

Program Logic**IBM System/360
Disk and Tape Operating Systems
PL/I Subset Language
Program Logic Manual****Volume 3 of 3****Program Numbers:****360 N-PL-464 (DOS)****360 N-PL-410 (TOS)**

This publication provides information on the internal logic of the IBM System/360 DOS/TOS PL/I compiler. It is intended for use by persons involved in programming maintenance and by system programmers who wish to alter the program design. The information contained herein is not required for the use of, and the operation with, the PL/I compiler. Therefore, distribution of this publication is restricted to users with the aforementioned requirements.

The publication is divided into three volumes. Volume 1 contains the description of the compiler phases; volumes 2 and 3 contain the corresponding flow charts. The form numbers of the three volumes are:

Volume 1: Y33-9010

Volume 2: Y33-9011

Volume 3: Y33-9012

All information regarding the library subroutines of the DOS/TOS PL/I compiler is contained in the publication IBM System/360, Disk and Tape Operating Systems, PL/I Subset-Library Routines, Program Logic Manual, Form Y33-9013.

The reader must be thoroughly familiar with the IBM System/360 Disk and Tape Operating Systems and with the PL/I Subset language. A list of all publications that provide pertinent information is contained in the introduction to volume 1 of this PLM.

Restricted Distribution

RESTRICTED DISTRIBUTION: This publication is intended primarily for use by IBM personnel and may not be made available to others without the approval of local IBM management.

First Edition

Specifications contained herein are subject to change from time to time. Any such change will be reported in subsequent revisions or Technical Newsletters.

This publication was prepared for production using an IBM computer to update the text and to control the page and line format. Page impressions for photo-offset printing were obtained from an IBM 1403 printer using a special print chain.

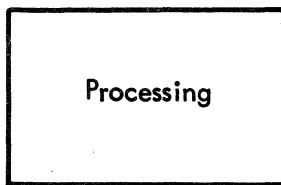
Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form is provided at the back of this publication for readers' comments. If the form has been removed, comments may be addressed to IBM Laboratories, Programming Publications, 703 Boeblingen/Germany, P.O. Box 210.

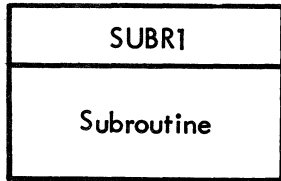
TABLE OF CONTENTS

Phase D05.	5	Phase F35.319
Phase D10.35	Phase F75.333
Phase D11.70	Phase F90.346
Phase D15.83	Phase F95.358
Phase D17.	123	Phase G00.374
Phase D20.	177	Phase G01.389
Phase D40.	184	Phase G15.393
Phase D70.	204	Phase G17.415
Phase D75.	232	Phase G20.417
Phase D80.	247	Phase G25.432
Phase E25.	253	Phase G30.446
Phase E50.	263	Phase G31.459
Phase E60.	282	Phase G40.461
Phase F25.	293	Phase G55.476

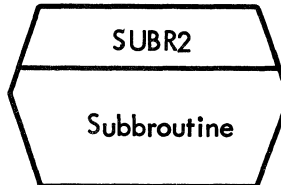
USE OF FLOW CHART SYMBOLS



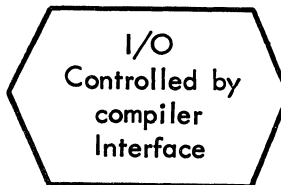
A group of program instructions that performs a processing function of the program.



Name or description of a subroutine, or parameters used by a subroutine, that is shown in detail on some other flow chart of the same phase. The entry point of the subroutine is given in the upper division of the box.



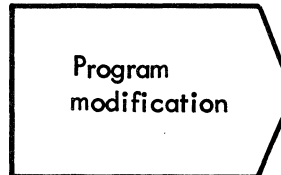
Name or description of a subroutine, or parameters used by a subroutine, that is shown in detail on some other flow chart of another phase. The entry point of the subroutine is given in the upper division of the box.



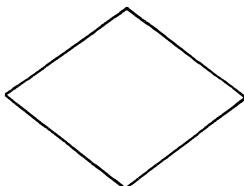
Name or description of subroutines that are part of the compiler interface.



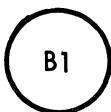
Any function of an input/output device.



An instruction or group of instructions that changes portions of a routine or initializes a routine for given conditions.



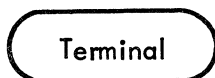
Points in the program where branches to alternate processing are made based on variable conditions, program switch settings, and test results.



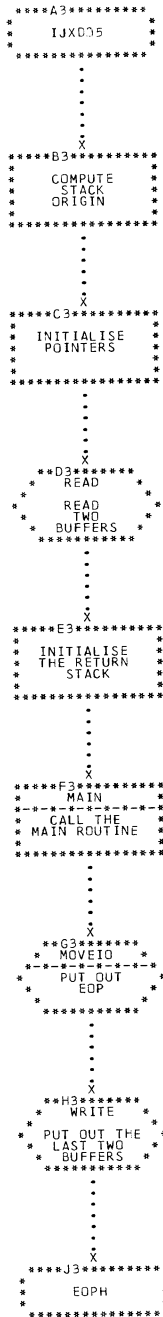
On-page connector. An entry from, or exit to, another block on the same flow chart. The characters in the connector identify the corresponding entry or exit on the flow chart.

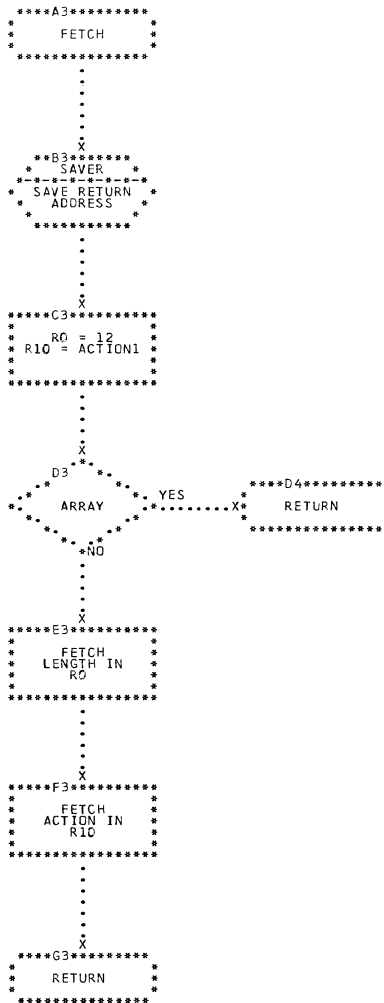


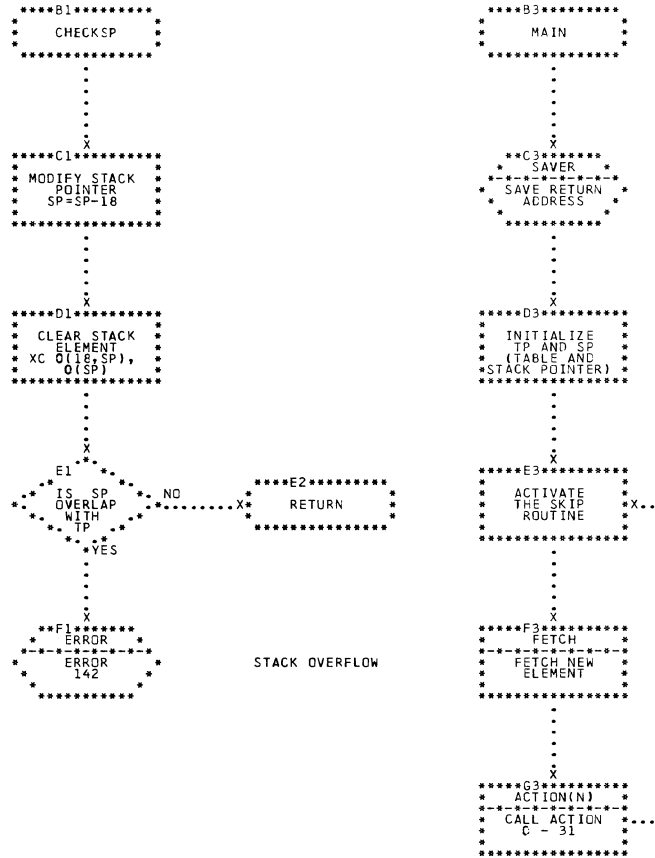
Off-page connector. An entry from, or exit to, a given point on another flow chart. The characters in the connector identify the flow chart and the block therein.

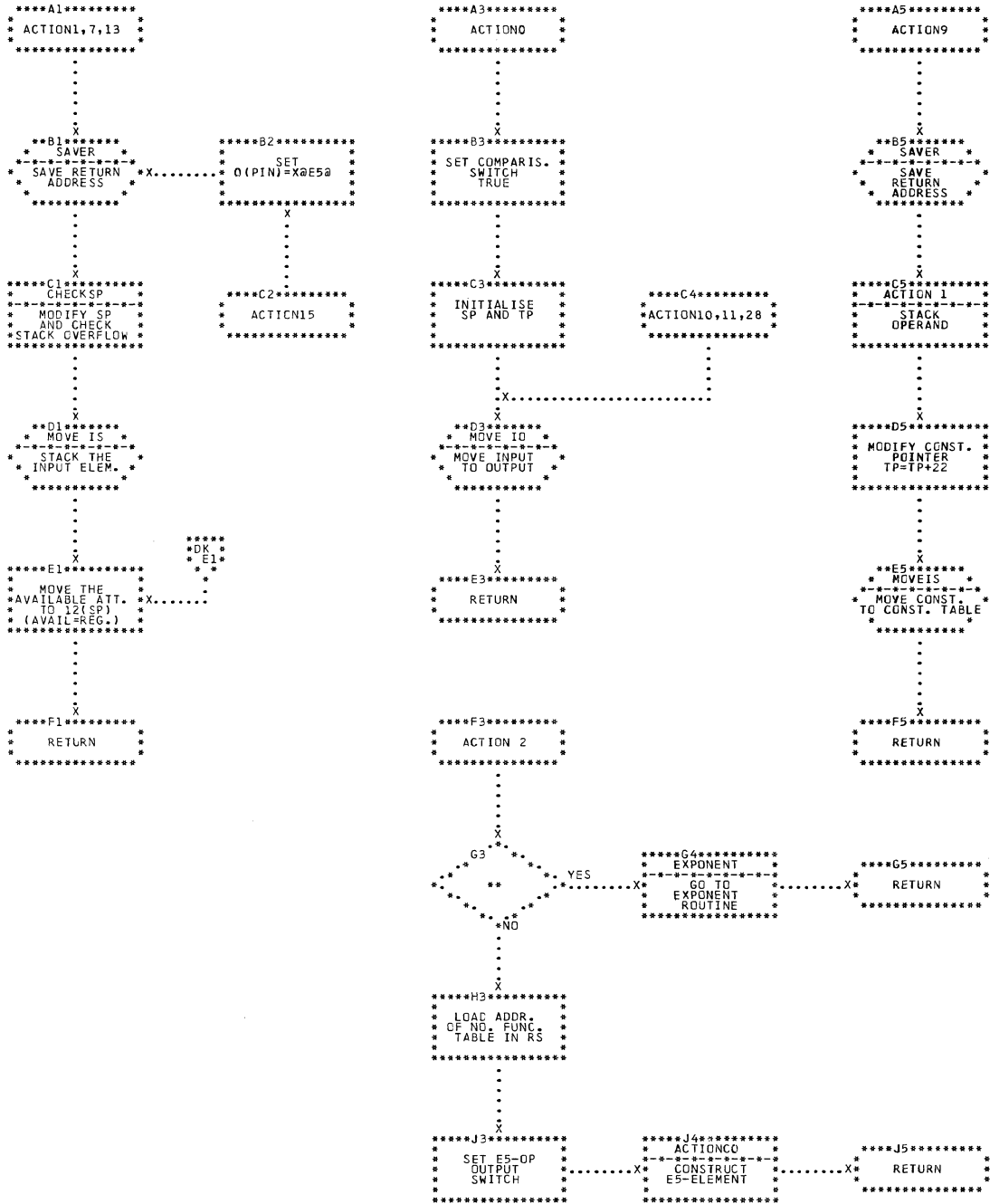


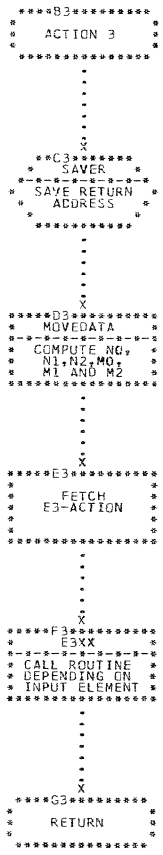
The beginning or end of a phase and the entry points or exits of a subroutine.

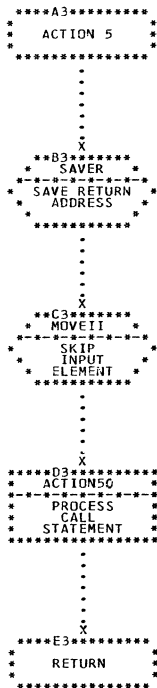


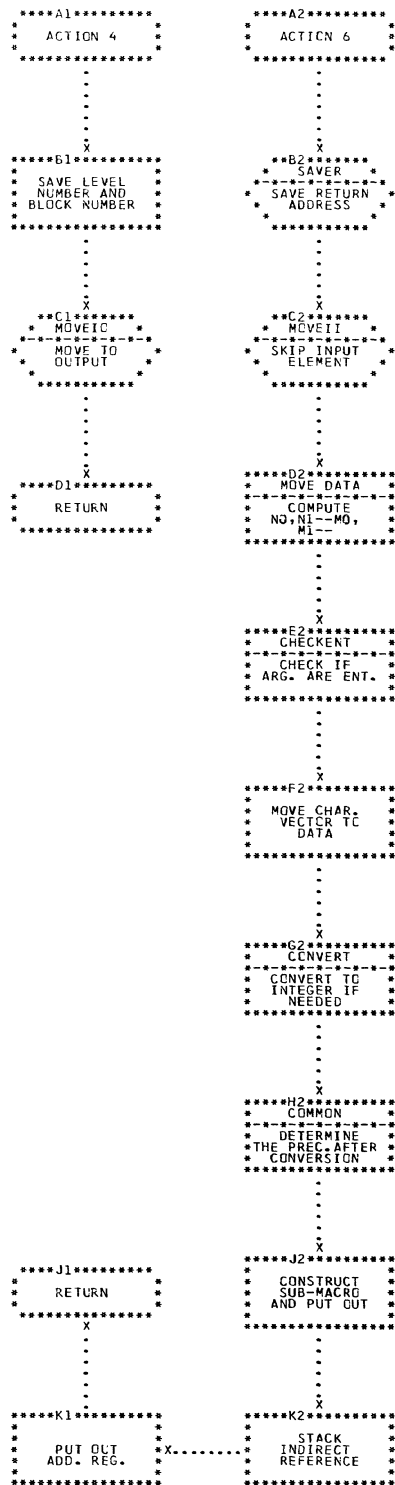












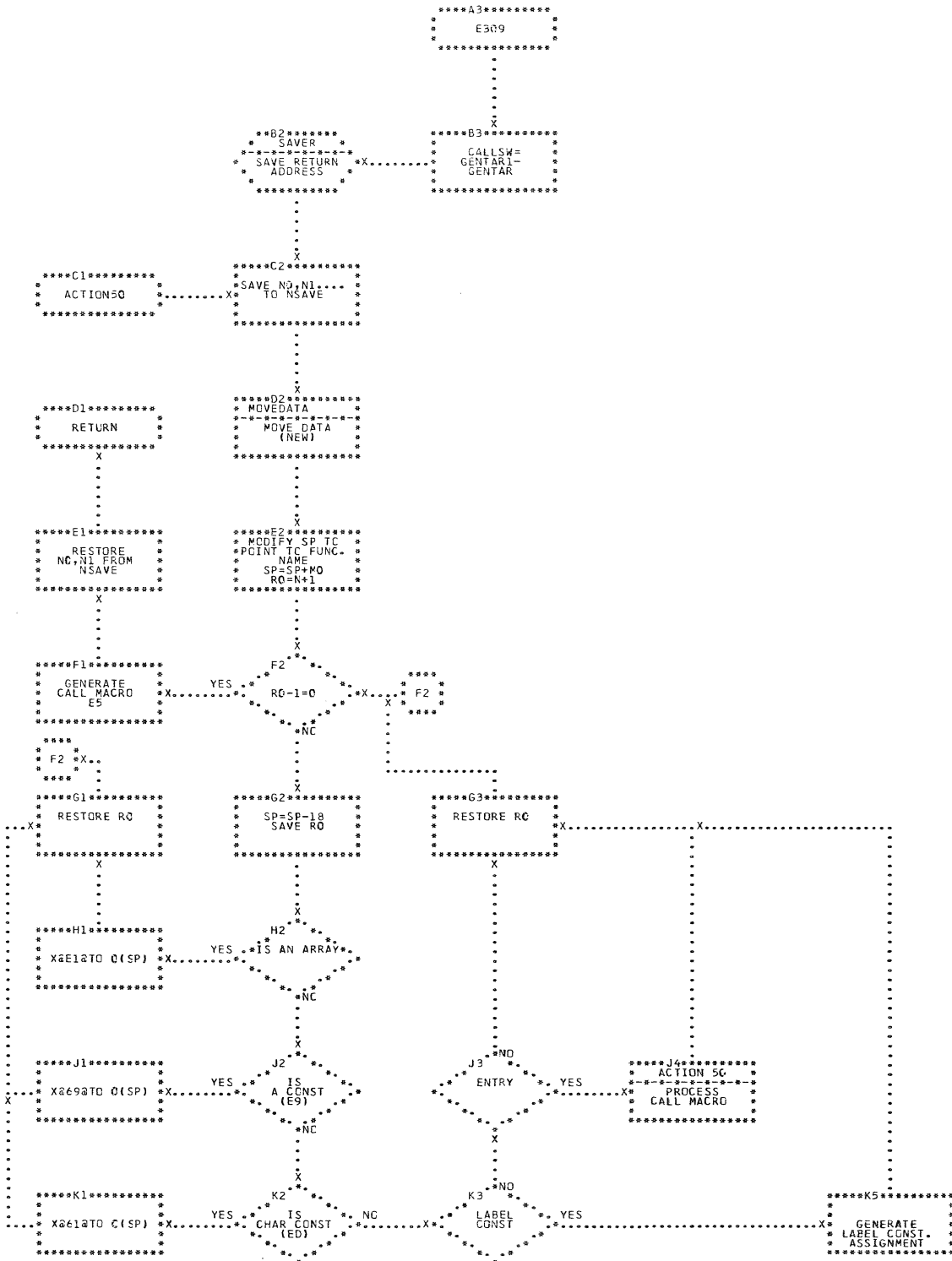
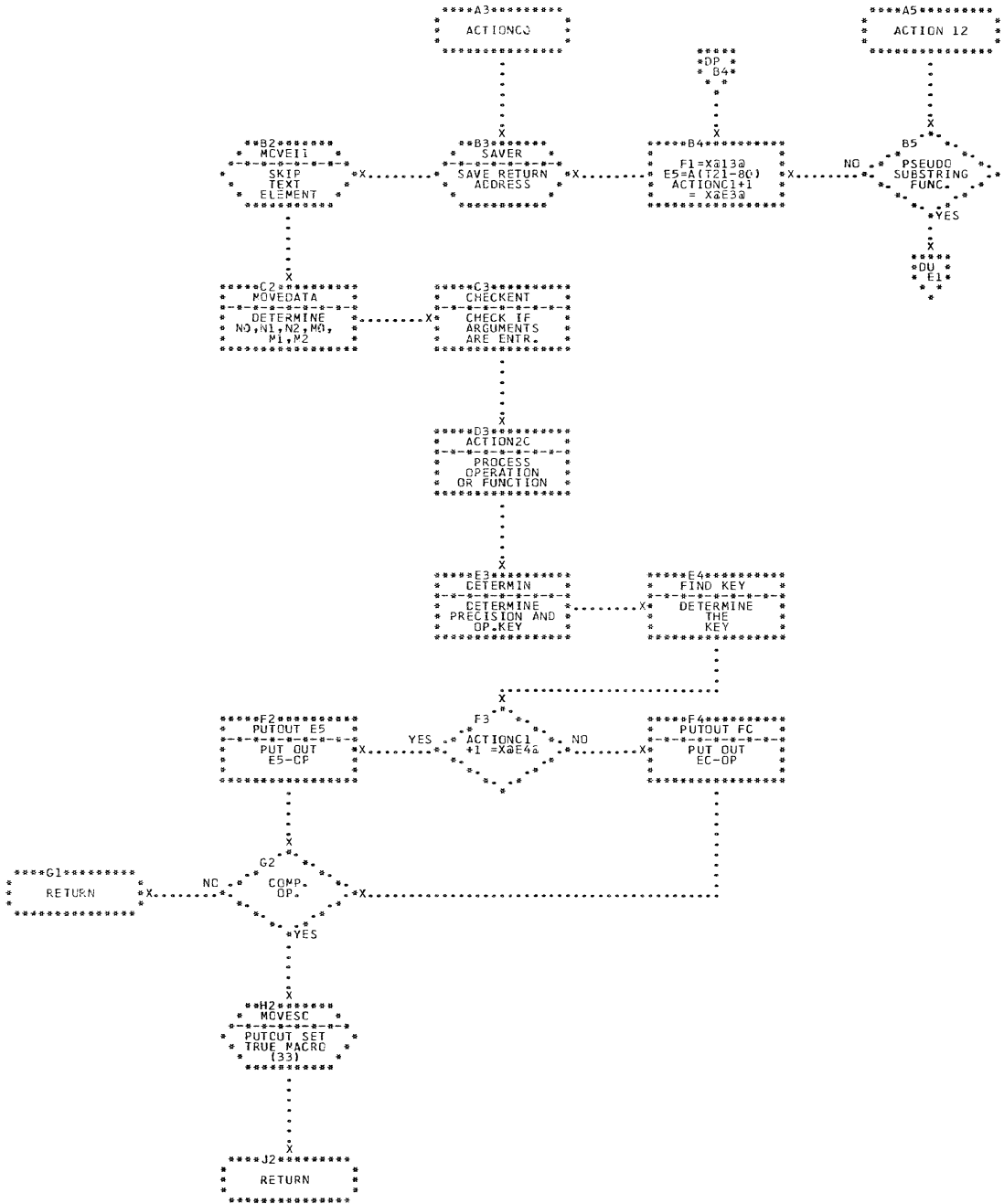
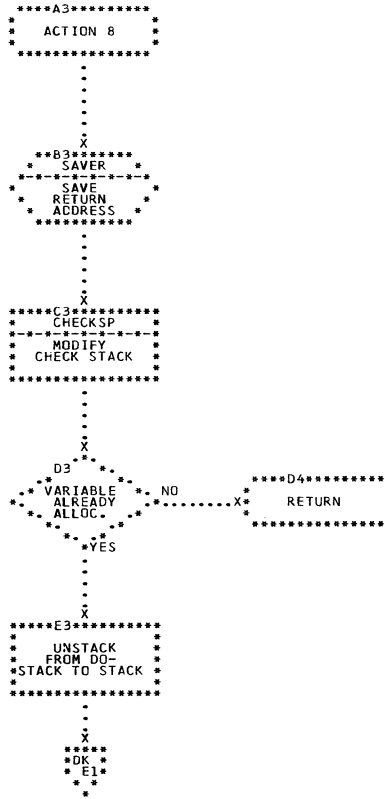
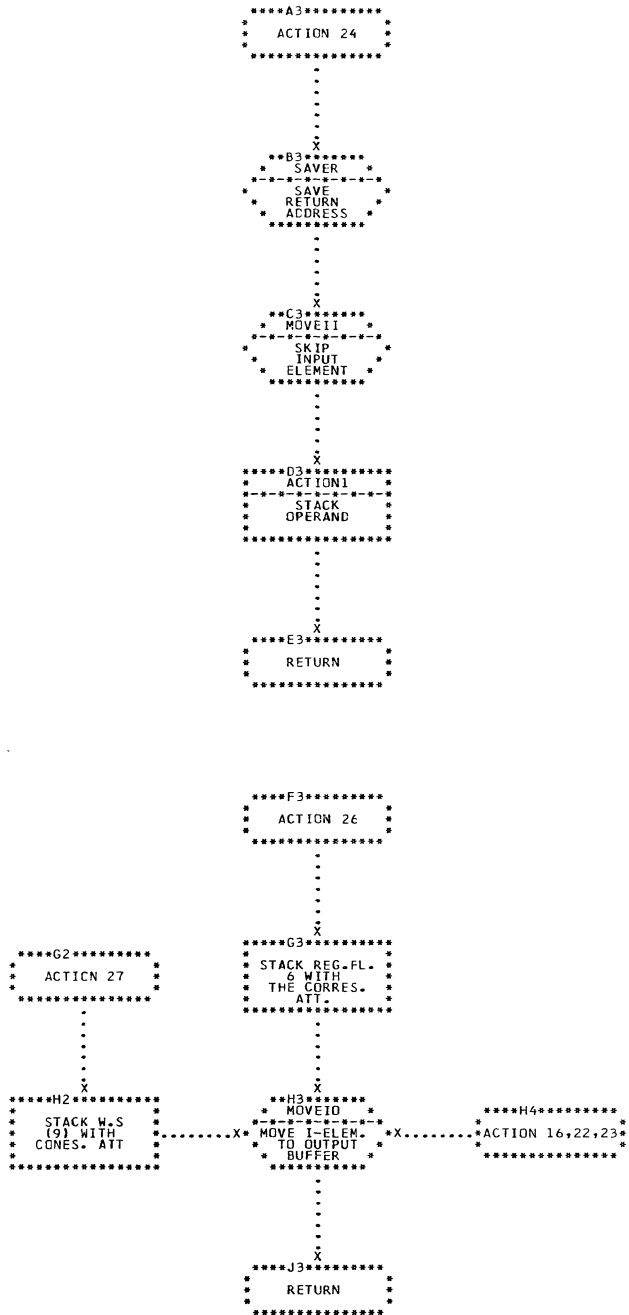
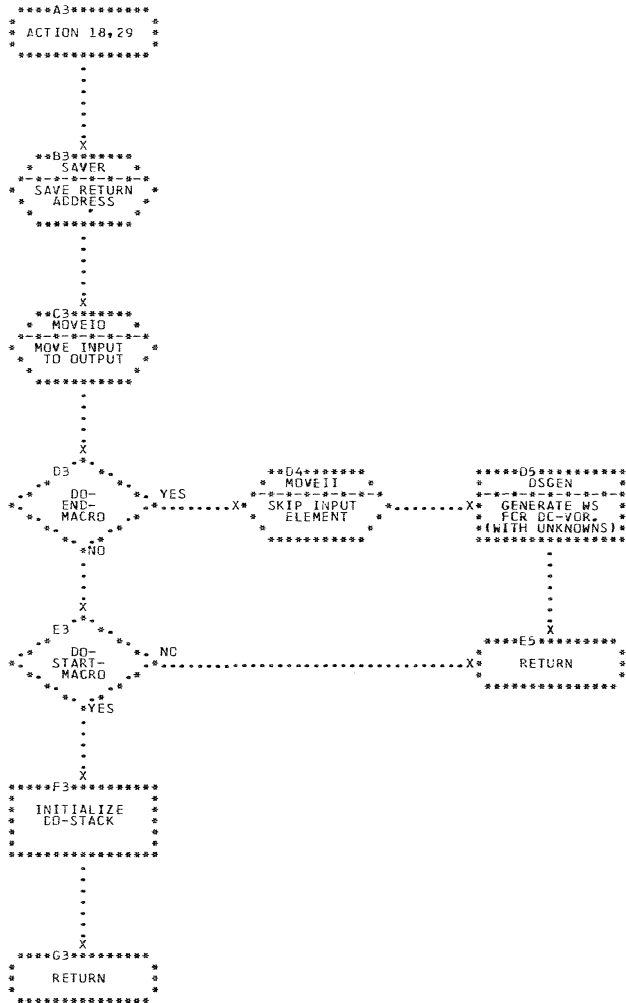


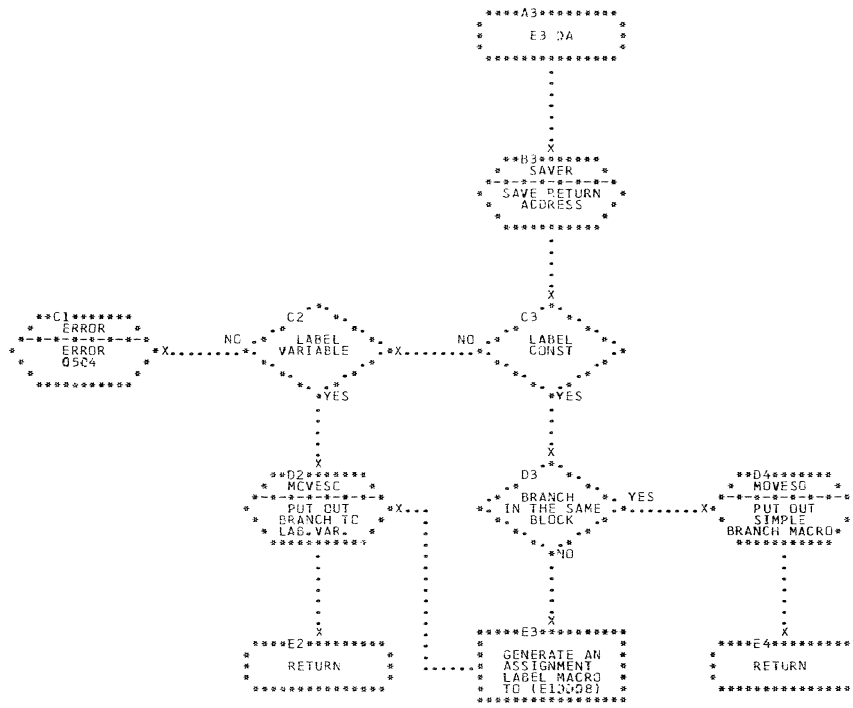
CHART DO. IJXD05 ENTRY NAME CALLS

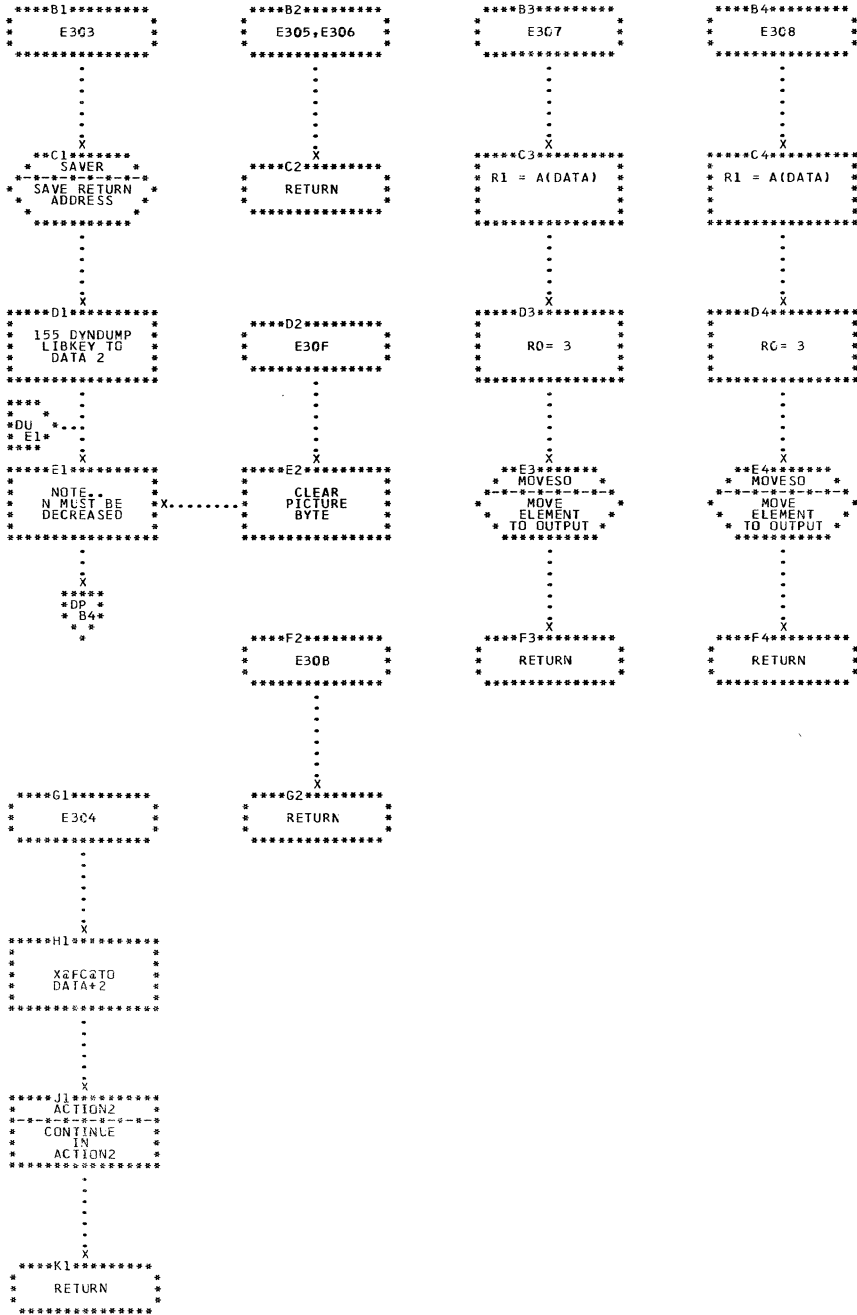


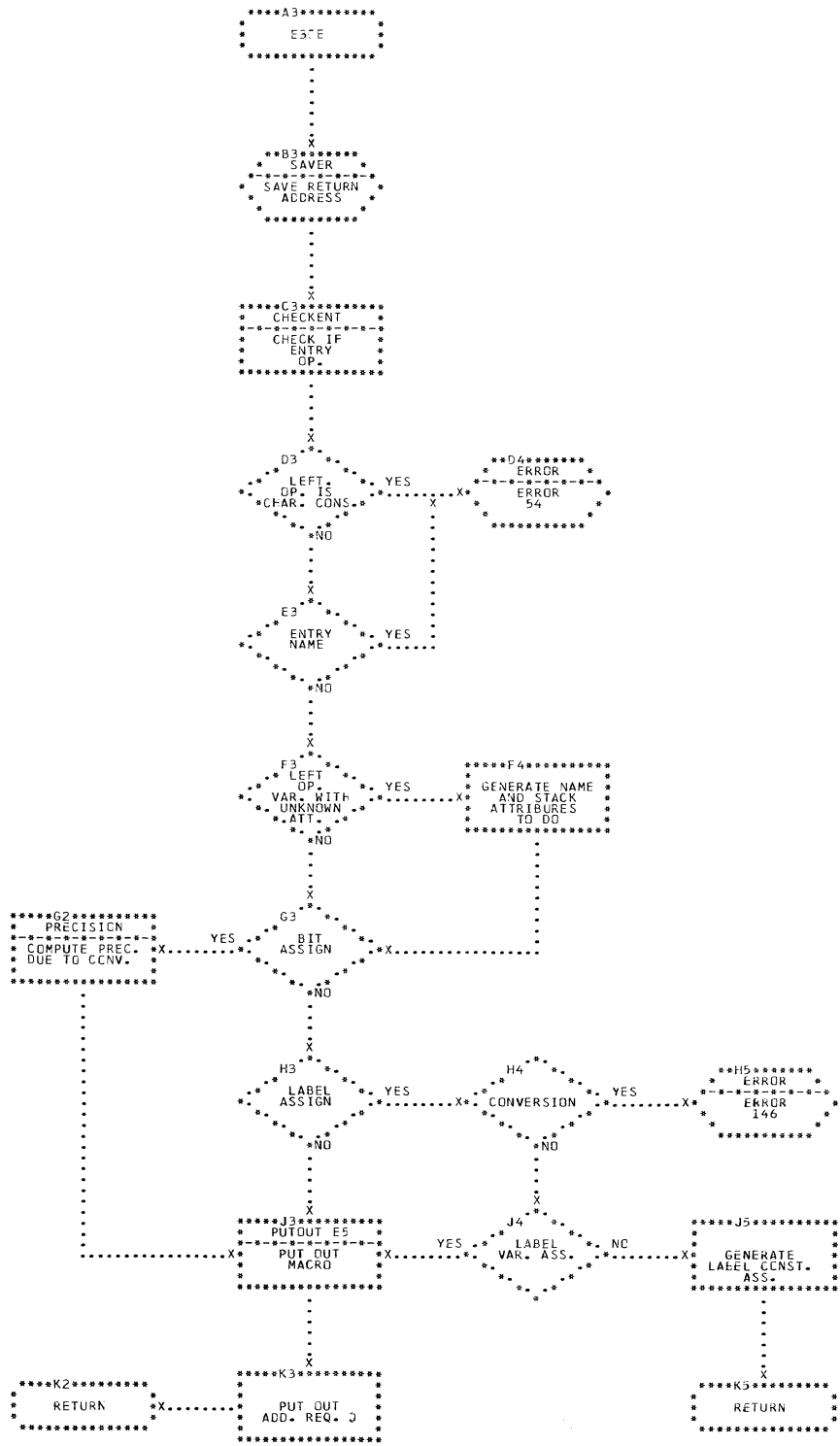


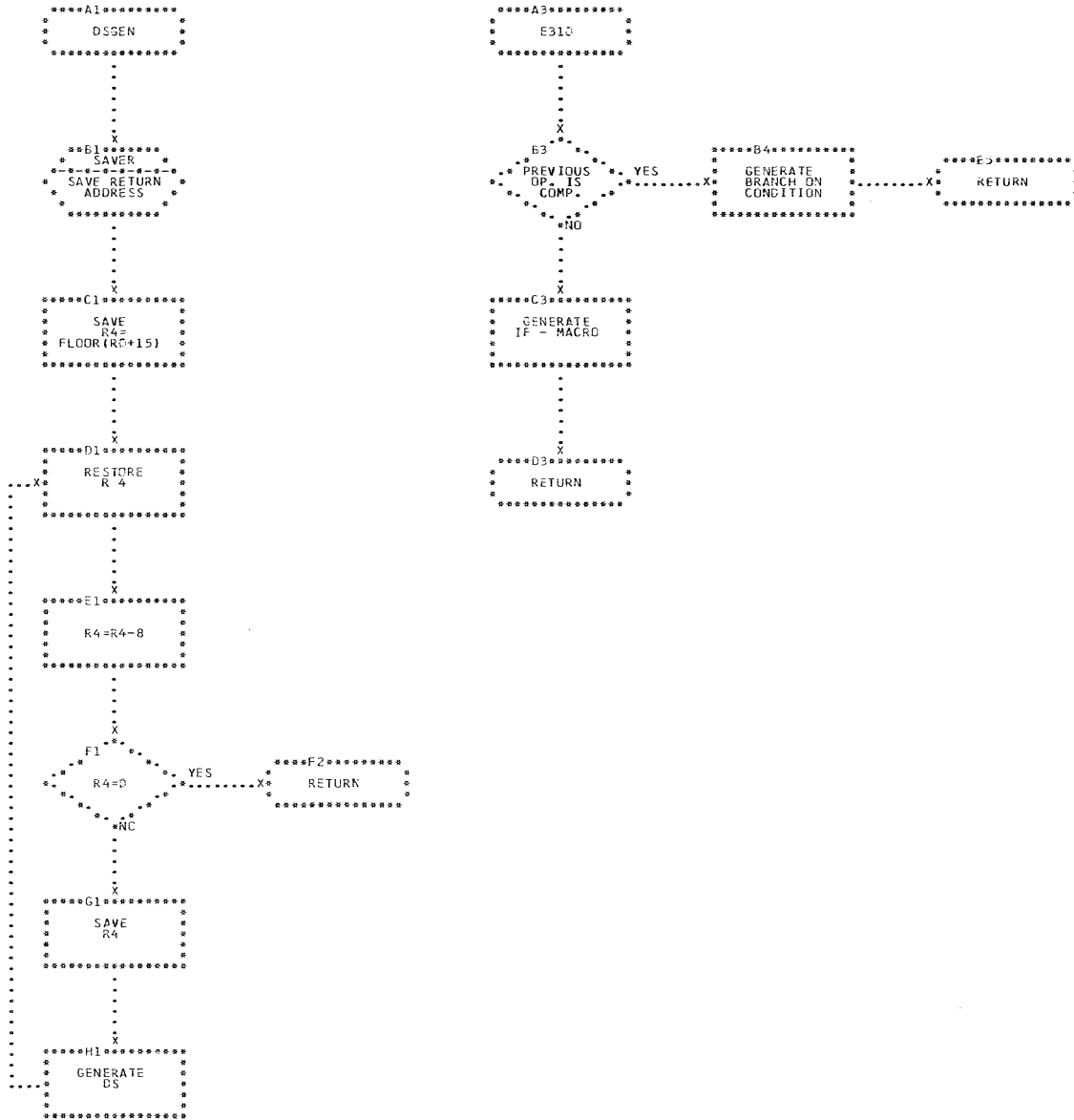


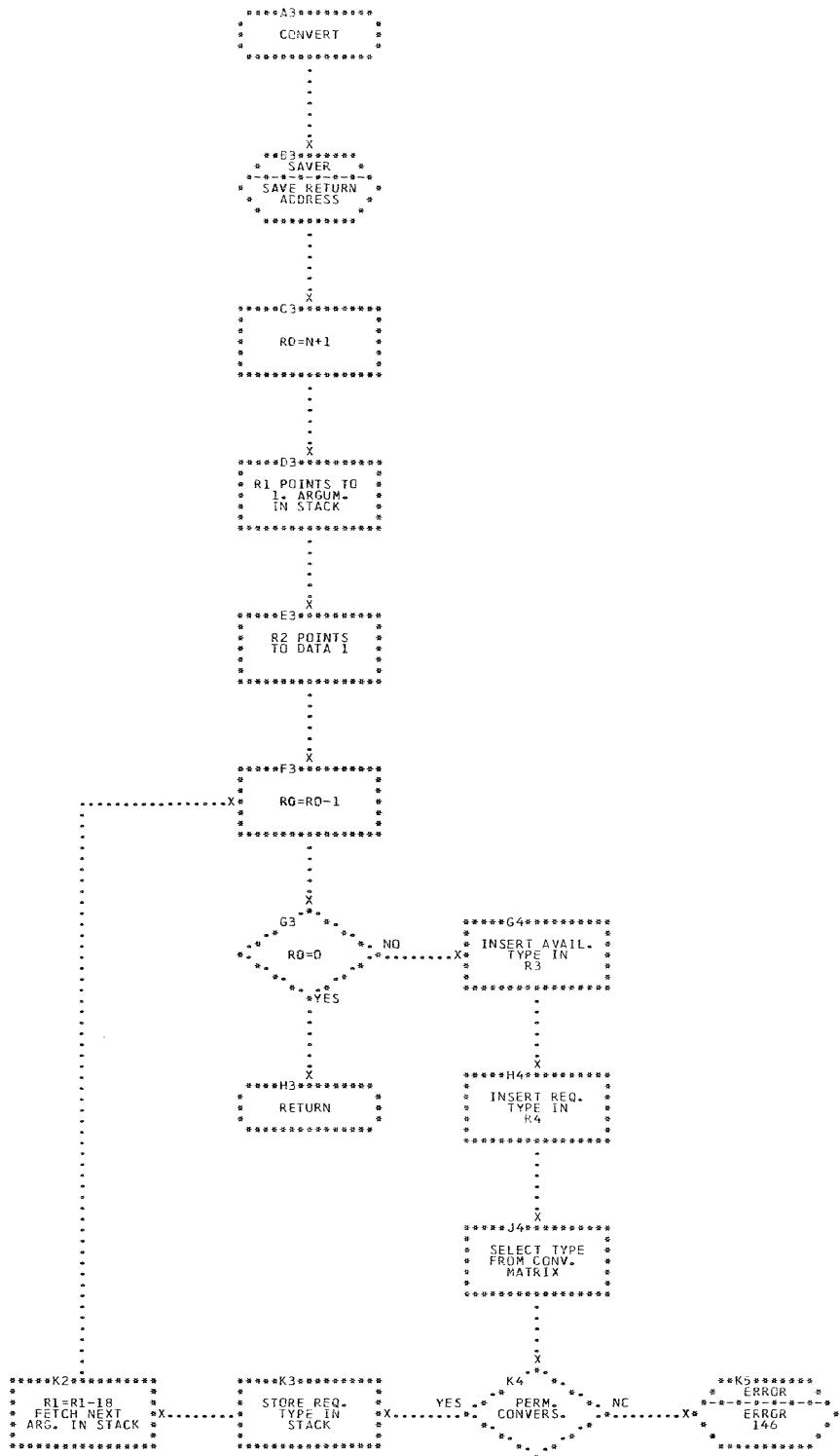


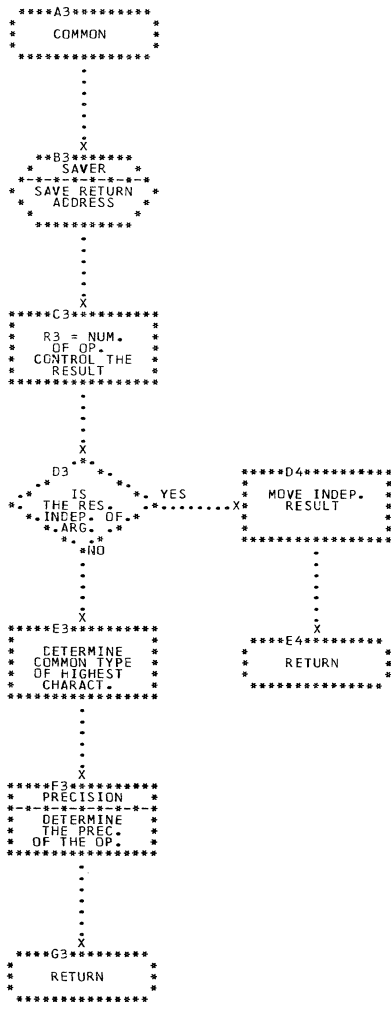












```

*****A1*****
PC 2
.
.
.
X
*****B1*****
PC
DETERMINE
P,Q,L AND
P-Q FOR 2. OP
.
.
.
X
*****C1*****
STORE P,Q,L
AND P-Q
FOR 2. OP
IN LPQ2
.
.
.
X
*****D1*****
R1 = R1+16
.
.
.
X
*****E1*****
PQ1
DETERMINE
P,Q,L AND
P-Q FOR 1. OP
.
.
.
X
*****F1*****
INSERT NUM.
OF ARG. IN
R3
.
.
.
X
*****G1*****
RETURN

