้พารามิเตอร์ลิสท์ โดยที่เอนทรีแรกของลิสท์ แสดงชื่อของโปรแกรมย่อยที่ถูกเรียก ส่วนในคับเบิลเวิร์ด (doublewords) อื่น ในพารามิเตอร์ลิสท์ อาจเป็นตัวอ้างอิง (arguments) ที่จะส่งไปยังโปรแกรมนั้น

รีจิสเตอร์ 13: เก็บตำแหน่งของ 24-fullword Save area ของโปรแกรมที่เป็นตัวเรียก เนื้อที่นี้จัดไว้สำหรับลิงเกจของระบบปฏิบัติงาน ØS และ DØS แต่ซีเอมเอสจะไม่ใช้รีจิสเตอร์ 13 เพราะโปรแกรมเอสวีซีจะจัดการเก็บรีจิสเตอร์เหล่านั้นให้เอง

รีจิสเตอร์ 14: เก็บตำแหน่งที่จะกลับเข้ามาทำต่อไปในโปรแกรม SVC (return address ของ SVC handling routines) นั่นคือเมื่อจะออกจากโปรแกรม จะต้องส่งการควบคุมกลับไปยังตำแหน่งนี้ และเมื่อรูทีนซีเอมเอส (CMS routine) ได้รับการควบคุมแล้วจะปิดแฟ้มข้อมูล แก้ไข ปรับปรุงตารางรายละเอียดเกี่ยวกับแฟ้มข้อมูลที่อยู่บนจานแม่เหล็กของผู้ใช้คนนั้น และคำนวณเวลาที่ใช้ในการดำเนินการ (execute) ซึ่งสิ่งเหล่านี้จะปรากฏให้เห็นในข่าวสารความพร้อมของระบบซีเอมเอส หลังจากที่โปรแกรมของเราได้ทำการดำเนินการเสร็จแล้ว

R; T = n.nn/x.xx hh: mm:ss

n.n เป็นเวลาเครื่องของซีเอมเอส (วินาที)

x.xx เป็นเวลาเครื่องของซีพีและซีเอมเอส

hh:mm:ss เป็นเวลาในขณะนั้น ชั่วโมง นาทีและวินาที ตามลำดับ

รีจิสเตอร์ 12 และ 15 : เก็บตำแหน่งในโปรแกรมที่จะโอนการทำงานไปให้ (program's entry point address)

รีจิสเตอร์ 15 ไม่สามารถใช้เป็นเบส รีจิสเตอร์ (base register) เพราะ CMS SVC ทั้งหมดจะใช้รีจิสเตอร์ 15 เพื่อติดต่อกับโปรแกรมของเรา

ตัวอย่างโปรแกรมภาษาแอสเซมบลีของระบบซีเอมเอส

```
PROGRam   CSECT
          USING  PROGRAM, 12    Establish addressability
          ST     14,SAVRET      Save return address in R14
          .
          .
          .
          L      14, SAVRET     Load return address
          LA     15             Set return code in R15
          BR     14             GO
SAVRET    DS     F              Save area
```

การควบคุมรหัสที่ส่งกลับมา

รีจิสเตอร์ 15 ถูกใช้เป็นรีจิสเตอร์ที่เก็บรหัสที่ส่งกลับมาในระบบซีเอมเอส โปรแกรมภายในของระบบซีเอมเอสทั้งหมด (CMS internal routines) จะส่งรหัสโดยส่งผ่านรีจิสเตอร์ 15 โปรแกรม SVC จะรับการควบคุมเมื่อโปรแกรมดำเนินการเสร็จสิ้นลง โดยพิจารณาจากรีจิสเตอร์ 15 ถ้ารีจิสเตอร์ 15 ไม่มีค่าเป็น 0 ค่านั้นจะปรากฏในข่าวสารความพร้อมในระบบซีเอมเอส

ภาคผนวก ค.

คำสั่งแมคโครในระบบซีเอมเอส (CMS/Macro)

FSCB

ใช้คำสั่ง FSCB เพื่อสร้างบลอคควบคุมแฟ้มข้อมูล (file system control block) สำหรับแฟ้มข้อมูลนำเข้าหรือส่งออกในระบบซีเอมเอส

รูปแบบของคำสั่ง

[label] | FSCB | [fileid] [,RECFM= format] [,BUFFER= buffer]
| | [,BSIZE=size][,RECNO=number] [,NOREC=number record]

label หมายถึง ชื่อที่ตั้งไว้สำหรับอ้างถึงบรรทัดหนึ่ง อาจมีหรือไม่มีก็ได้ (optional statement label)

feleid หมายถึง ข้อกำหนดในการอ้างถึงแฟ้มข้อมูลในระบบซีเอมเอส ซึ่งบอกอยู่ในเครื่องหมายคำพูด ('ชื่อแฟ้มข้อมูล ชนิดแฟ้มข้อมูล หมู่ของแฟ้มข้อมูล') แต่ละส่วนจะคั่นด้วยช่องว่างอย่างน้อย 1 ช่อง ถ้าไม่ระบุหมู่ของแฟ้มข้อมูลจะถือเป็น A1

RECFM = format เป็นตัวระบุว่าระเบียนจะมีความยาวคงที่หรือเปลี่ยนแปลง (F or V) ถ้าไม่กำหนดจะถือเป็น F

BUFFER = buffer กำหนดตำแหน่งของที่เก็บข้อมูลชั่วคราวที่ใช้เก็บข้อมูลที่จะนำเข้าและส่งออก

BSIZE = size บอกจำนวนไบต์ที่จะถูกอ่านหรือบันทึกในแต่ละครั้ง

RECNO = number บอกหมายเลขของระเบียนถัดไปที่จะใช้ โดยนับจากจุดเริ่มต้นของแฟ้มข้อมูล ระเบียนแรกจะกำหนดหมายเลขเป็น 1 ถ้าไม่กำหนด RECNO จะถือเป็น 0 ซึ่งหมายถึงให้อ่านระเบียนอย่างเรียงลำดับ

NOREC = numrec บอกจำนวนระเบียนที่จะถูกอ่านในคราวต่อไป ถ้าไม่ระบุจะถือเป็น 1

หมายเหตุ

1. RECNO, RECFM, BOFFER, BSIZE และ NOREC จะต้องเป็นตัวเลขโดยตรง (self-defining terms)

2. เราสามารถใช้ FSCB ตัวเดียวกันในการอ้างถึงแฟ้มข้อมูลที่แตกต่างกันได้ ถ้าอ้างถึงแฟ้มข้อมูล โดยใช้ FSCB เราสามารถเปลี่ยนแปลงข้อกำหนดในการอ้างถึงแฟ้มข้อมูลได้เมื่อใช้คำสั่งแมคโคร FSOPEN, FSWRITE หรือ FSSTATE ถ้าใช้ FSOPEN จะต้องใช้ BSIZE และ RECFM ใน FSCB เพื่อบอกคุณลักษณะของแฟ้มข้อมูลที่ต้องการ

3. สามารถใช้ FSCB หลายตัวในการอ้างถึงแฟ้มข้อมูลเดียวกันได้ ถ้าหากกำหนด RECNO ใน FSCB การอ่านหรือบันทึกจะยังคงอยู่ที่ระเบียนนั้นเสมอจนกว่าจะบง RECNO ใหม่ให้กับ FSCB นั้นหรือไปใช้ FSCB อื่นที่อ้างถึงแฟ้มข้อมูลเดียวกันนั้น

FSCBD

ใช้ FSCBD เพื่อใช้ DSECT กับบลอคควบคุมแฟ้มข้อมูล

รูปแบบ :