```

```

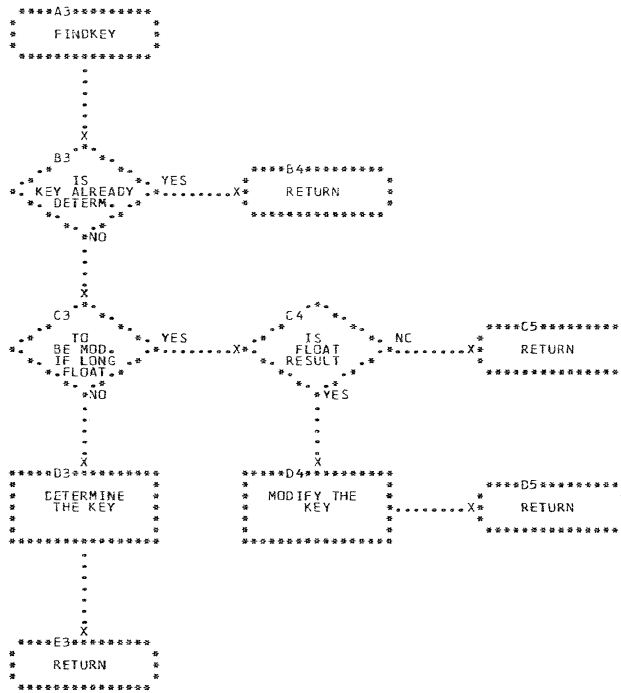
*****A3*****
PQ
.
.
.
X
*****B3*****
R4 = 15(R1)
R5 = 16(R1)
(P,Q)
.
.
.
X
*****C3*****
R3 = L
(IN BYTES)
.
.
.
X
*****D3*****
R2=P-Q
.
.
.
X
*****E3*****
RETURN

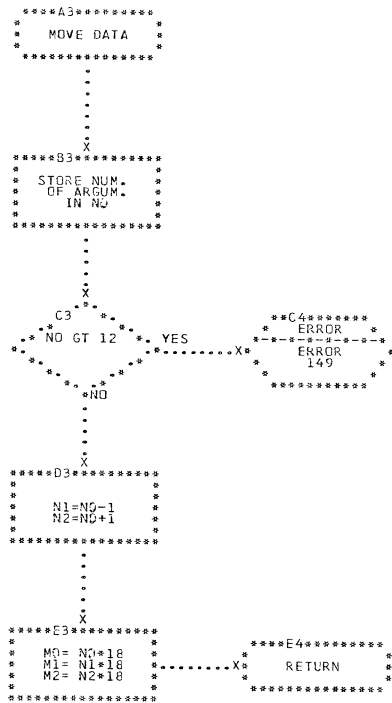
```

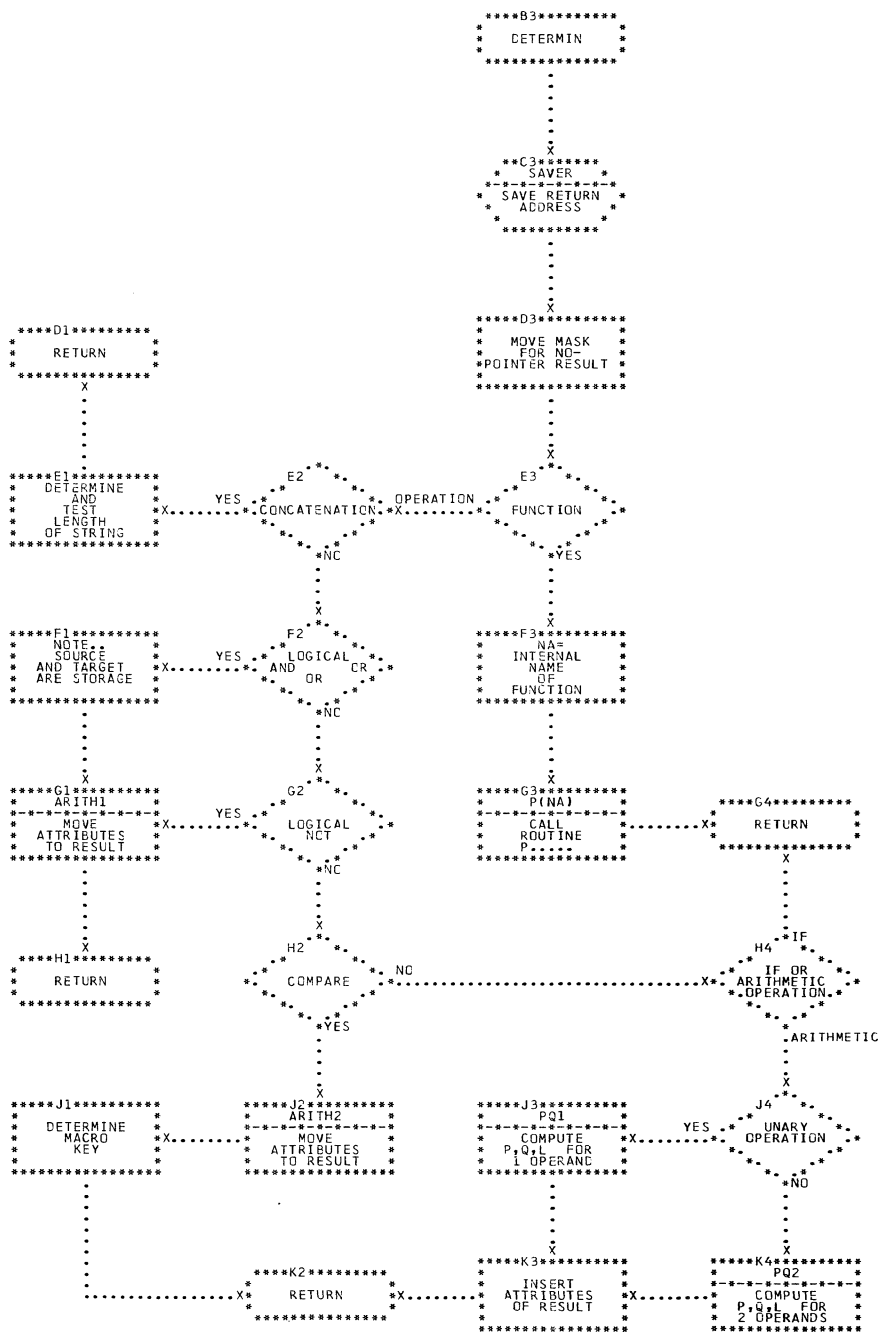
```

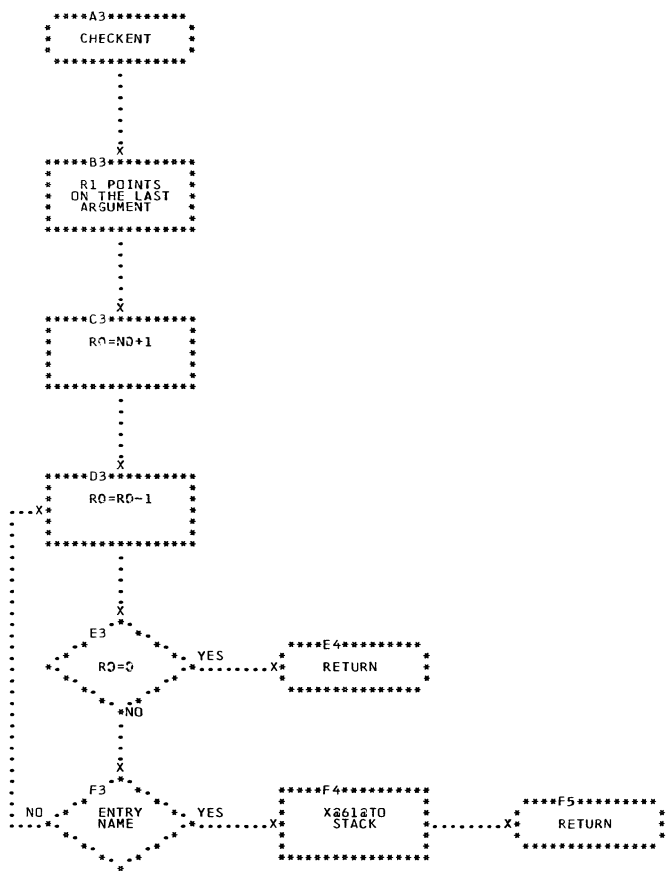
*****A5*****
PC 1
.
.
.
X
*****B5*****
PC
DETERMINE
P,Q,L AND
P-Q
.
.
.
X
*****C5*****
STORE P,Q,L
AND P-Q IN
LPQ1
.
.
.
X
*****D5*****
RETURN

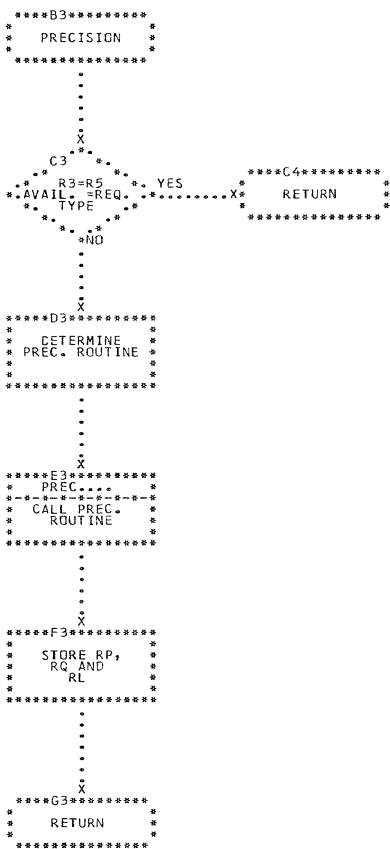
```

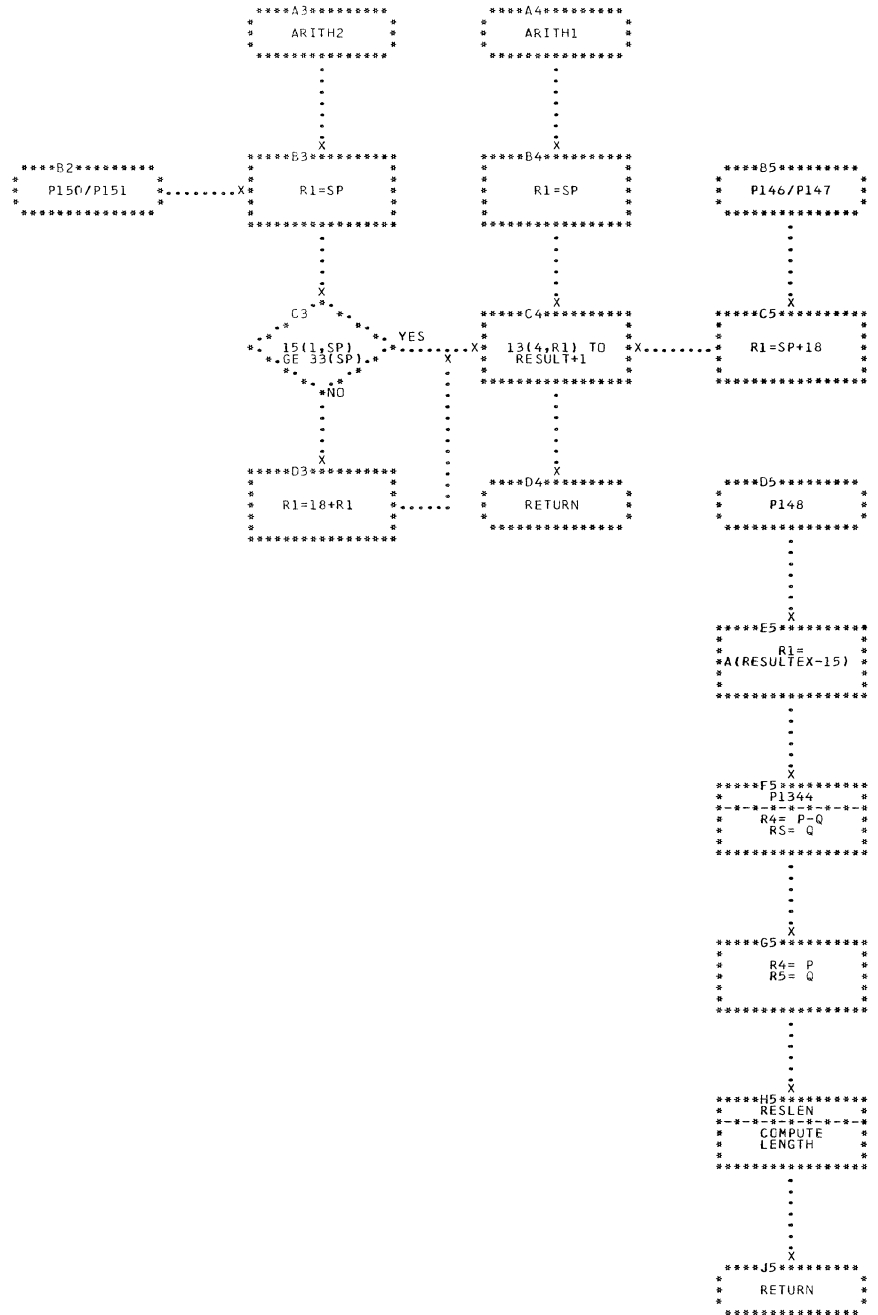


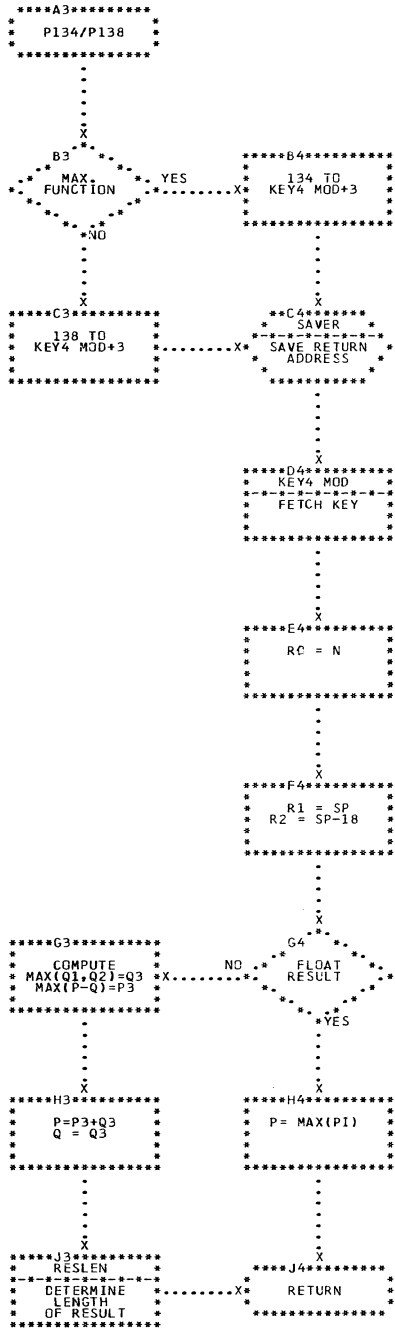












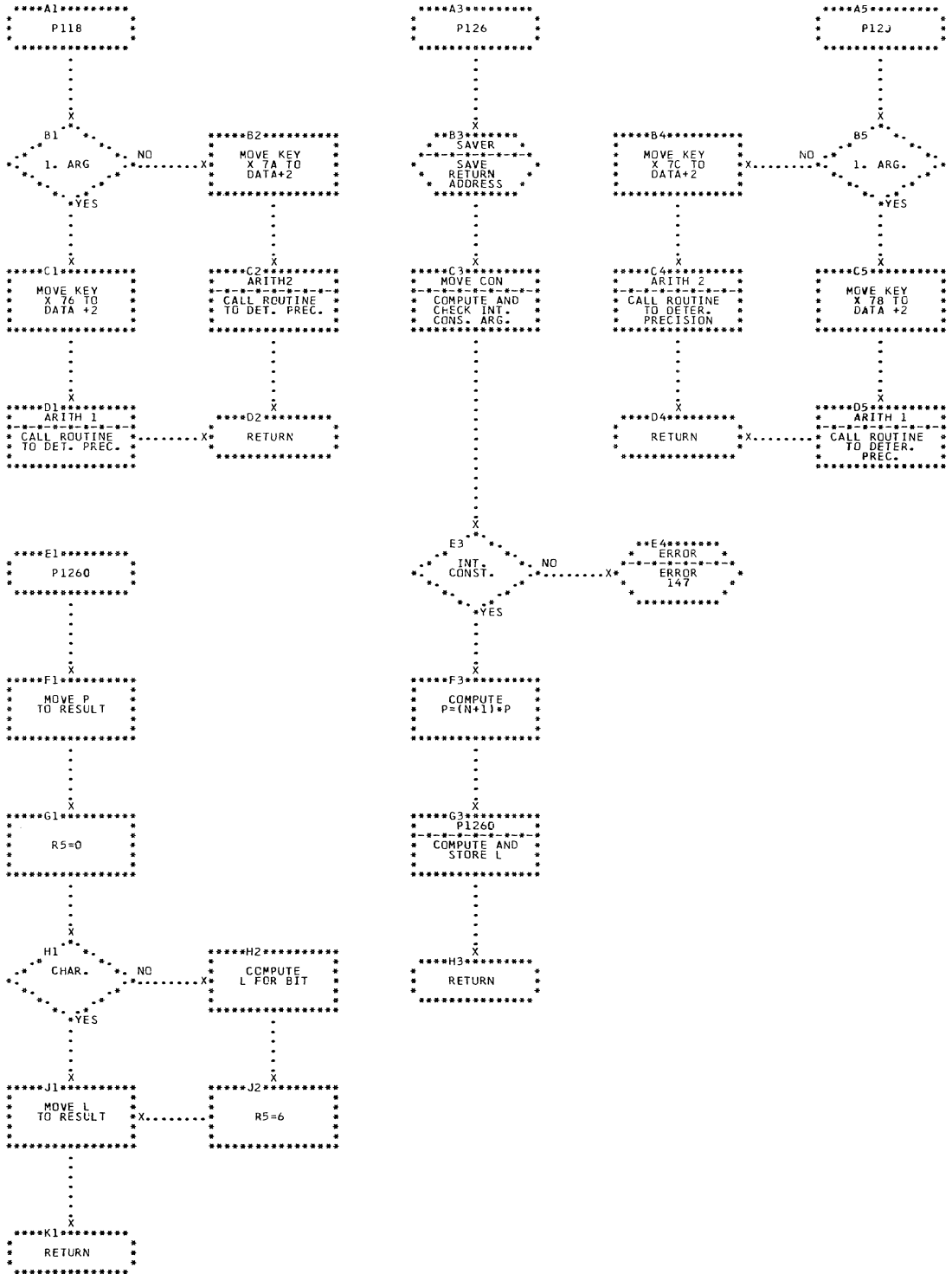
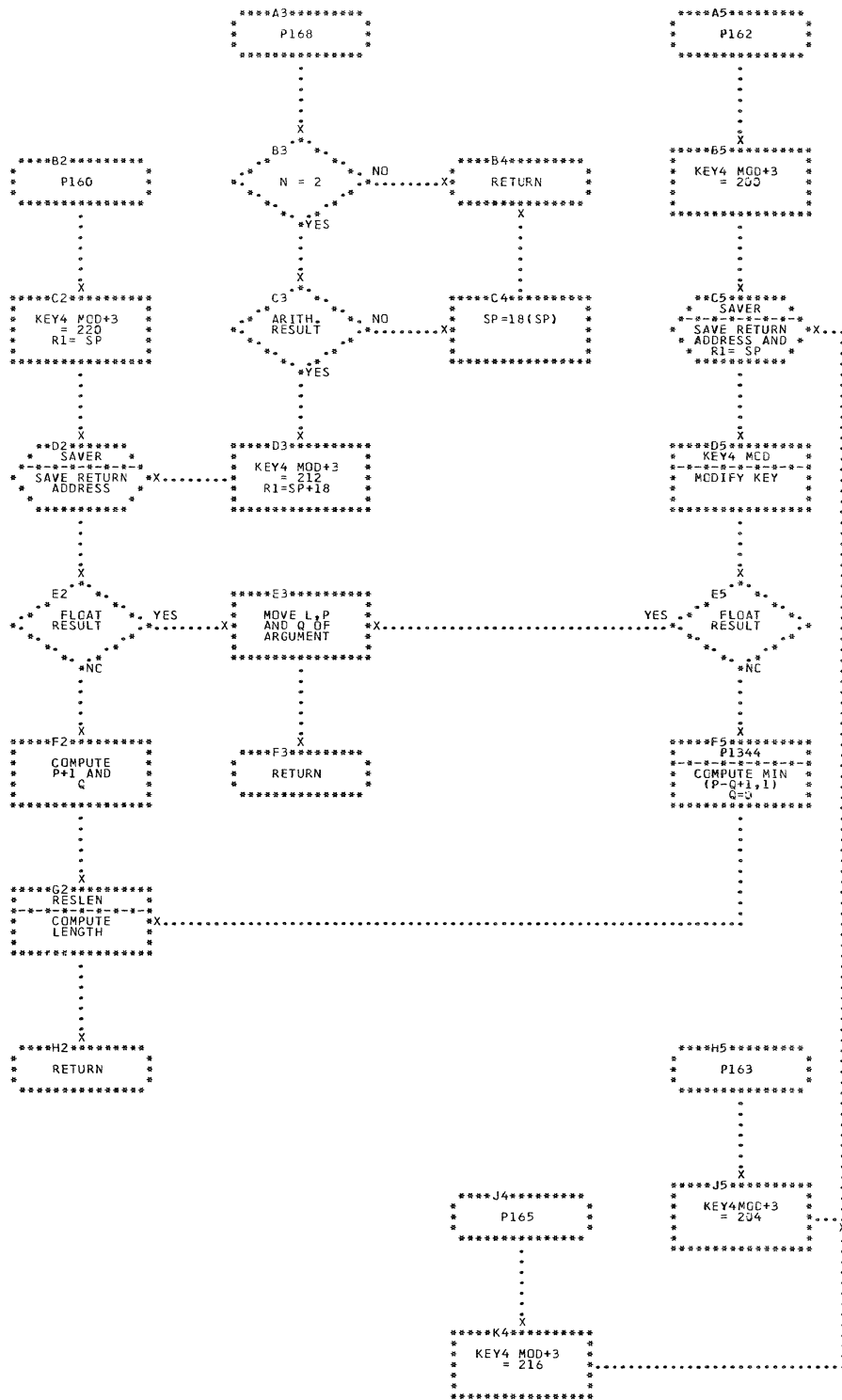
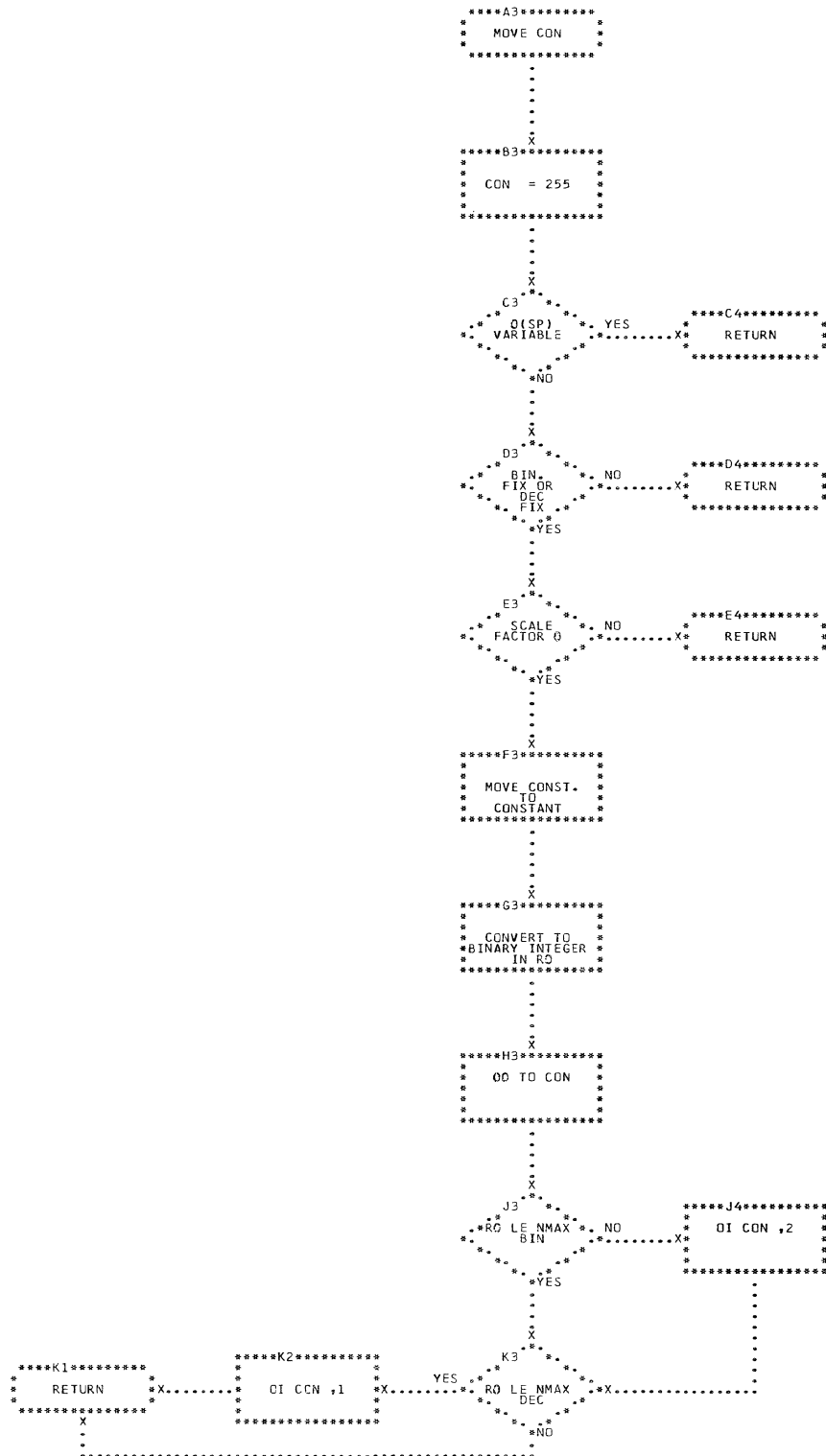
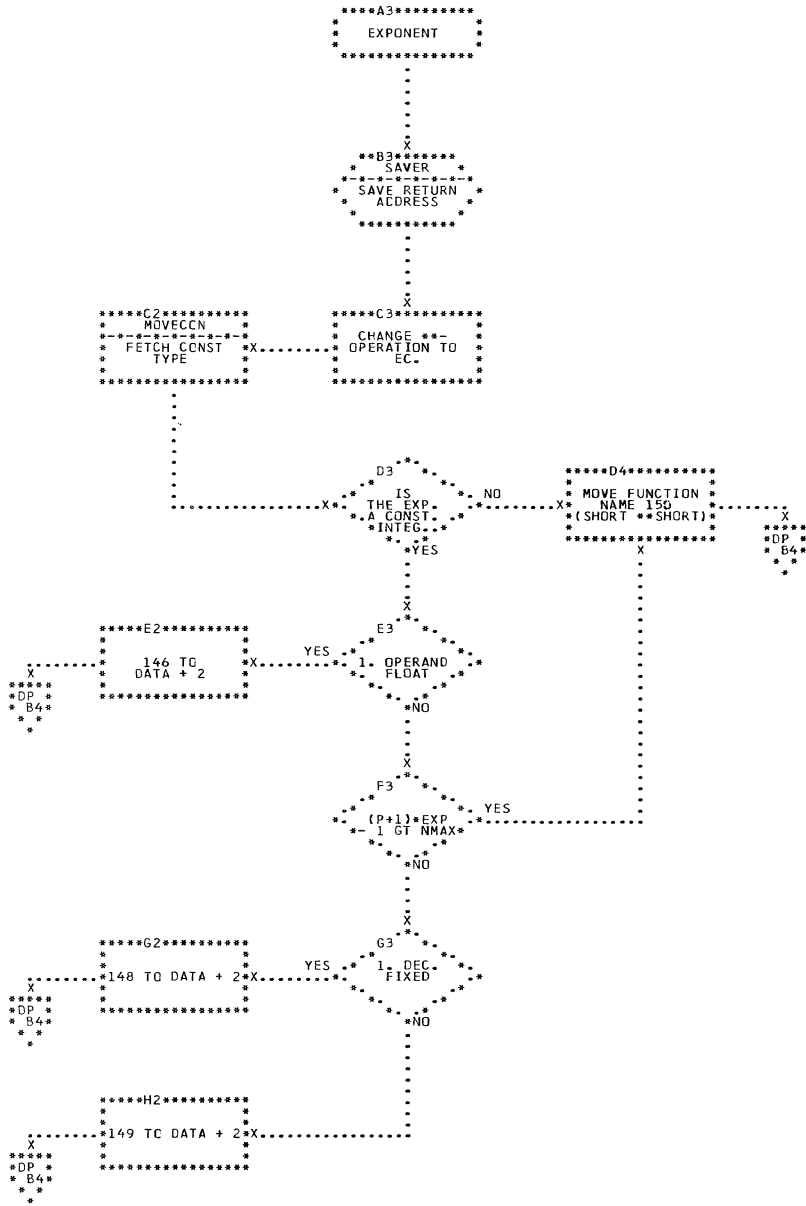


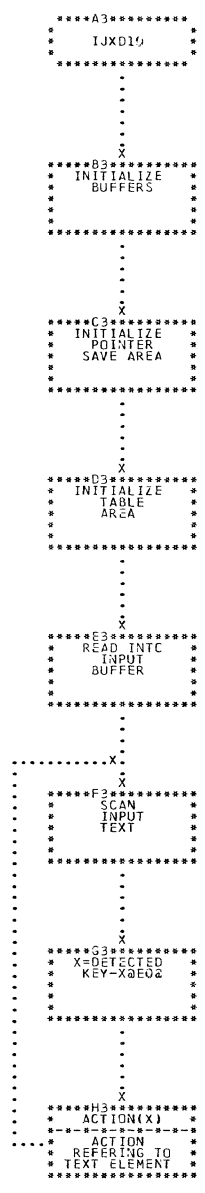
CHART EK. IJXD05

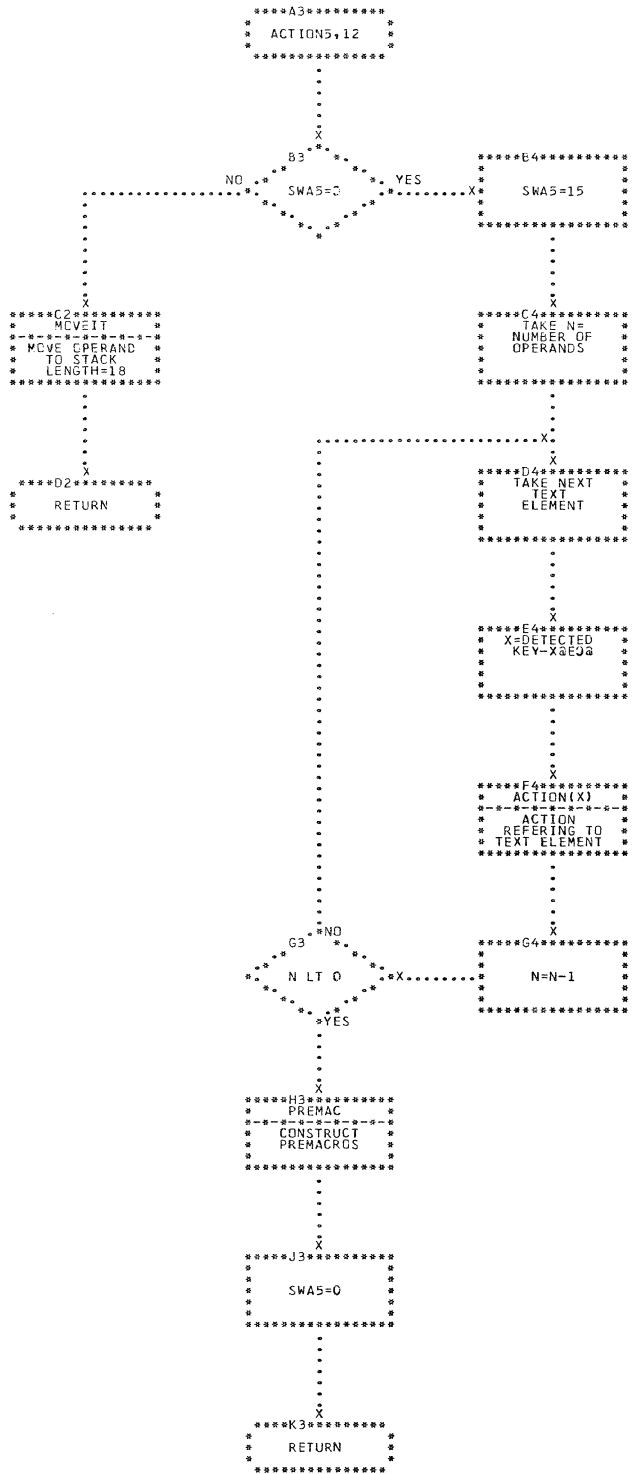
PRECISION ROUTINES

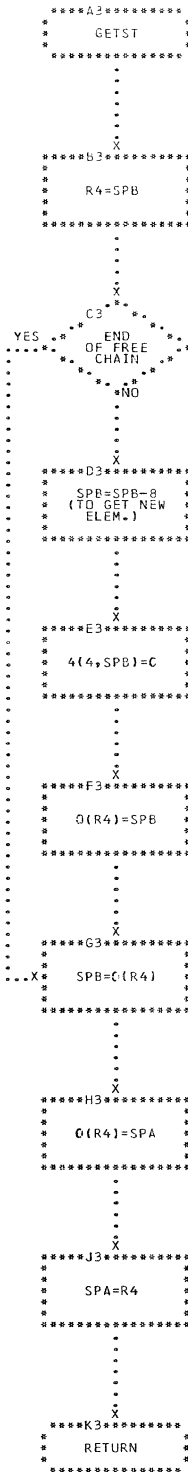


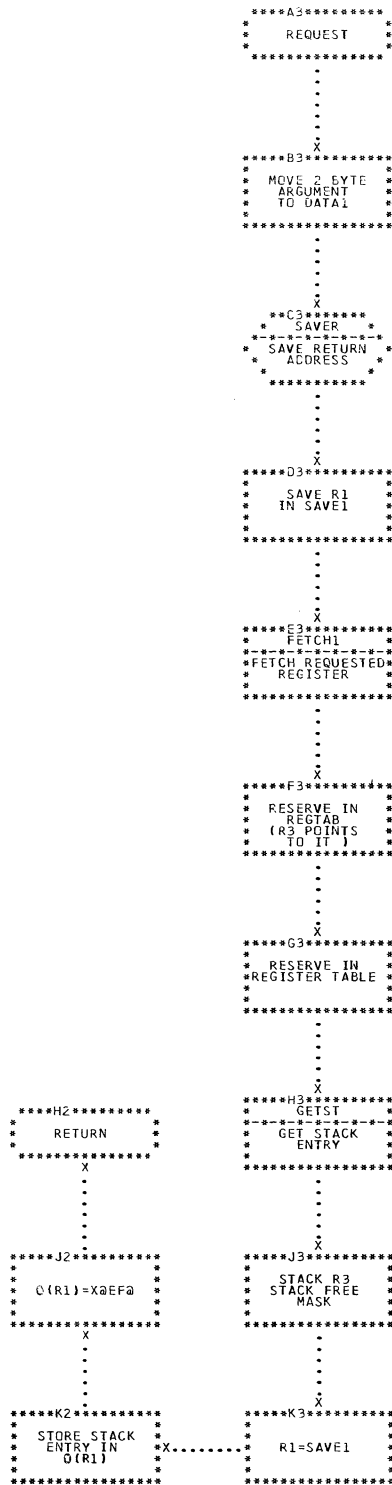


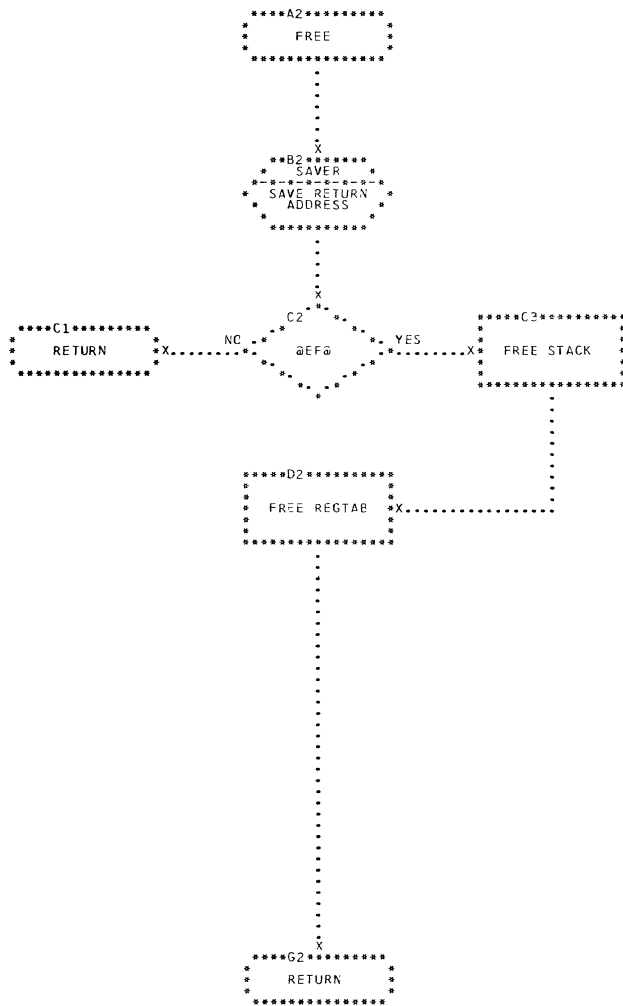


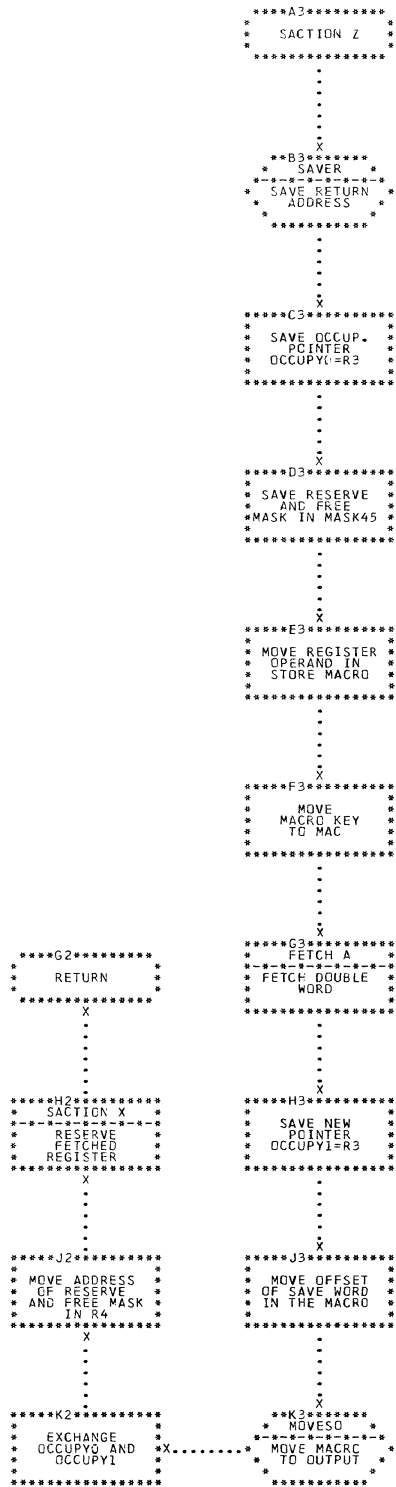


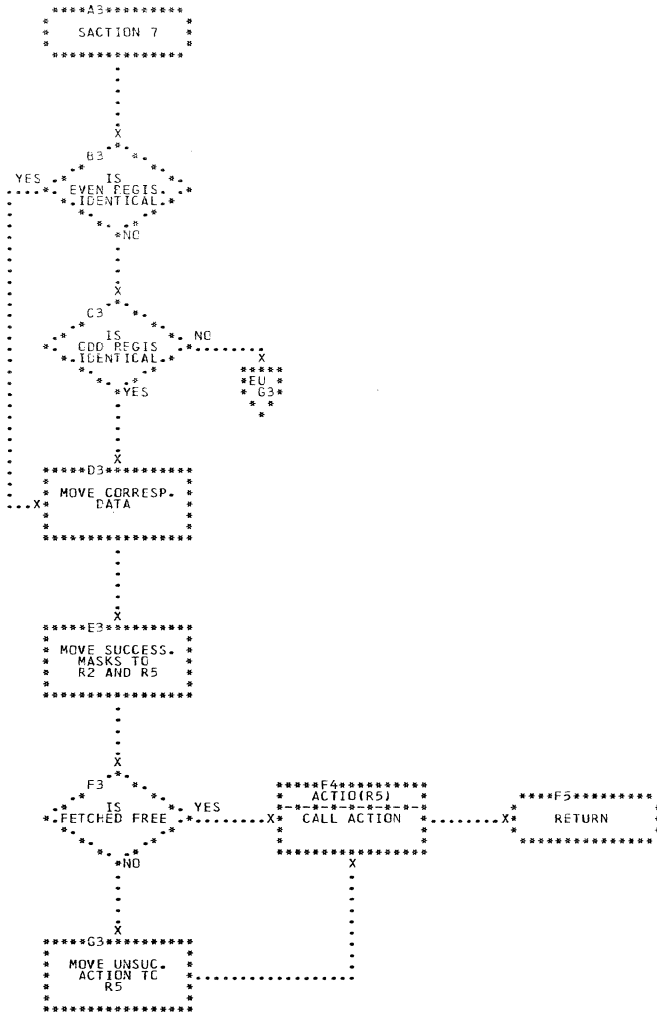


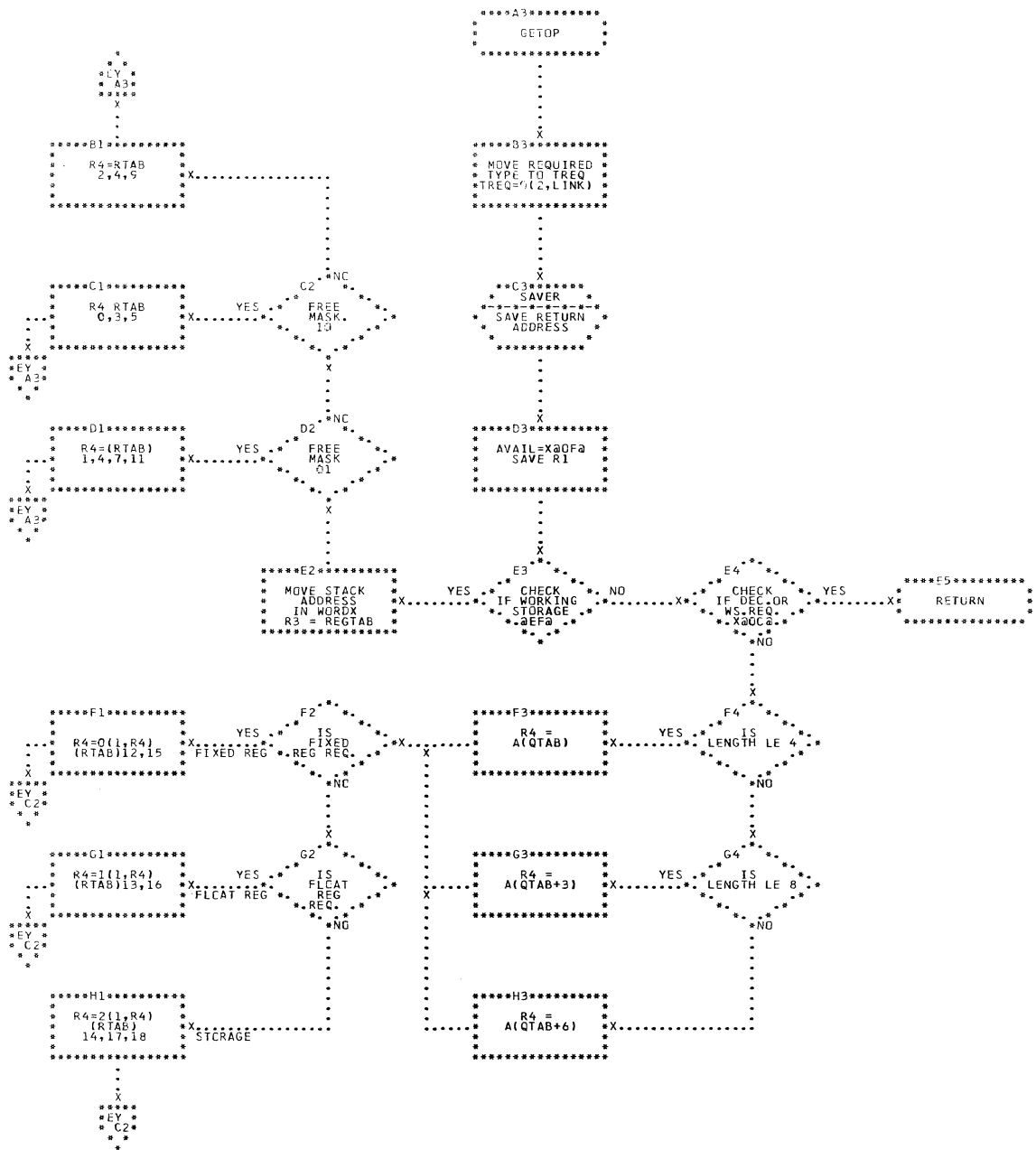


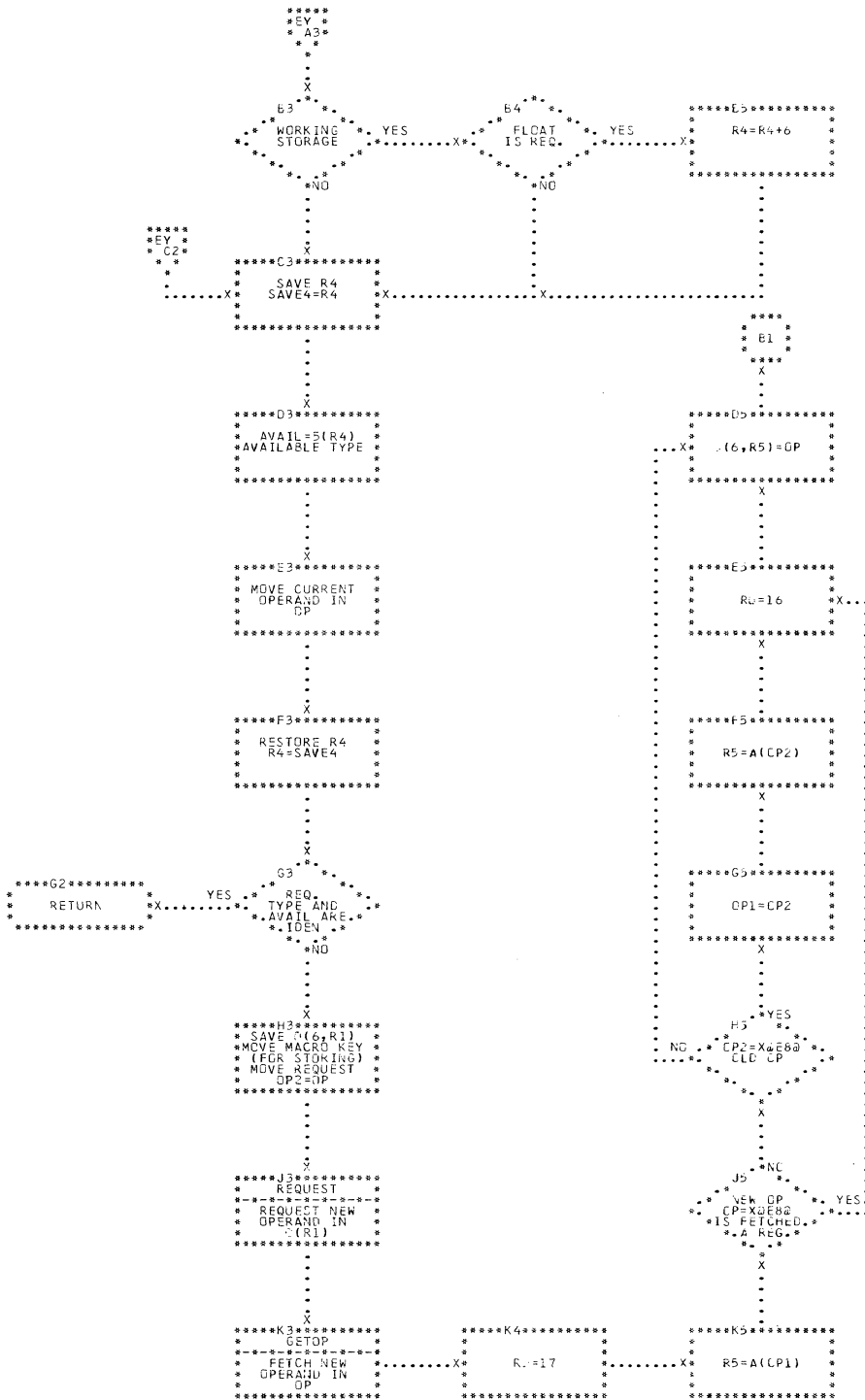
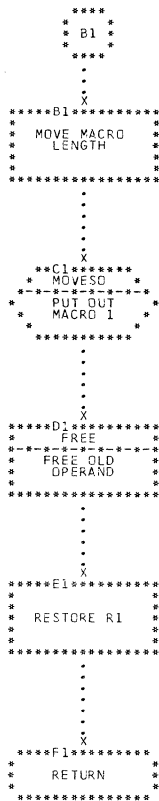