[label] | FSCBD |

หมายเหตุ

คำสั่งแมคโคร FSCBD จะขยายออกได้เป็น

```
          FSCBD
FSCBD     DSECT
FSCBCOMM  DS    CL8   คำสั่ง
FSCBFN    DS    CL8   ชื่อของแฟ้มข้อมูล
FSCBFT    DS    CL8   ชนิดของแฟ้มข้อมูล
FSCBFM    DS    CL2   หมู่ของแฟ้มข้อมูล
FSCBITNO  DS    H     หมายเลขสัมพันธ์ของแต่ละระเบียน
FSCBBUFF  DS    A     ตำแหน่งของที่เก็บข้อมูลชั่วคราวซึ่งเก็บข้อมูลที่อ่านเข้ามาหรือ
                      กำลังจะบันทึกลง
FSCBSIZE  DS    F     ความยาวของที่เก็บข้อมูลชั่วคราว
```

```
FSCBFV   DS   CL2   รูปแบบของระเบียน (F หรือ V)
FSCBNOIT DS   H     จำนวนระเบียนที่ถูกอ่าน/บันทึก
FSCBNORD DS   A     จำนวนไบท์ที่อ่านไปจริง ๆ
```

FSCLOSE

ใช้ FSCLOSE เพื่อปิดแฟ้มข้อมูลและเก็บสถานะของแฟ้มข้อมูลลงในฐานแม่เหล็ก ให้ถูกต้องอยู่เสมอ

รูปแบบ:

```
[label] | FSCLOSE | { fileid [ ,FSCB=fscb ] } [ ,ERROR=erraddr ]
        |          | {      FSCB=fscb      }
```

fileid หมายถึง ข้อกำหนดในการอ้างถึงแฟ้มข้อมูลในระบบซีเอมเอส อาจอยู่ในรูป ชื่อแฟ้มข้อมูล ชนิดแฟ้มข้อมูล หมู่ของแฟ้มข้อมูล หรือ (รีจิสเตอร์) กรณี (รีจิสเตอร์) จะต้องไม่ใช้รีจิสเตอร์ 0 หรือ 1 ในการเก็บตำแหน่งของข้อกำหนดในการอ้างถึงแฟ้มข้อมูล ข้อกำหนดดังกล่าวจะต้องยาว 18 ตัวอักษร กล่าวคือ ชื่อแฟ้มข้อมูลยาว 8 ตัวอักษร ชนิดแฟ้มข้อมูลยาว 8 ตัวอักษร หมู่ของแฟ้มข้อมูลยาว 2 ตัวอักษร

FSCB = fscb บอกตำแหน่งของ FSCB ซึ่งอาจบอกเป็นลาเบลของคำสั่ง FSCB macro หรือบอกเป็น (รีจิสเตอร์) ที่เก็บตำแหน่งของ FSCB

ERROR=erraddr บอกตำแหน่งของโปรแกรมตรวจสอบความผิดพลาด (error routine) ที่จะส่งการควบคุมไปให้ กรณีที่เกิดความผิดพลาดขึ้น เมื่อใช้คำสั่งนี้ ถ้าไม่ใช้ ERROR = erraddr กรณีเกิดความผิดพลาดขึ้น ตัวควบคุมจะไปยังคำสั่งถัดไปในโปรแกรมนั้น ราวกับว่าไม่มีความผิดพลาดใด ๆ เกิดขึ้น

หมายเหตุ

กรณีเกิดความผิดพลาดรีจิสเตอร์ 15 จะเก็บรหัสเป็นเลข 6 ซึ่งหมายถึงว่าแฟ้มข้อมูลที่ต้องการปิดนั้นไม่ได้เปิดไว้