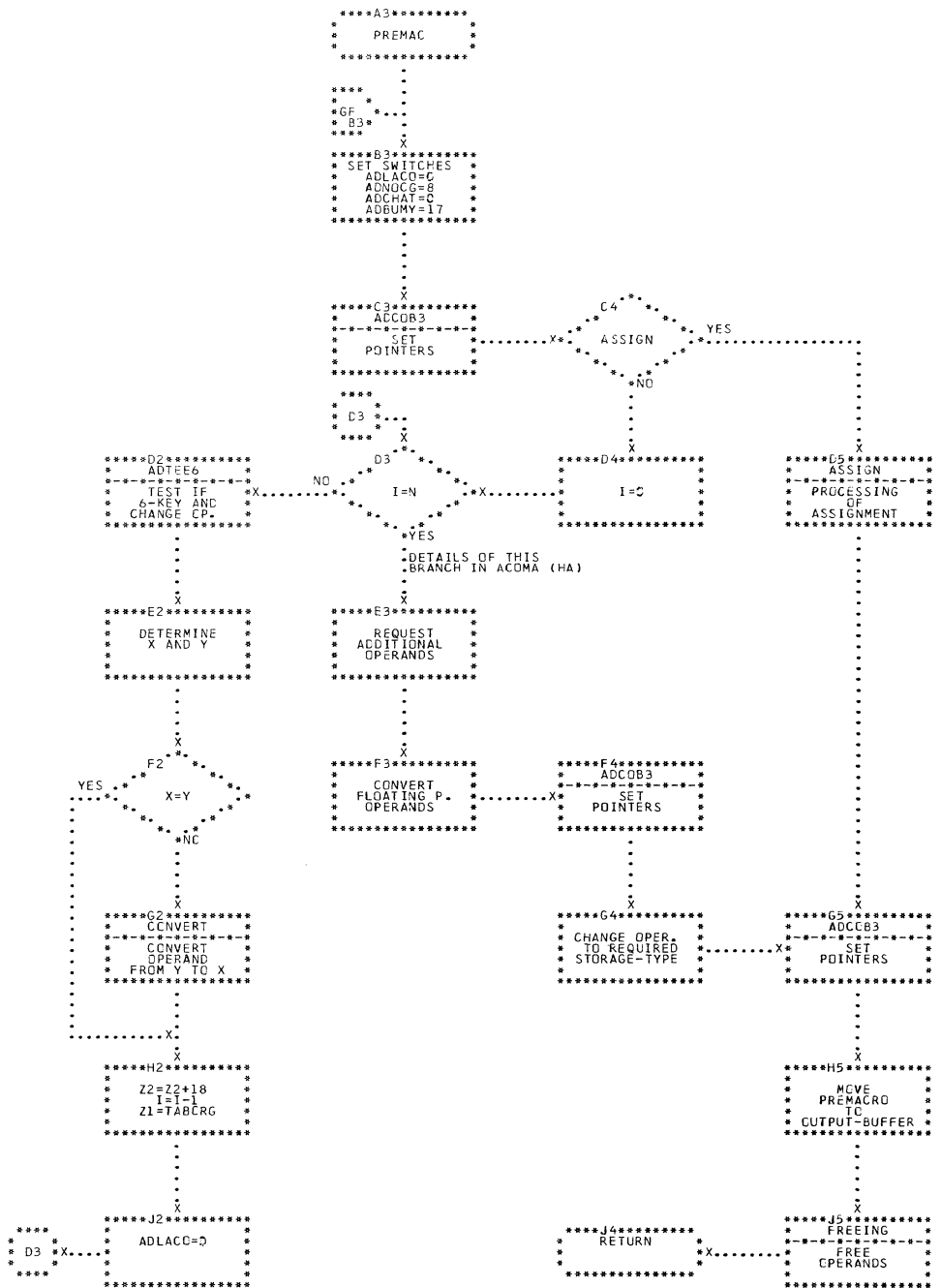












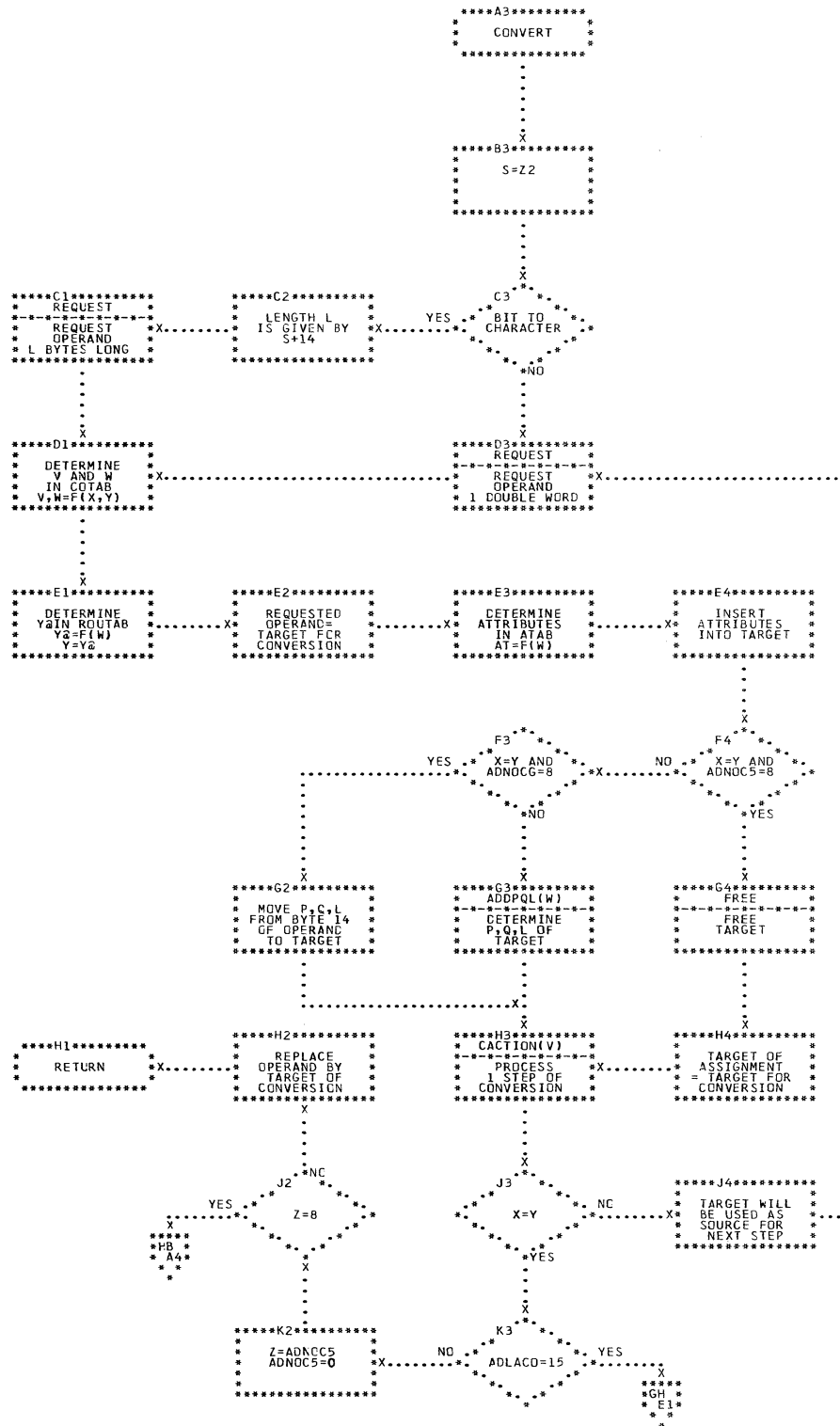


CHART 66. IJXD10

CONVERSION OF OPERANDS

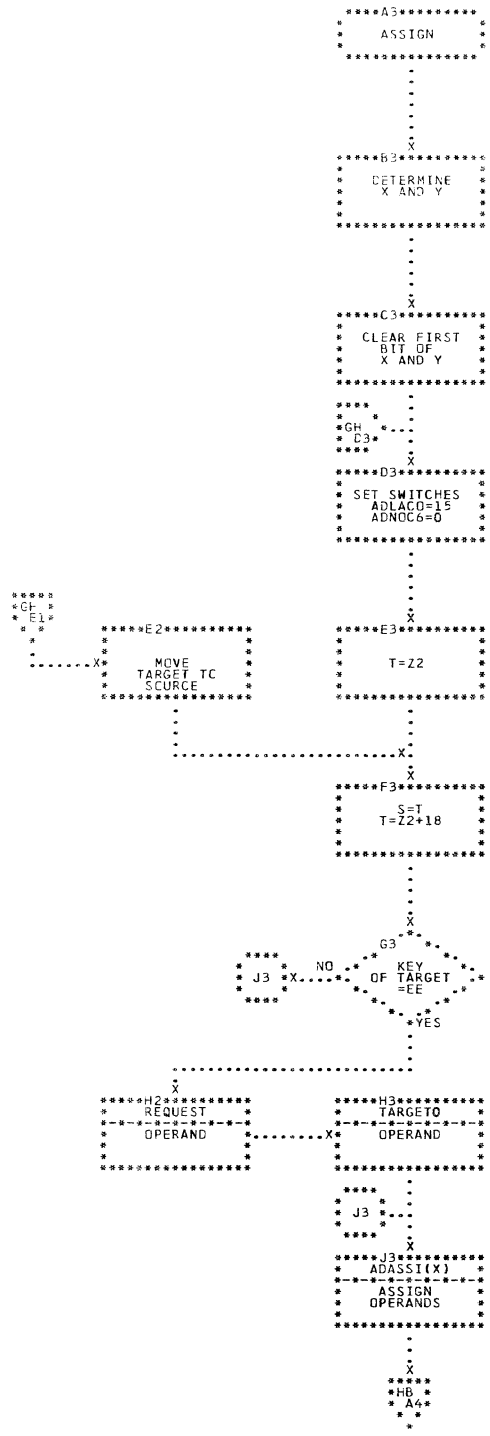
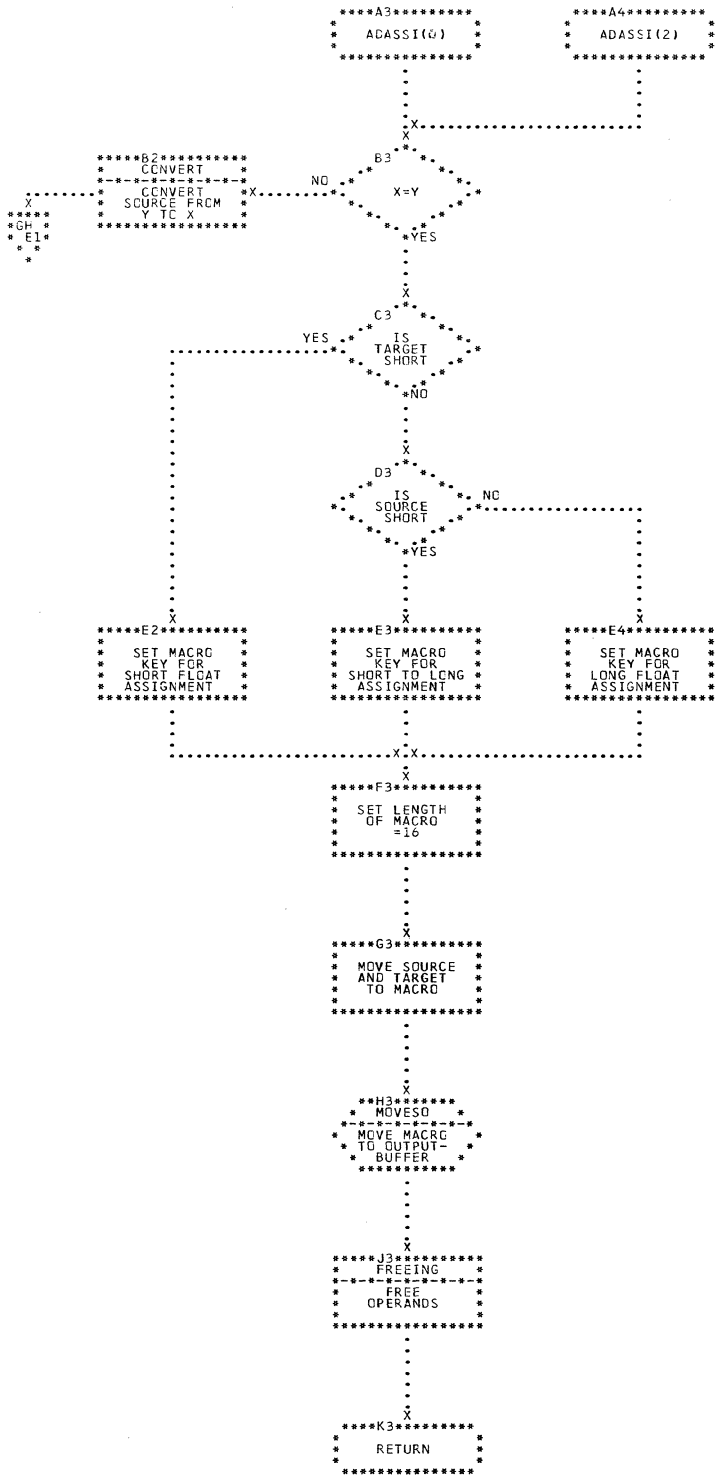
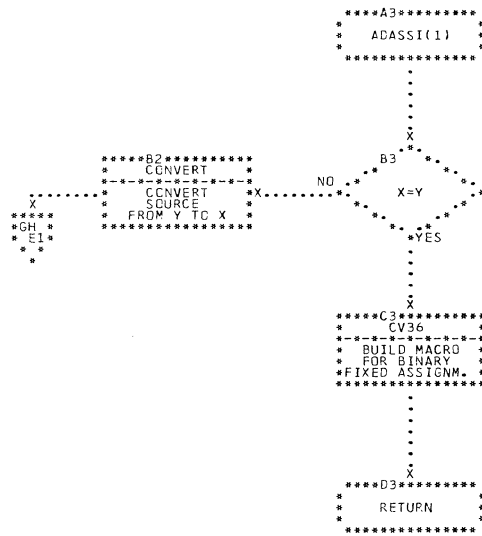


CHART GH. IJXD10 ASSIGNMENTS





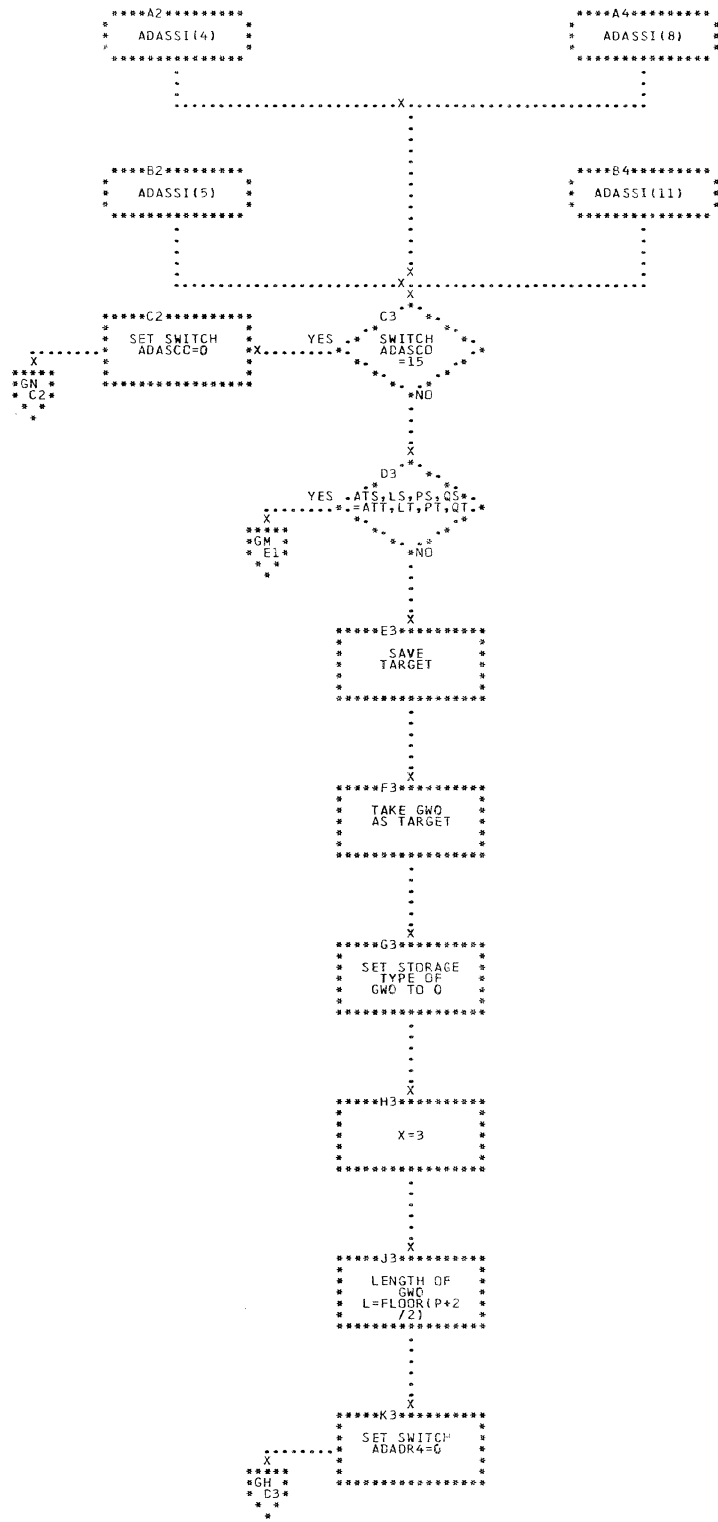
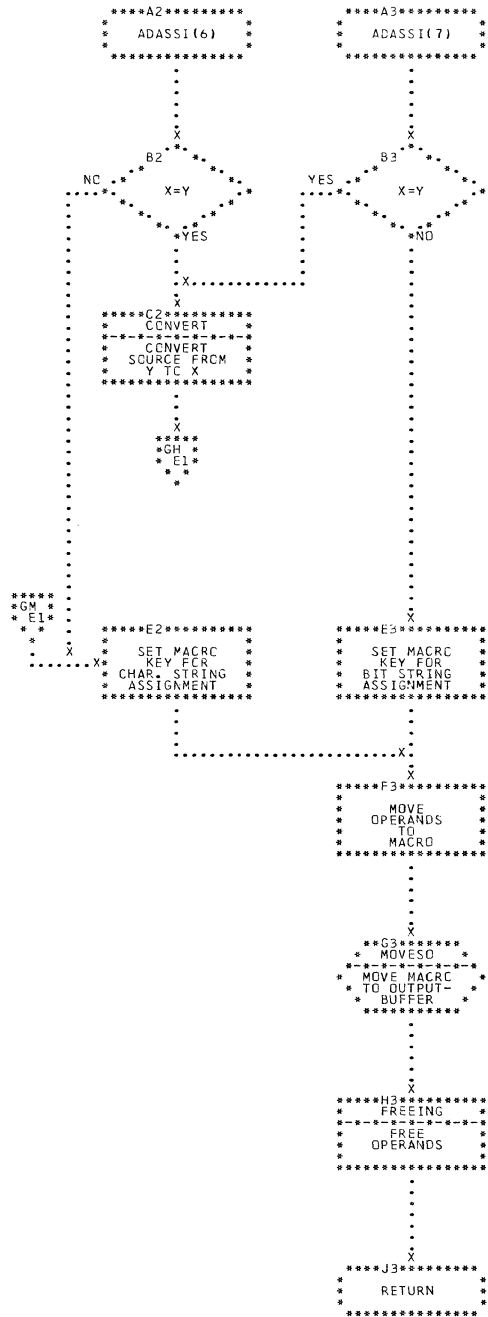
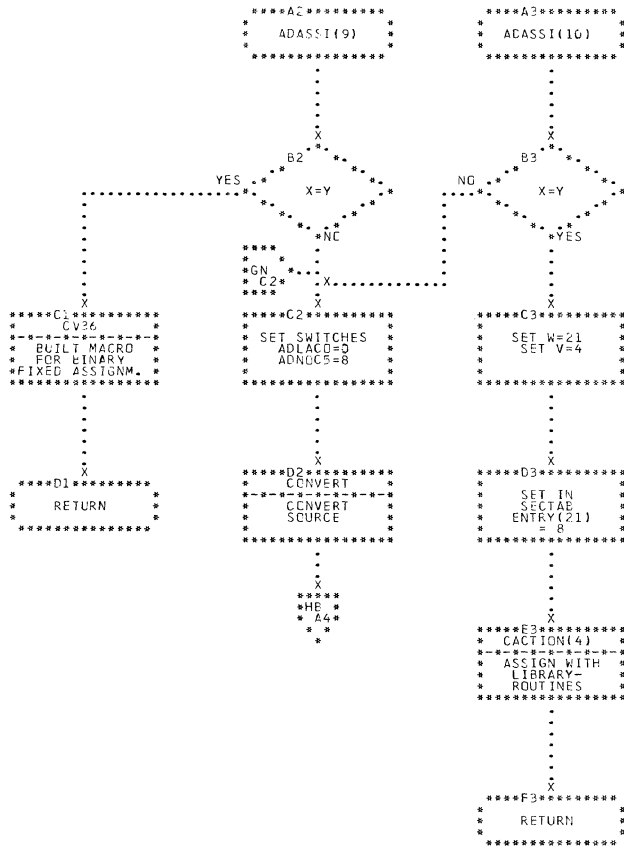
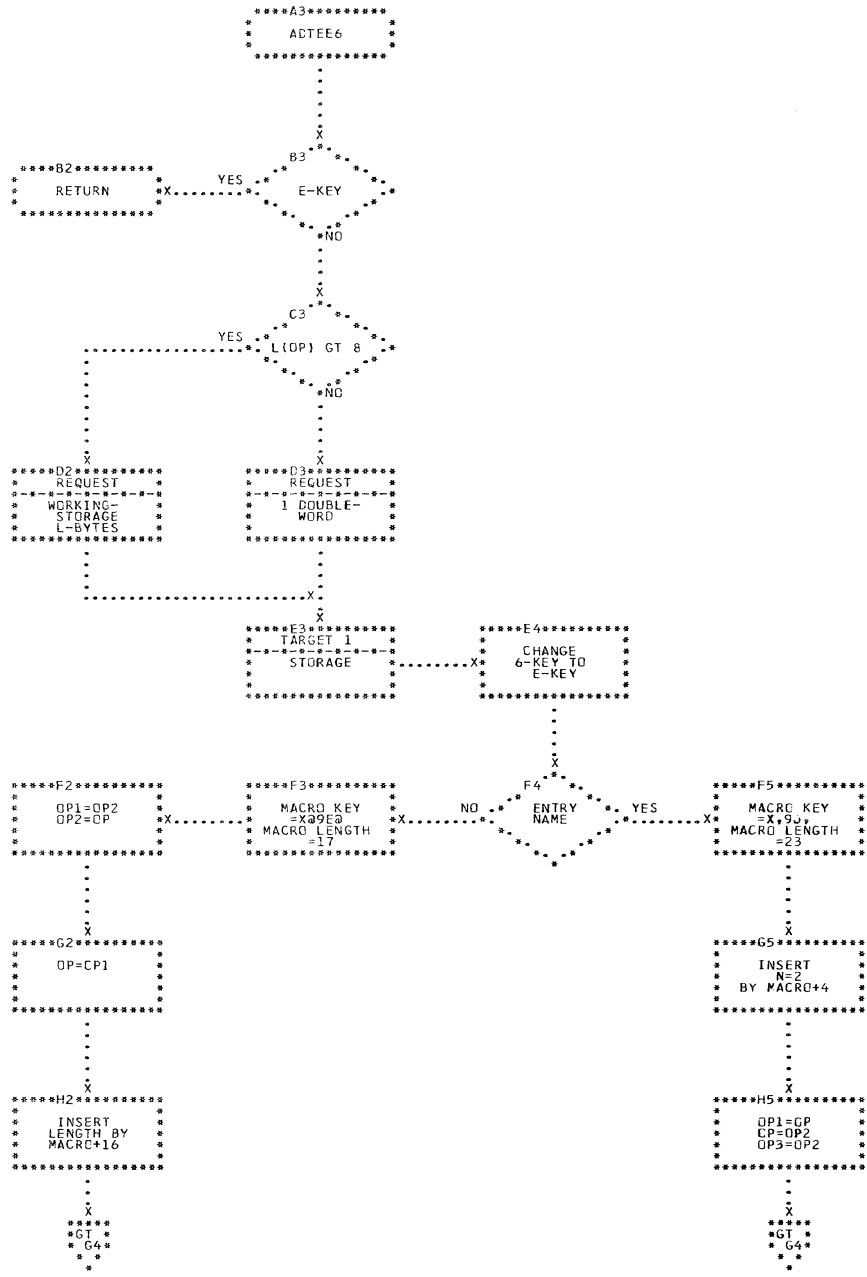


CHART GL. IJXD10

ZONED AND NUMERIC ASSIGNMENTS



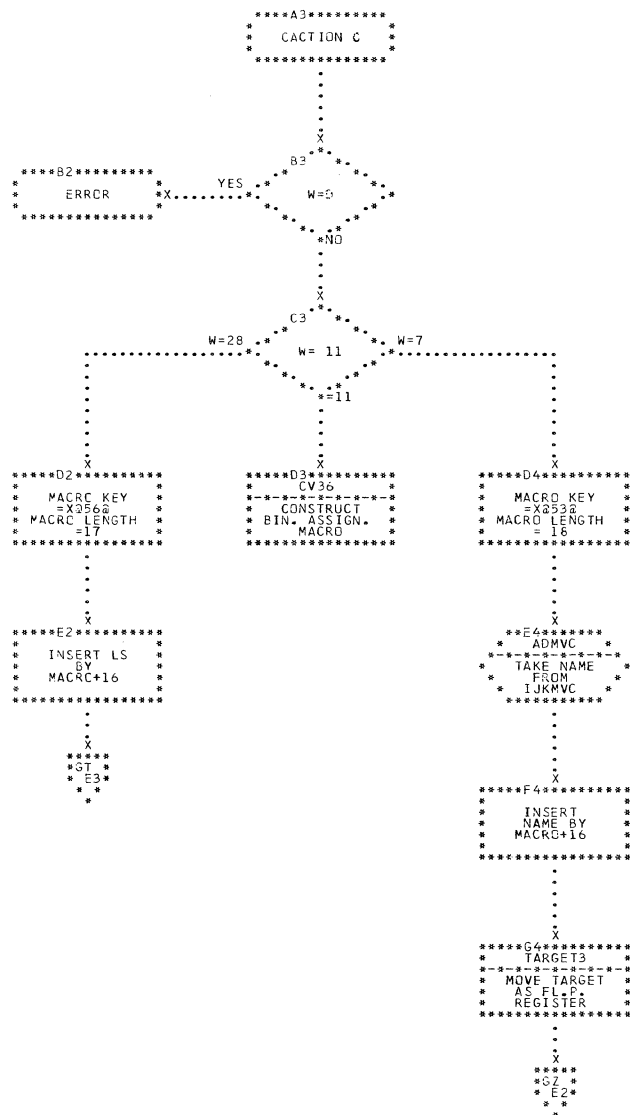


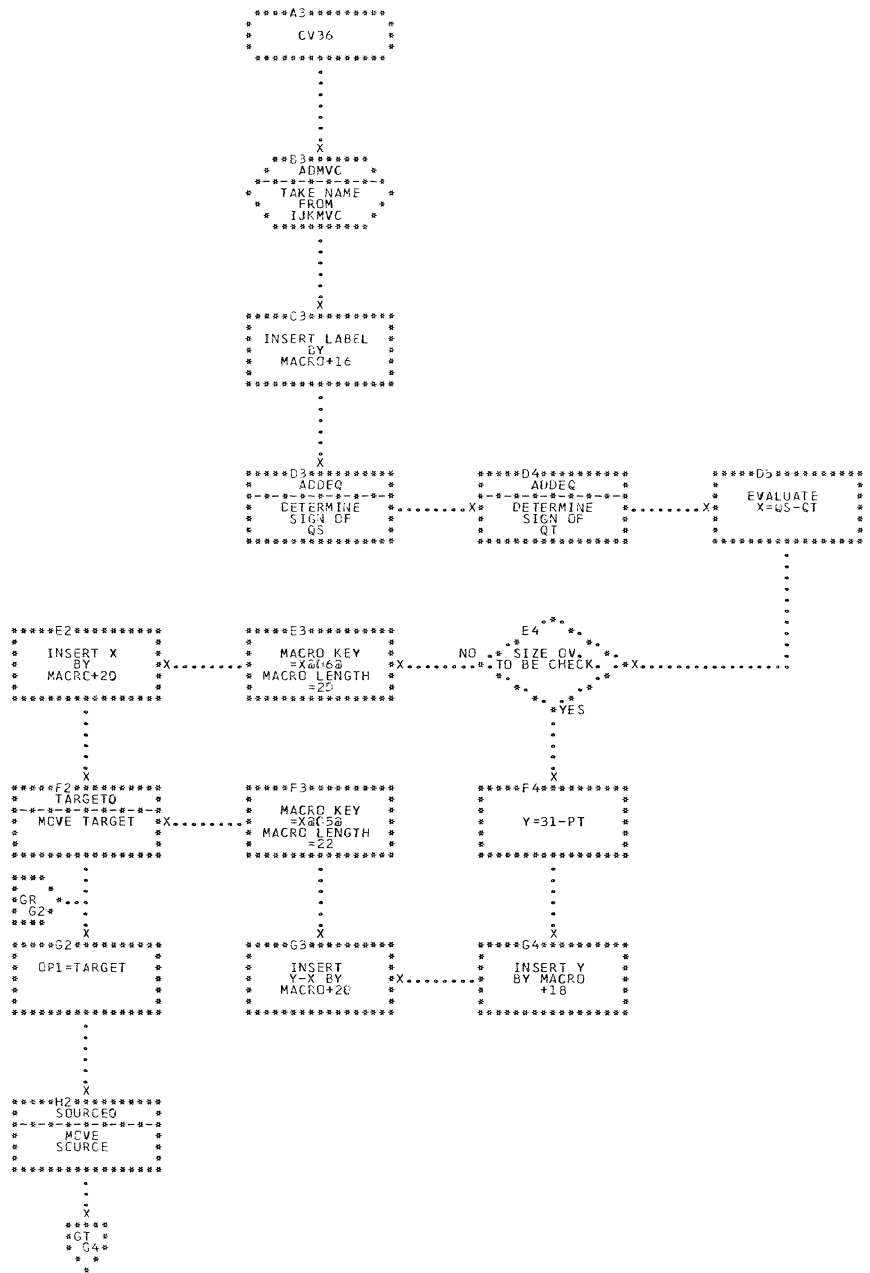


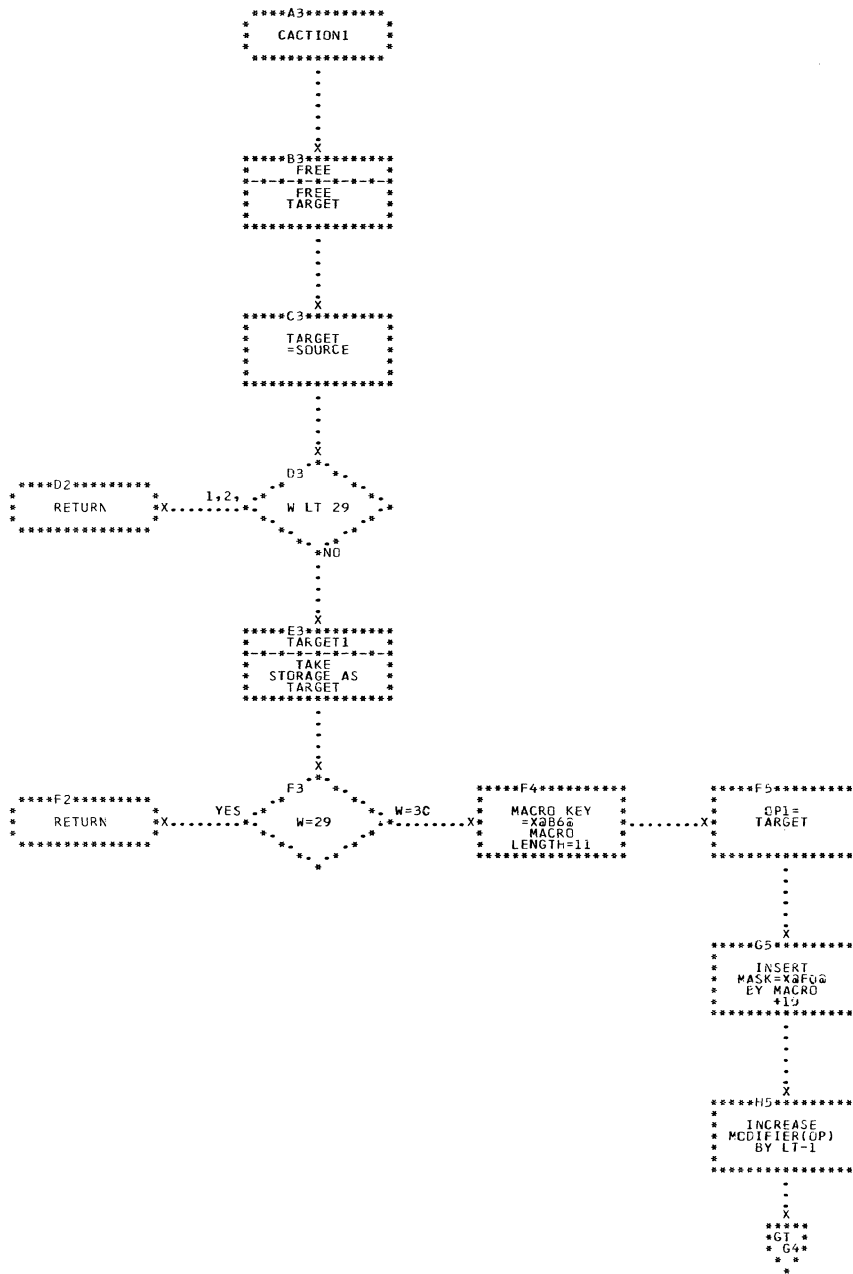
```

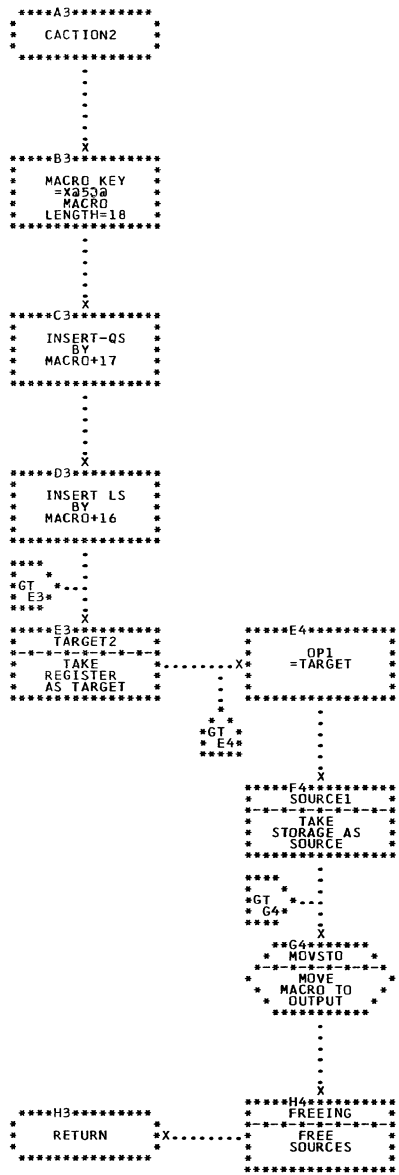
*****A3*****
* ACCOB3 *
*****
*
*
*
*
*****B3*****
* Z1=IABORG *
* Z2=Z1+11 *
*****
*
*
*
*
*****C3*****
* N IS *
* GIVEN BY *
* Z1+7 *
*****
*
*
*
*
*****D3*****
* RETURN *
*****

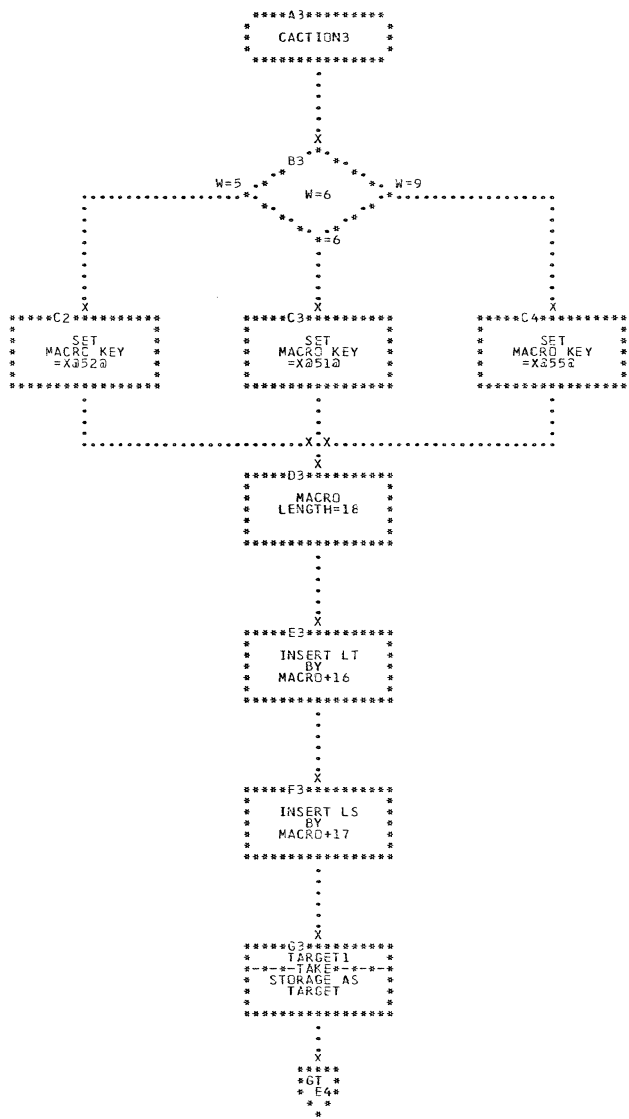
```











```

*****A2*****
* CACTION 4,5,6 *
*****

```

```

X

```

```

*****B2*****
* SET BIT *
* (AN=F1(W)) *
* IN LIBRARY- *
* BIT STRING *
*****

```

```

X

```

```

*****C2*****
* FUN=F2(W) *
*****

```

```

X

```

```

*****D2*****
* SET *
* BIT (FUN) *
* IN LIBRARY *
* BIT STRING *
*****

```

```

*****D3*****
* INSERT *
* FUN BY *
* MACRO+5 *
*****

```

```

*****D4*****
* INSERT O *
* BY *
* MACRO+4 *
*****

```

```

X

```

```

*****E4*****
* SOURCE1 *
* TAKE *
* STORAGE AS *
* SOURCE *
*****

```

```

X

```

```

*****F2*****
* MOVE NAME *
* FROM IJXWVC *
* TO MACRO+18 *
*****

```

```

*****F3*****
* OP1=TARGET *
* OP2=SOURCE *
*****

```

```

*****F4*****
* TARGET1 *
* TAKE *
* STORAGE AS *
* TARGET *
*****

```

```

X

```

```

*****G2*****
* G2 *
* REGISTERS *
* FREE *

```

```

X

```

```

X

```

```

*****H2*****
* GENERATE *
* STM 14,4,GWG *
*****

```

```

X

```

```

*****J2*****
* MOVSTC *
* MOVE CCODE *
* TO OUTPUT *
*****

```

```

*****J3*****
* SET *
* SWITCH *
* ADAC40=L *
*****

```

```

*****J4*****
* MOVE NAME *
* FROM IJXWVC *
* TO DED+3 *
*****

```

```

*****J5*****
* SET LENGTH *
* OF MACRO *
* =20 *
*****

```

```

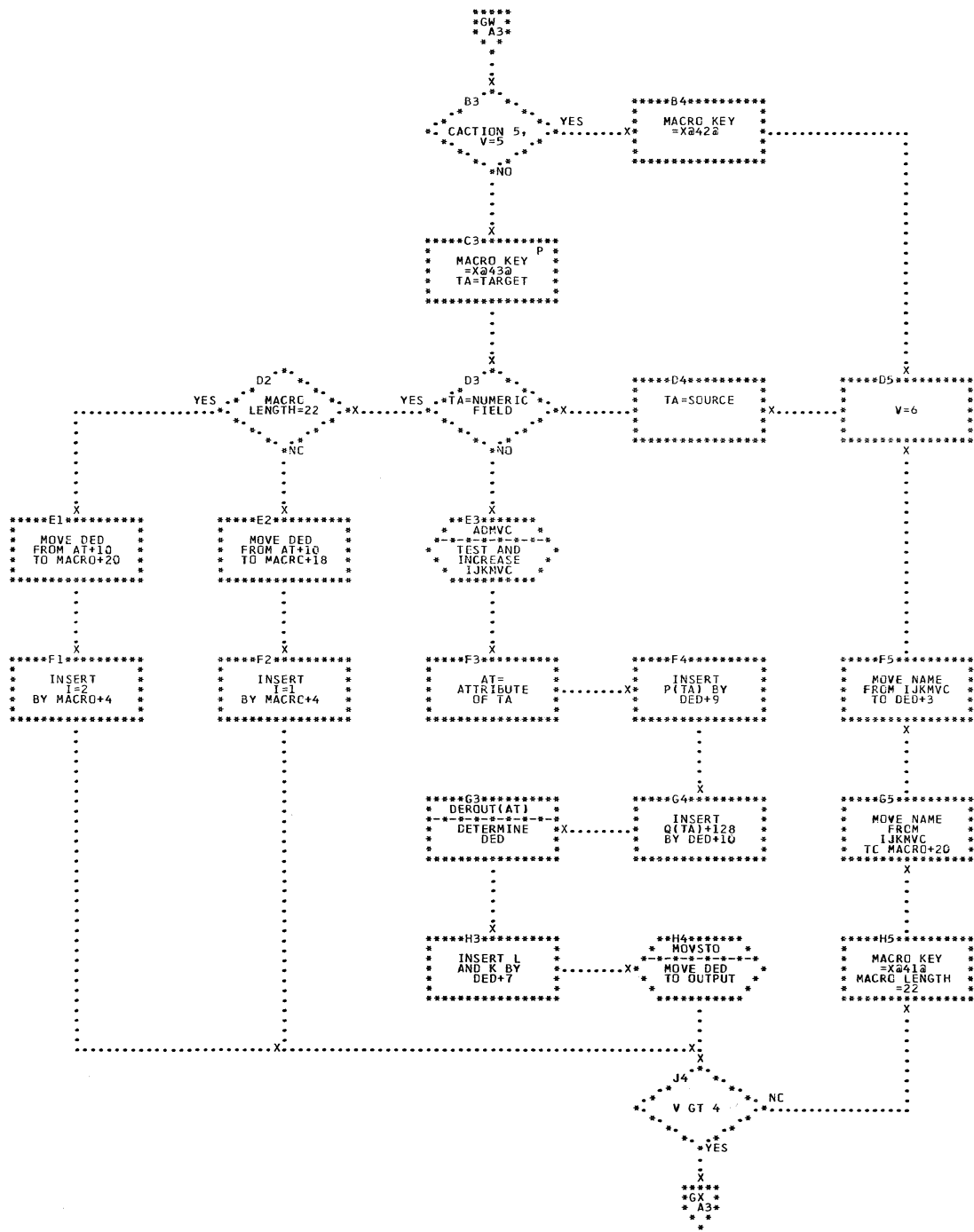
X

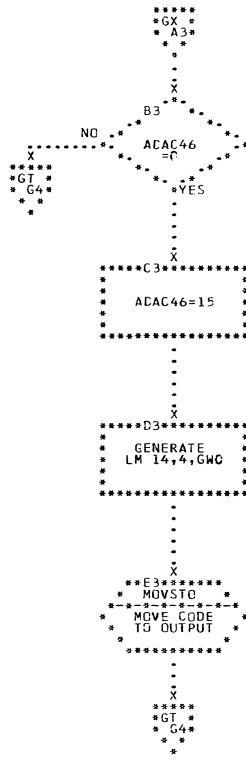
```

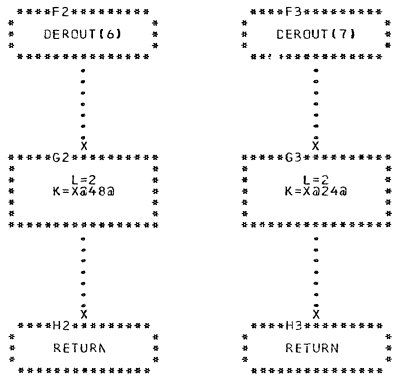
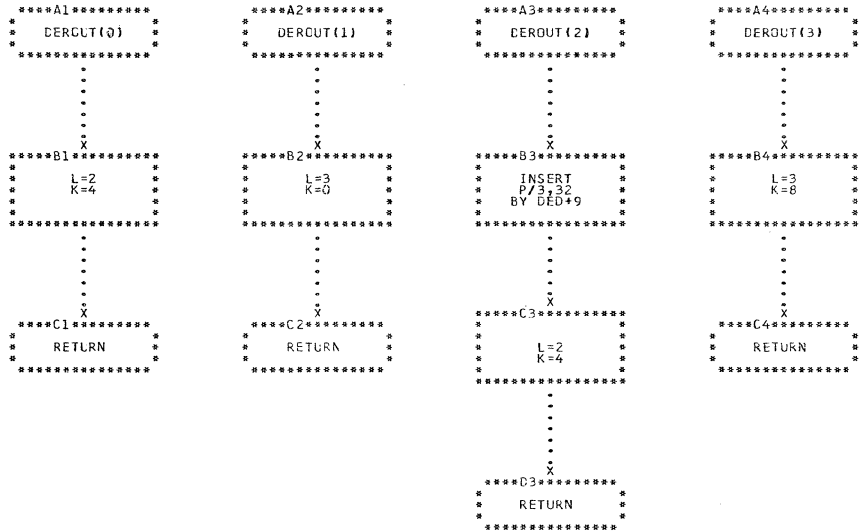
```

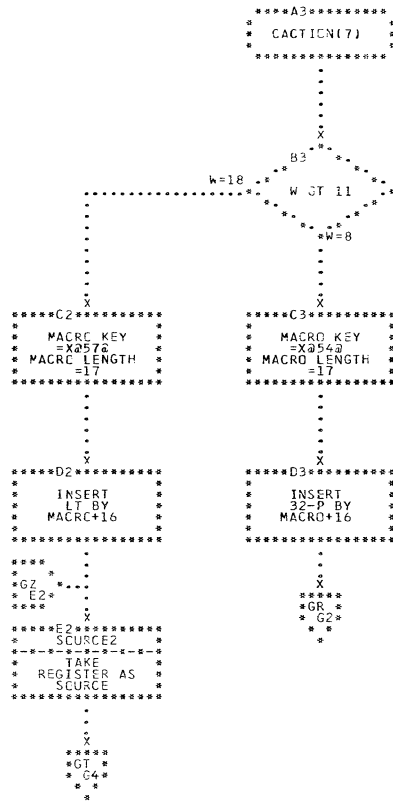
* GW *
* A3 *

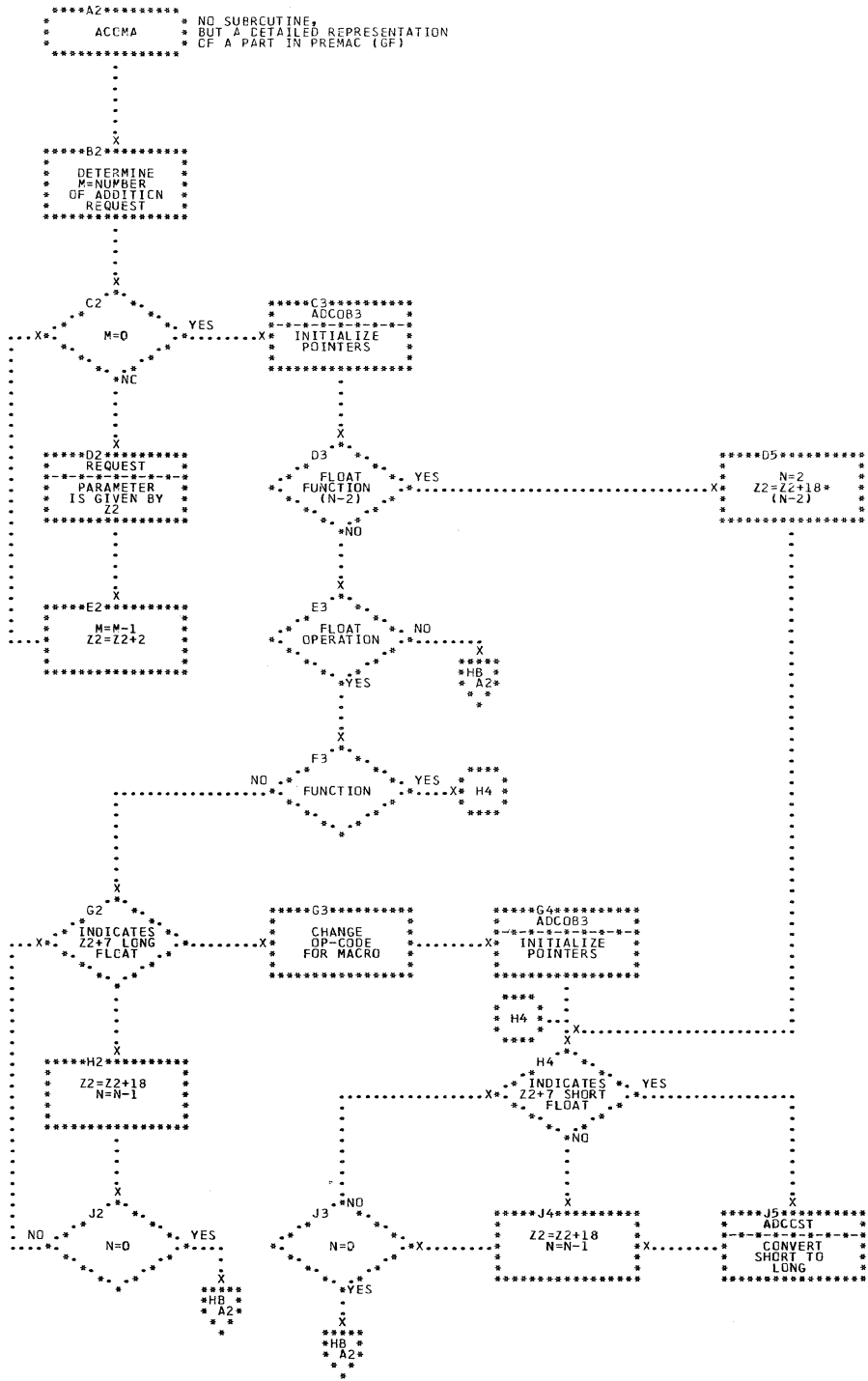
```











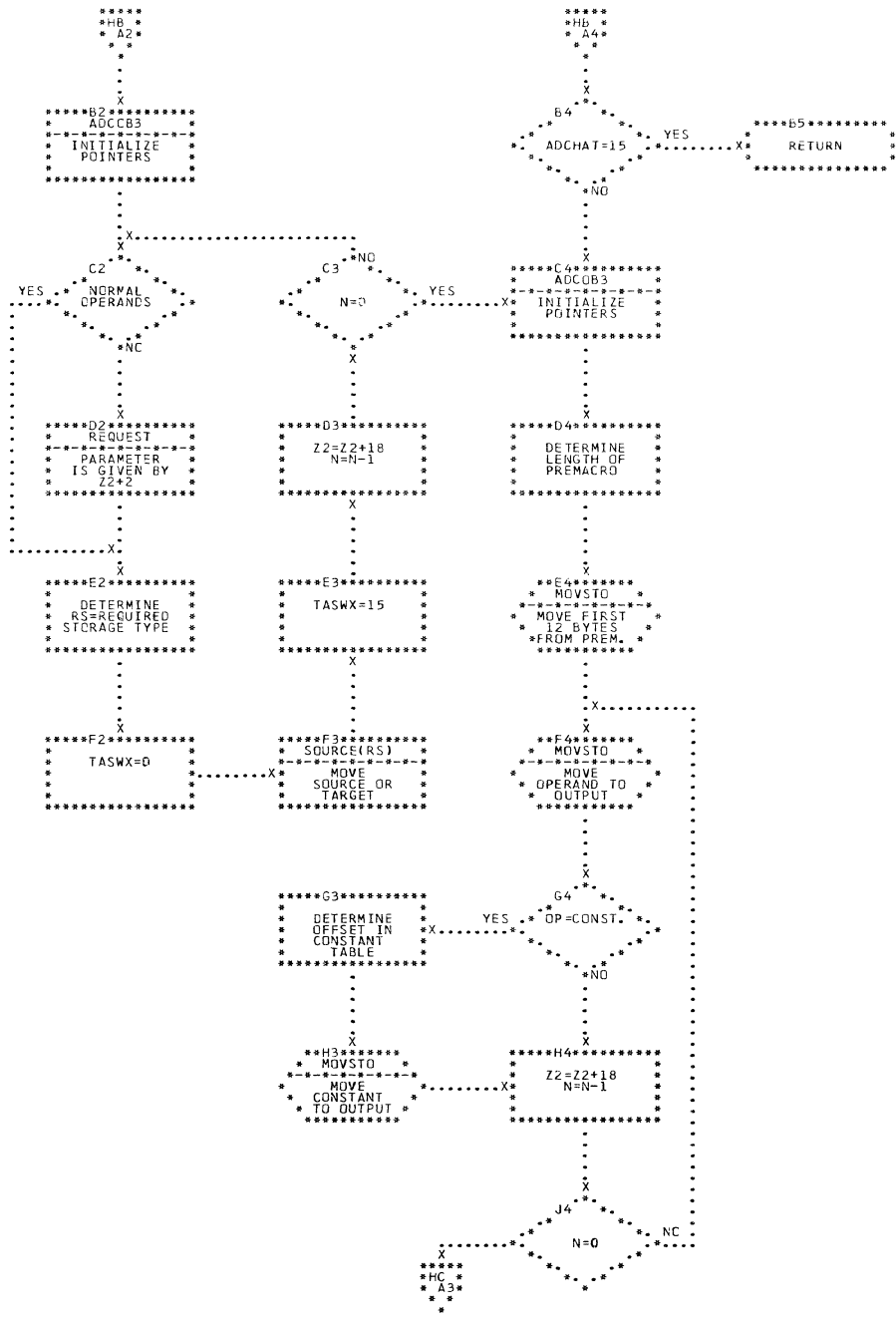
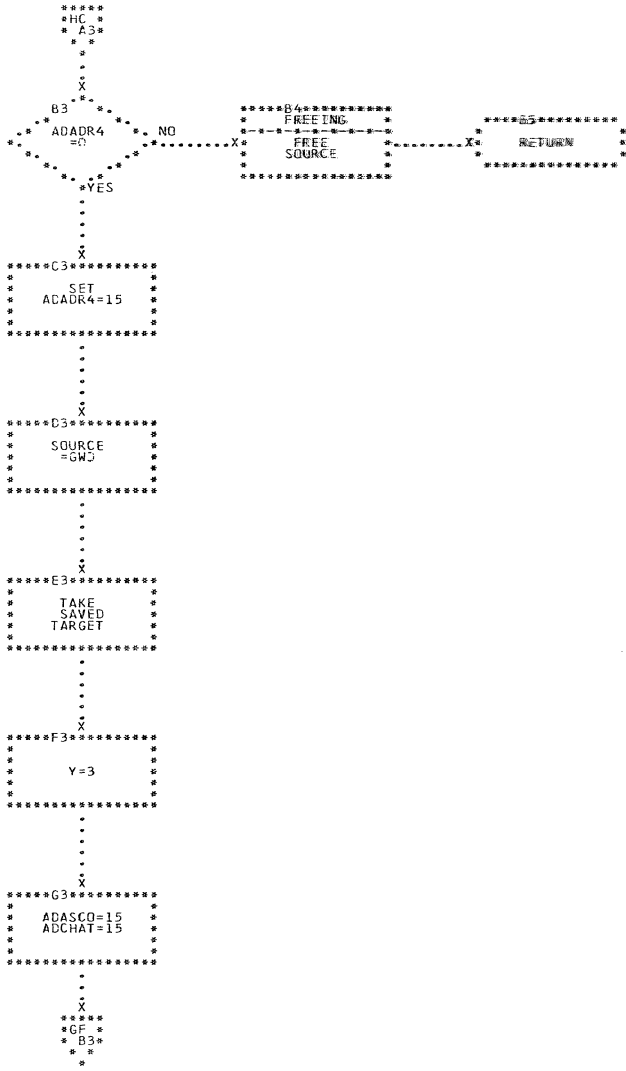
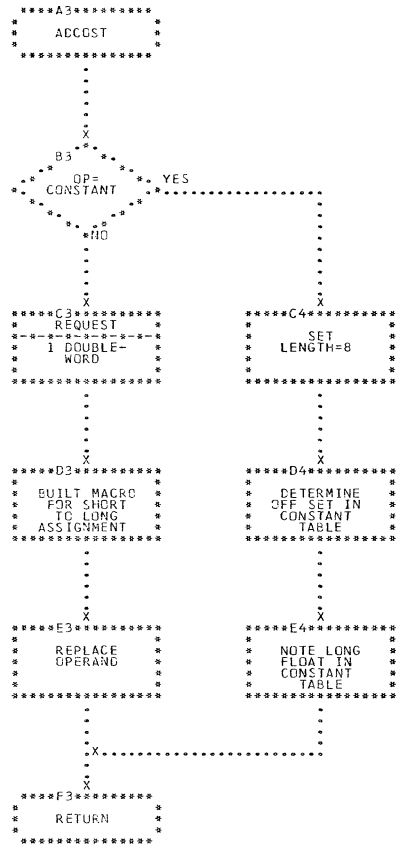


CHART HB. IJXD10

CONSTRUCTION OF PREMACROS





*****A1*****
IJXC11

*****B1*****
MOVE
(TABLE)
TO
(TABLEM)

C1
INTTABEN
+15 AND X 08
EQ ONE
YES NO

*****D1*****
SET REG1
TO
EQFAD +
TEXTIN

*****D2*****
SET REG1
TO
EQFADT +
TEXTIN

*****E1*****
SET
(REG1)
TO
NEOFAD+1

*****F1*****
SET
INTTABEN(1)
TO
ZERO

*****G1*****
POINTR..T
REGC
INTTABEN

*****H1*****
CHECK
T

*****J1*****
REAC..T
REG12
INTTABEN
+C

*****K1*****
CHECK
T

A3
INTTABEN
+15 AND X 08
EQ ONE
YES NO

*****B3*****
EXCP..T
BSR

C3
MOVE
TASAVA
TO
(TABLEM)

D3
INTTABEN
+15 AND X 08
EQ ONE
YES NO

*****E3*****
WAIT..T

F3
SET IJWC TO
INTTABEN+8
IJMTC TO
INTTABEN+16

G3
INTTABEN
+15 AND X 04
EQ ONE
NO YES

*****H3*****
IJKLB+2
OR
X 10

*****J3*****
IJKLB+5
OR
ADLTBIN

*****K3*****
SET IJKMT
TO
INTTABEN+14

A4
INTTABEN
+15 AND X 02
EQ ONE
NO YES

*****B4*****
IJKMT+2
OR
X C8

C4
READ..
TEXTIN
END
IJKMBL

*****D4*****
CHECK..
TEXTIN

*****E4*****
NOTE..
TEXTIN

*****F4*****
SET KTETA+8
TO
REG 1

G4
IJKMT+2
AND X 80
EQ ONE
NO YES

*****H4*****
SET KTETA+14
TO
REGC

*****A5*****
ENTRY.
NEOF

*****B5*****
POINTS..
TEXTIN

C5
IJKMT+2
AND X 80
EQ ONE
NO YES

*****D5*****
SET REG1
TO
EQFAD
+(TEXTIN)

*****E5*****
SET REG1
TO
EQFADT
+(TEXTIN)

*****F5*****
SET
(REG1)
TO
DUMPSAVE+1

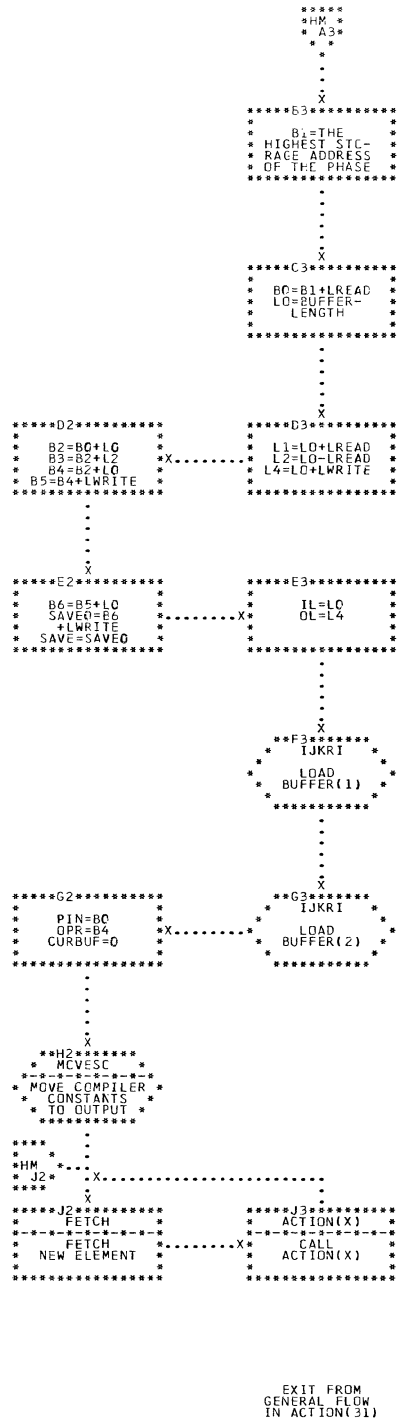
G5
IJKMT+2
AND X 80
EQ ONE
NO YES

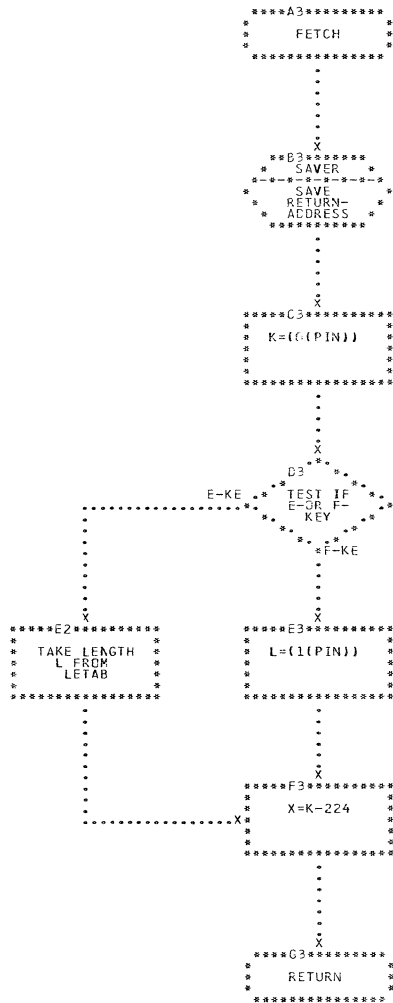
*****H5*****

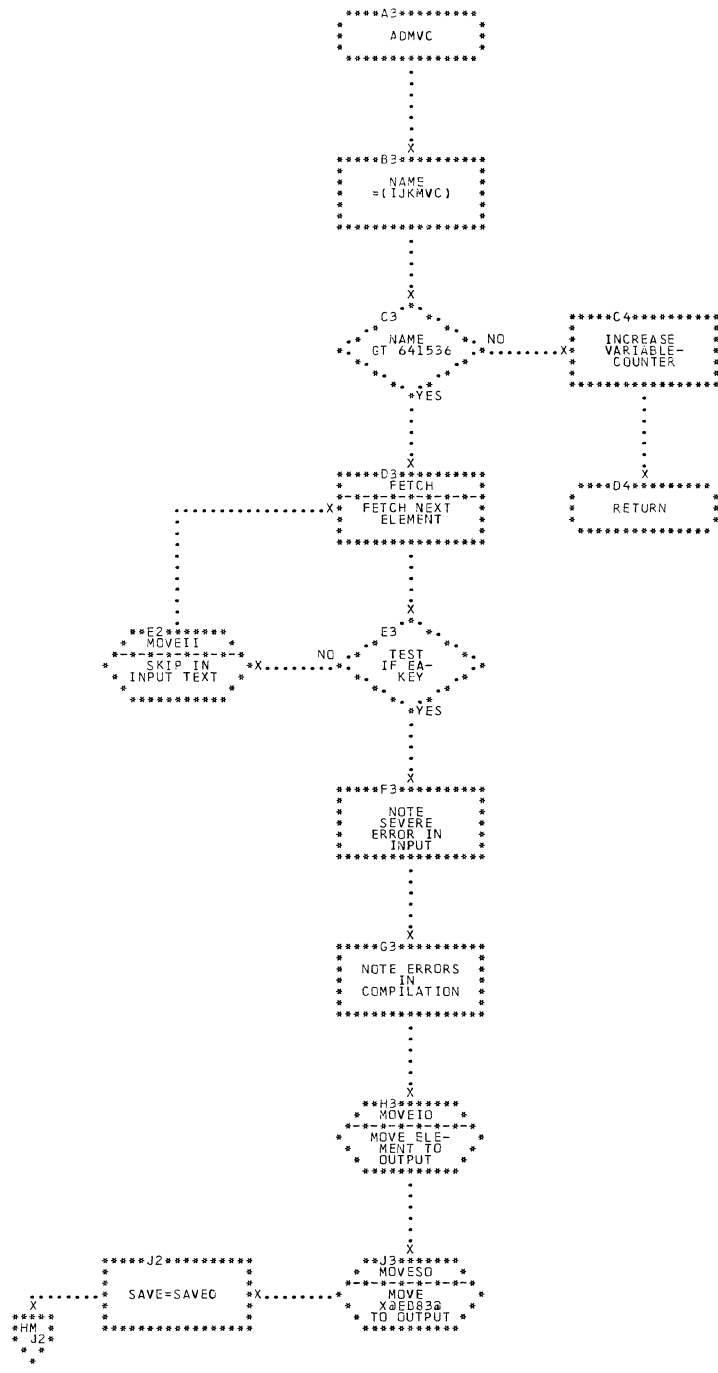
*****J5*****
SET REG1
TO EQFADT
+(TEXTOUT)

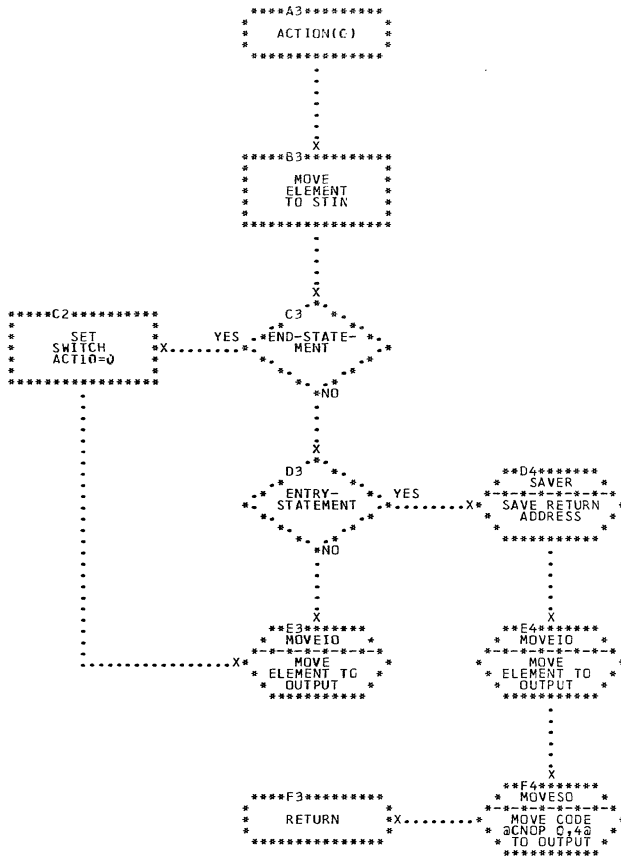
*****K5*****
SET
(REG1)
TO
DUMPSAVE+1

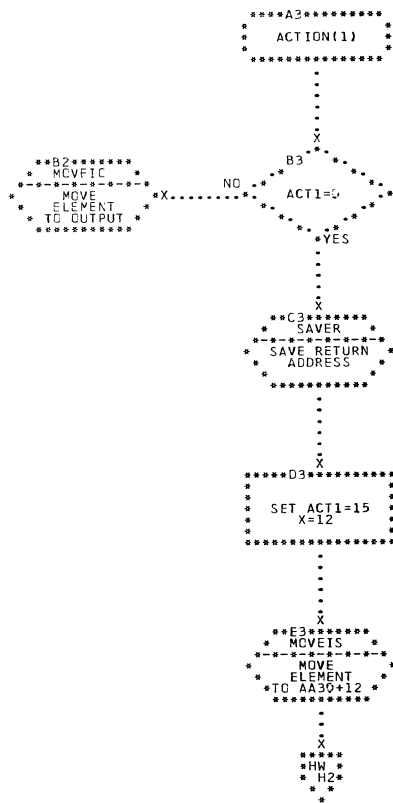
X
HM
A3
*

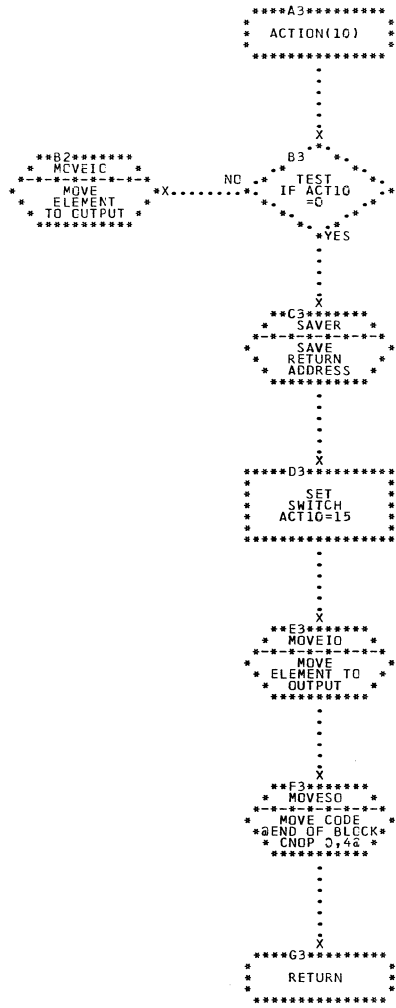


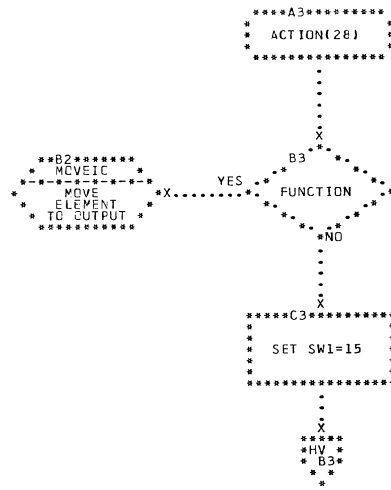


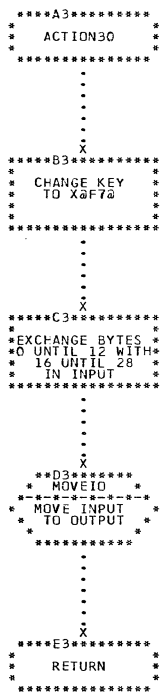


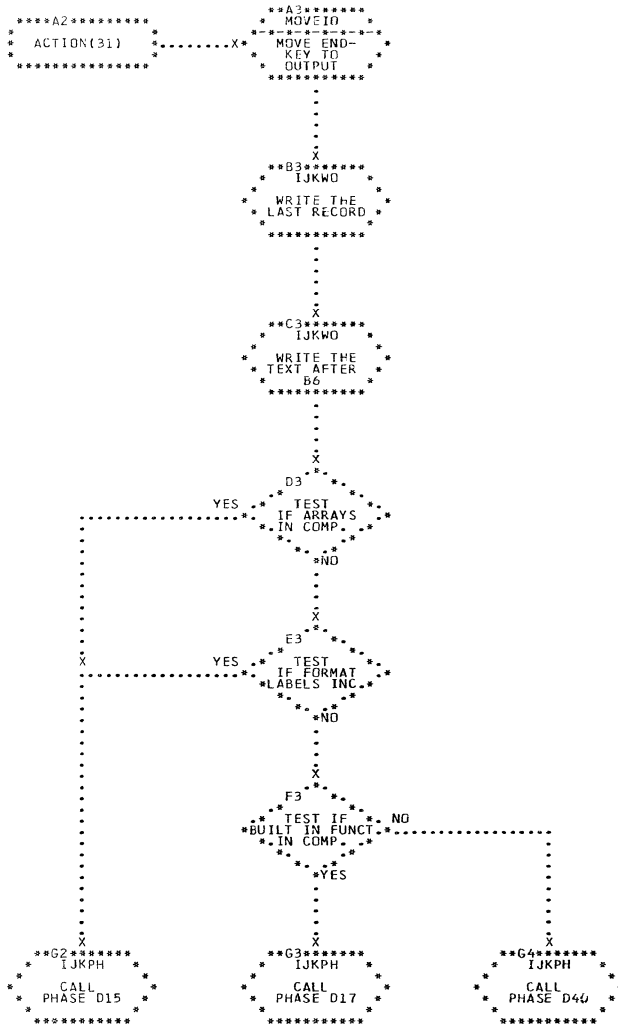


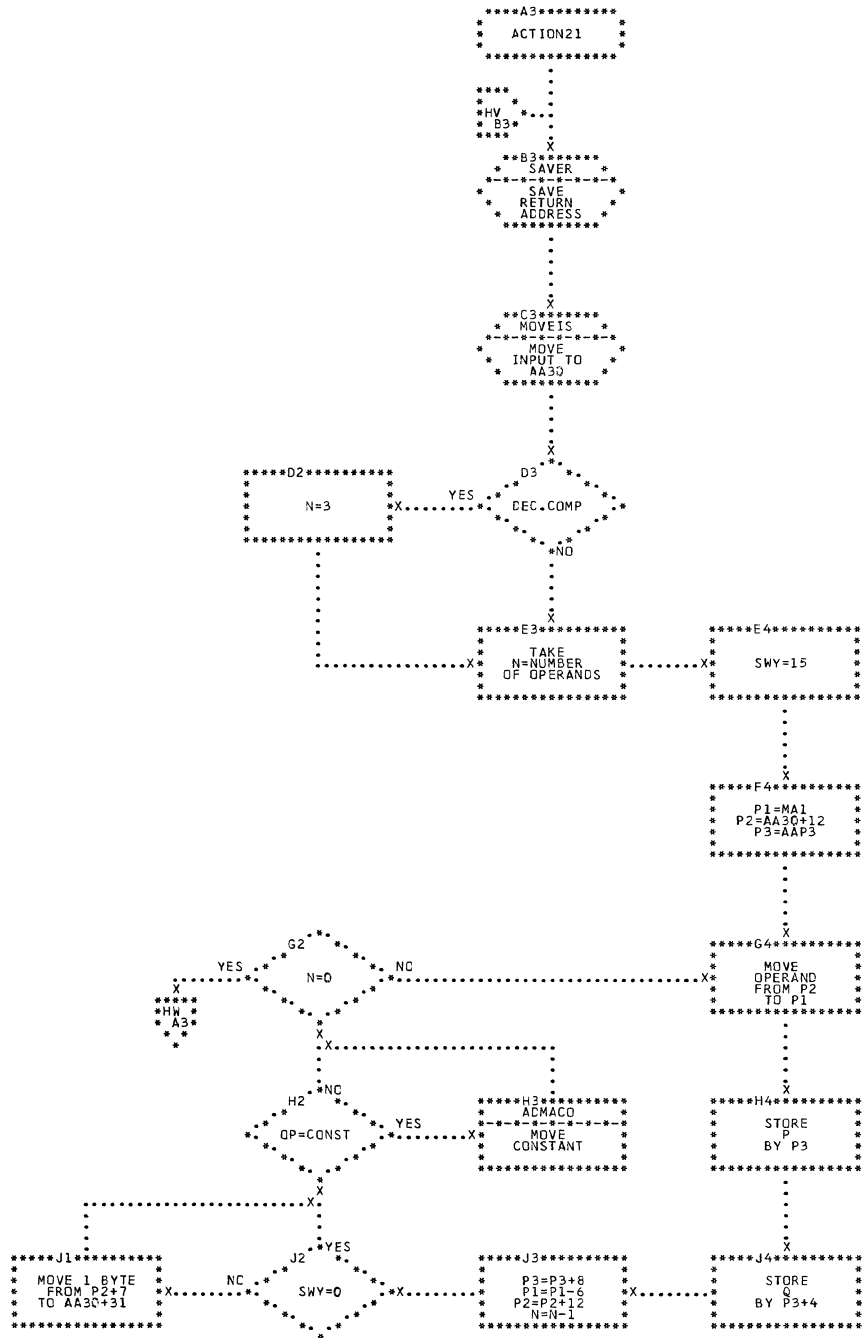


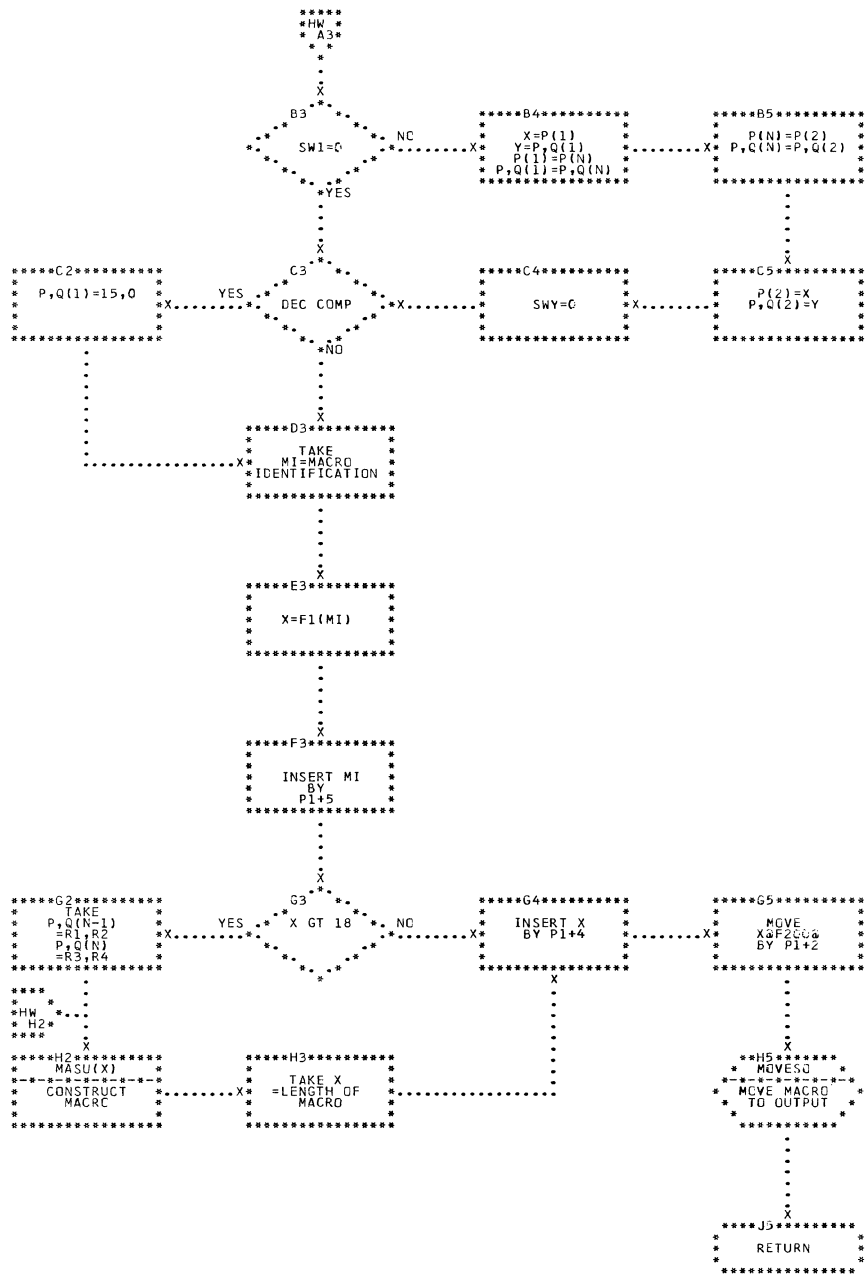








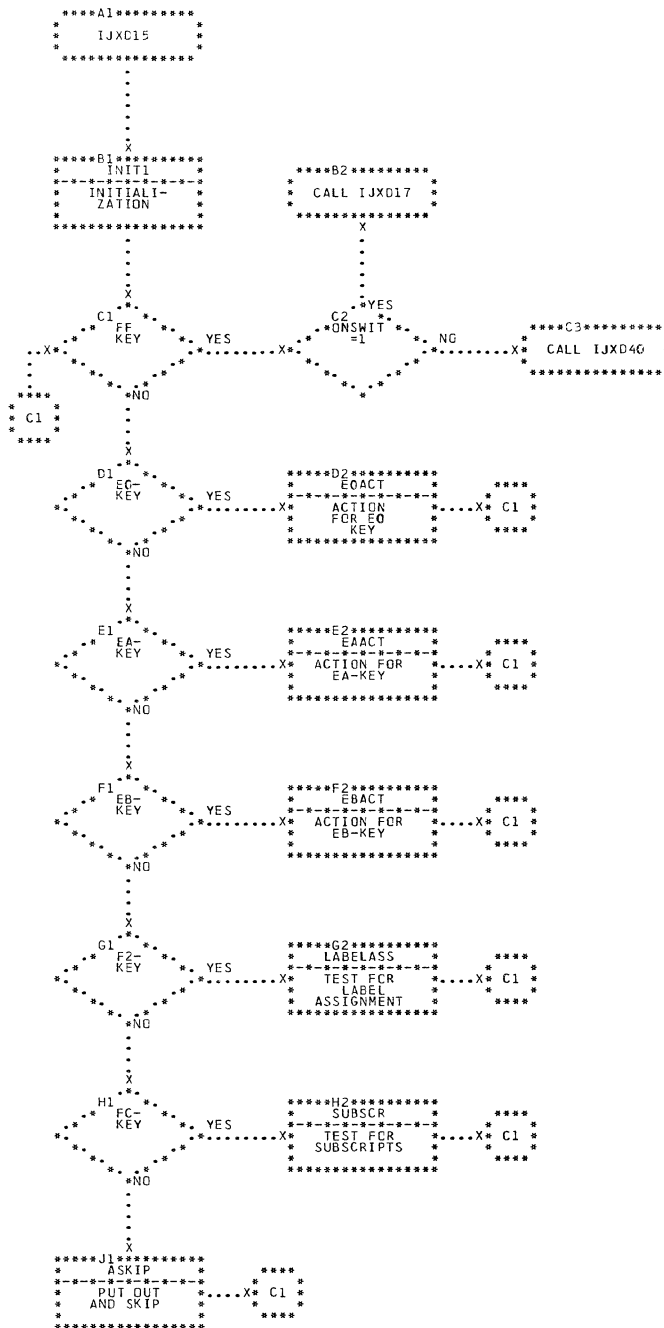


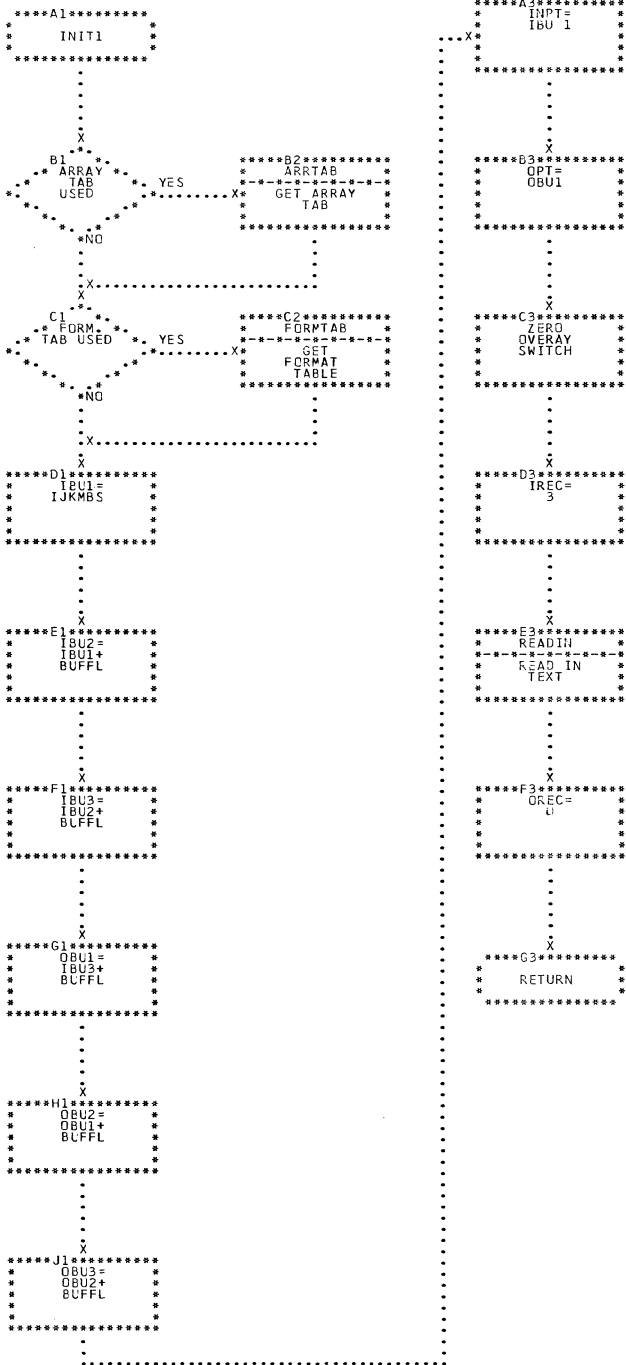


```

*****A3*****
*   ADMACO   *
*****
.
.
.
X
*****B3*****
*   CLEAR   *
*   DELETE  *
*   BIT     *
*****
.
.
.
X
*****C3*****
*   MOVESD  *
*   ----- *
*   MOVE    *
*   CONSTANT TO *
*   OUTPUT  *
*****
.
.
.
X
*****D3*****
*   P2=P2+22 *
*****
.
.
.
X
*****E3*****
*   SWY=0    *
*****
.
.
.
X
*****F3*****
*   RETURN   *
*****