FSOPEN

ใช้ FSOPEN เพื่อตรวจสอบว่าแฟ้มข้อมูลพร้อมที่จะนำเข้าหรือส่งออกหรือไม่

รูปแบบ:

```
[label] | FSOPEN | { fileid [,FSCB=fscb] } [,ERROR = erraddr] [,options]
        |        | { FSCB = fscb         }
```

options ได้แก่ BUFFER = buffer

RECNO = number

BSIZE = size

RECFM = format

NOREC = numrec

ค่าที่จะใช้อาจจะใช้ค่าจริงเลยหรือบอกเป็นรีจิสเตอร์ก็ได้ เช่น NOREC = 1 หรือ NOREC = (3) โดยที่รีจิสเตอร์ 3 เก็บค่า 1

หมายเหตุ

1. เมื่อคำสั่ง FSOPEN ทำเสร็จ รีจิสเตอร์ 1 จะชี้ไปยัง FSCB เพื่อดูแฟ้มข้อมูล ถ้าไม่ได้กำหนด FSCB เอาไว้ก่อน จะมีการกำหนด FSCB ขึ้นทันทีเมื่อใช้ แมคโคร FSOPEN แต่อย่างไรก็ตาม ถ้า FSOPEN ถูกใช้ตรวจสอบแฟ้มข้อมูลที่มีอยู่แล้ว BSIZE และ RECFM จะถูกให้ค่าใหม่เพื่อบอกคุณสมบัติที่ต้องการสำหรับแฟ้มข้อมูลนั้น
2. คำสั่ง FSCB ใช้ตรวจสอบว่ามีแฟ้มข้อมูลที่จะอ่านหรือบันทึกหรือไม่ และใช้สร้าง FSCB ที่ต้องการ
3. กรณีเกิดความผิดพลาด รีจิสเตอร์ 15 เก็บรหัสดังนี้

   20 หมายถึง ข้อกำหนดเพื่ออ้างถึงแฟ้มข้อมูลไม่ถูกต้อง

   28 หมายถึง ไม่มีแฟ้มข้อมูลนั้นอยู่

FSREAD

ใช้ในการอ่านระเบียนทีละระเบียน จากแฟ้มข้อมูลที่อยู่ในจานแม่เหล็ก แล้วไปเก็บไว้ในที่เก็บข้อมูล นำเข้าหรือส่งออกเป็นการชั่วคราว (I/O buffer)

รูปแบบ :

```
[label] | FSREAD| { fileid [,FSCB=fscb] } [,ERROR=erraddr ] [,options]
        |      | { FSCB=fscb           }
```

options ได้แก่ BUFFER = buffer

NOREC = numrec

BSIZE = size

RECNO = number

หมายเหตุ

1. ถ้าในโปรแกรมไม่ได้ใช้คำสั่ง FSCB จะต้องระบุ BUFFER= และ BSIZE = ใน FSREAD เพื่อบอกตำแหน่งของที่เก็บข้อมูลชั่วคราวและความยาวของระเบียนที่จะอ่าน กรณีที่อ่านระเบียนที่มีความยาวไม่คงที่ (variable length record) ถ้าระเบียนนั้นยาวกว่าหน่วยความจำที่เก็บ ส่วนที่เกินจะถูกตัดออก
3. รีจิสเตอร์ 1 จะชี้ไปยัง FSCB เพื่อหาแฟ้มข้อมูล ถ้าไม่ได้ระบุ FSCB ไว้ก่อน FSCB จะถูกสร้างขึ้นต่อจากคำสั่ง FSREAD
3. รีจิสเตอร์ 0 จะเก็บจำนวนไบต์ที่อ่านจริง (นอกจากนี้ FSCBNORD ก็เก็บสารสนเทศนี้เช่นกัน
4. การอ่านระเบียนอย่างเรียงลำดับโดยเริ่มจากระเบียนที่ต้องการ จะใช้ RECNO เพื่อบอกระเบียนแรกที่จะอ่าน แล้วใช้คำสั่ง FSREAD อีกครั้งหนึ่ง ให้ RECNO= 0 เพื่ออ่านข้อมูลอย่างต่อเนื่อง ต่อจากระเบียนแรกที่อ่านไปแล้ว
5. รีจิสเตอร์ 15 เก็บรหัสดังต่อไปนี้ กรณีที่เกิดความผิดพลาดขึ้น