```





*****A1*****
EQ ACT

.....
X

*****B1*****
CLEAR
ERROR
SWITCHES
ERRSW,CT

.....
X

*****C1*****
CLEAR
ERROR
STACK
ERRDSTK(16)

.....
X

*****D1*****
LA RC,6
LR R1,INPT

.....
X

*****E1*****
MOVOUT
PUT OUT
STATEMENT
BEGIN

.....
X

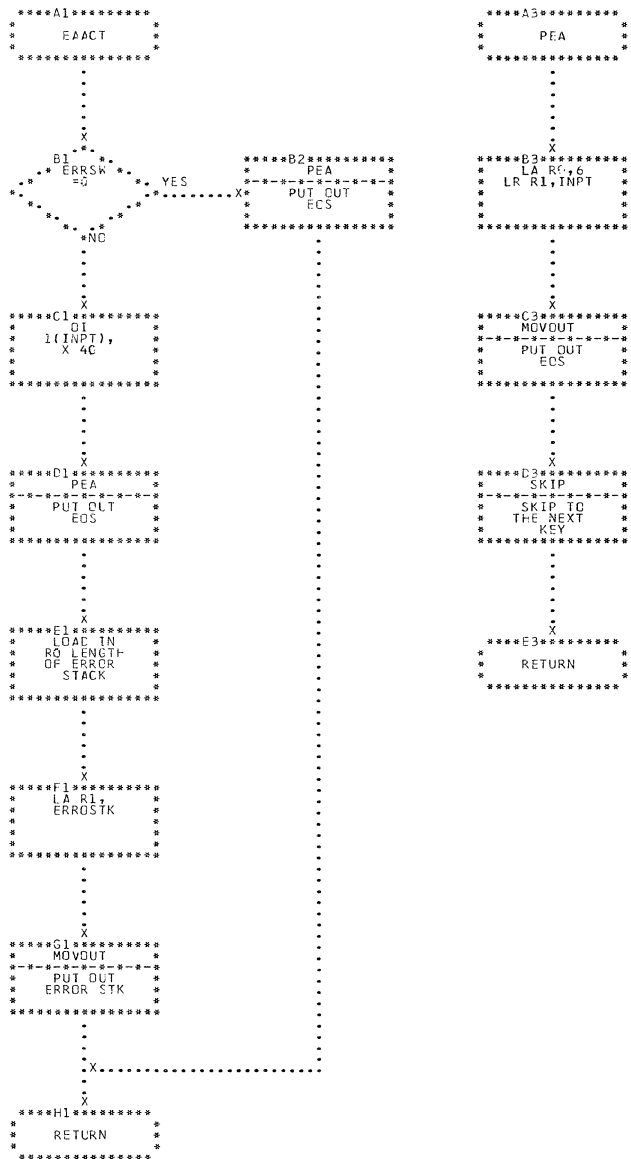
*****F1*****
LA RC,6

.....
X

*****G1*****
SKIP
INCREASE
INPUT POINT

.....
X

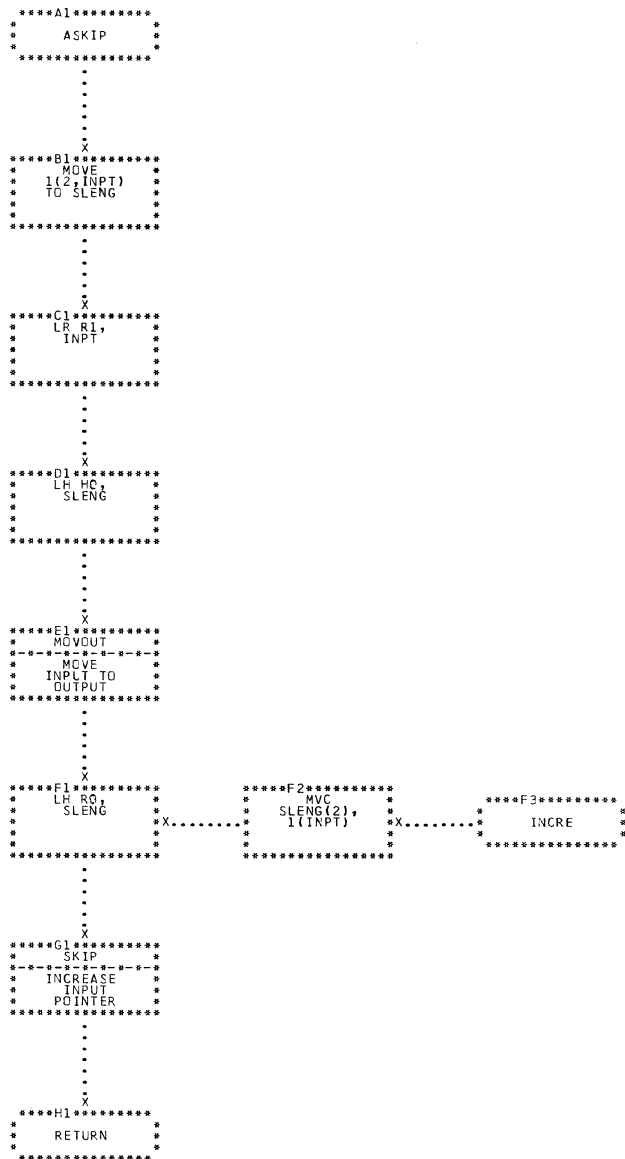
*****H1*****
RETURN

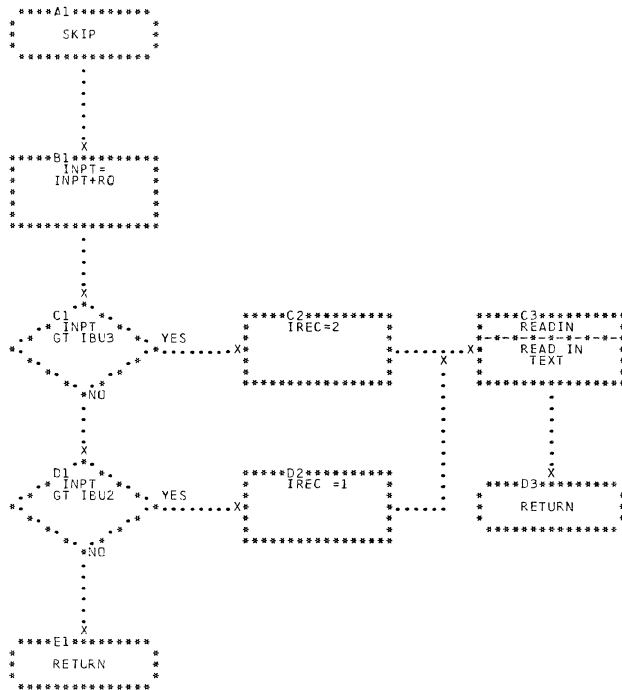


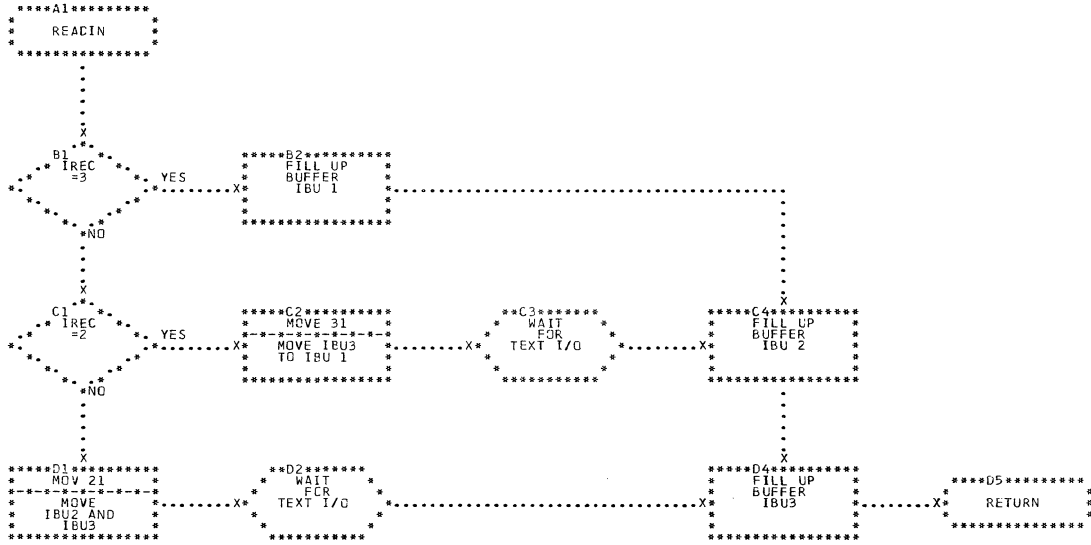
```

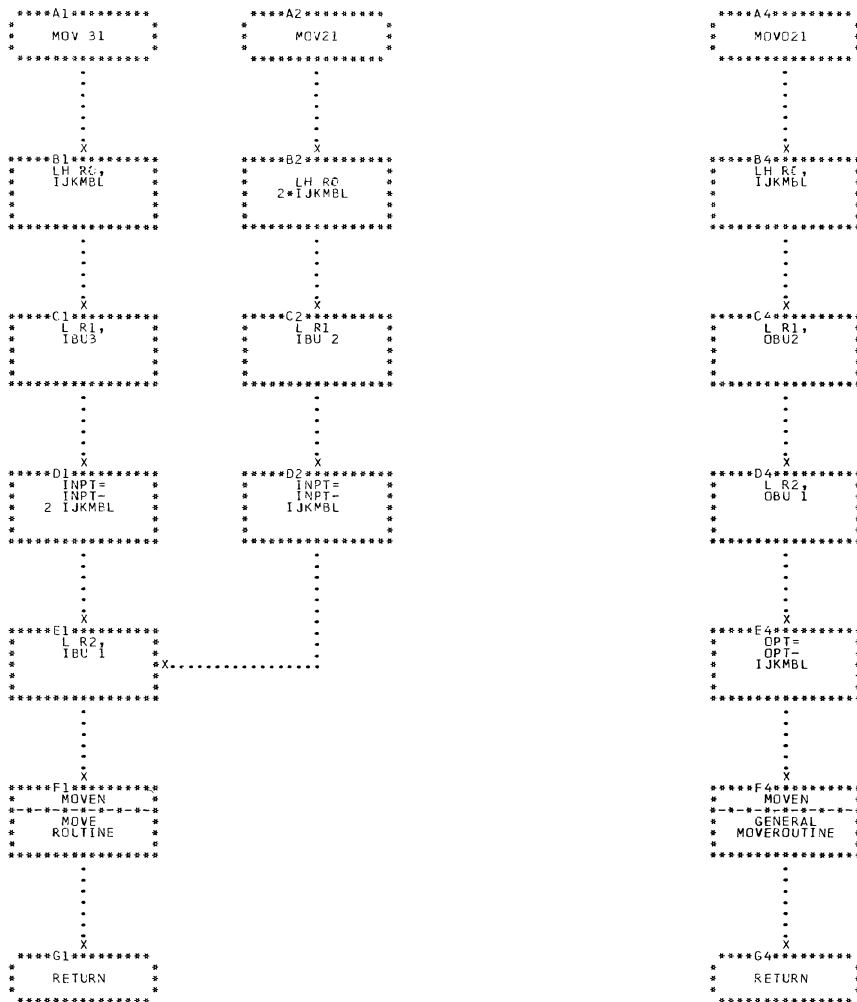
.....A1.....
*   EBACT   *
*   ..... *
*   .      *
*   .      *
*   X      *
*   ..... *
*   B1..... *
*   LA R0,Z *
*   ..... *
*   .      *
*   .      *
*   X      *
*   ..... *
*   C1..... *
*   LR R1,  *
*   INPT    *
*   ..... *
*   .      *
*   .      *
*   X      *
*   ..... *
*   D1..... *
*   MOVQUI  *
*   PUT OUT *
*   ERROR   *
*   BYTES   *
*   ..... *
*   .      *
*   .      *
*   X      *
*   ..... *
*   E1..... *
*   SKIP    *
*   INCREASE *
*   INPUT   *
*   POINTER *
*   ..... *
*   .      *
*   .      *
*   X      *
*   ..... *
*   F1..... *
*   RETURN  *
*   ..... *

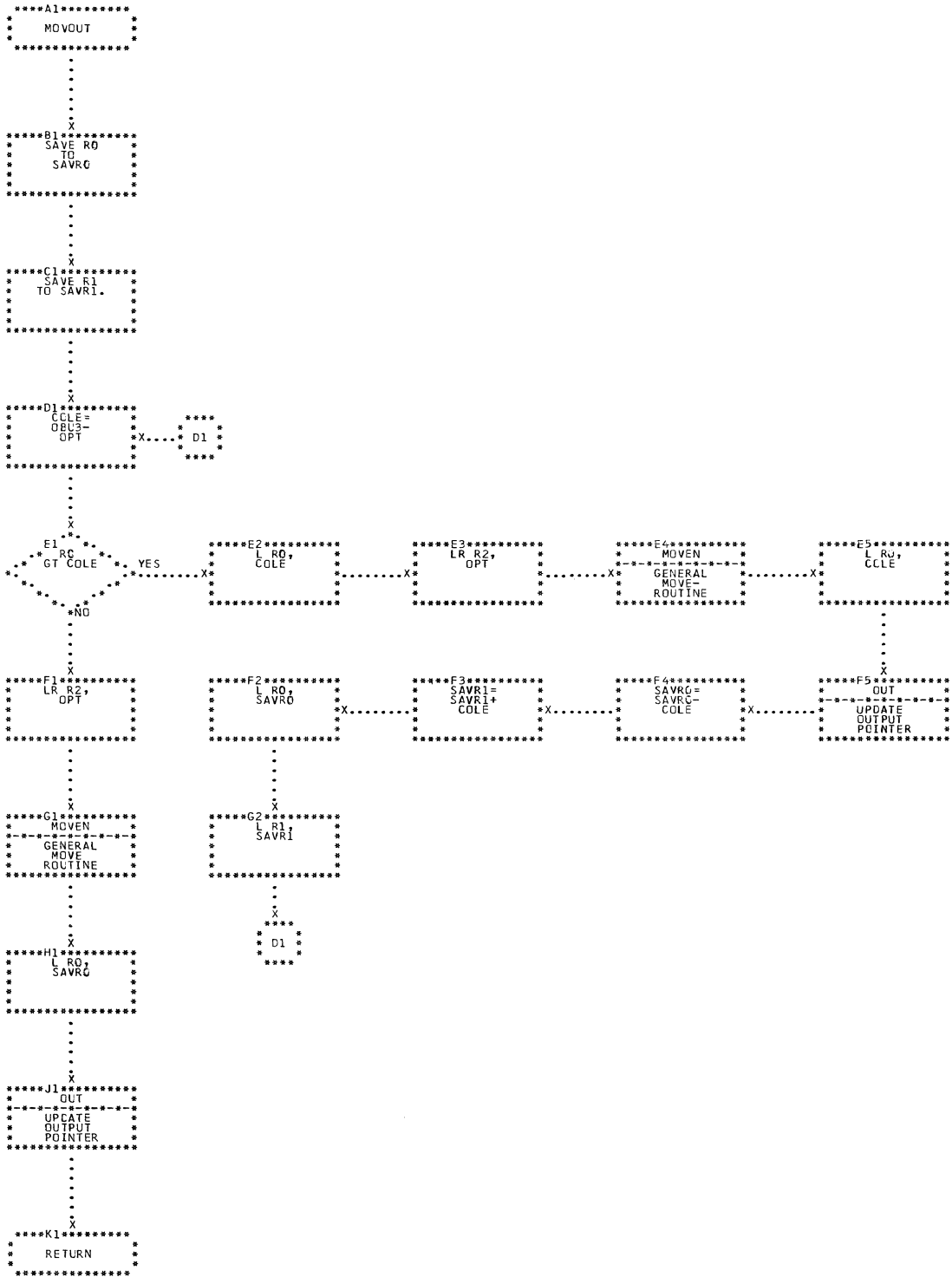
```

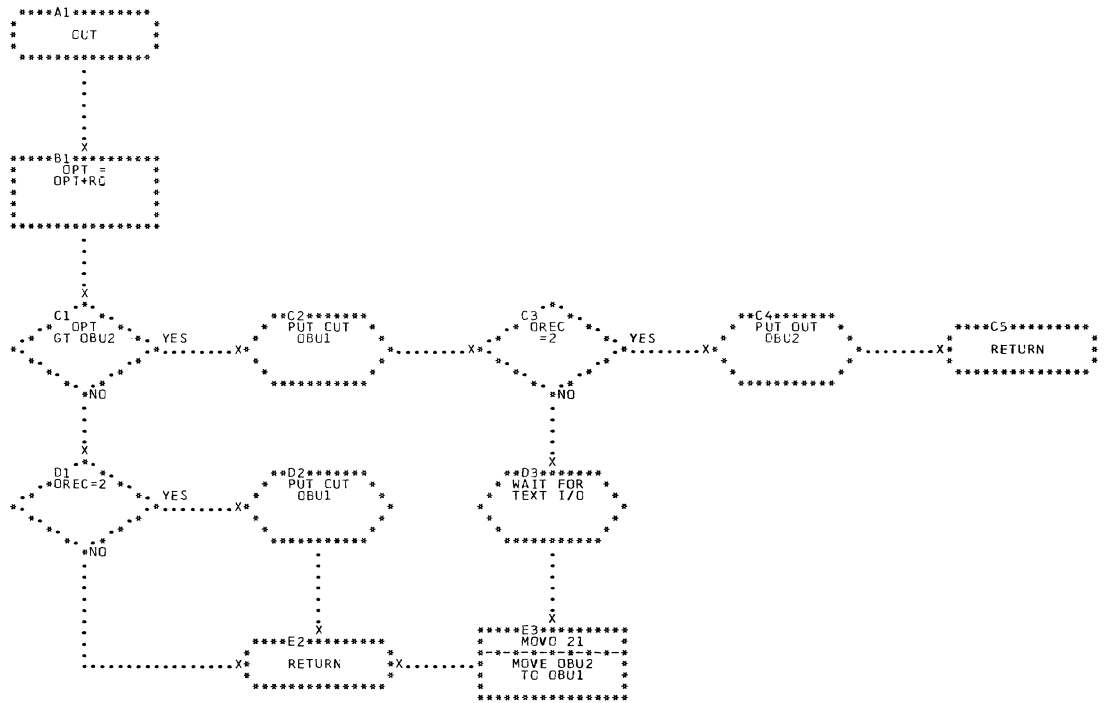












*****A1*****
EOPACT

.....
X

*****B1*****
MVI
O(GPT),
EOP KEY

.....
X

*****C1*****
LA RC,1
OREC=2

.....
X

*****D1*****
DLT
PUT PUT
LAST
RECORD

.....
X

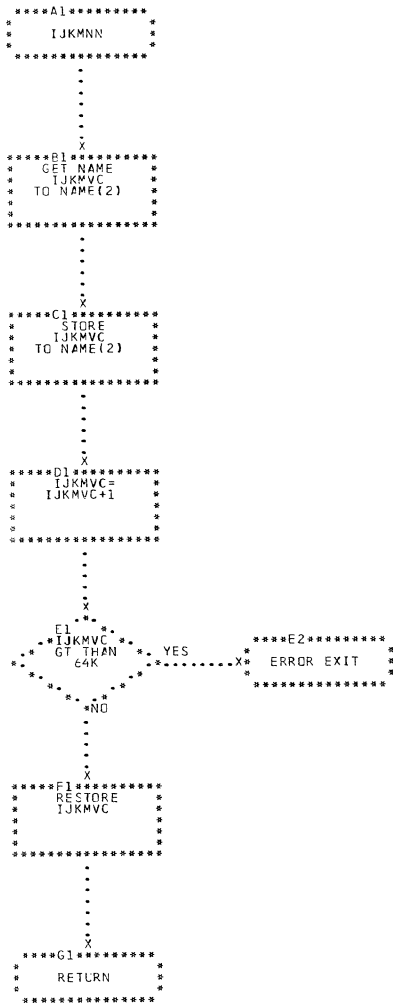
*****E1*****
OVSWIT
=0

NO

*****E2*****
IJX D20

.....
X

*****F1*****
IJXD40



```
*****A1*****
*   ARRTAB   *
*****
```

.....
X

```
*****B1*****
*   ARPU=   *
*   IJKMES+ *
*   5 BUFFL *
*****
```

.....
X

```
*****C1*****
*   TABTAB *
*   READ IN *
*   TABTAB *
*****
```

.....
X

```
*****D1*****
*   NI BIT2 *
*   IN TABTAB *
*   ENTRY   *
*****
```

.....
X

```
*****E1*****
*   LA RD,1 *
*****
```

.....
X

```
*****F1*****
*   L RL,   *
*   AREU   *
*****
```

.....
X

```
*****G1*****
*   LA R2  *
*   ZTABG3 *
*****
```

.....
X

```
*****H1*****
*   READ IN *
*   ARRAY TAB *
*****
```

.....
X

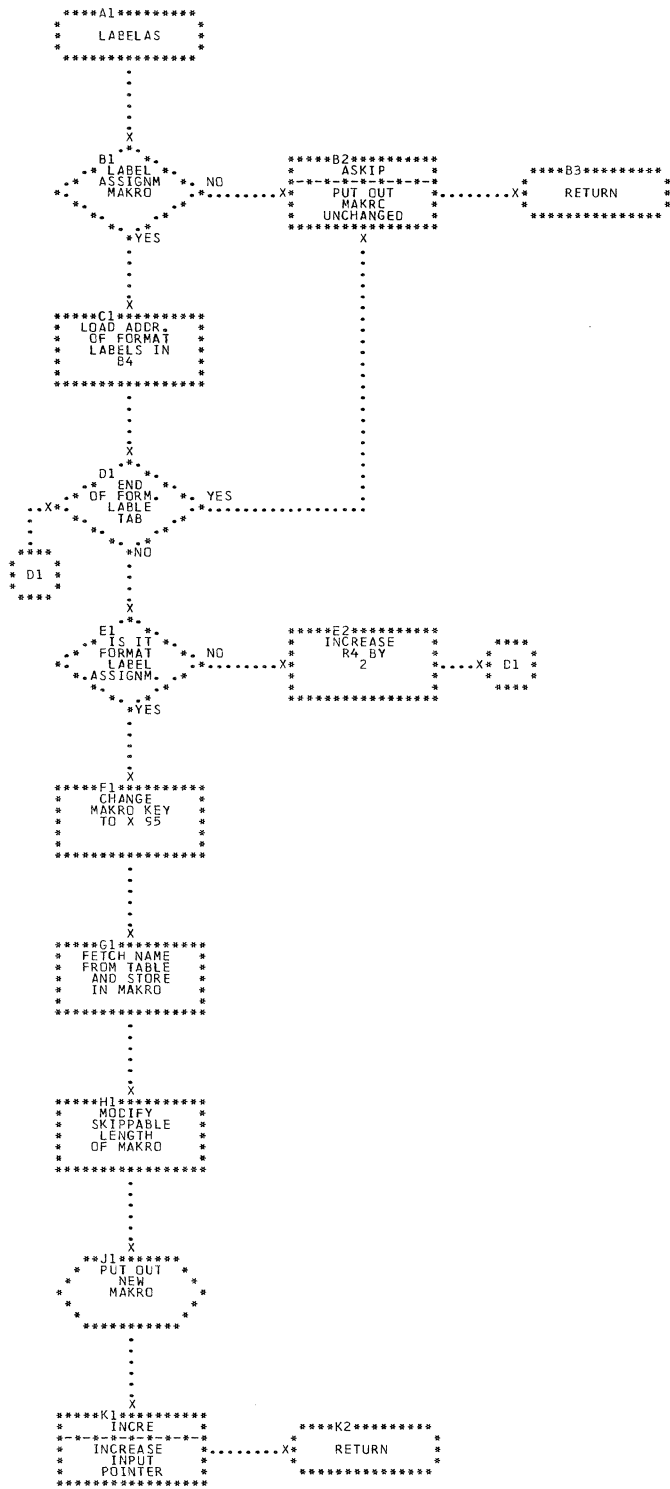
```
*****J1*****
*   RETURN *
*****
```

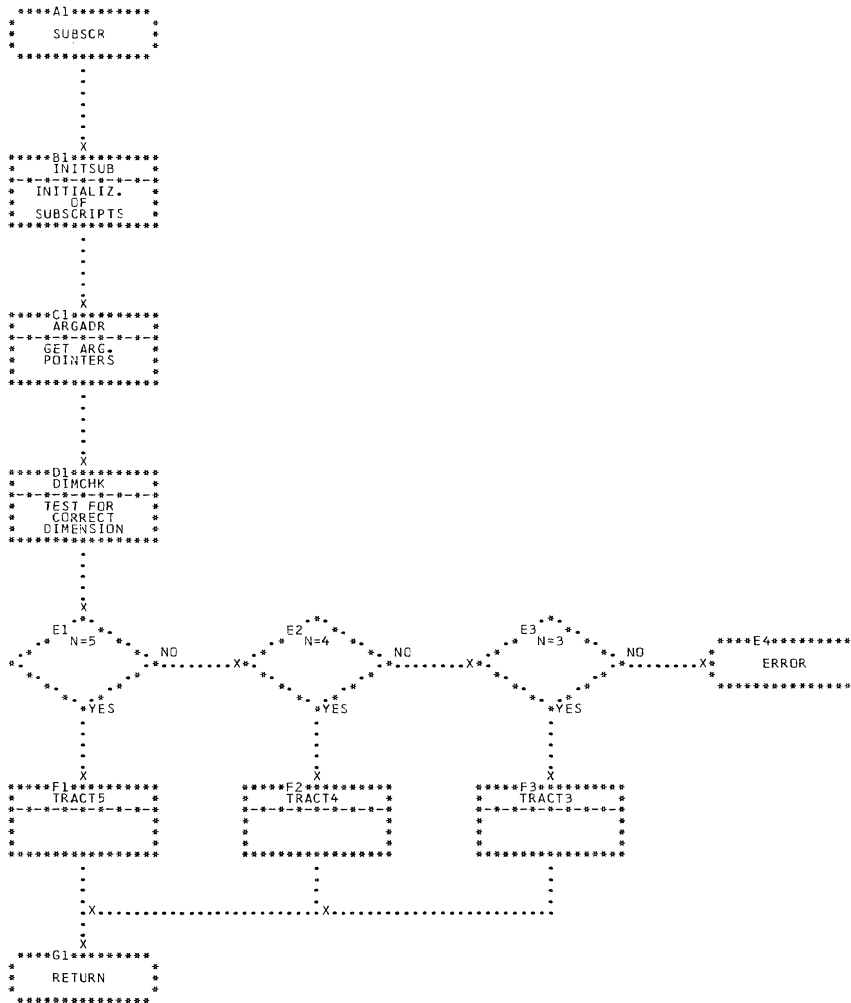


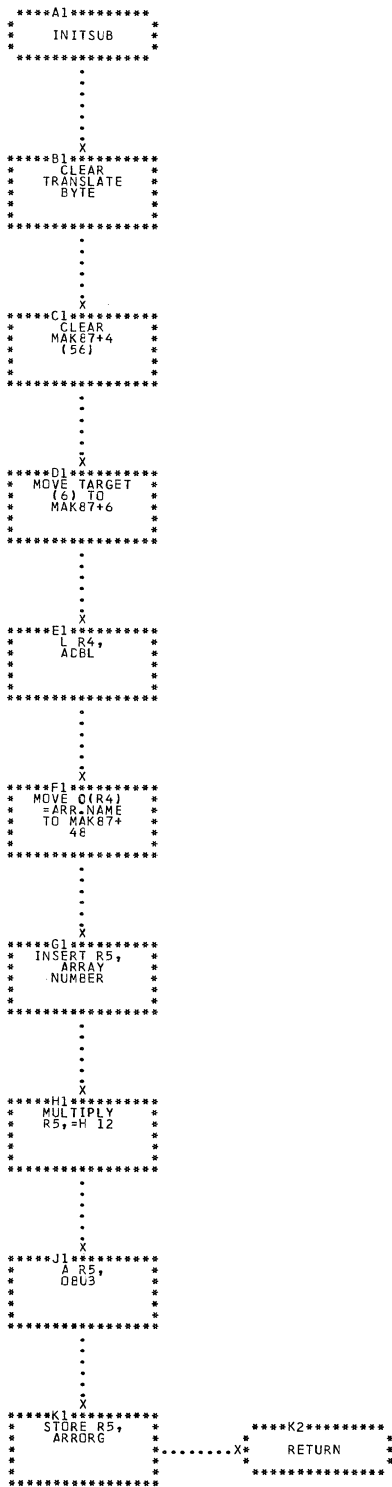
```

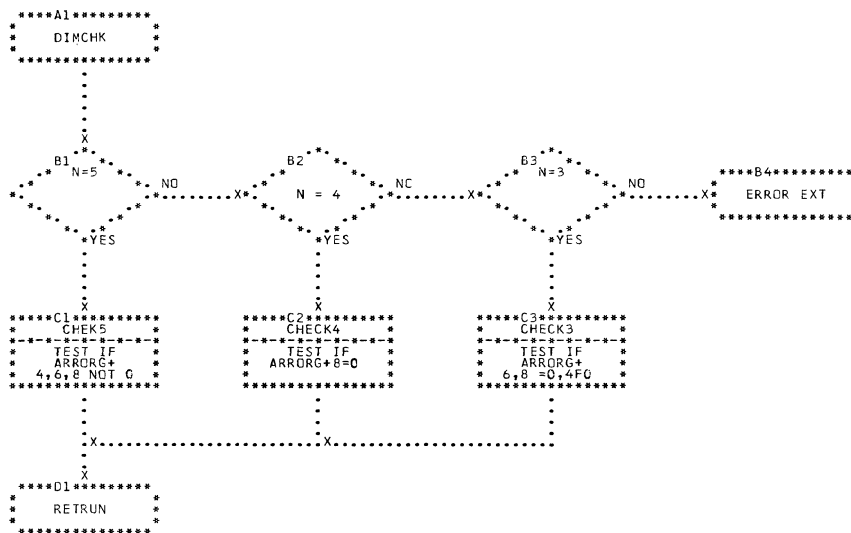
*****A1*****
*  FORMTAB  *
*****
*  .  *
*  .  *
*  X  *
*****B1*****
*  FORMBU=  *
*  IJKMS   *
*  -384    *
*****
*  .  *
*  .  *
*  X  *
*****C1*****
*  TABTAB  *
*  REAC IN *
*  TABTAB  *
*****
*  .  *
*  .  *
*  X  *
*****D1*****
*  IN GIT2  *
*  IN TABTAB *
*  ENTRY    *
*****
*  .  *
*  .  *
*  X  *
*****E1*****
*  LA RC,1  *
*****
*  .  *
*  .  *
*  X  *
*****F1*****
*  L R1,    *
*  FORMBU  *
*****
*  .  *
*  .  *
*  X  *
*****G1*****
*  LA R2,   *
*  ZTABIB  *
*****
*  .  *
*  .  *
*  X  *
*****H1*****
*  REAC IN *
*  FORMAT  *
*  TABLE  *
*****
*  .  *
*  .  *
*  X  *
*****J1*****
*  RETURN  *
*****

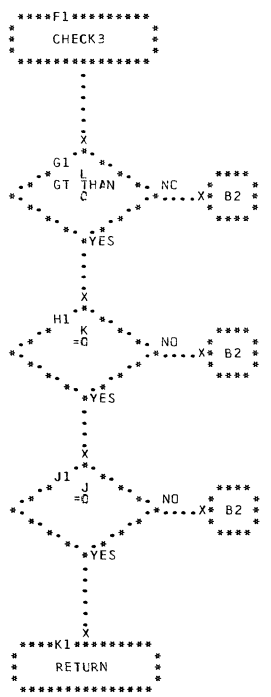
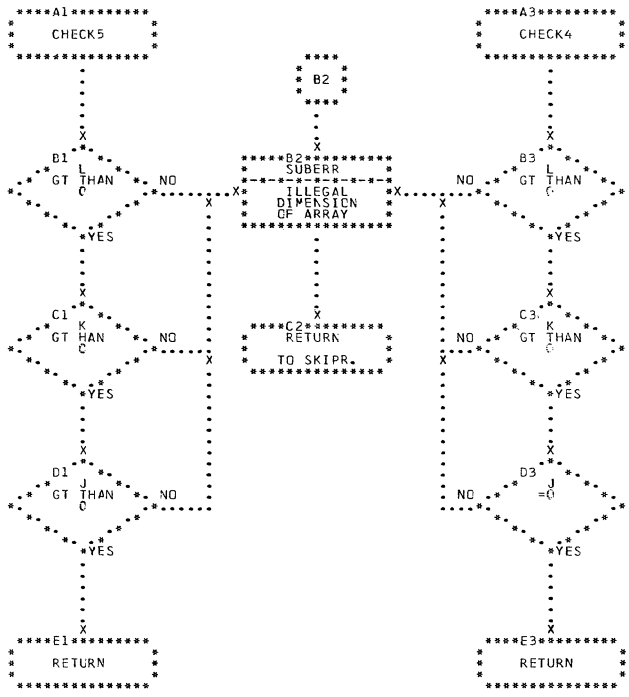
```

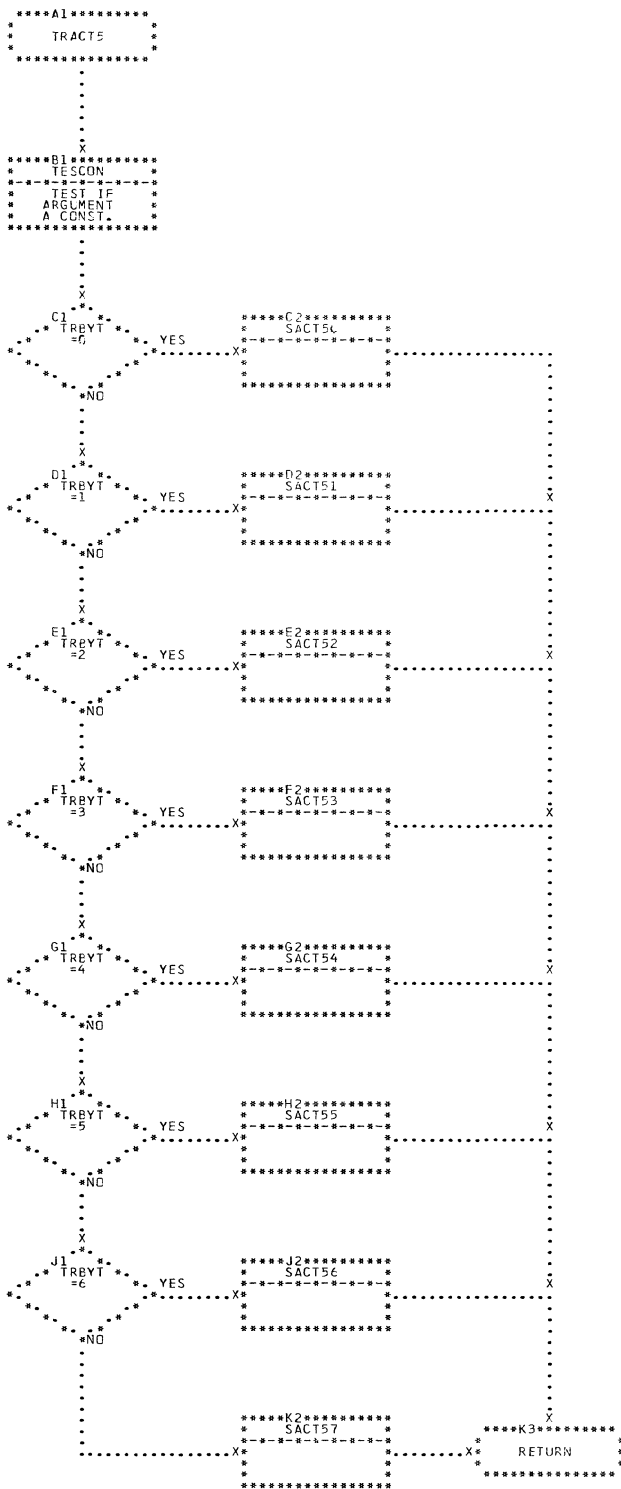


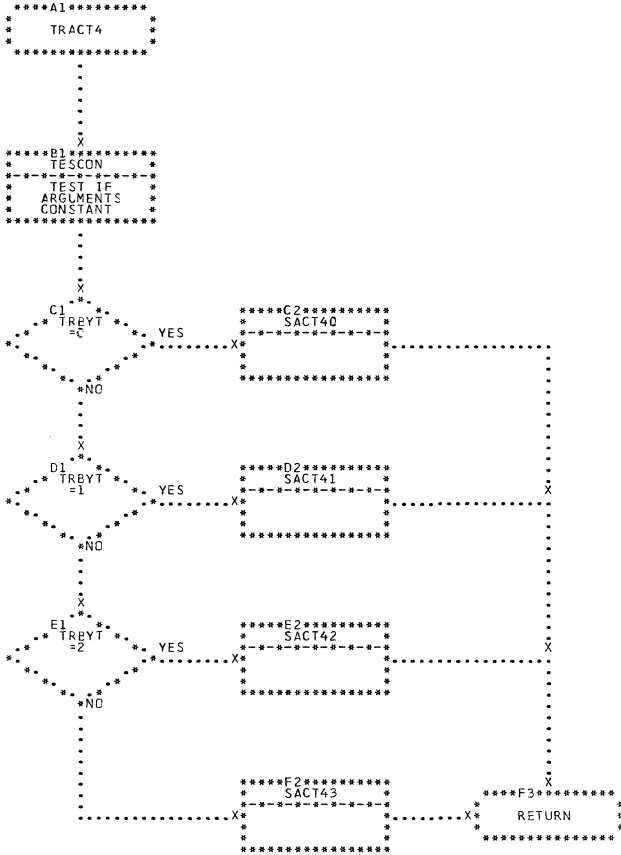


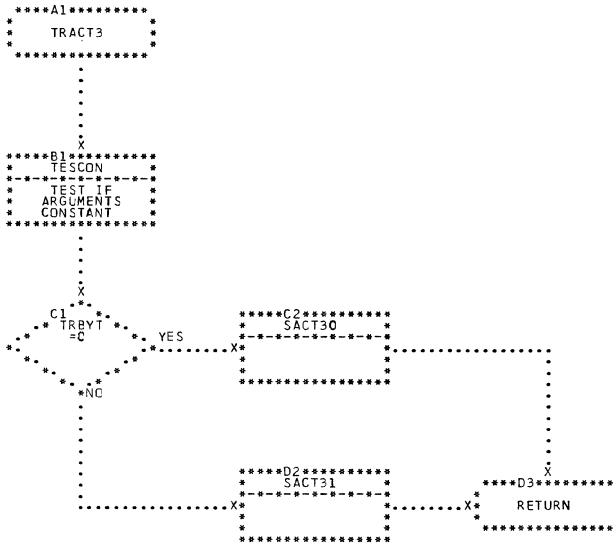


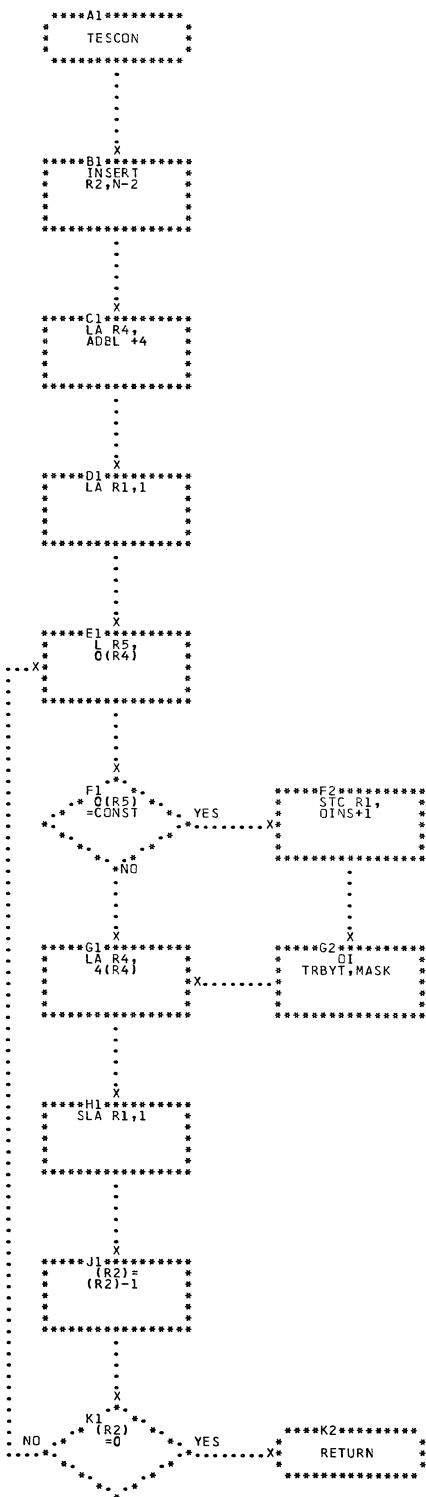












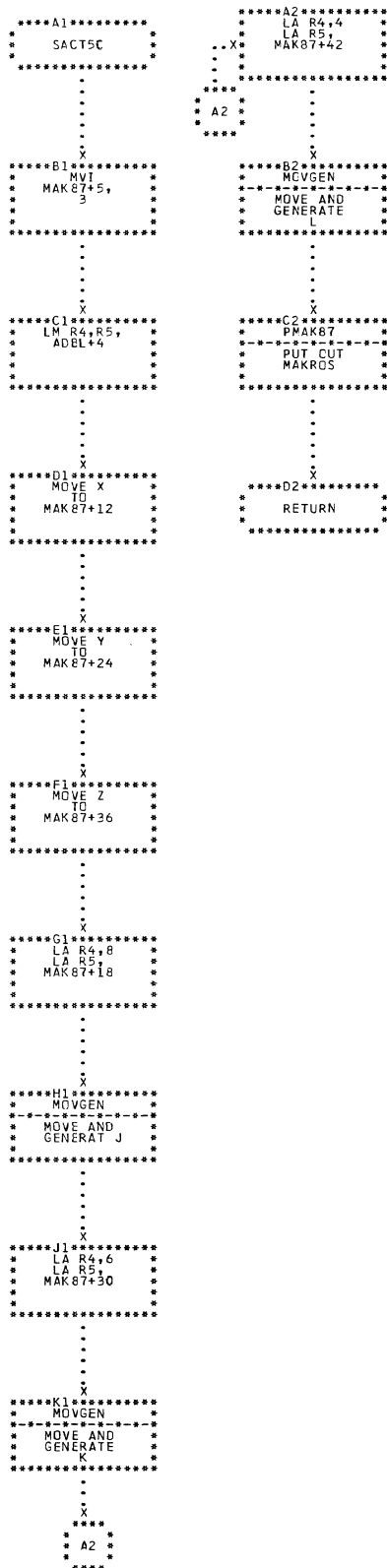
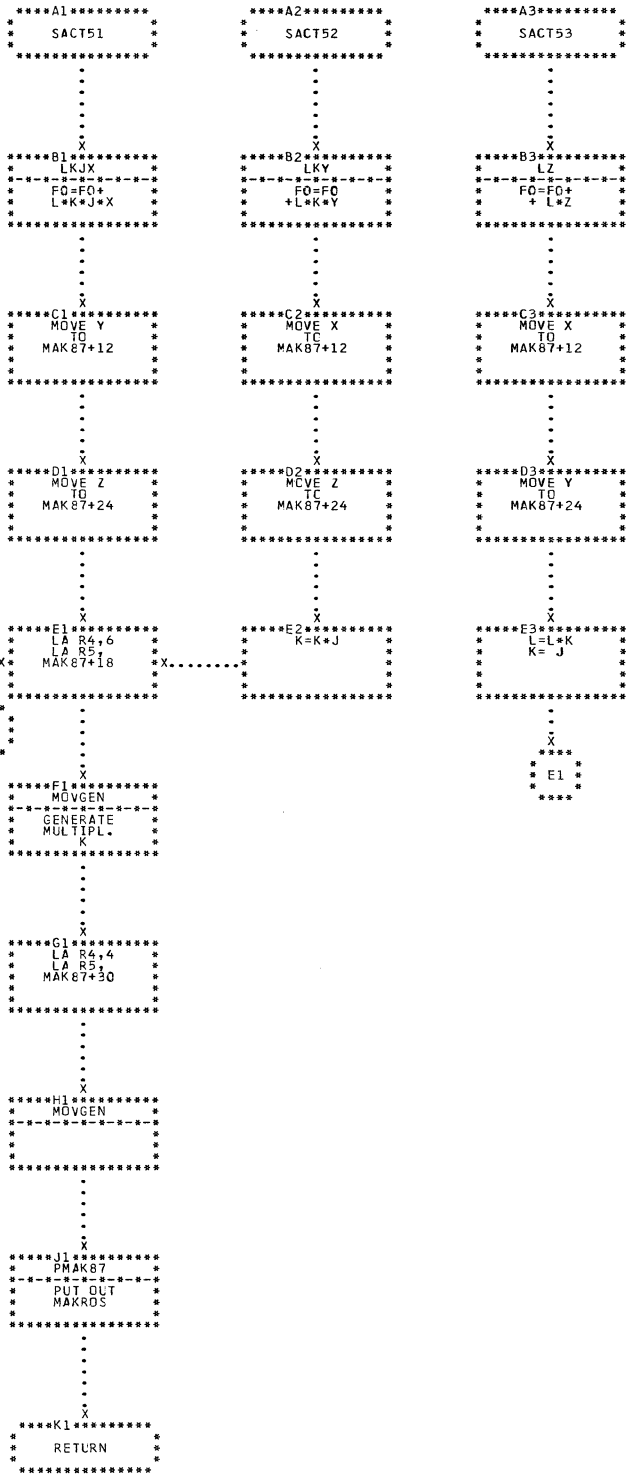
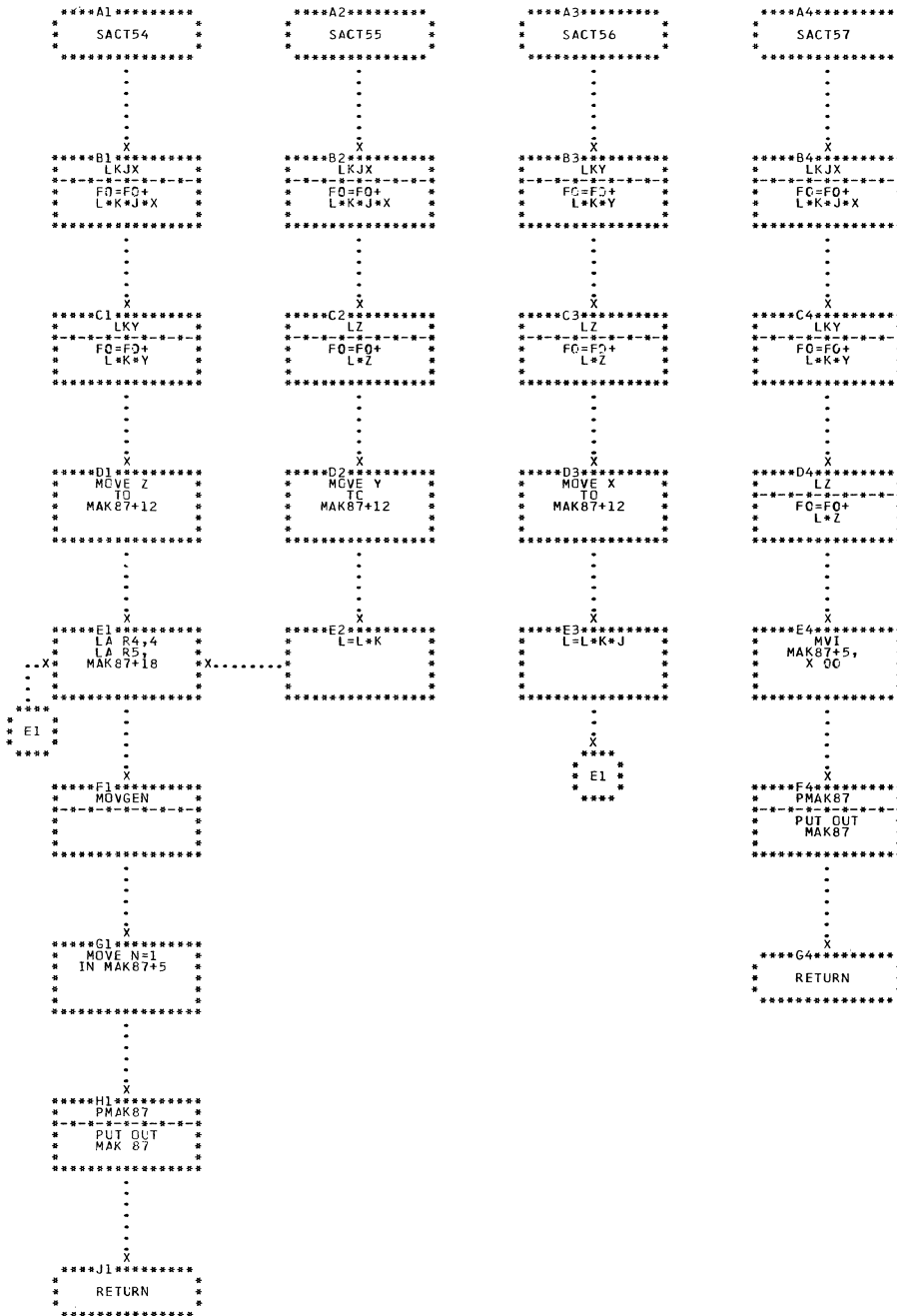
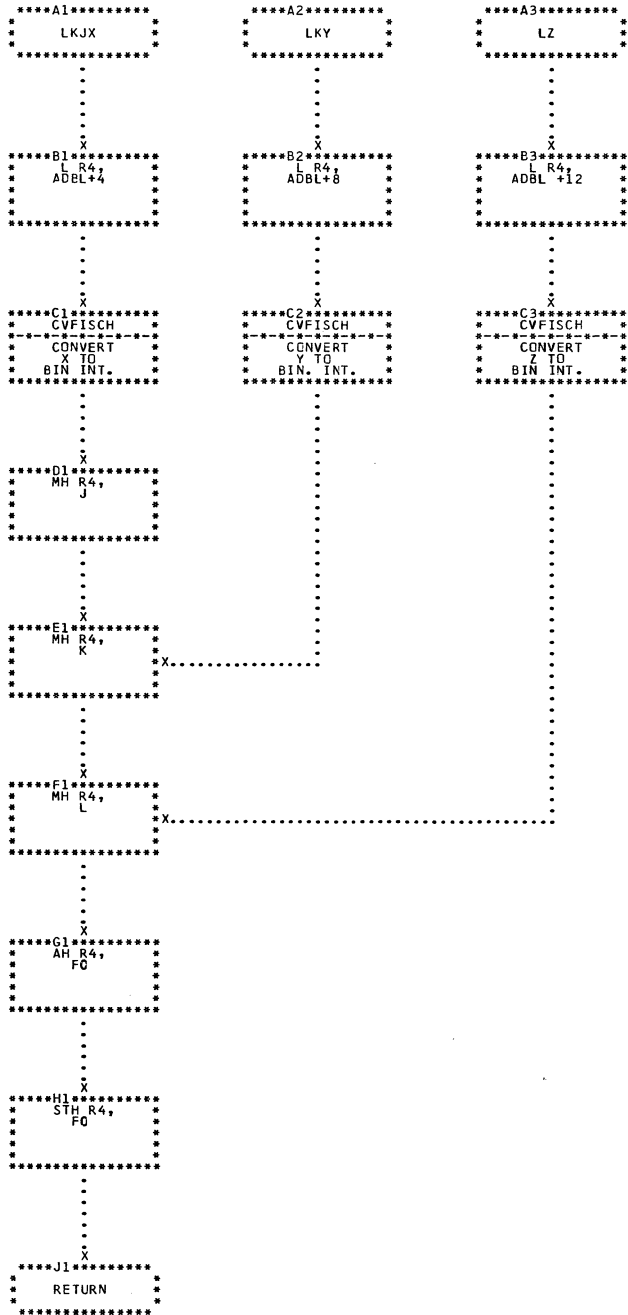


CHART JY. IJXD15

SACT50







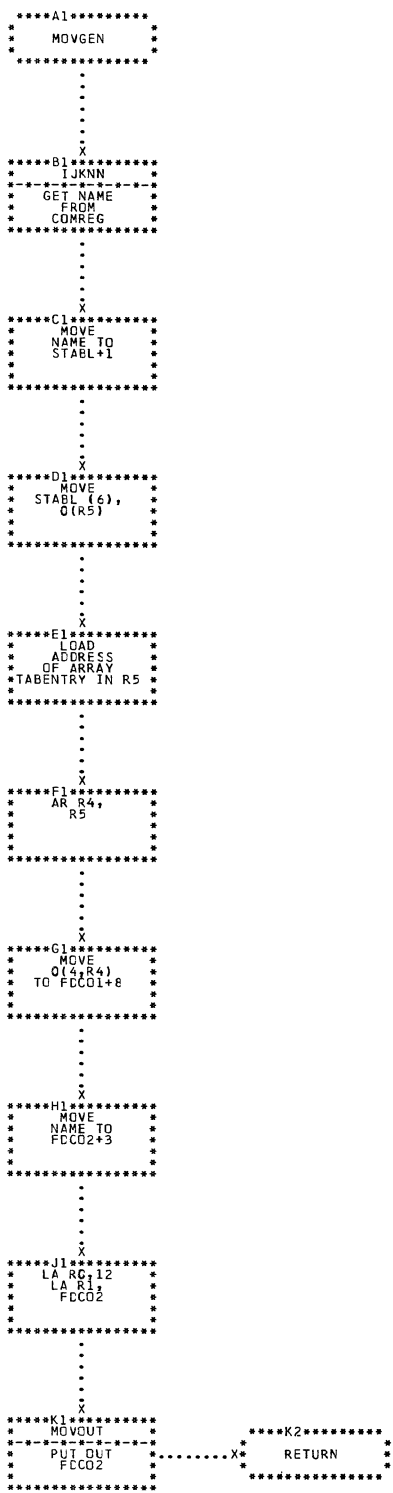
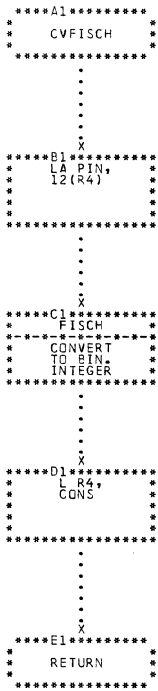
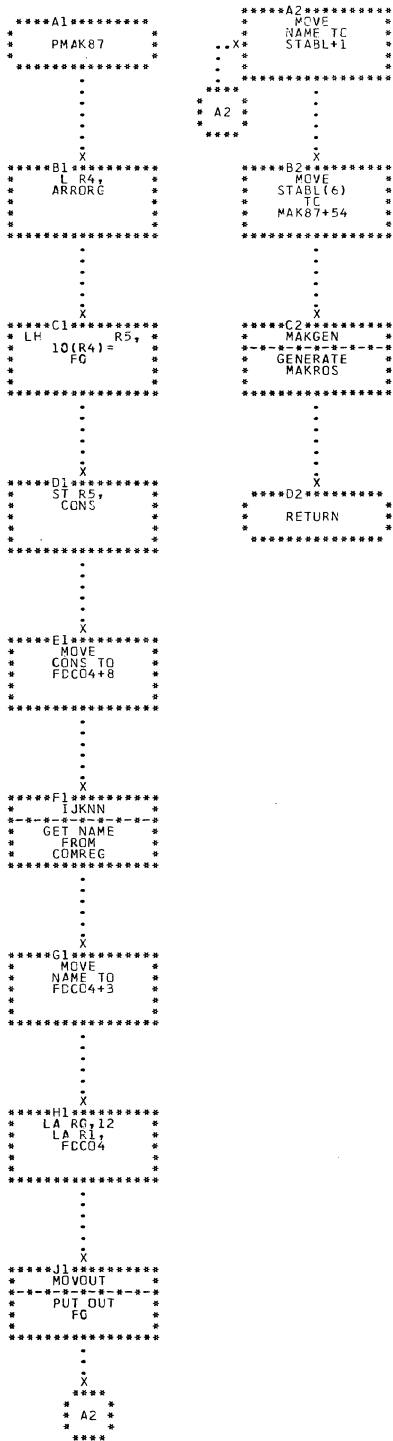
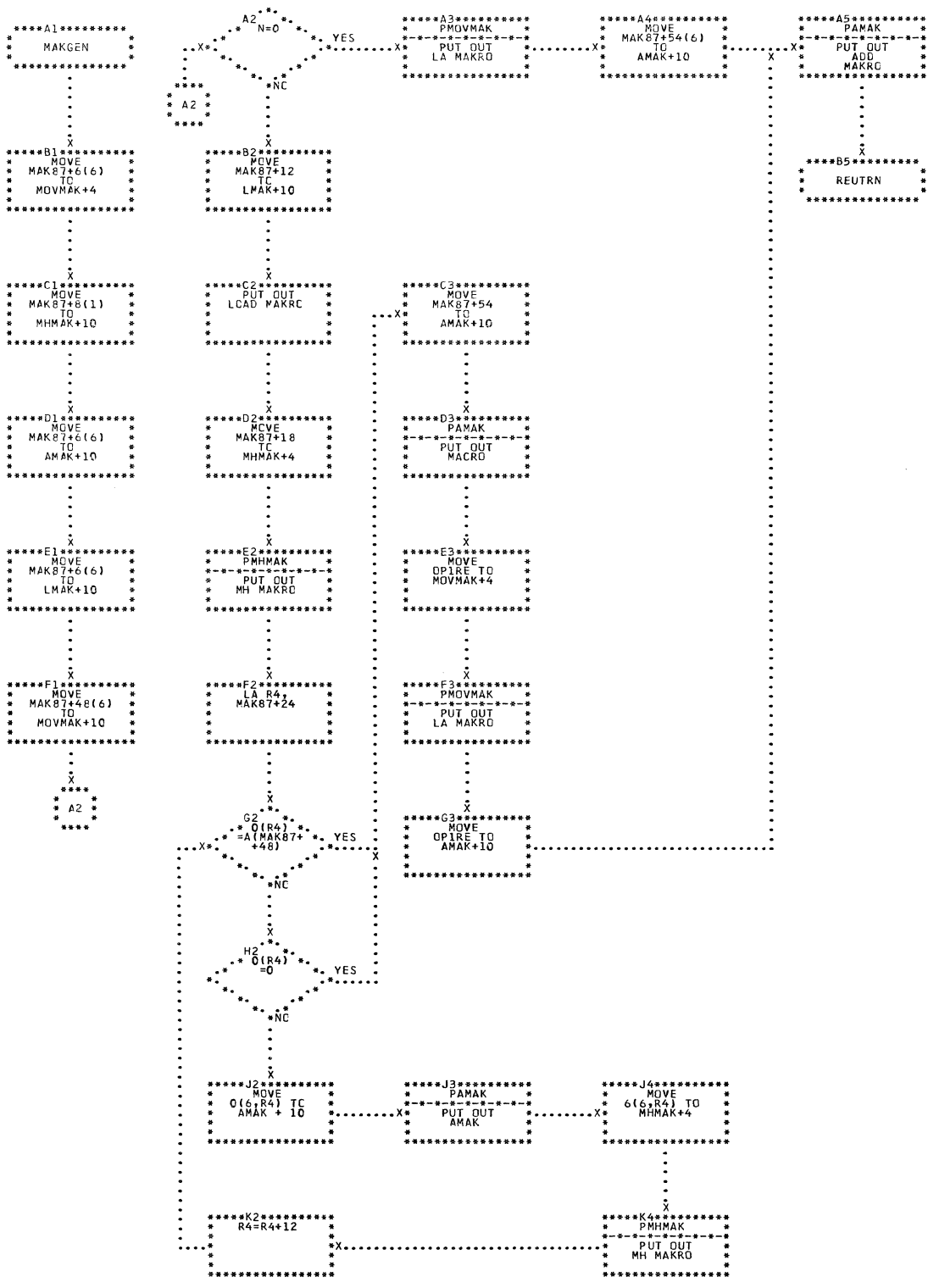
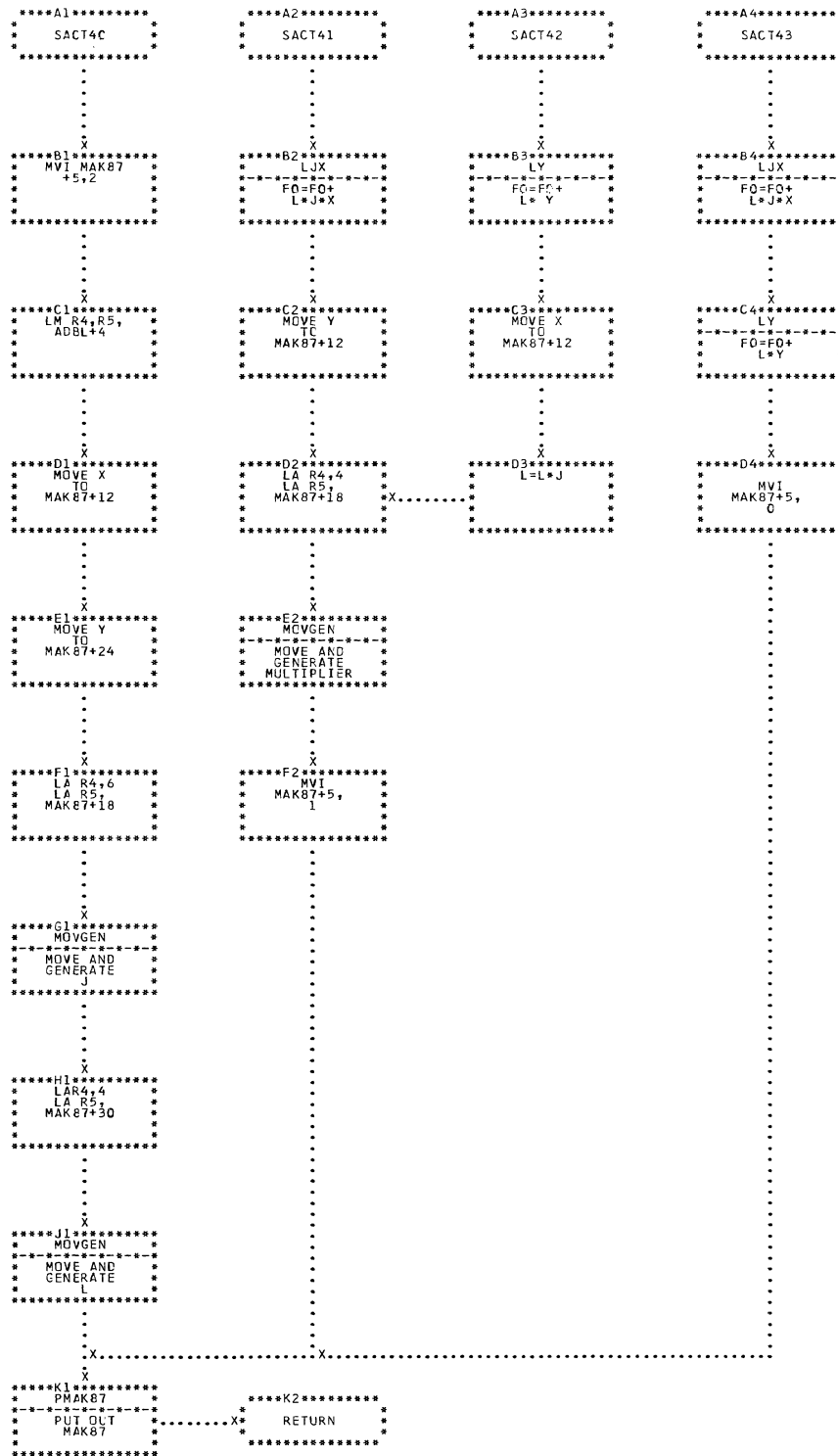


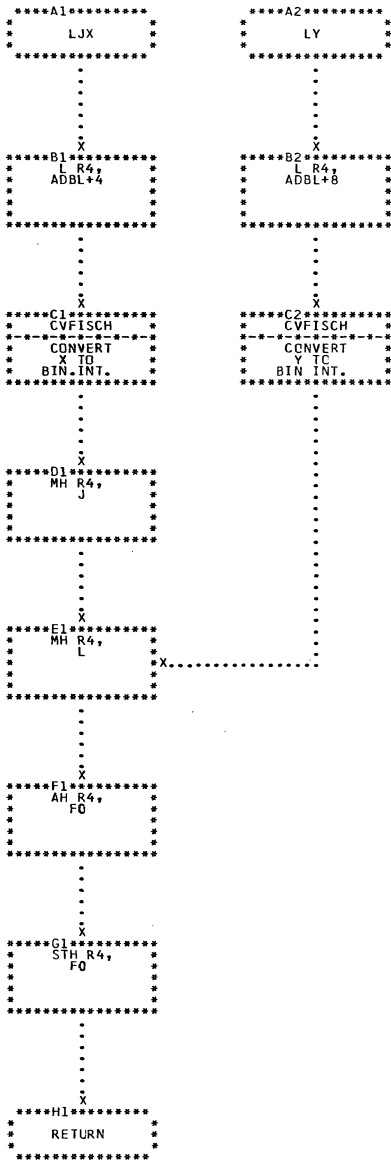
CHART KC. IJXD15 MOVGEN

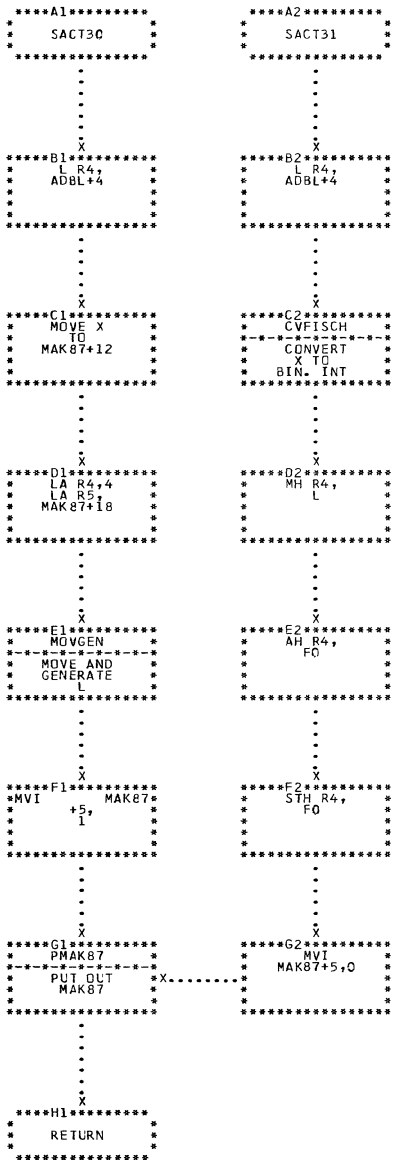


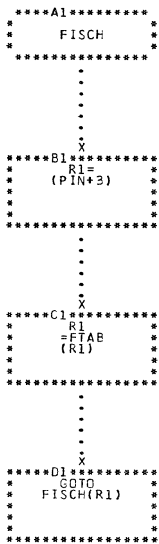


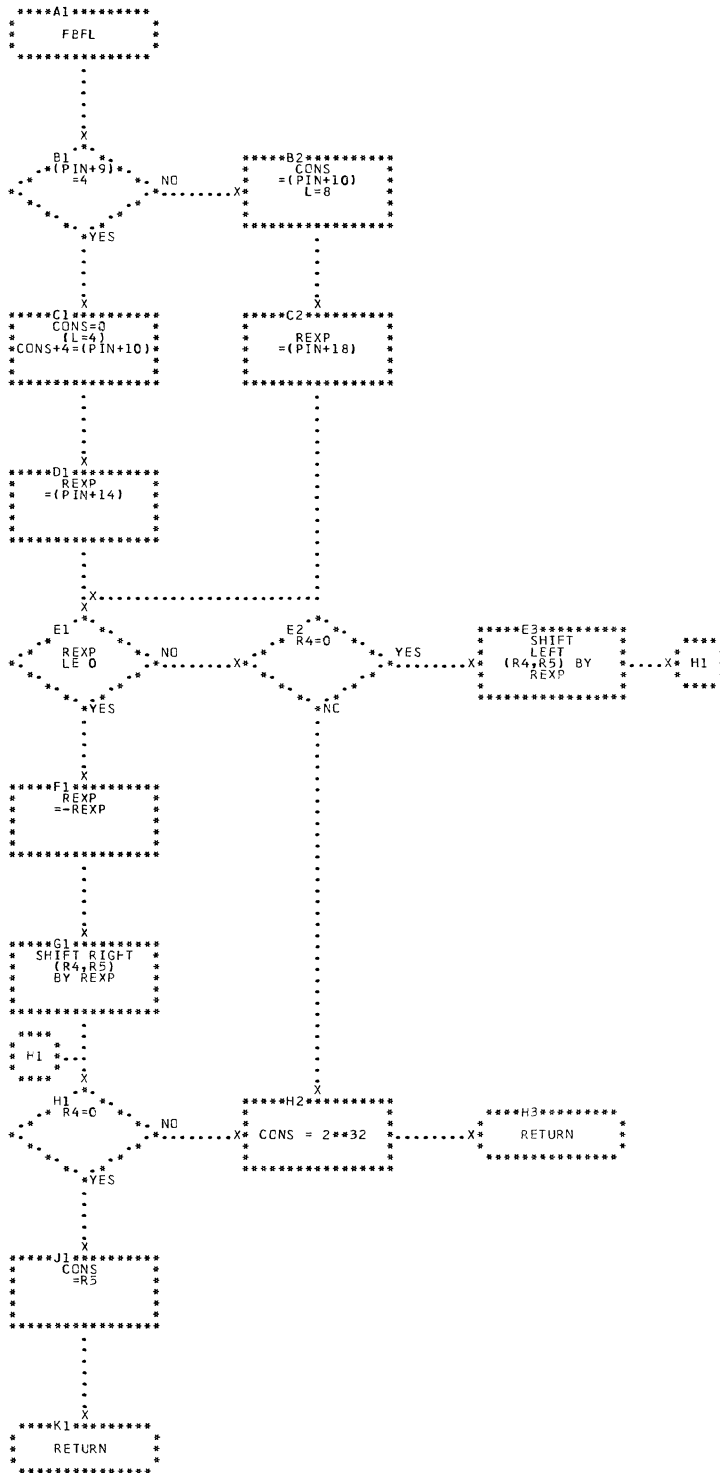


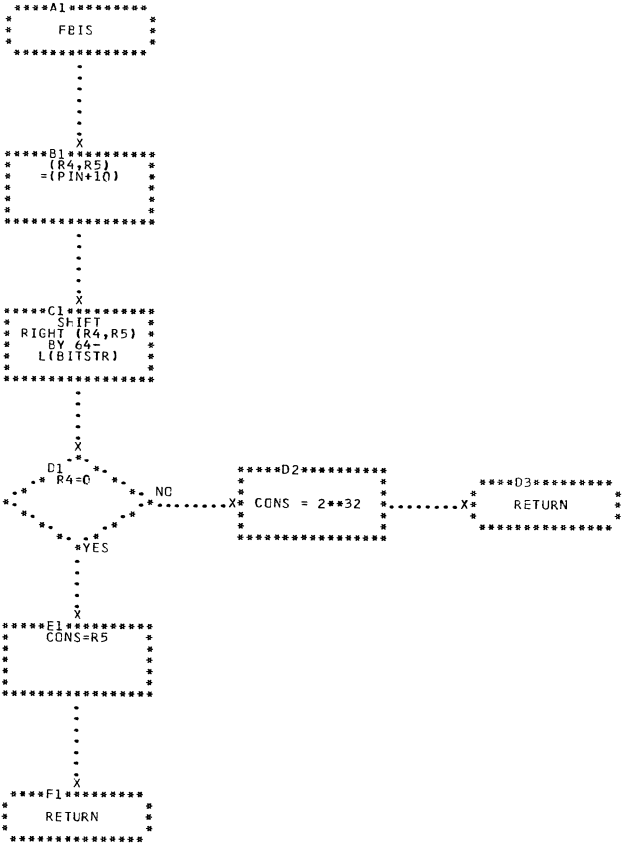


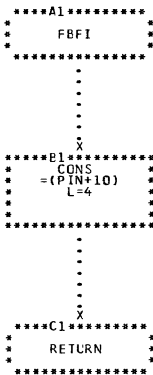


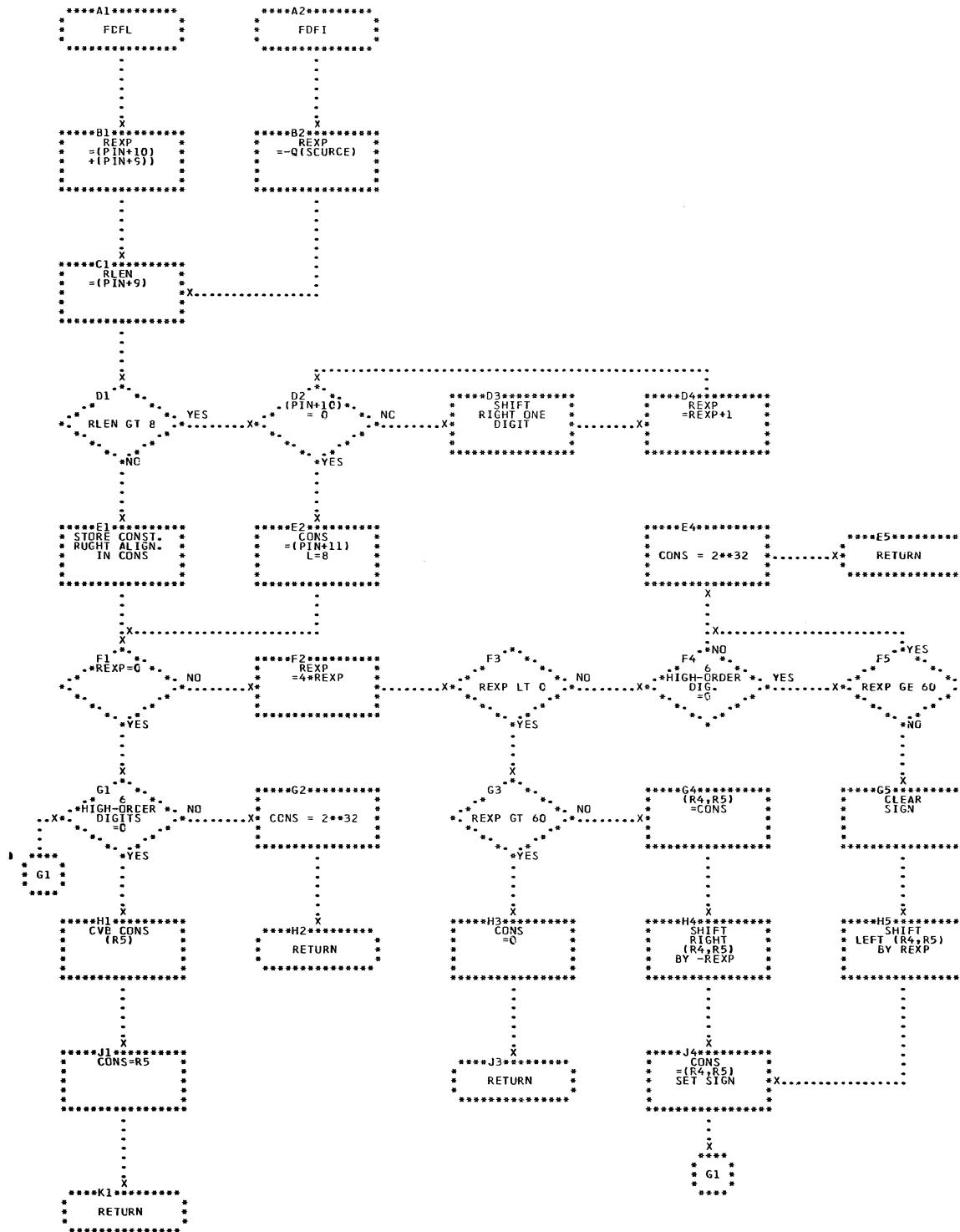












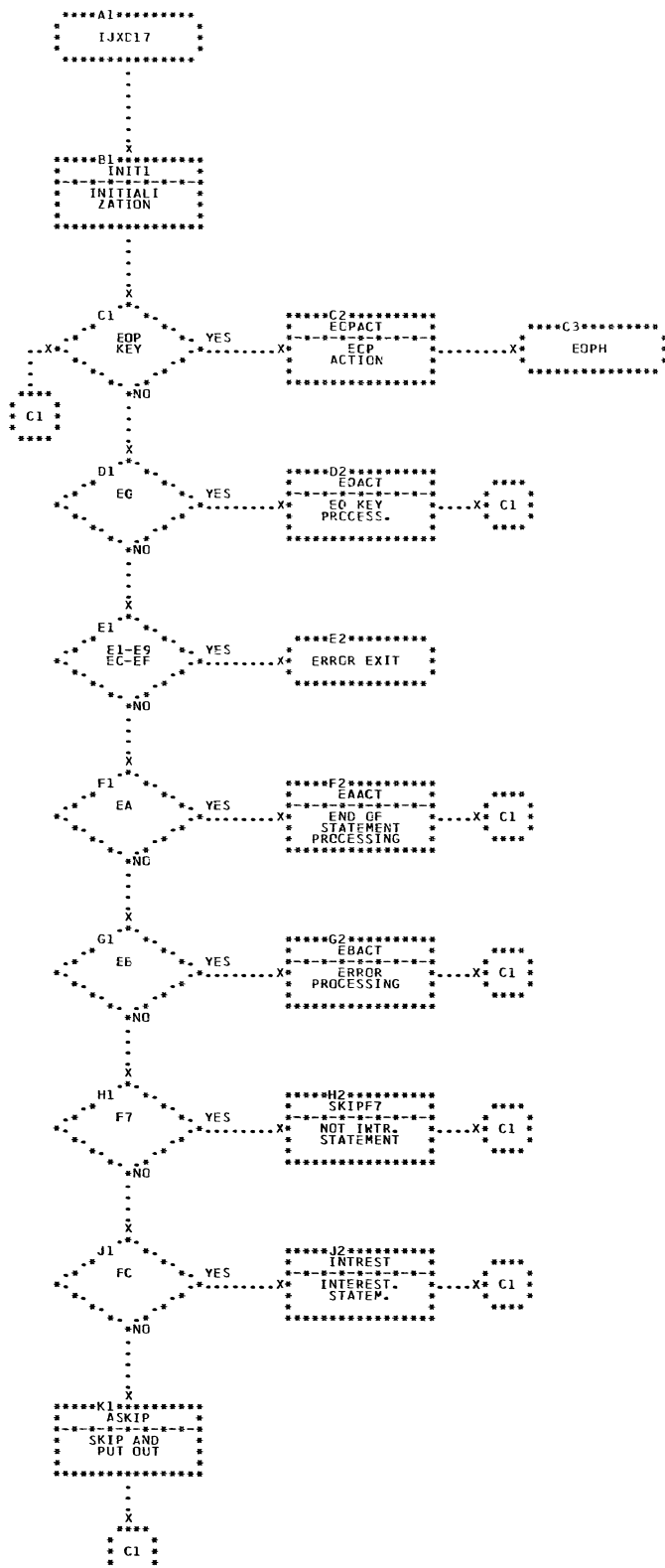
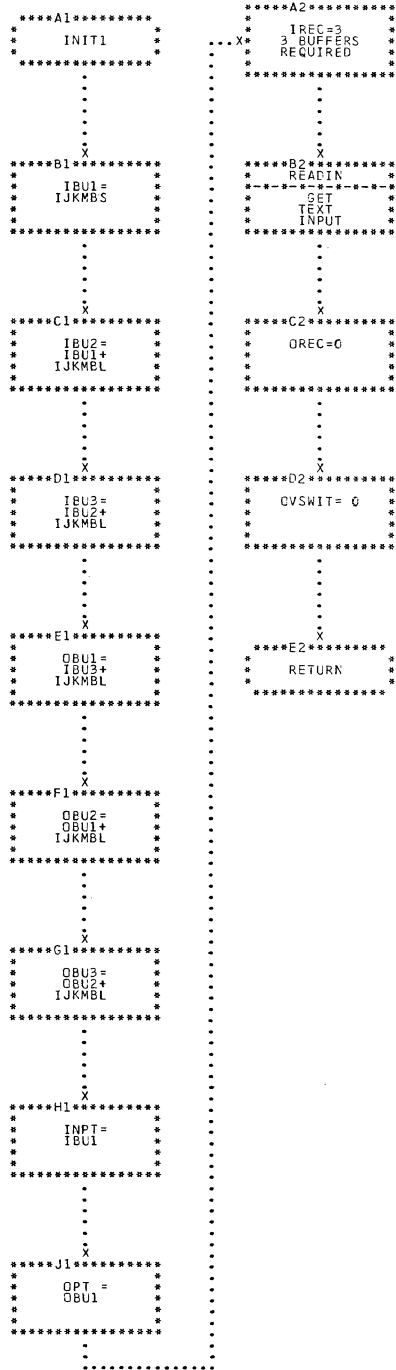


CHART LA. IJXD17

GENERAL FLOW



```
*****A1*****
      EOACT
*****
```

.....
X

```
*****B1*****
      CLEAR
      ERROR
      SWITCHES
      ERRS,CT
*****
```

.....
X

```
*****C1*****
      CLEAR
      ERROR
      STACK
      ERRS,CT(16)
*****
```

.....
X

```
*****D1*****
      LA R0,6
      LR R1,INPT
*****
```

.....
X

```
*****E1*****
      MOVOUT
      PUT OUT
      STATEMENT
      BEGIN
*****
```

.....
X

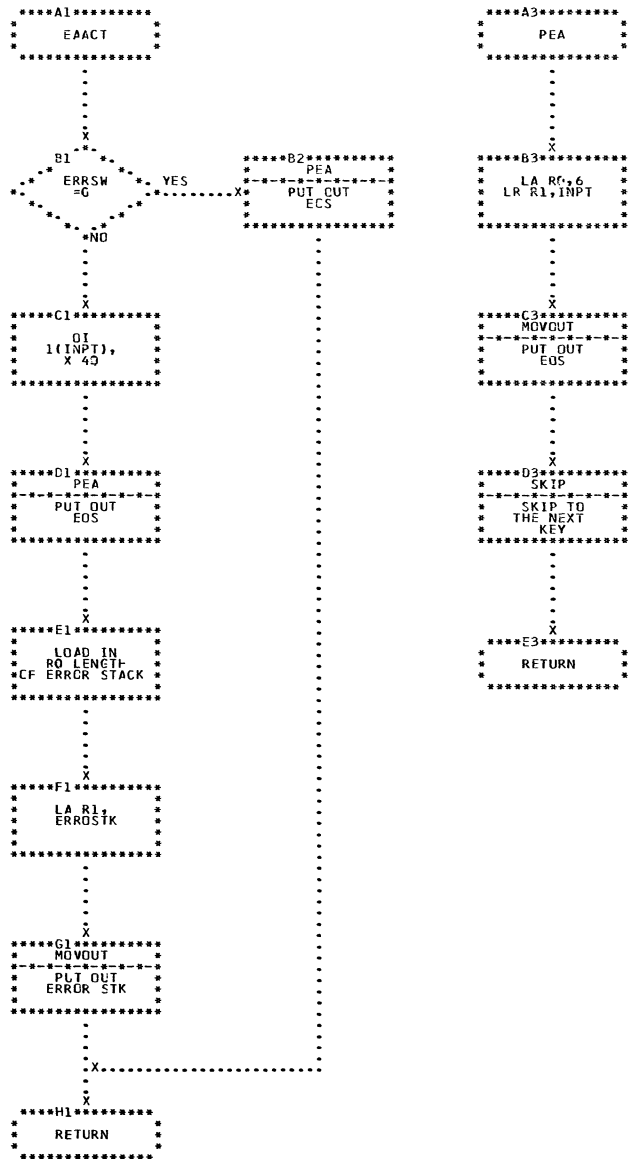
```
*****F1*****
      LA
      RC,6
*****
```

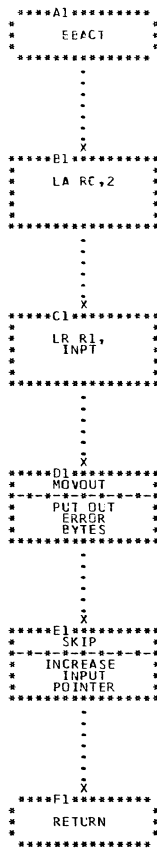
.....
X

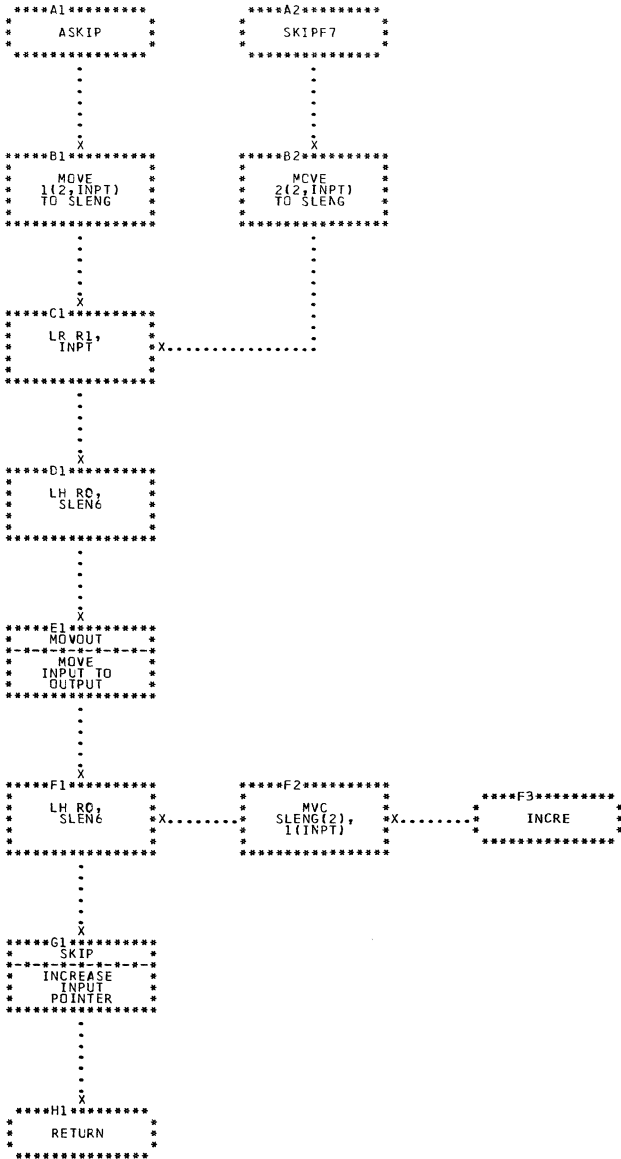
```
*****G1*****
      SKIP
      INCREASE
      INPUT POINT
*****
```