รหัส	ความหมาย
1	ไม่มีแฟ้มข้อมูลที่ระบุ
2	เกิดความผิดพลาดเกี่ยวกับตำแหน่งของหน่วยความจำชั่วคราว
3	เกิดความผิดพลาดที่ข้อมูลที่ส่งมาหรือจะส่งออก
5	จำนวนระเบียนที่อ่านน้อยกว่าหรือเท่ากับ 0 หรือมากกว่า 32768
7	รูปแบบของระเบียนผิดพลาด (ตรวจสอบตอนเปิดแฟ้มข้อมูลครั้งแรกเพื่ออ่าน)
8	ความยาวผิดพลาด
9	แฟ้มข้อมูลเปิดสำหรับข้อมูลส่งออก
11	จำนวนระเบียนมากกว่า 1 สำหรับแฟ้มข้อมูลที่มีความยาวเปลี่ยนแปลงได้
12	จบแฟ้มข้อมูลหรือหมายเลขระเบียนเกินจำนวนระเบียนที่มีอยู่ (ซึ่งไม่เกิน 65,533)
13	แฟ้มข้อมูลแบบความยาวเปลี่ยนแปลงได้ใช้ดิสเพลสเมนท์ผิด
14	มีอักษรที่ห้ามใช้ในชื่อแฟ้มข้อมูล
15	มีอักษรที่ห้ามใช้ในชนิดของแฟ้มข้อมูล

FSSTATE

ใช้ตรวจดูว่ามีแฟ้มข้อมูลที่ต้องการอยู่หรือไม่

รูปแบบ:

```
[label] | FSSTATE | { fileid [,FSCB = fscb] } [,ERROR=erraddr]
        |         | { FSCB = fscb           }
```

หมายเหตุ

1. ถ้ามีแฟ้มข้อมูลนั้นจริง รีจิสเตอร์ 15 จะเก็บค่า 0
2. เมื่อ FSSTATE ทำงานเสร็จ รีจิสเตอร์ 1 จะเก็บตำแหน่งของตารางเก็บสถานะของแฟ้มข้อมูล (file status table; FST)

FST ประกอบด้วย

ดิสเพลสเมนท์ฐาน 10	รายละเอียดของเขตข้อมูลนั้น
0	ชื่อแฟ้มข้อมูล
8	ชนิดของแฟ้มข้อมูล
16	วันที่บันทึกครั้งล่าสุด (mmdd)
18	เวลาที่บันทึกครั้งล่าสุด (hhmm)
20	ตัวชี้ตำแหน่งที่จะบันทึก
22	ตัวชี้ตำแหน่งที่จะอ่าน
24	หมู่ของแฟ้มข้อมูล
26	จำนวนระเบียนในแฟ้มข้อมูล
28	ตำแหน่งในจานแม่เหล็กที่จะโยงไปหา
30	รูปแบบของระเบียน (F/V)
32	ความยาวของโลจิคอล เรคคอด
36	จำนวนบลอคซึ่งเก็บข้อมูลยาว 800 ไบท์
38	ปีล่าสุดที่บันทึก (yy)

3. กรณีเกิดความผิดพลาด รีจิสเตอร์ 15 จะมีรหัสดังนี้

20 แสดงว่า ข้อกำหนดเพื่ออ้างถึงแฟ้มข้อมูลไม่ถูกต้อง

24 แสดงว่า หมู่ของแฟ้มข้อมูลไม่ถูกต้อง

28 แสดงว่า ไม่มีแฟ้มข้อมูลนั้น

36 แสดงว่า เข้าถึงจานแม่เหล็กไม่ได้

FSWRITE

ใช้บันทึกระเบียนจากที่เก็บข้อมูลนำเข้าหรือส่งออกเป็นการชั่วคราว ไปยังแฟ้มข้อมูลบนจานแม่เหล็กในระบบซีเอมเอส