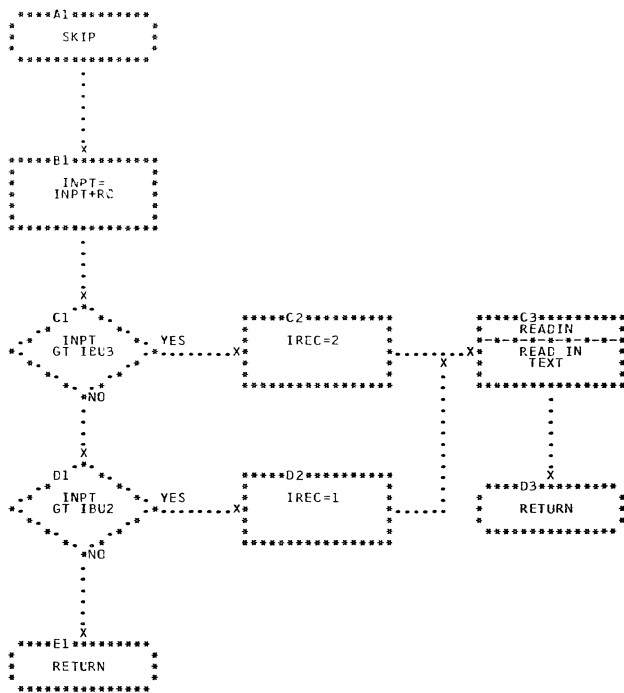
.....
X

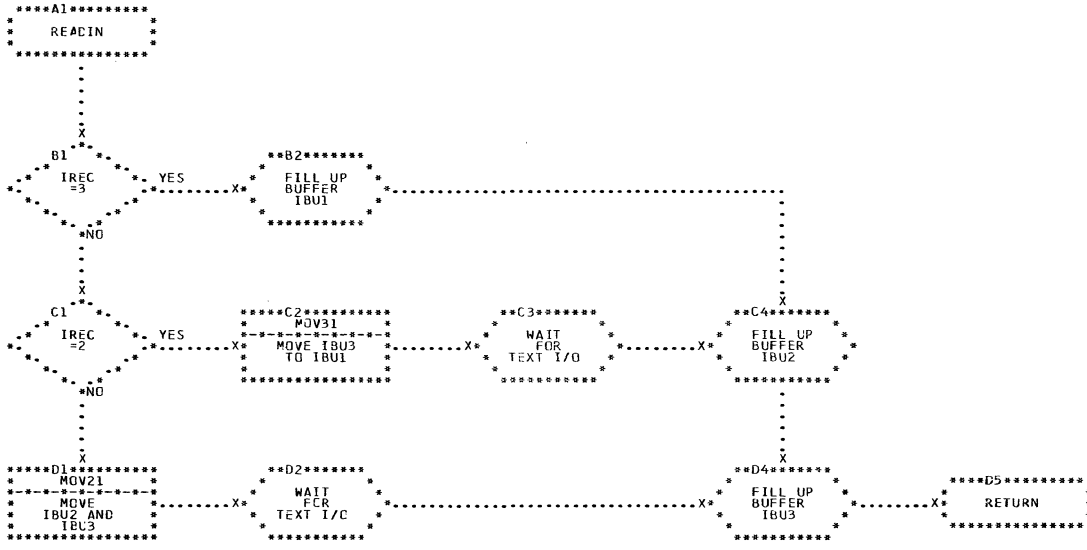
```
*****H1*****
      RETURN
*****
```

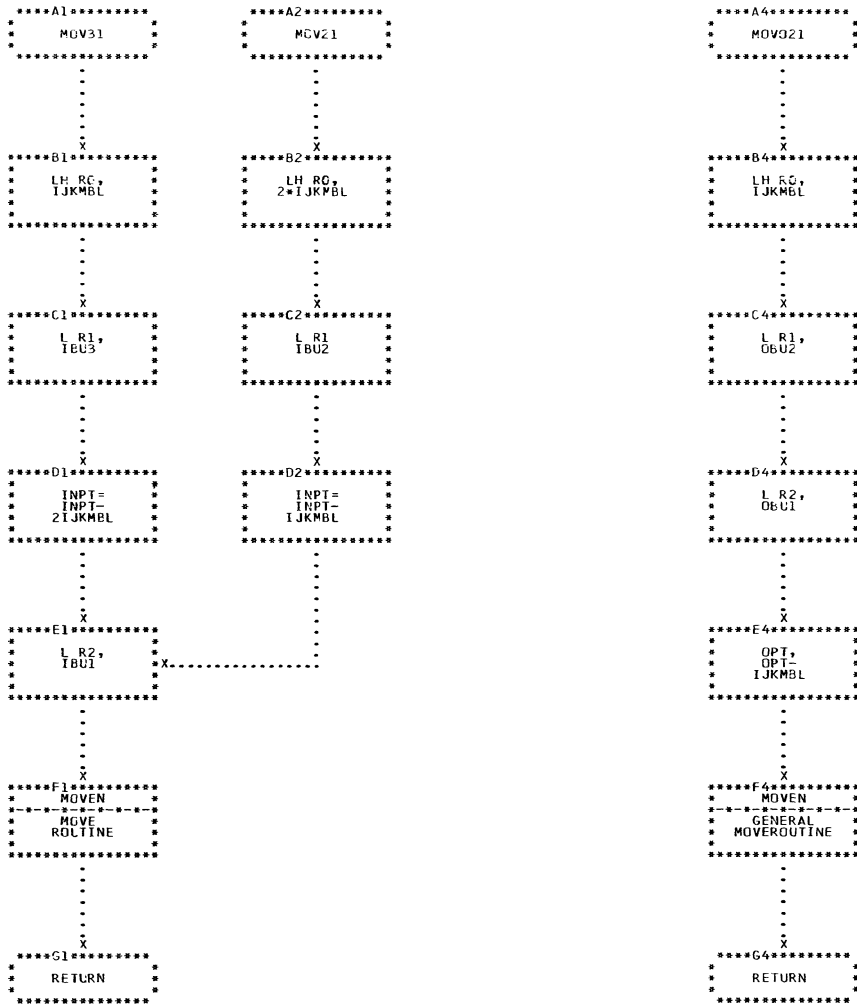


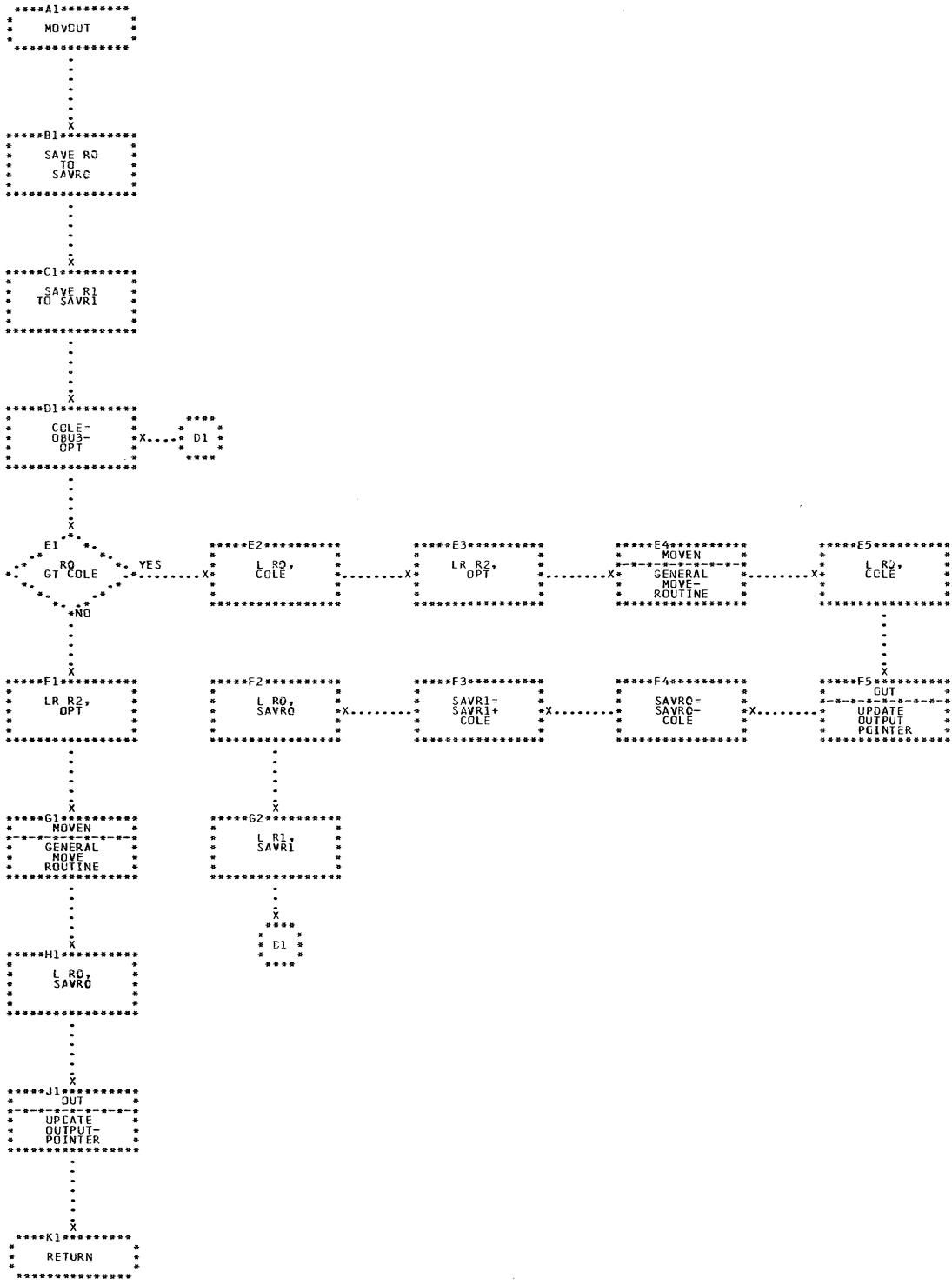


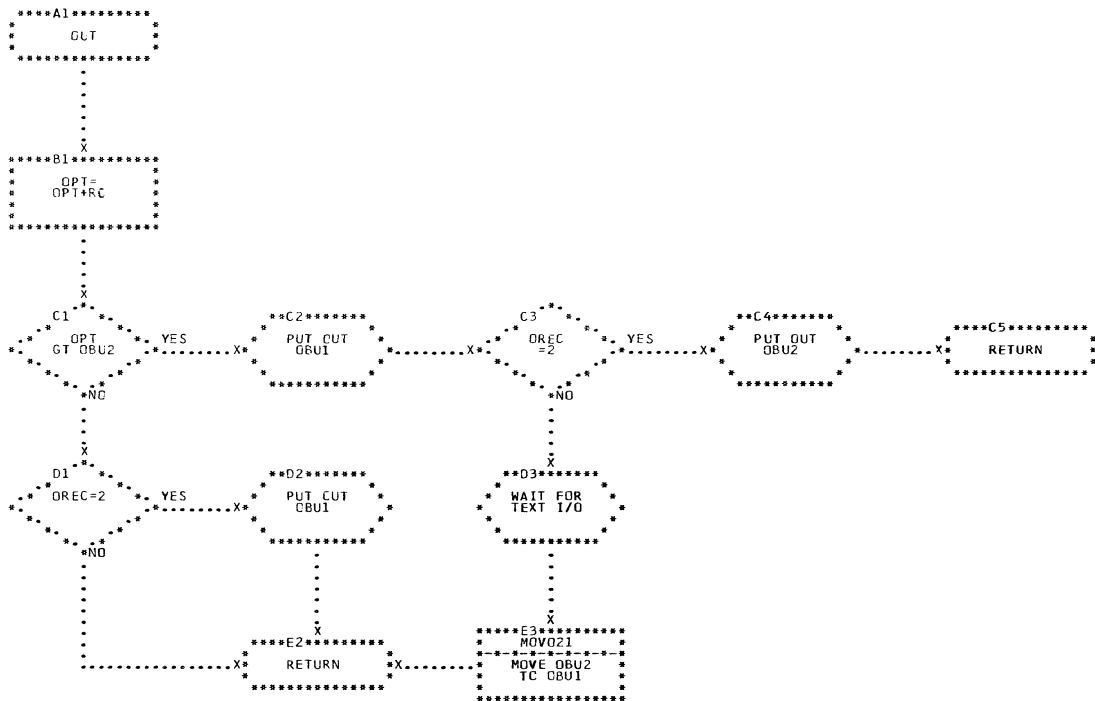


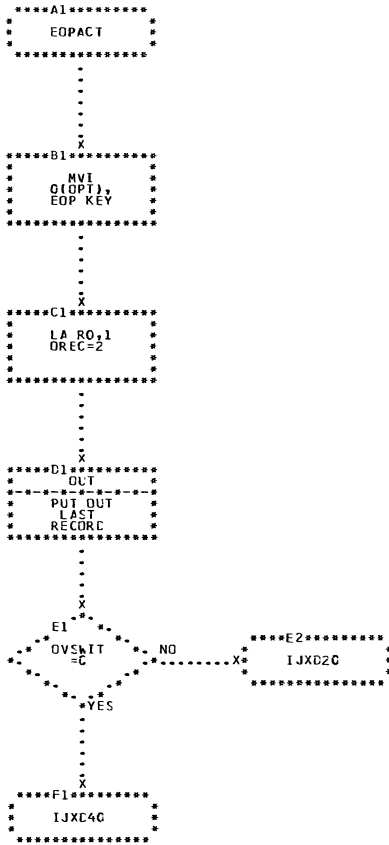


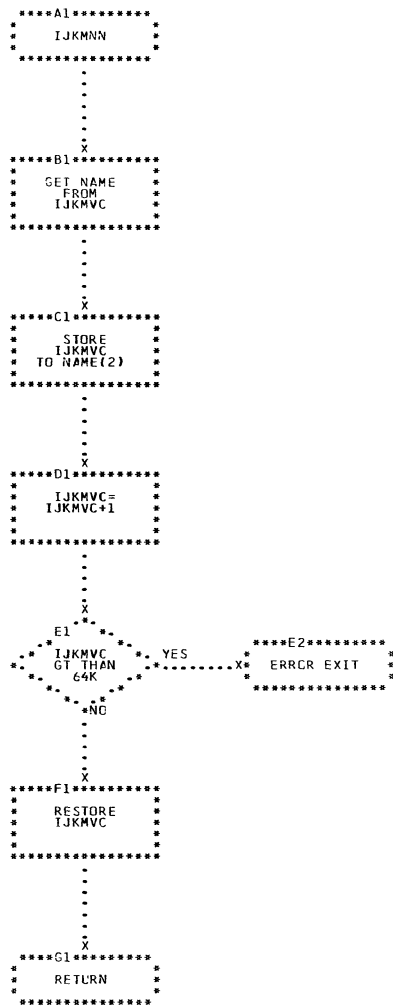


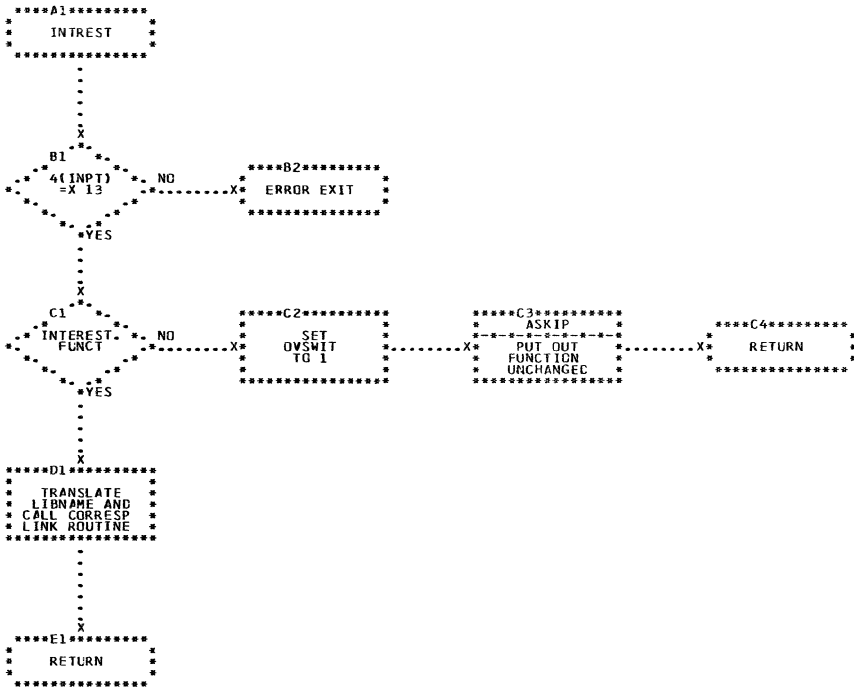


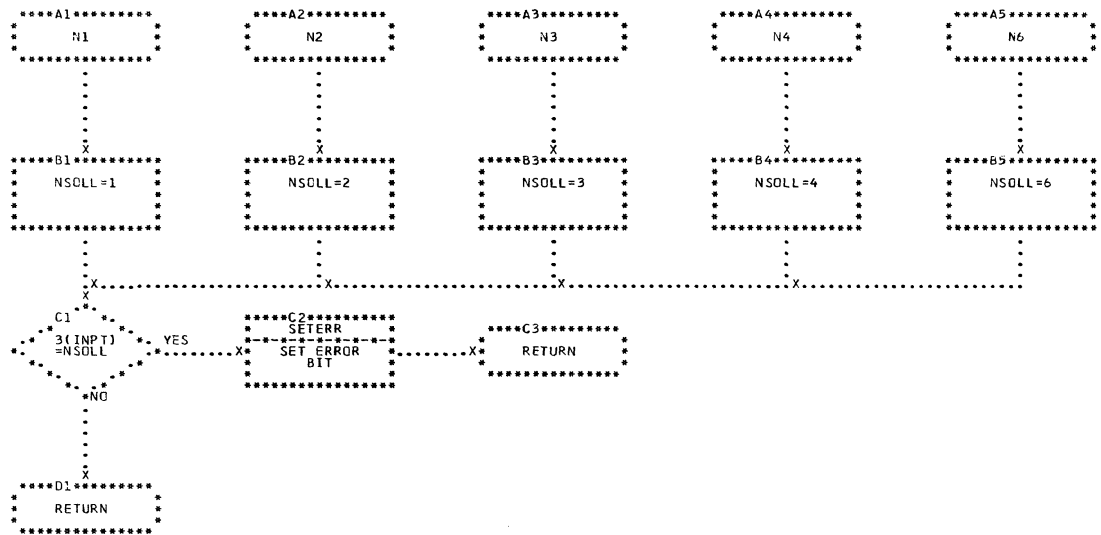


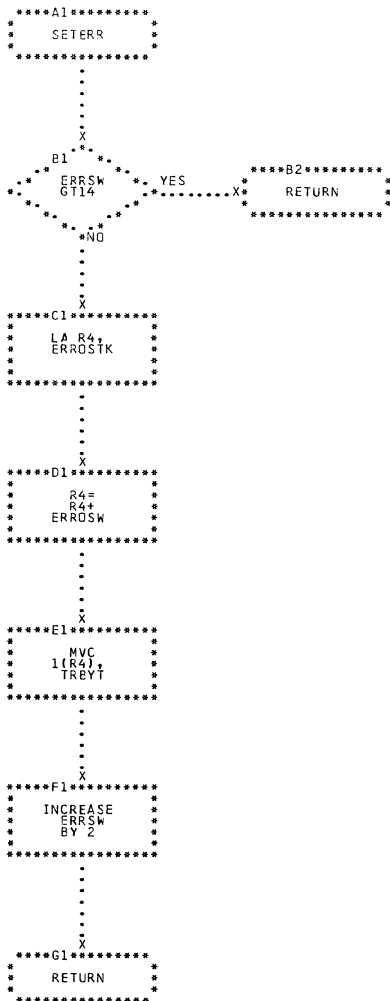


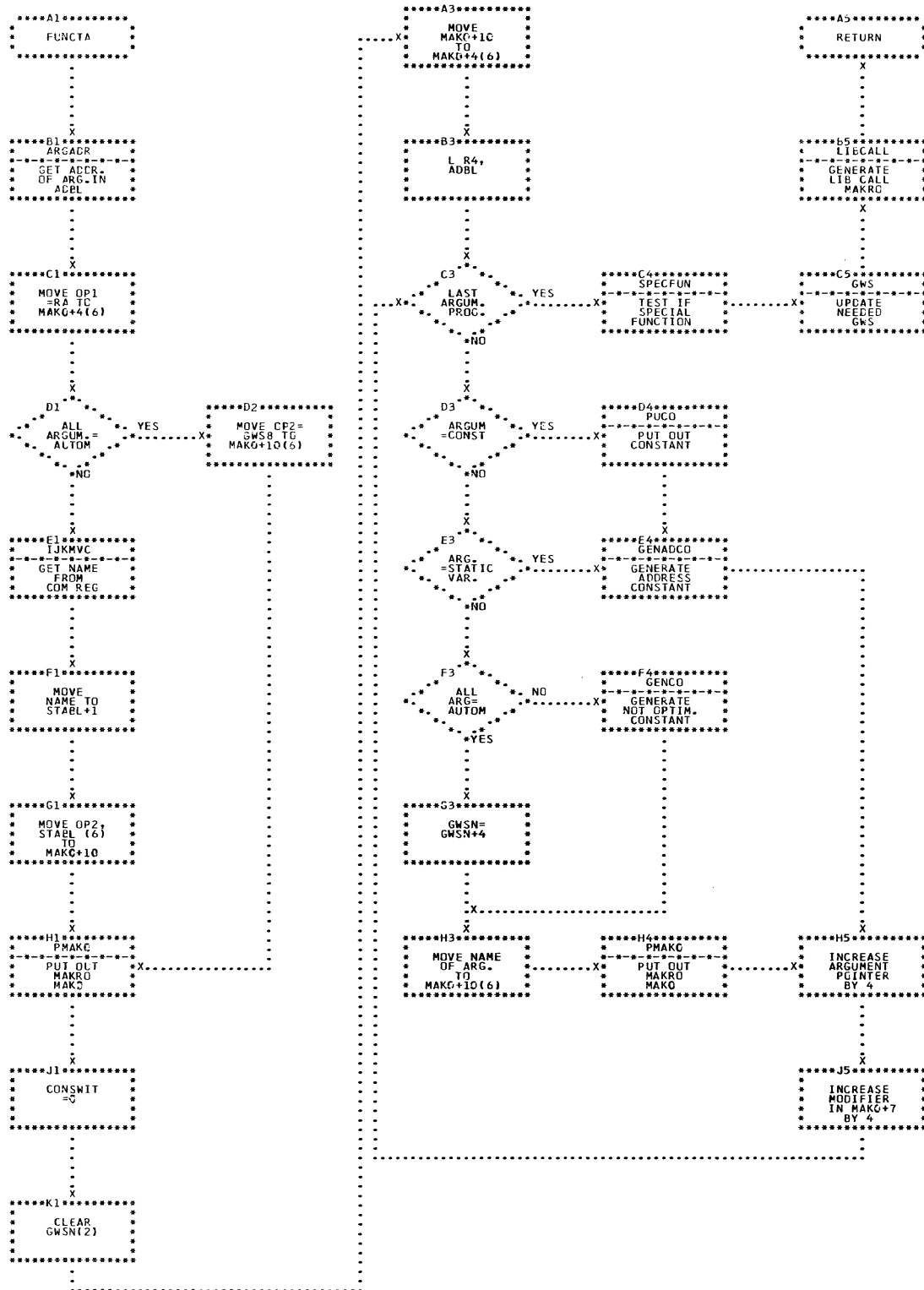












*****A1*****
REPROUN

.....
X

*****B1*****
ARCADR
GET ADR.
OF ARG

.....
X

*****C1*****
MOVE
ADBL+4
TO
ADBL+12

.....
X

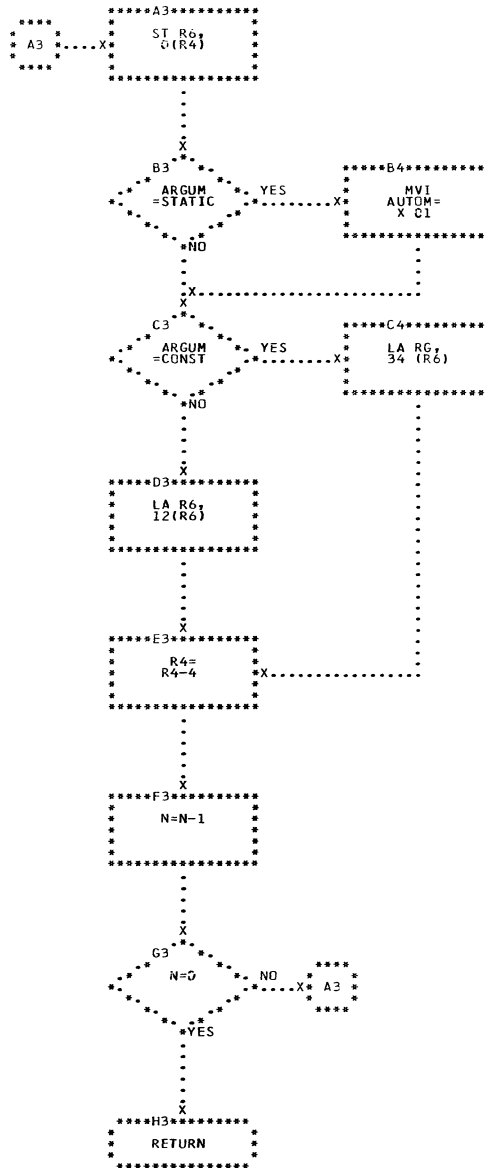
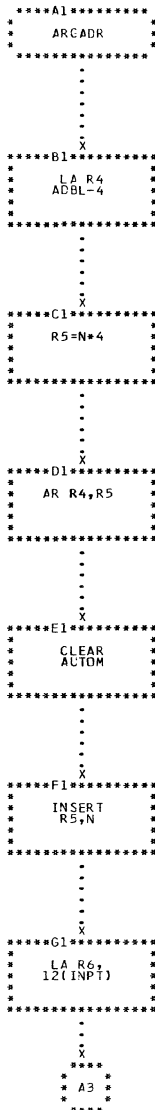
*****D1*****
MOVE
ADBL+8 TO
ADBL+4

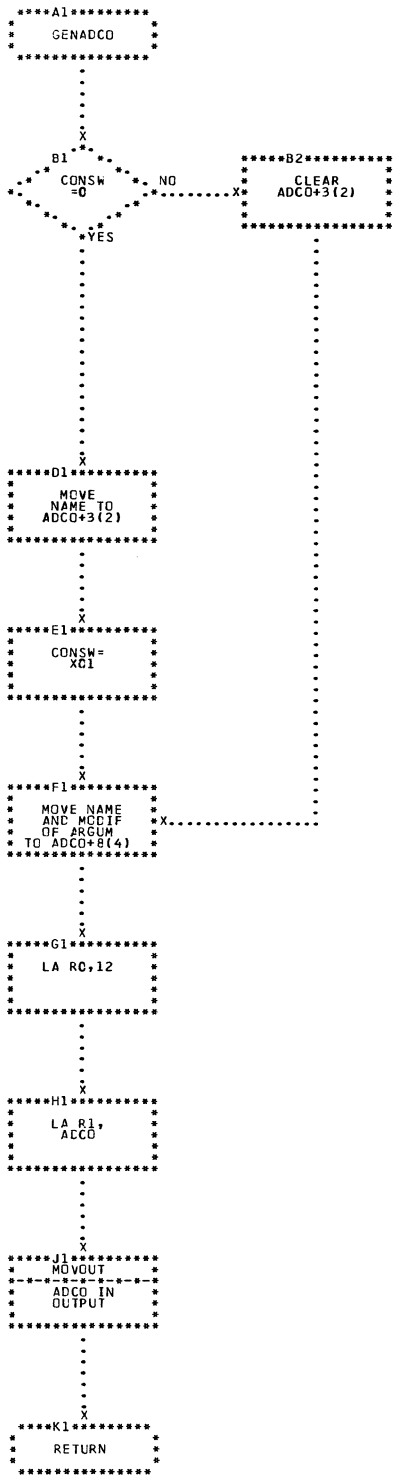
.....
X

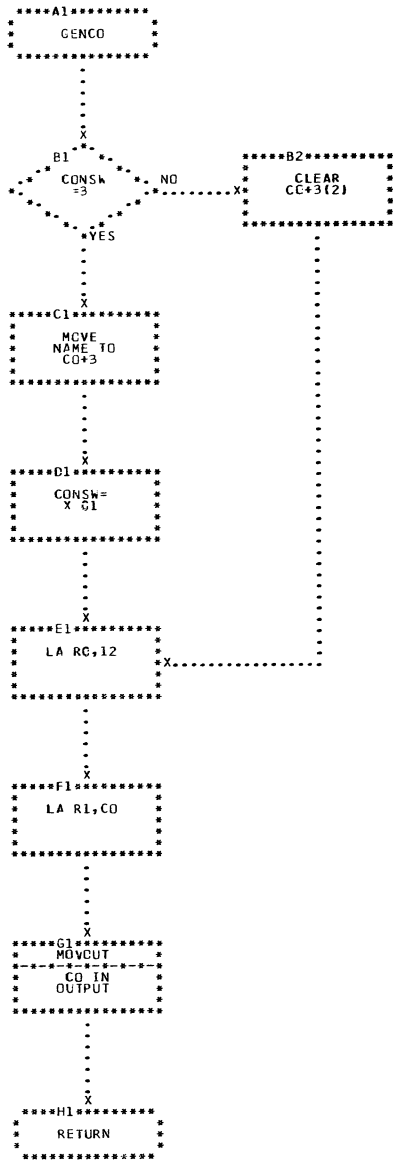
*****E1*****
CLEAR
ADBL+8(4)

.....
X

*****F1*****
RETRUN







```
*****A1*****
*   PMAKC   *
*****
```

```
.....
*   X   *
.....
```

```
*****B1*****
* LOAD    *
* LENGTH *
* OF MAKO *
* IN RG   *
*****
```

```
.....
*   X   *
.....
```

```
*****C1*****
* LA     *
* R1,MAK *
*****
```

```
.....
*   X   *
.....
```

```
*****D1*****
* MOVOUT *
*-----*
* GENERAL *
* MOVE    *
* ROUTINE *
*****
```

```
.....
*   X   *
.....
```

```
*****E1*****
* RETURN *
*****
```

```
*****A3*****
*   PUCO   *
*****
```

```
.....
*   X   *
.....
```

```
*****B3*****
* INSERT  *
* LENGTH *
* OF CONST *
* IN RO   *
*****
```

```
.....
*   X   *
.....
```

```
*****C3*****
* ADD FIXED *
* LENGTH=  *
* 13       *
*****
```

```
.....
*   X   *
.....
```

```
*****D3*****
* STORE   *
* SKIPP   *
* LENGTH *
*****
```

```
.....
*   X   *
.....
```

```
*****E3*****
* MOVE F3 *
* KEY AND *
* SKIP, L *
* ENGL.  *
* TO 9(R *
* S)     *
*****
```

```
.....
*   X   *
.....
```

```
*****F3*****
* L R1,  *
* 9(R5)  *
*****
```

```
.....
*   X   *
.....
```

```
*****G3*****
* MOVOUT *
*-----*
* GENERAL *
* MOVE    *
* ROUTINE *
*****
```

```
.....
*   X   *
.....
```

```
*****H3*****
* RETURN *
*****
```

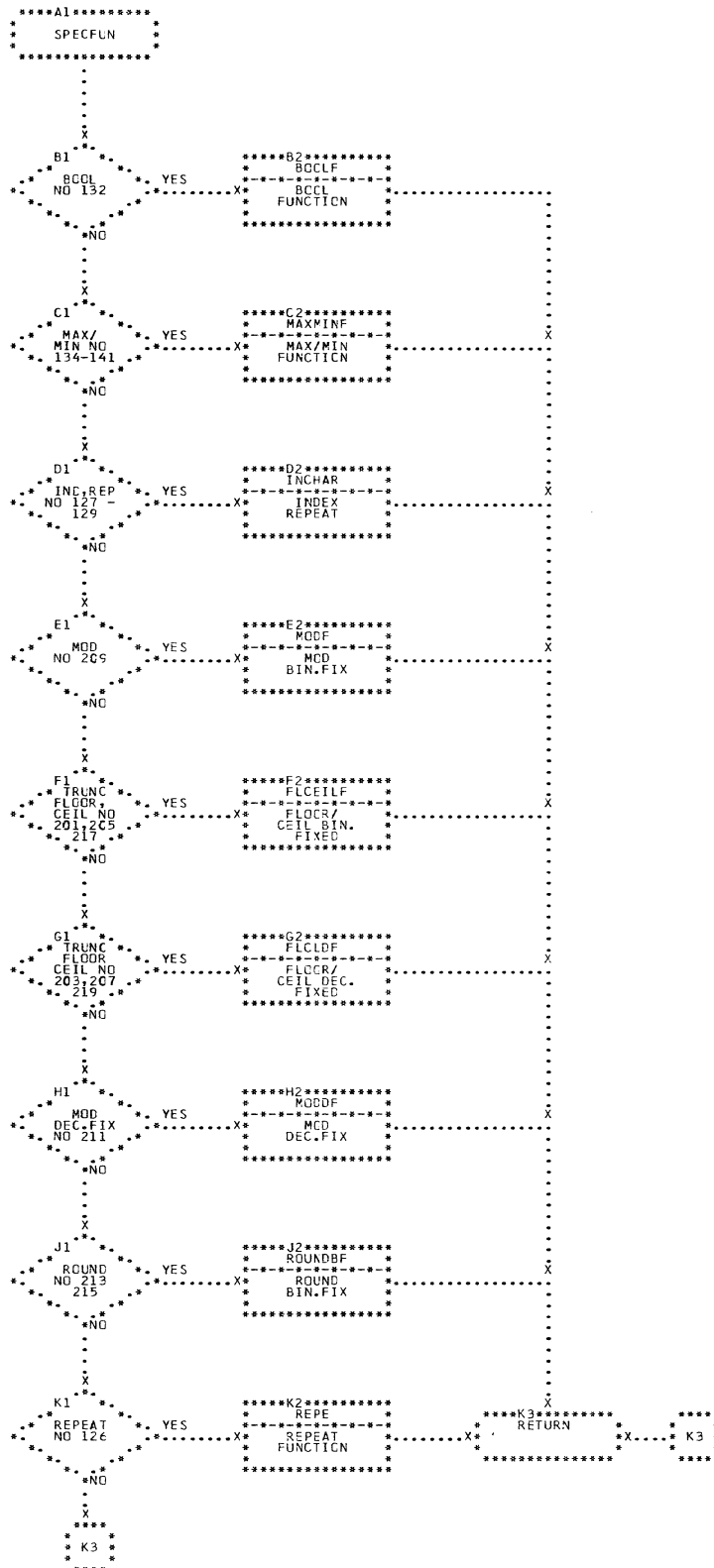
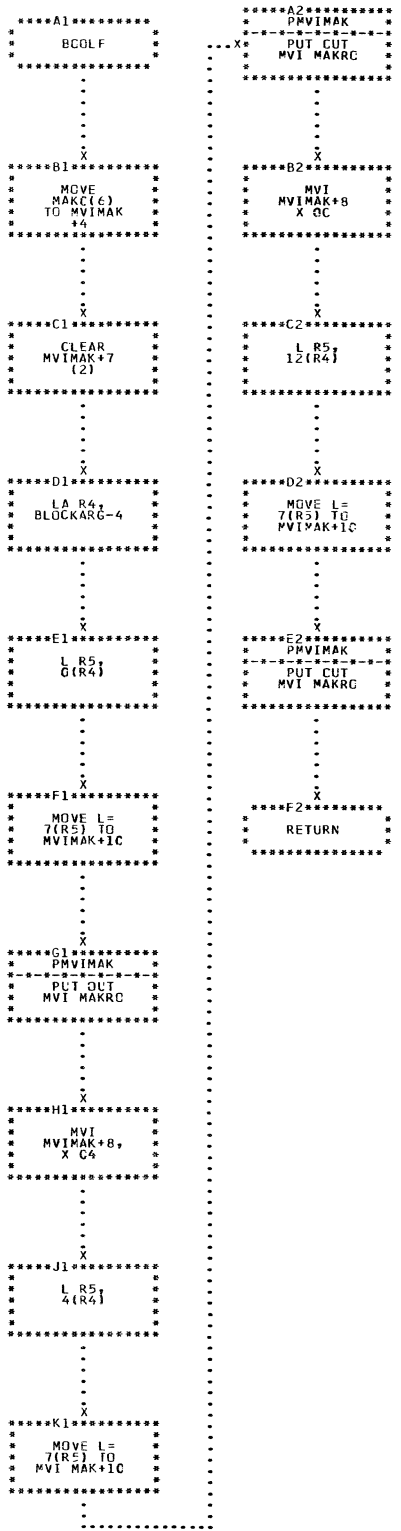
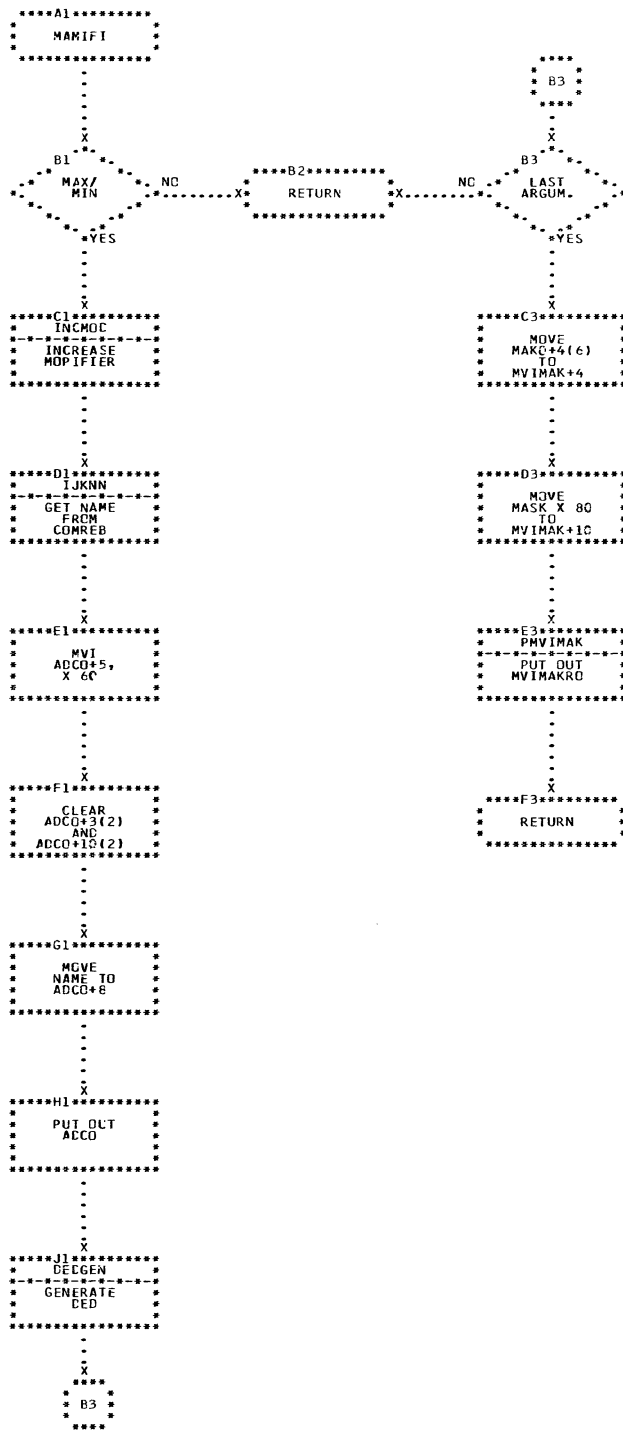
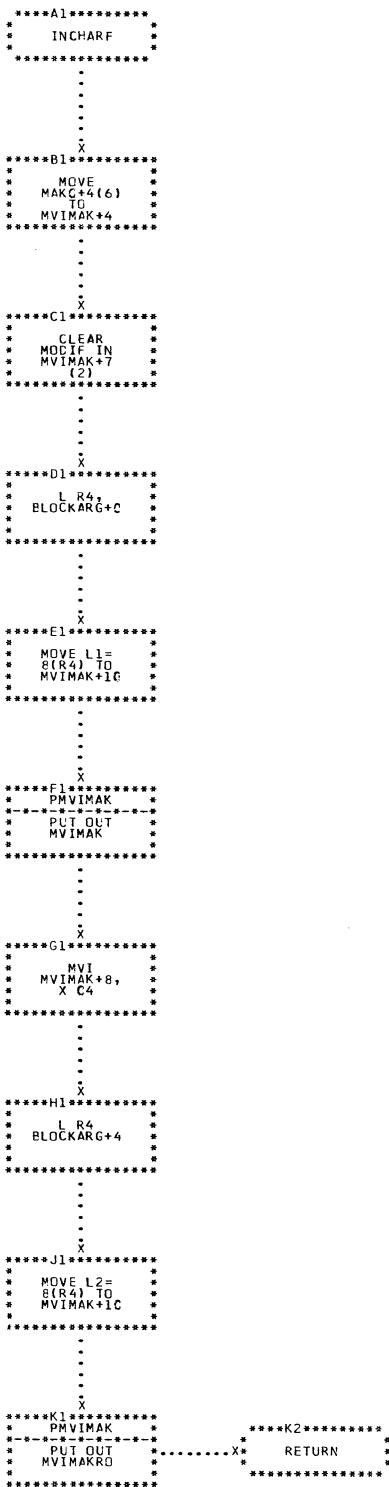
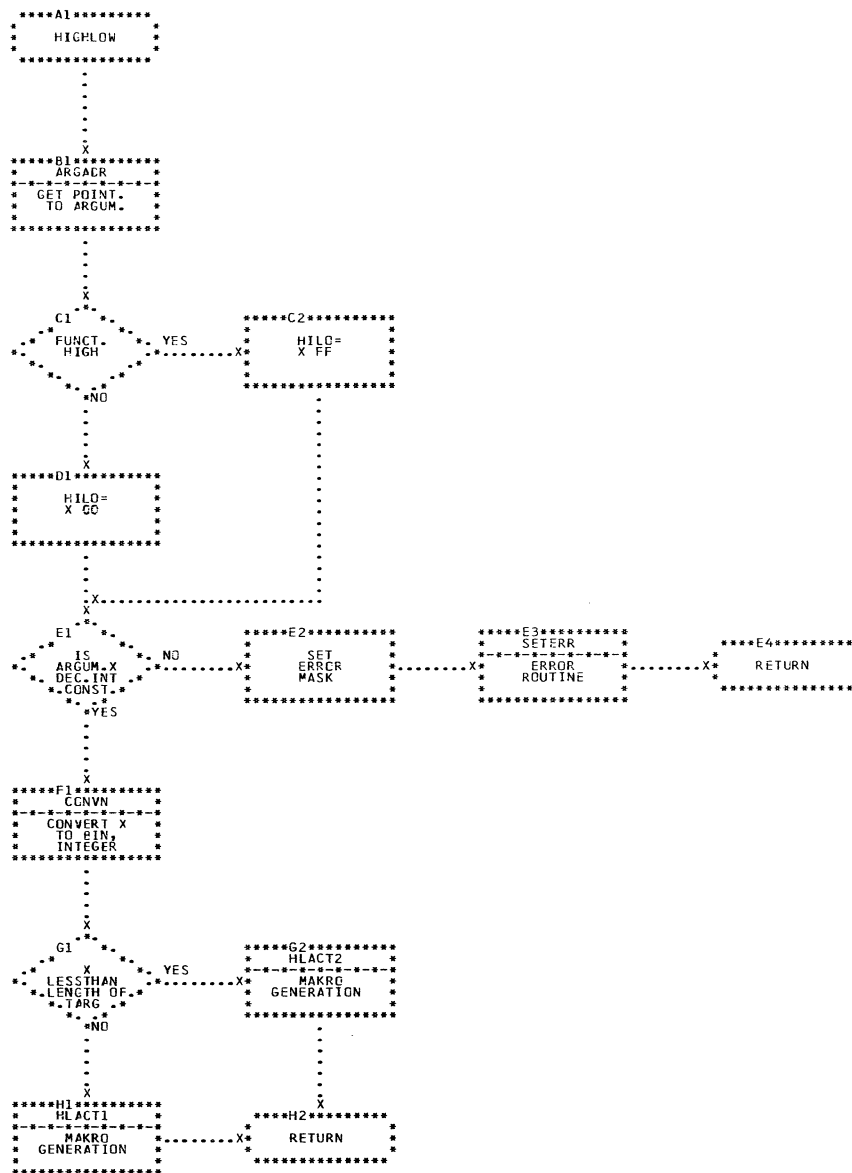



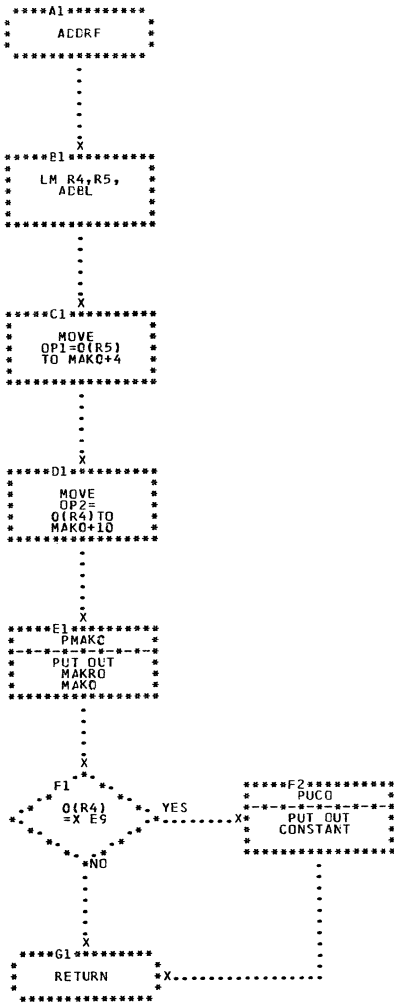
CHART LW. IJXD17 SPECFUN

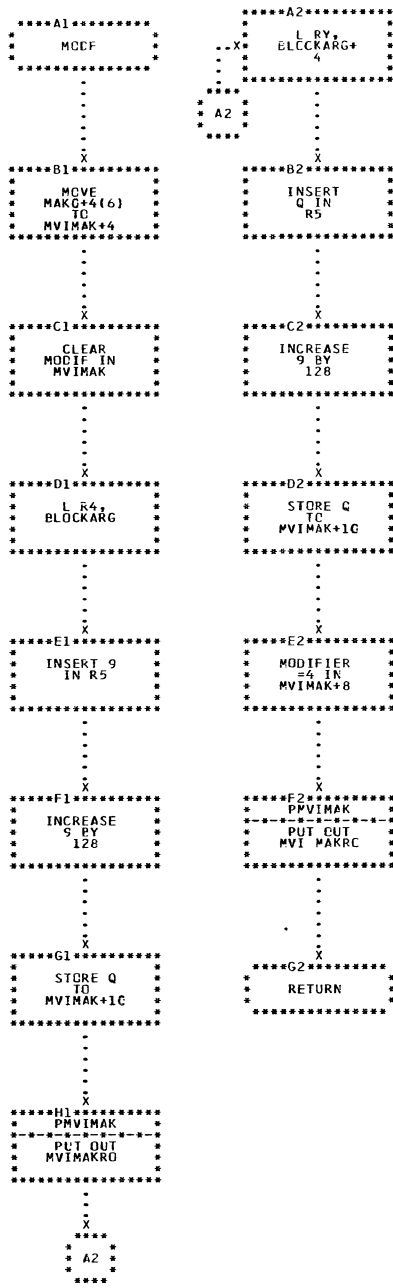












.....A1.....
FLCEILF
.....

.....
X

.....B1.....
MCVE
MAK C+4(6)
TO
MVIMAK+4
.....

.....
X

.....C1.....
CLEAR
MODIFIER
IN
MVIMAK+8
.....

.....
X

.....D1.....
L R4
BLOCKARG
.....

.....
X

.....E1.....
INSERT
Q IN
R5
.....

.....
X

.....F1.....
INCREASE
Q
R5
.....

.....
X

.....G1.....
STORE
Q TO
MVIMAK+10
.....

.....
X

.....H1.....
PWHIMAK
PUT_OLT
MVI MAKRO
.....

.....
X

.....J1.....
RETURN
.....

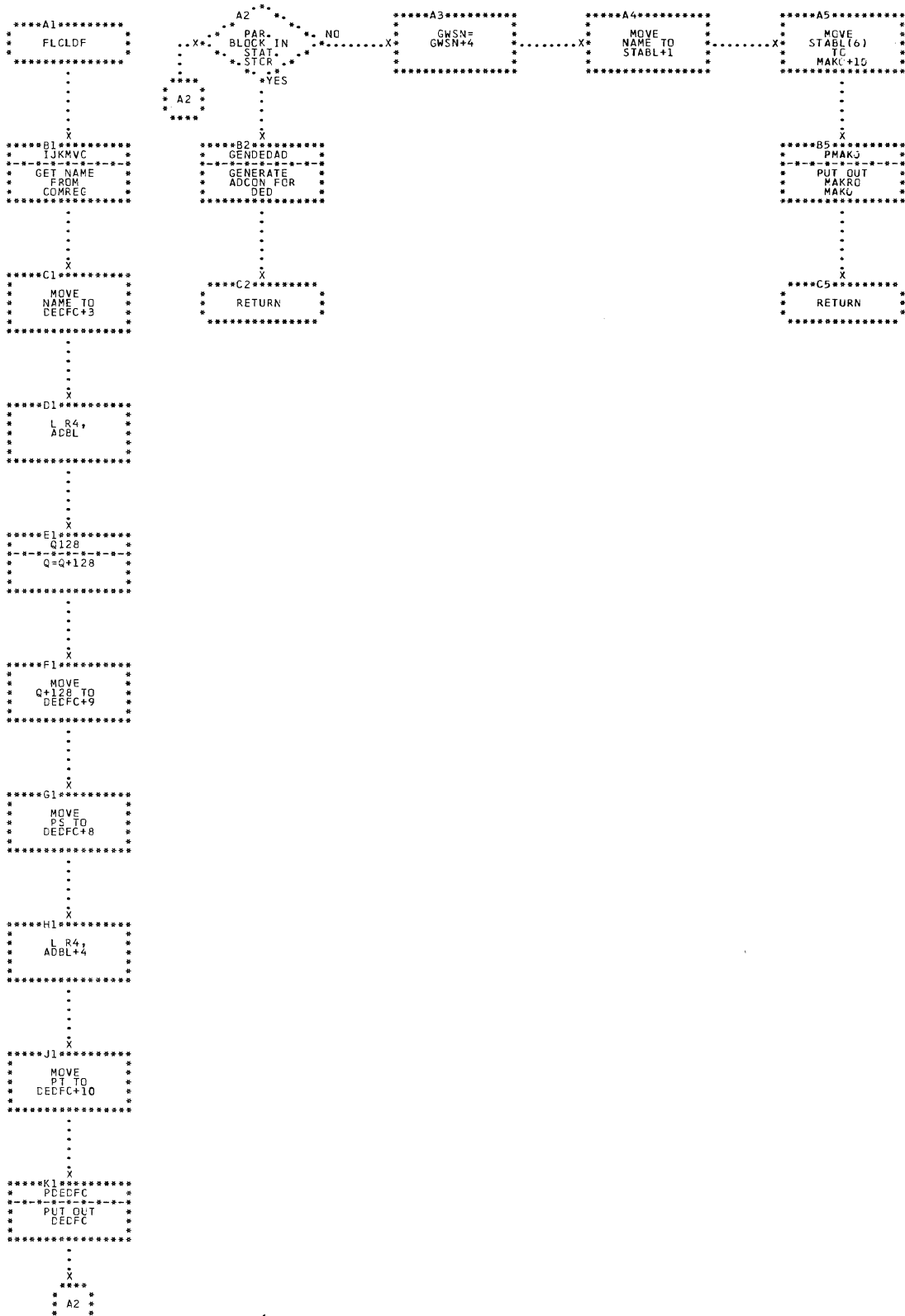
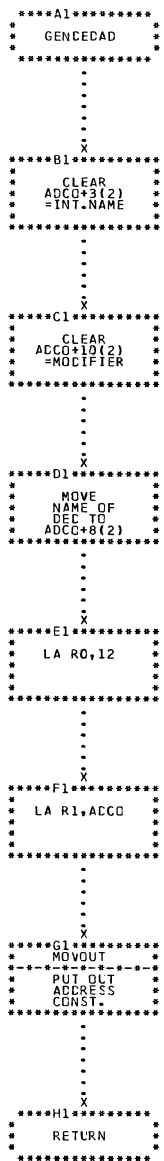


CHART NC. IJXD17

FLCLDF



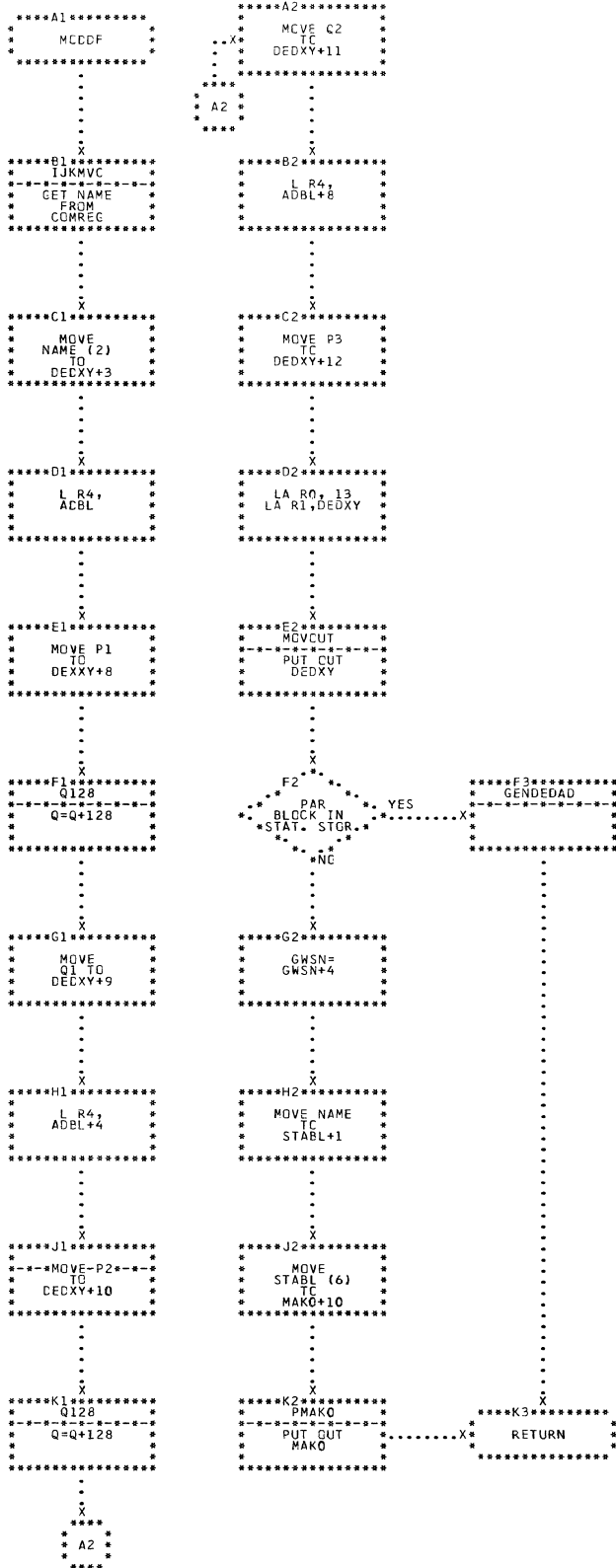
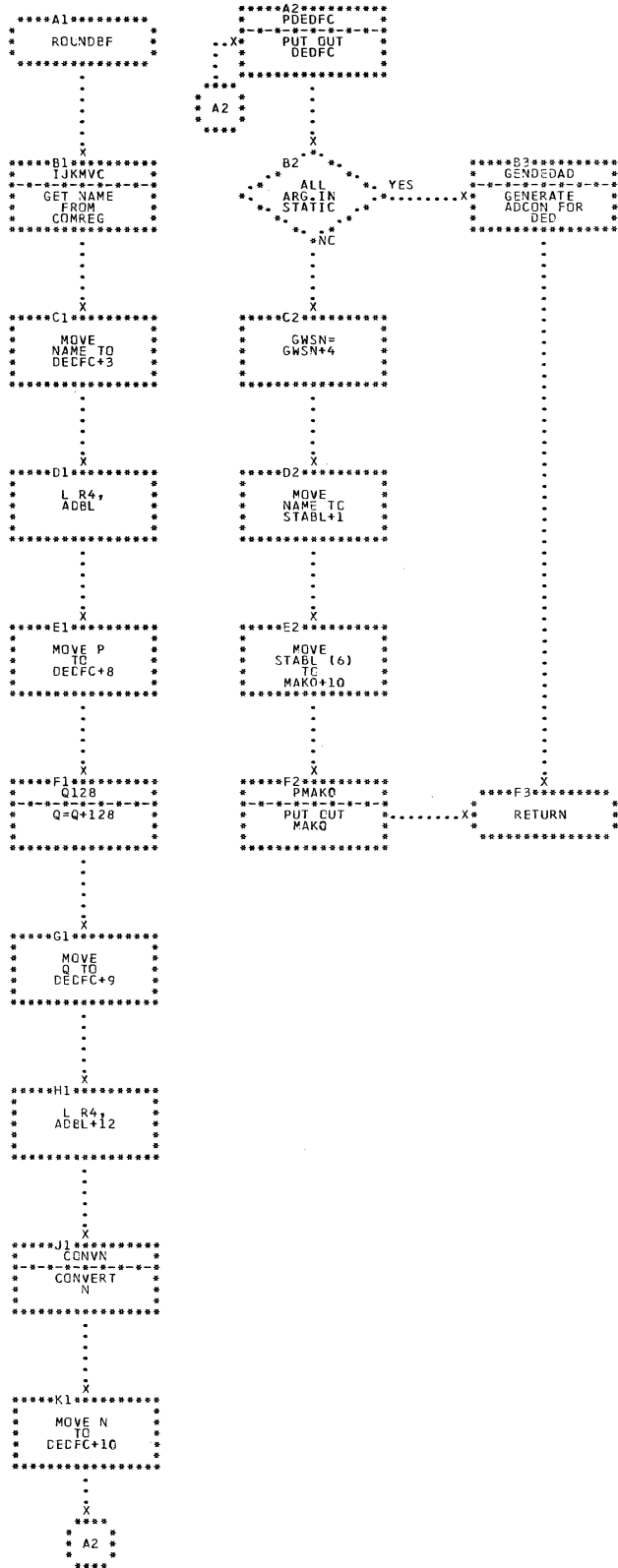
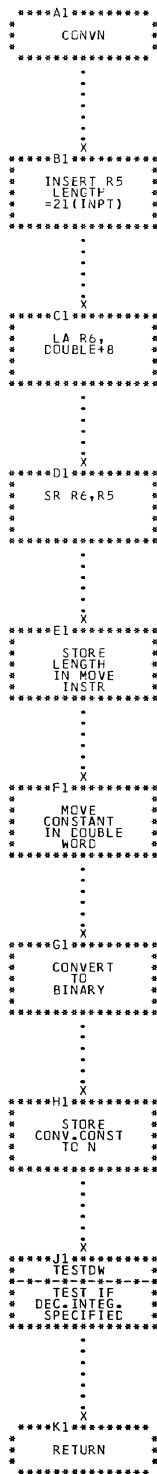
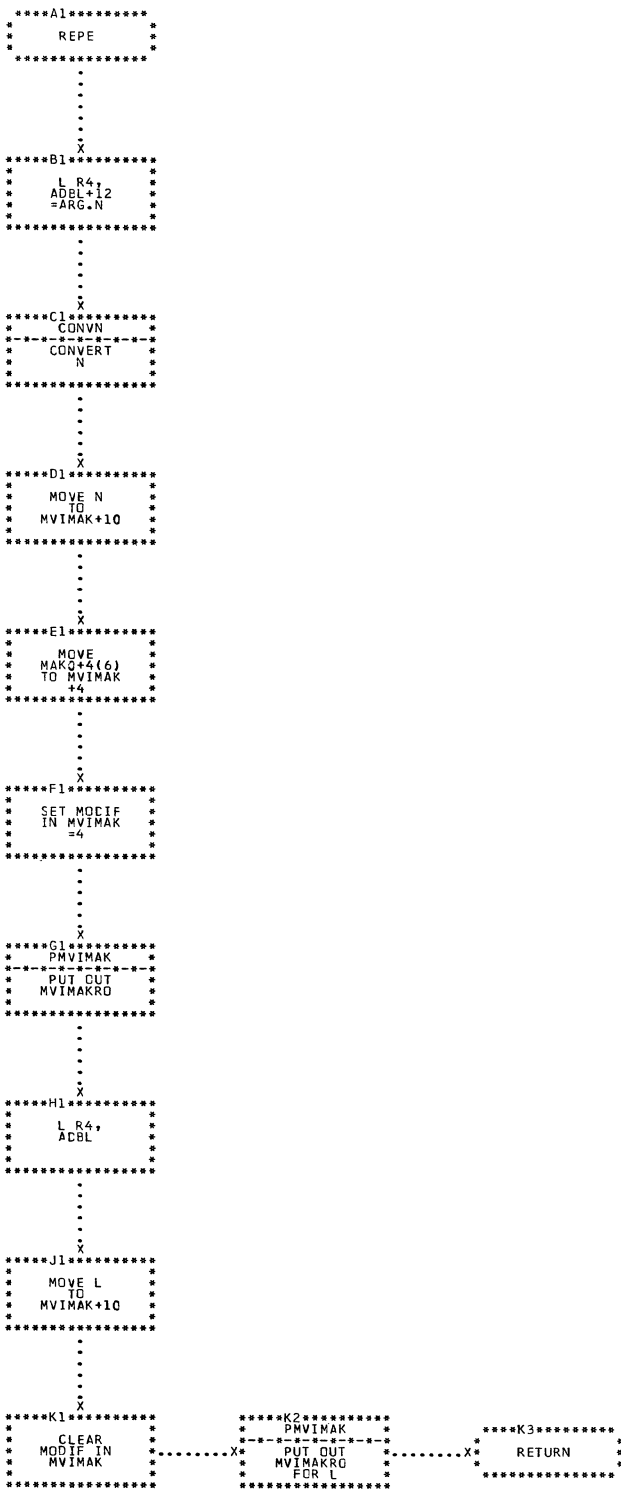