รูปแบบ:

```
[label] | FSWRITE | {fileid [,FSCB=fsch]}  [,ERROR=erraddr] [,options]
        |         | {  FSCB = fsxb      }
```

options ได้แก่ BUFFER = buffer

RECNO = number

BSIZE = size

NOREC = numrec

RECFM = format

หมายเหตุ

1. ถ้า FSCB ไม่ได้ระบุไว้ก่อน และ FSCB ไม่ได้กำหนดอยู่ใน FSWRITE ด้วย เรา จะต้องระบุ BUFFER = และ BRIZE = เพื่อชี้ตำแหน่งของที่เก็บข้อมูลที่จะอ่านหรือ บันทึก และความยาวของระเบียนที่จะบันทึกนั้น สำหรับหมู่ของแฟ้มข้อมูลจะต้องบอก ทั้งตัวอักษรและตัวเลข ถ้าแฟ้มข้อมูลเป็นแบบความยาวไม่คงที่ จะต้องระบุ RECFM = V ลงไปด้วย
2. รีจิสเตอร์ 1 เก็บตำแหน่งของ FSCB ถ้าไม่ได้กำหนด FSCB ไว้ FSCB จะถูก สร้างขึ้นเองต่อจากคำสั่ง FSWRITE
3. การบันทึกระเบียนลงในตำแหน่งที่กำหนด แล้วบันทึกต่อไปแบบเรียงลำดับ จะใช้หลักการ ในทำนองเดียวกับของ FSREAD
4. สำหรับแฟ้มข้อมูลที่มีความยาวคงที่ เราอาจระบุ BSIZE และ NOREC เพื่อบ่งถึง ขนาดของบลอคและจำนวนของระเบียนต่อ 1 บลอค เช่น ในการบันทึก ระเบียนที่ยาว 80 ไบท์ หลาย ๆ ระเบียนลงในบลอคที่มีความยาว 800 ไบท์ จะต้องระบุ BSIZE = 800 NOREC = 10
5. กรณีเกิดความผิดพลาด รีจิสเตอร์ 15 จะเก็บรหัสดังนี้

รหัส	ความหมาย
2	เกิดผิดพลาดที่ตำแหน่งของหน่วยความจำชั่วคราว
4	เกิดผิดพลาดที่อักษรตัวแรกในหมู่แฟ้มข้อมูล
5	เกิดผิดพลาดที่อักษรตัวที่ 2 ในหมู่แฟ้มข้อมูล
6	หมายเลขของระเบียนสูงเกิน 65,533
7	พยายามข้ามระเบียน (ประเภทความยาวเปลี่ยนแปลงได้) ที่ไม่สามารถถูก บันทึก

รหัส	ความหมาย
8	ไม่ได้ระบุขนาดของที่เก็บข้อมูลชั่วคราว
9	เปิดแฟ้มข้อมูลสำหรับข้อมูลส่งออก
10	ถึงจำนวนแฟ้มข้อมูลที่มากที่สุดในมินิดิสค์แล้ว (3400)
11	รูปแบบของระเบียนไม่ได้เป็น F หรือ V
12	พยายามบันทึกข้อมูลลงในจานแม่เหล็กที่ยอมให้อ่านได้อย่างเดียว
13	บรรจุข้อมูลลงจานแม่เหล็กขนเต็มแล้ว
14	จำนวนไบท์ที่จะถูกบันทึกหารด้วยจำนวนระเบียนที่จะถูกบันทึกไม่ลงตัว
15	ความยาวของระเบียนประเภทความยาวคงที่ เกิดไม่เท่ากันทุกระเบียน
16	ระบุรูปแบบของระเบียนไม่ตรงกับที่เป็นจริงในแฟ้มข้อมูล
17	ระเบียนประเภทความยาวไม่คงที่มีมากกว่า 65K ไบท์
18	จำนวนระเบียนมากกว่า 1 สำหรับแฟ้มข้อมูลประเภทความยาวไม่คงที่
19	ถึงจำนวนบลอกสูงสุดที่เก็บข้อมูลสำหรับแฟ้มข้อมูลนั้นแล้ว (16060)
20	อักษรในชื่อแฟ้มข้อมูลผิดพลาด
21	อักษรในชนิดของแฟ้มข้อมูลผิดพลาด
22	เกินความจุของหน่วยความจำเสมือน
25	ที่เก็บตารางเก็บรายละเอียดของแฟ้มข้อมูลมีไม่พอ

LINEDIT

ใช้แปลงค่าฐาน 20 ไปเป็น EBCDIC หรือเลขฐาน 16 (hexadecimal) แล้วแสดงผลทางขดเทอร์มินอล

รูปแบบ:

```
[label] | LINEDIT | [,TEXT='messagetext'] [,DOT={YES|NO}] [,COMP={YES|NO}]
                  [,TEXTA=address]
                  [,SUB=(substitutionlist)
                  [,DISP={TYPE|NONE|SIO|PRINT|CPCOMM|ERRMSG}]
                  [,BUFFA=({address|(reg)})]
                  [,MF={I|L|({E,address|(reg)})}] [,MAXSOES=auster]
                  [,RENT={YES|NO}]
```

TEXT = 'message Text' บอกข่าวสารที่จะแสดงทางจอ

TEXTA= address บอกตำแหน่งที่สามารถอ้างถึงข่าวสารนั้นได้ อาจเป็น
- ลาเบล ซึ่งเป็นตำแหน่งของข่าวสาร
- (รีจิสเตอร์) เก็บตำแหน่งของข่าวสาร

DOT บ่งว่าจะให้จุดอยู่ท้ายข่าวสารหรือไม่

COMP บ่งว่าจะให้มีช่องว่างหลาย ๆ ช่องต่อกันในบรรทัดได้หรือไม่

SUB ระบุว่าจะใช้ค่าในตำแหน่งไหน

DISP ระบุว่าบรรทัดที่จะแก้ไข เปลี่ยนแปลง (edited line) จะถูกใช้อย่างไร

BUFFA ระบุตำแหน่งของที่เก็บข้อมูลชั่วคราว

MF ระบุรูปแบบของแมคโคร

MAXSUBS ระบุจำนวนมากที่สุดที่จะมีตัวแทนค่า

RENT ระบุว่าจะใช้รหัสหรือไม่

หมายเหตุ

1. ห้ามใช้รีจิสเตอร์ 0, 1, 15 เพราะ LINEDIT จะใช้รีจิสเตอร์เหล่านี้
2. จุดในข่าวสาร จะเป็นตำแหน่งที่จะมีการนำค่ามาใส่ จำนวนจุดก็หมายถึง จำนวนตัวอักษรที่จะแสดงออกมา ตัวแปรที่เก็บค่าอยู่จะระบุใน SUB =

RDTERM

ใช้อ่านบรรทัดจากเทอร์มินอลมาไว้ในที่เก็บข้อมูลนำเข้าหรือส่งออกเป็นการชั่วคราว

รูปแบบ:

[label] | RDTERM | buffer [,EDIT=code] [,LENGTH=length] [,ATTREST={YES / NO}]

buffer ระบุตำแหน่งของที่เก็บข้อมูลชั่วคราวที่จะนำบรรทัดที่อ่านจากเทอร์มินอลมาเก็บไว้ ที่เก็บข้อมูลชั่วคราวนี้ยาว 130 ไบต์ แต่ถ้าจะเปลี่ยนแปลงจะใช้ ตำแหน่งที่เก็บข้อมูลชั่วคราวนี้อาจกำหนดเป็นสัญญลักษณ์ (symbolic address) หรือ (รีจิสเตอร์)

EDIT=code code อาจเป็น

- NO หมายถึง อ่านทีละบรรทัด (logical line) และไม่ต้องแก้ไข
- PAD หมายถึง บรรทัดที่อ่านเข้ามา (input line) จะถูกเติม (pad) ด้วยช่องว่างจนกว่าจะถึงความยาวที่กำหนด
- UPCASE หมายถึง ให้แปลงตัวอักษรในบรรทัดให้เป็นตัวพิมพ์ใหญ่
- YES หมายถึง ให้เติม (pad) และแปลงเป็นตัวพิมพ์ใหญ่
- PHYS หมายถึง อ่านแบบฟิสิคอล ไลน์ ถ้าใช้ EDIT = PHYS จะต้องบอก

LENGTH และ ATTREST = NO

LENGTH=length หมายถึง ความยาวของที่เก็บข้อมูลชั่วคราวนี้ โดยปรกติจะถือว่าเป็น 130 ไบท์ ความยาวสูงสุดที่เป็นได้คือ 2030 ไบท์ ถ้าเราจะกำหนดความยาวเอง จะต้องมี EDIT=PHYS ด้วย ความยาวอาจบอกเป็น n หมายถึง ตัวเลขที่บอกความยาวของที่เก็บข้อมูลชั่วคราว,(รีจิสเตอร์) หมายถึง รีจิสเตอร์ที่เก็บค่าความยาวของที่เก็บข้อมูลชั่วคราว

ATTREST=YES/NO ระบุว่าจะให้เกิดการขัดจังหวะ (attention interupt) ระหว่างการอ่านเป็นผลให้เริ่มต้นอ่านใหม่หรือไม่

หมายเหตุ

1. รีจิสเตอร์ 0 เก็บจำนวนตัวอักษรที่อ่านทั้งหมด เมื่อการทำงานของคำสั่ง RDTERM สิ้นสุดลง
2. กรณีที่เกิดผิดพลาดขึ้น รีจิสเตอร์ 15 จะเก็บรหัสดังนี้
   2 พารามิเตอร์ไม่ถูกต้อง
   4 การอ่านยุติโดยสัญญาณขัดจังหวะ

WRTERM

ใช้แสดงผลแต่ละบรรทัดทางเทอร์มินอล

รูปแบบ:

[label] |WRTERM| line[,length][,EDIT=code]

line อาจมีรูปแบบได้ 3 แบบ คือ

1. 'linetext' สิ่งที่ต้องการแสดงทางจอจะอยู่ในเครื่องหมายคำพูดทั้งหมด
2. linetext เป็นลาเบลของบรรทัดที่มีสิ่งที่ต้องการแสดงทางจอ
3. (รีจิสเตอร์) เป็นรีจิสเตอร์ที่เก็บตำแหน่งของบรรทัดที่จะแสดงทางจอ

EDIT=code code อาจกำหนดเป็น

YES ถ้าต้องการไม่ให้มีเทรลลิง แบลงค์ (trailing blank) แต่ต้องการให้แคริเอจ รีเทอน (carriage return) ไปต่อท้ายบรรทัดสุดท้าย ถ้าไม่กำหนด EDIT = จะถือว่า EDIT = YES'

NO ถ้าต้องการให้ เทรลลิง แบลงค์ยังคงอยู่ และ แคริเอจ รีเทอนไม่ต่อท้ายบรรทัดสุดท้าย

LONG ถ้าบรรทัดที่จะแสดงเกิน 130 ไบท์

ประวัติผู้เขียน

นางสาววัชรินทร์ ศรสันทิสุข เกิดเมื่อวันที่ 5 กรกฎาคม พ.ศ.2503 สำเร็จการศึกษาวิทยาศาสตรบัณฑิต (คณิตศาสตร์) จากคณะวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ ปีการศึกษา 2524–25 เข้าศึกษาระดับปริญญามหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมศาสตร์คอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ในปี พ.ศ.2525