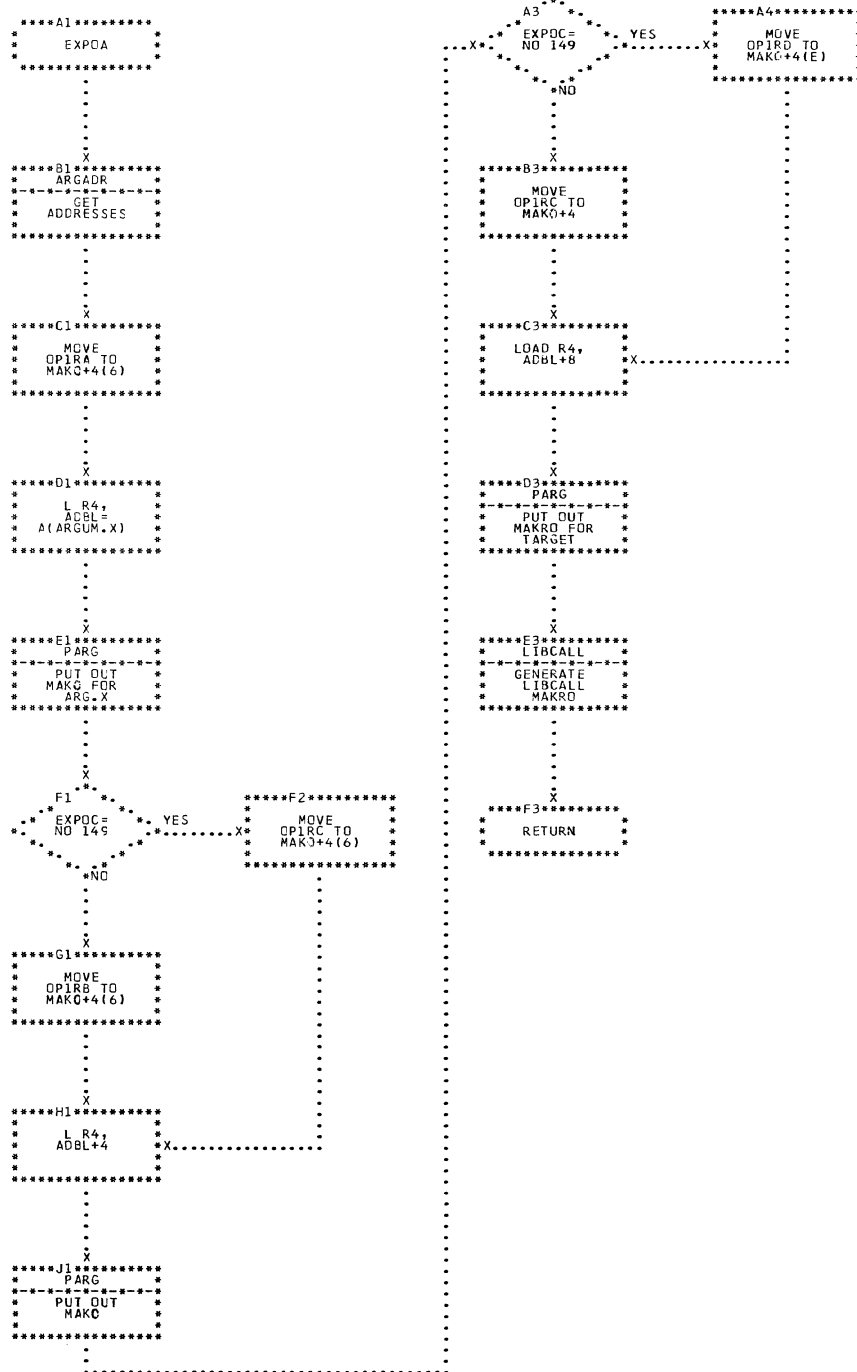
CHART NE. IJXD17

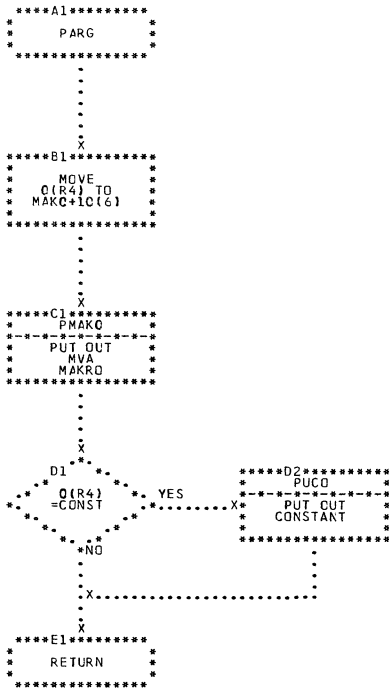
MODDF

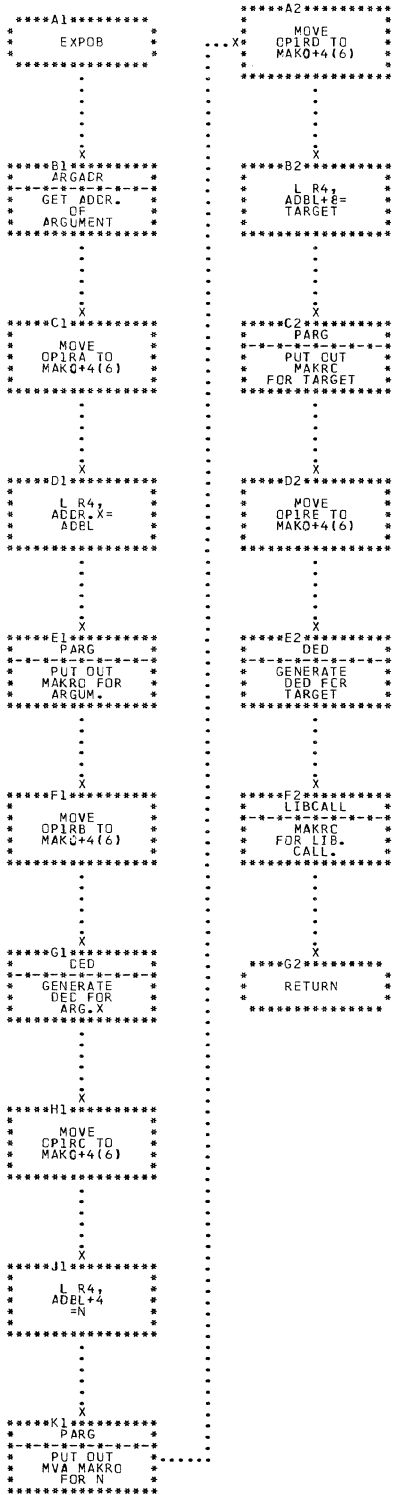












```
*****A1*****  
CED  
*****
```

.....
X

```
*****B1*****  
IJKMVC  
-----  
GET NAME  
FROM  
CDMREG  
*****
```

.....
X

```
*****C1*****  
MOVE NAME  
TO STABL+1  
*****
```

.....
X

```
*****D1*****  
MOVE  
STABL TO  
MAKO+1C(6)  
*****
```

.....
X

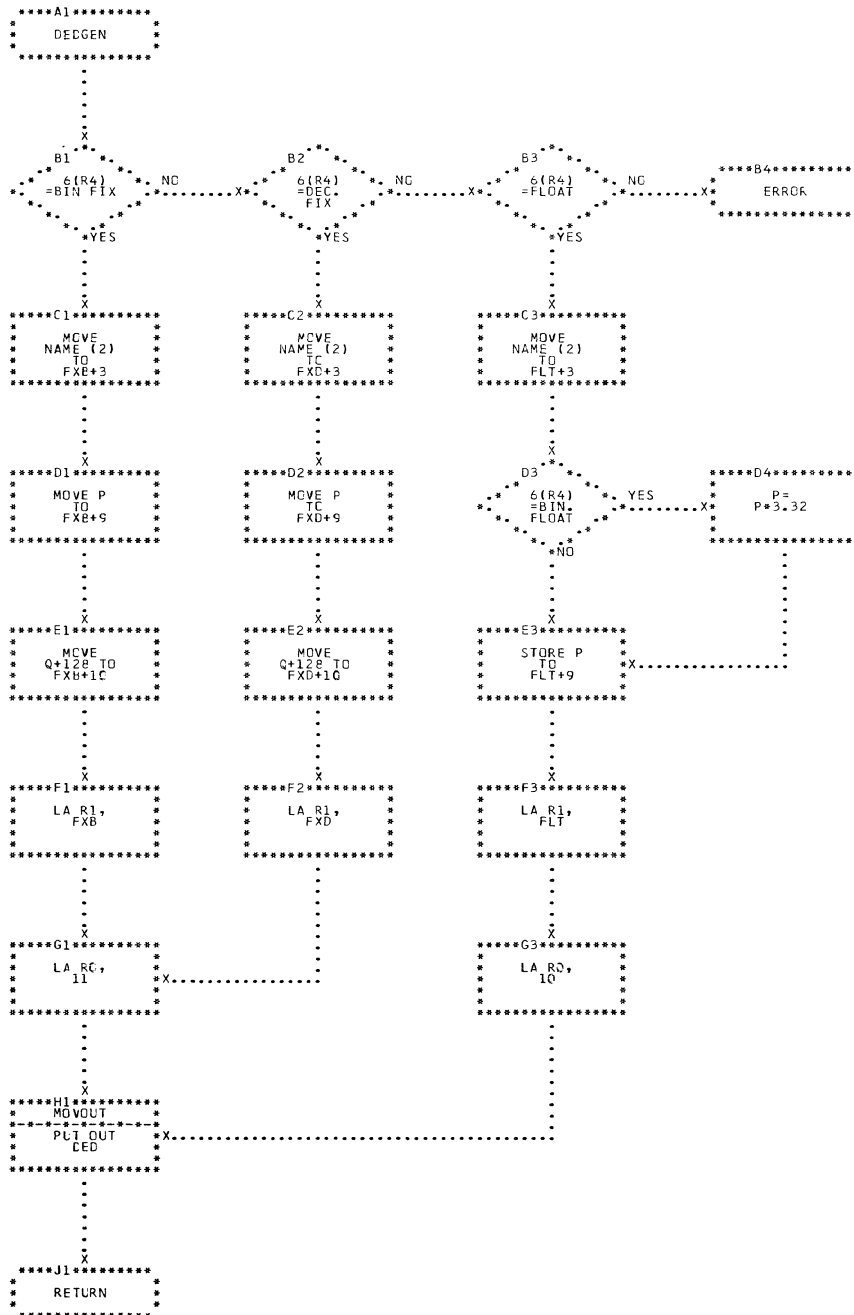
```
*****E1*****  
PMAKC  
-----  
PUT OUT  
MVA MAKRO  
FOR DED  
*****
```

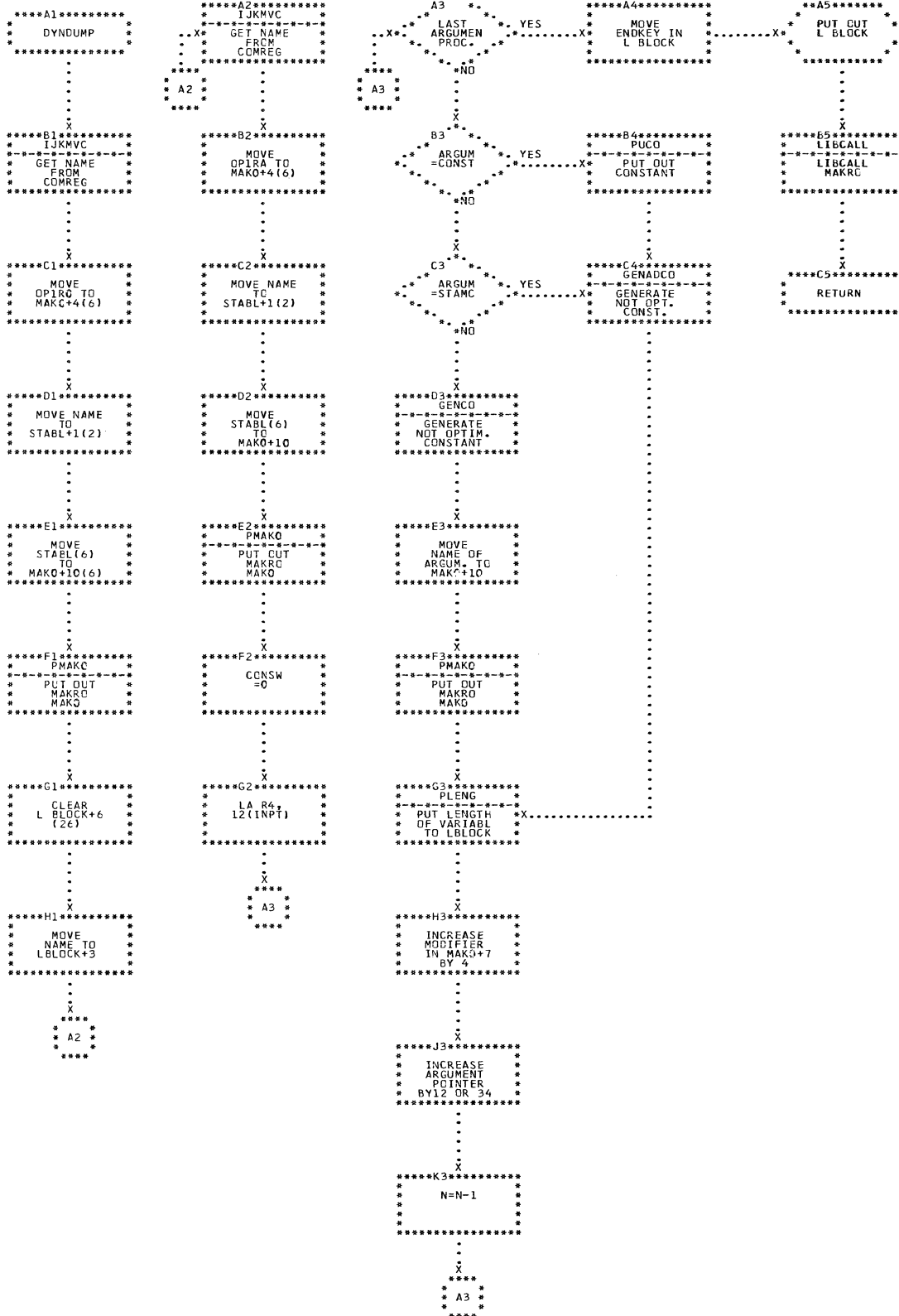
.....
X

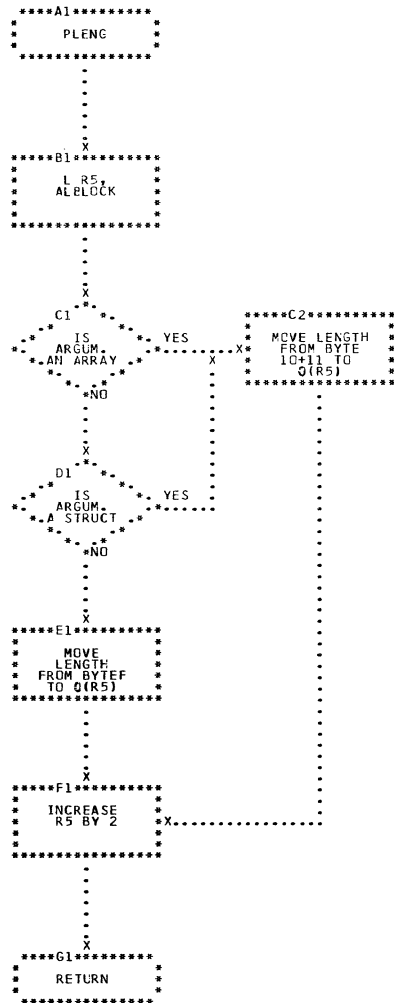
```
*****F1*****  
DECGEN  
-----  
GENERATE  
CED  
*****
```

.....
X

```
*****G1*****  
RETURN  
*****
```



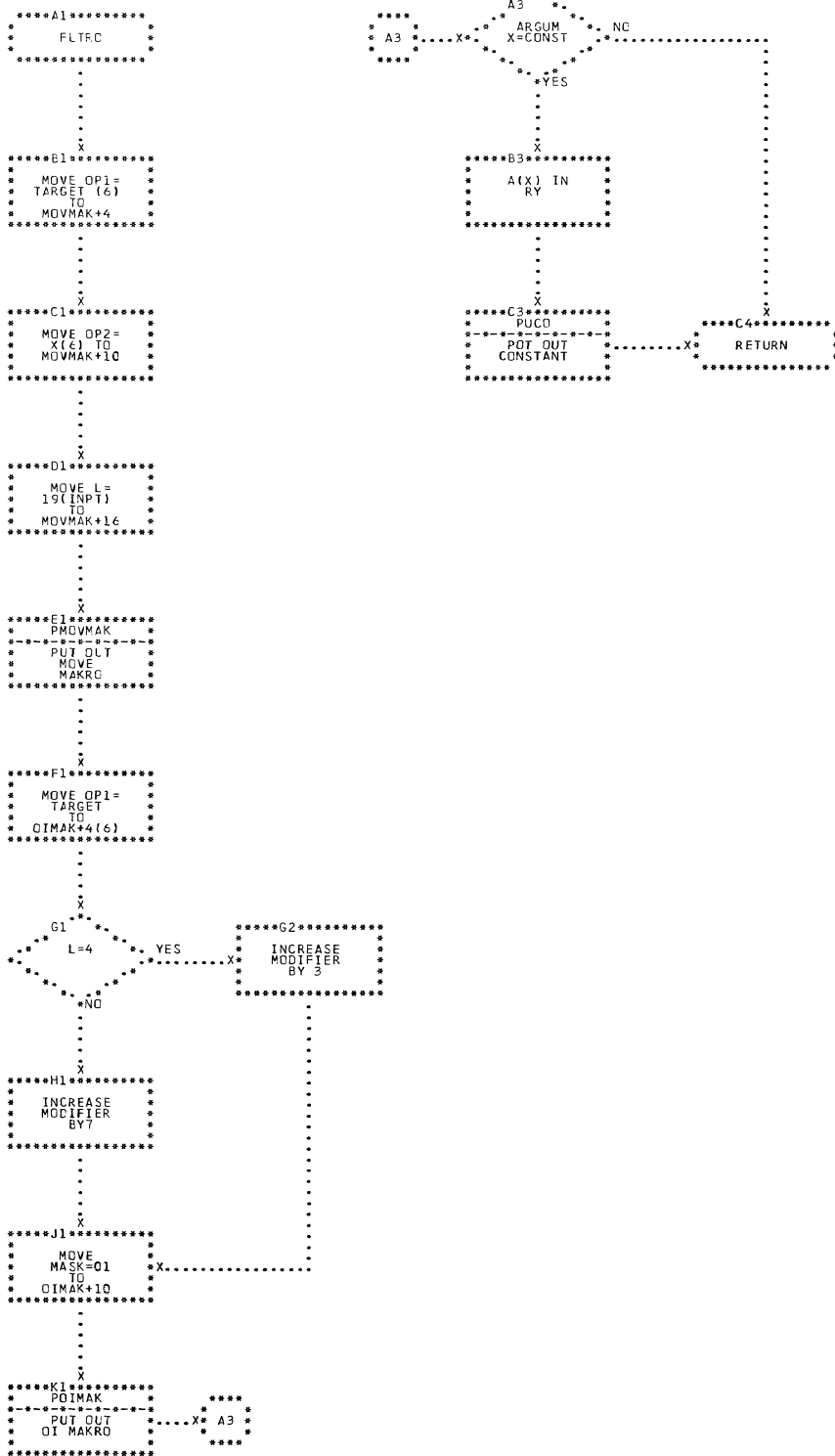


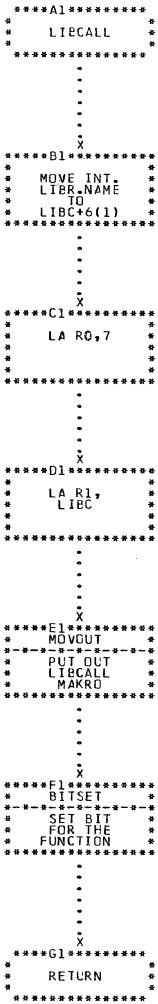


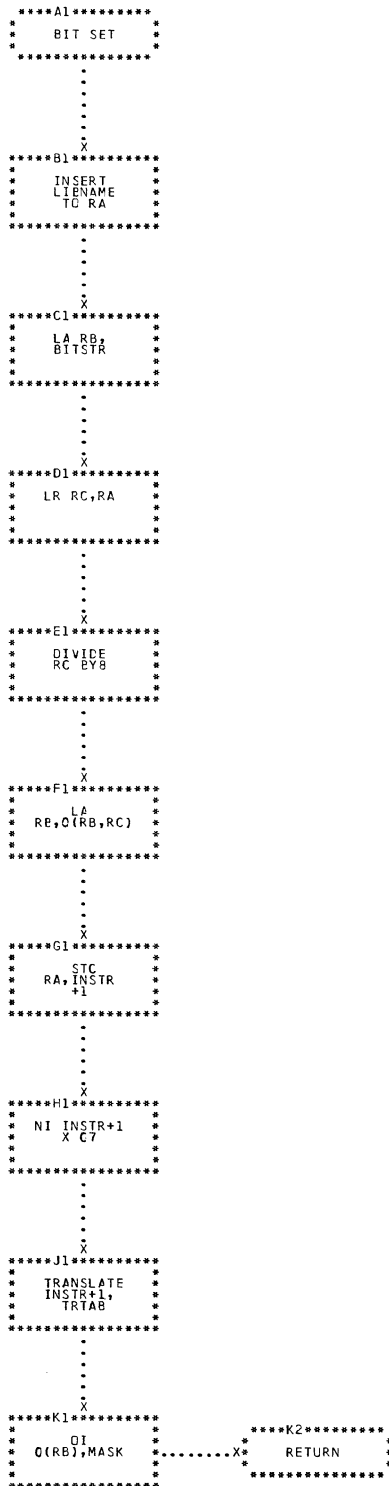
```

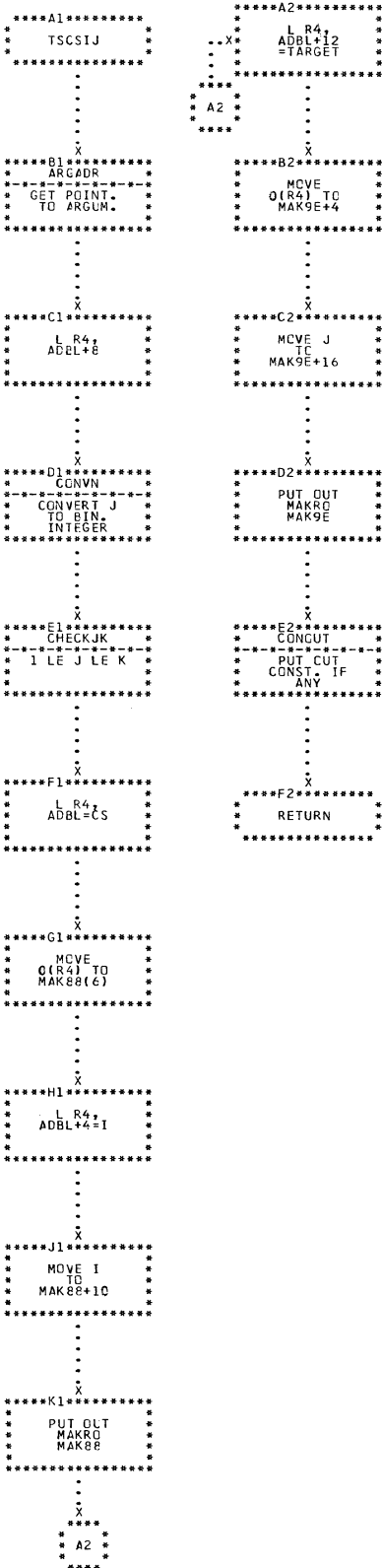
*****A1*****
TIMDAT
.....
X
*****B1*****
MOVE
OPRA TO
MARCB+4(6)
.....
X
*****C1*****
MOVE
12(INPT) TO
MAKO+10(6)
.....
X
*****D1*****
PMAKG
-----
PUT OUT
MAKRC
MAKC
.....
X
*****E1*****
LIBCALL
-----
GENERATE
LIBCALL
MACRO
.....
X
*****F1*****
RETURN
.....

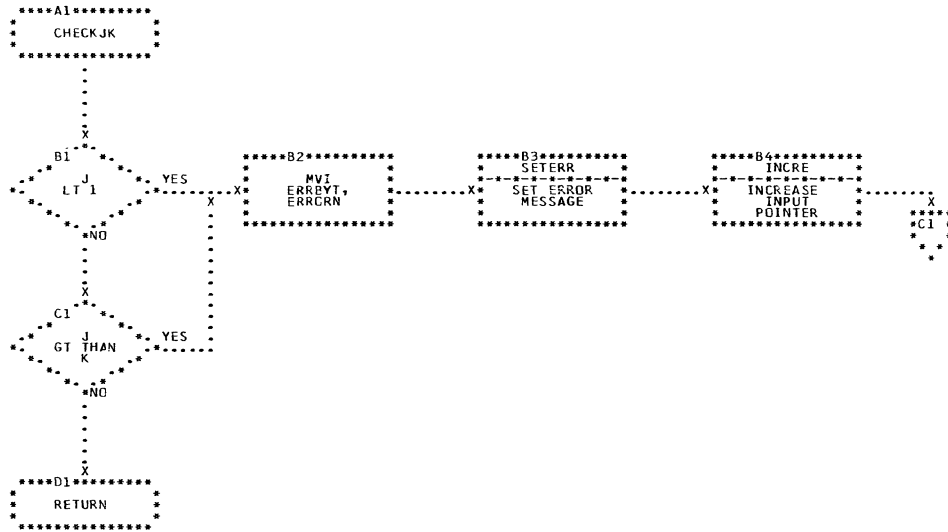
```

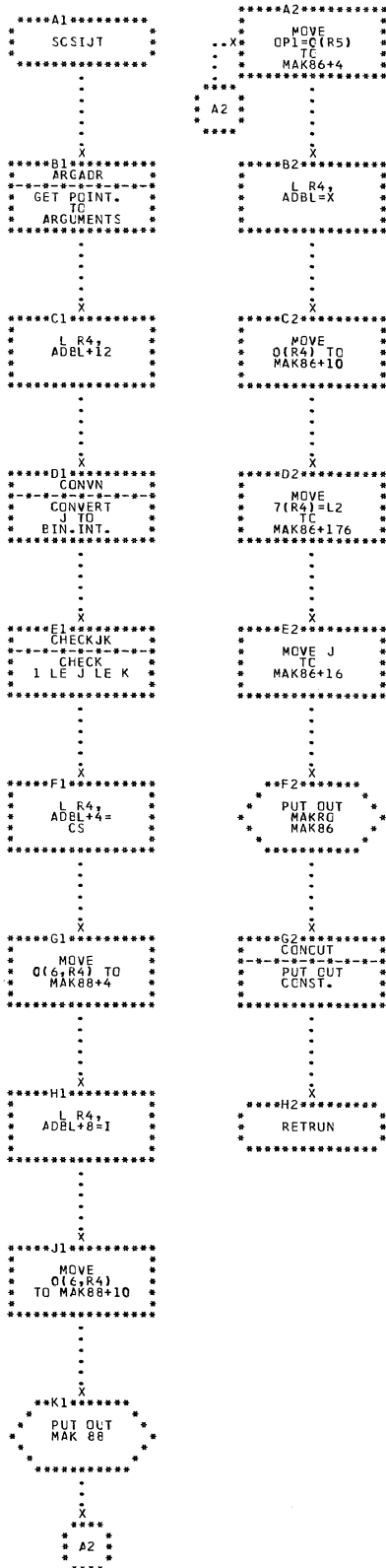












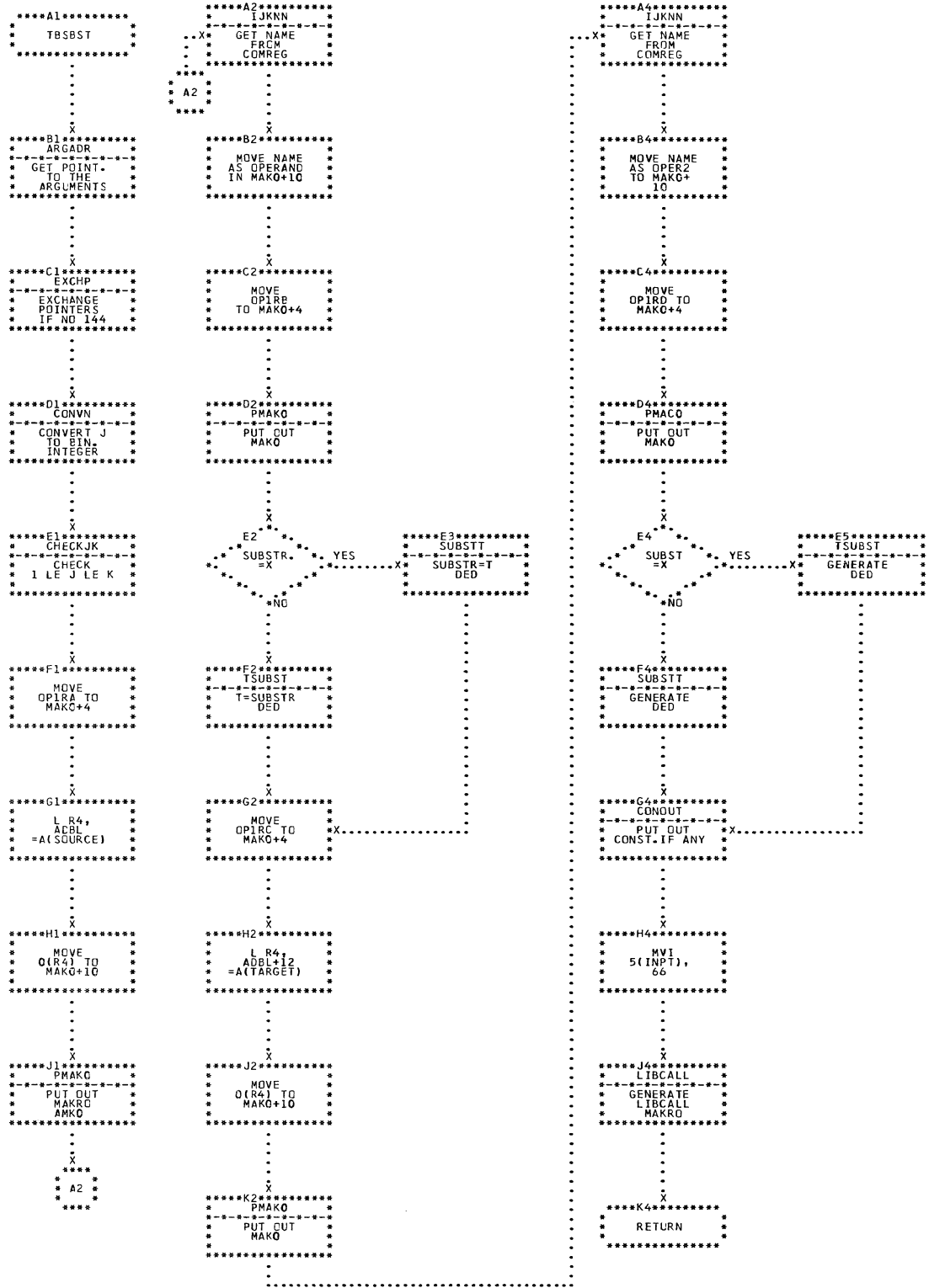
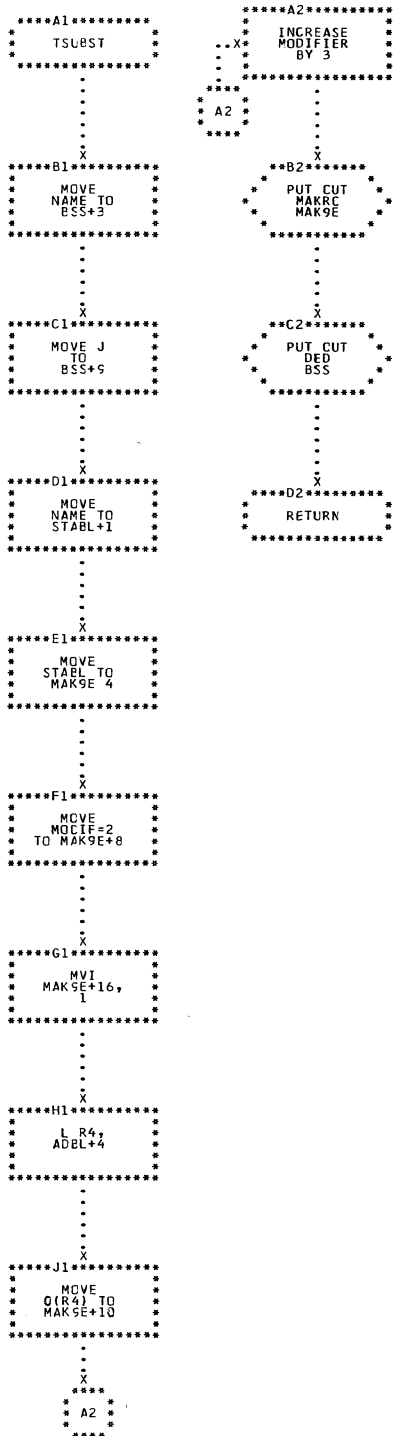
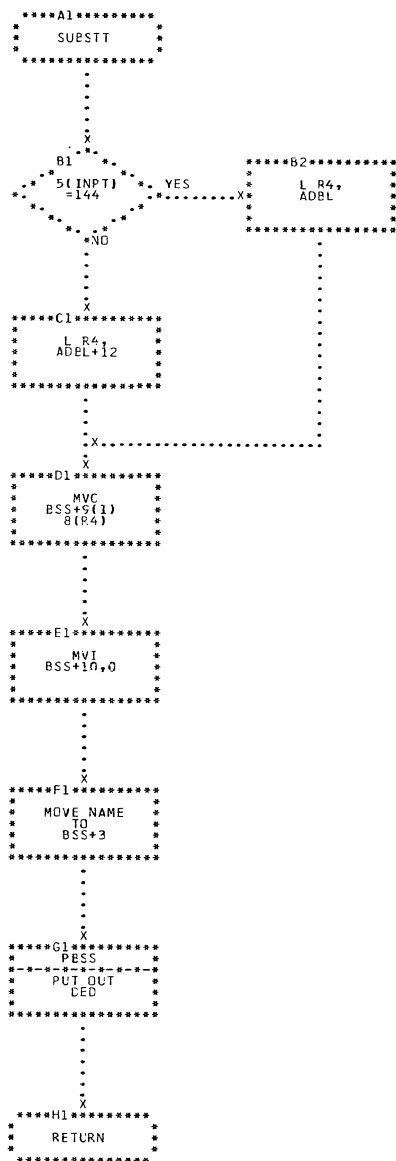
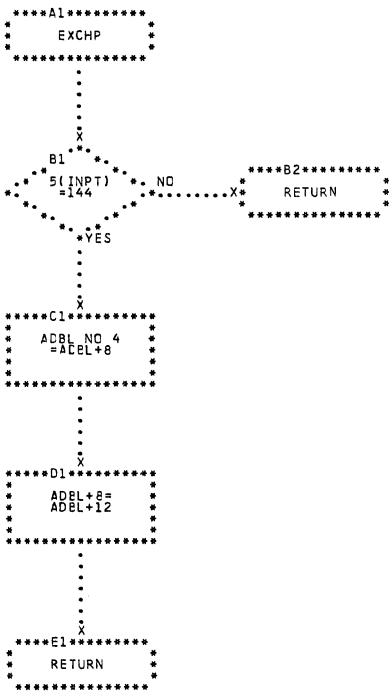
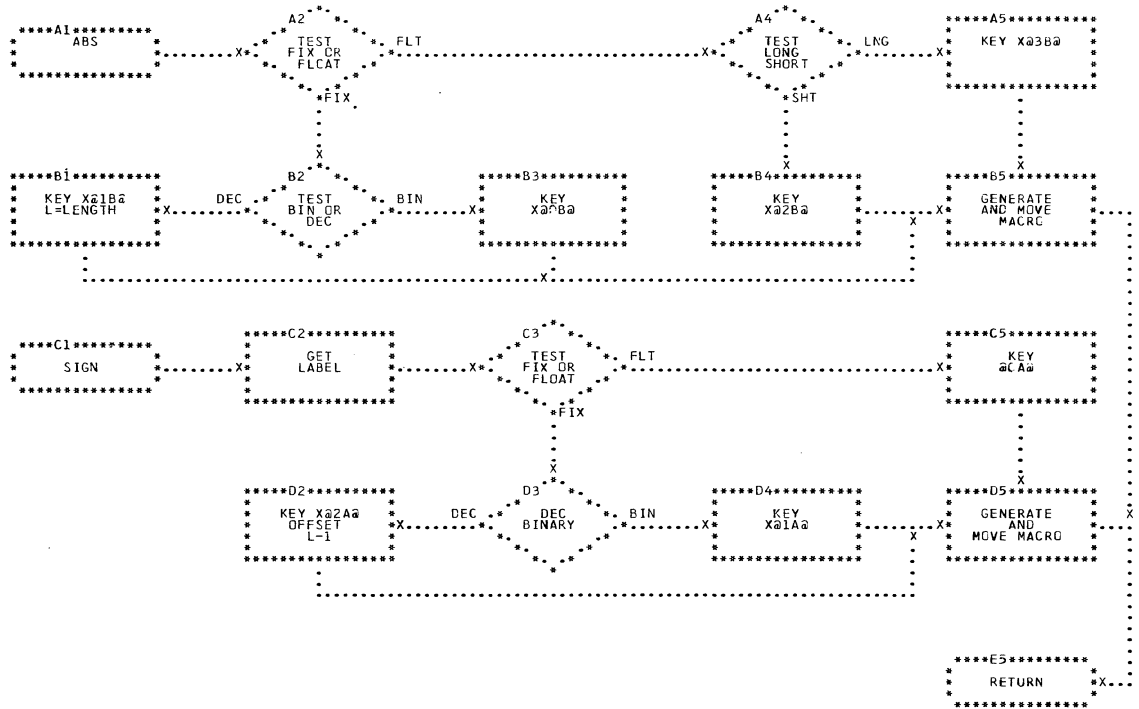


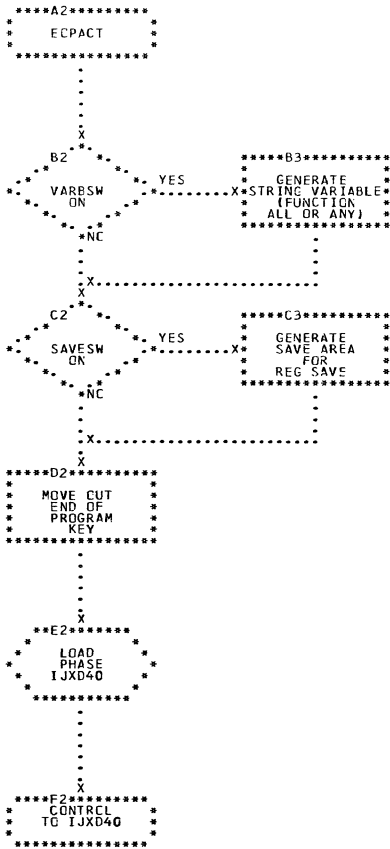
CHART NW. IJXD17 TBSBST

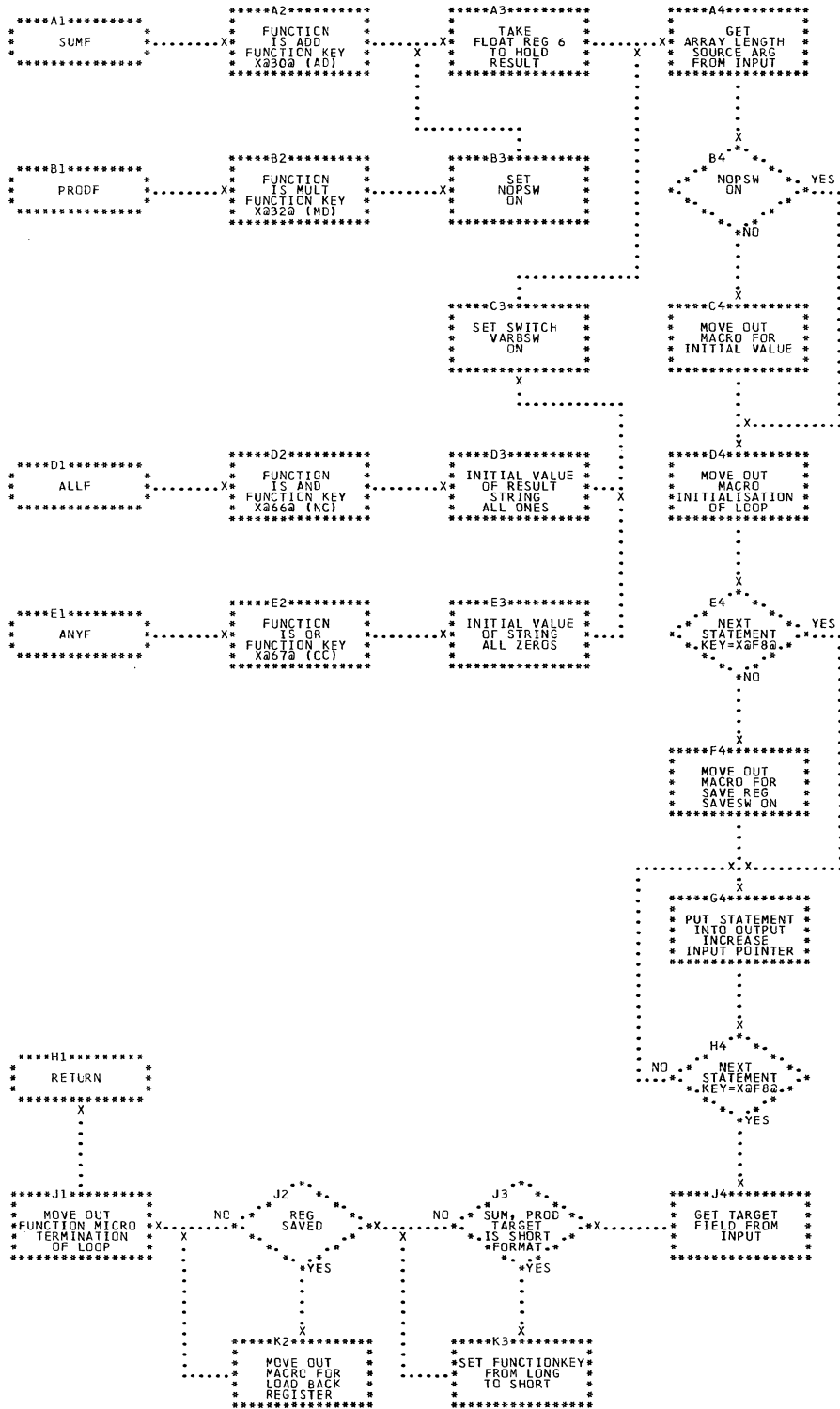


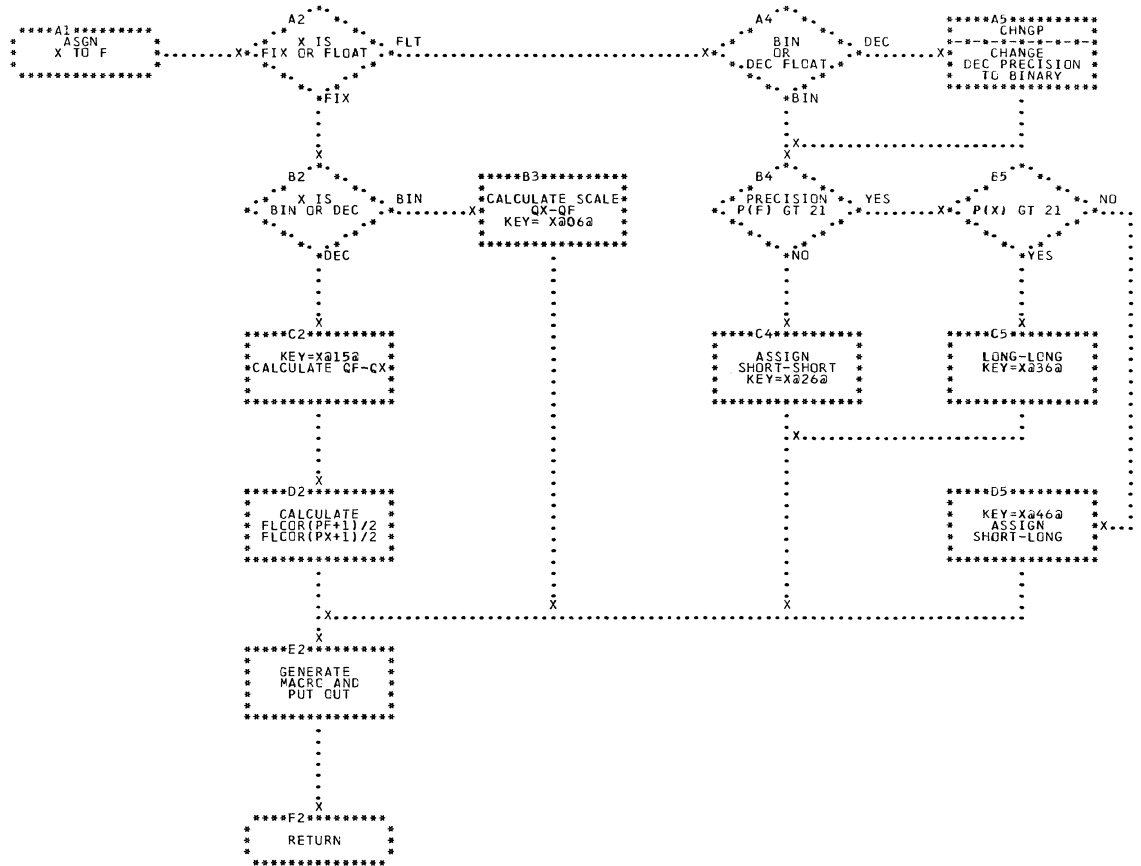


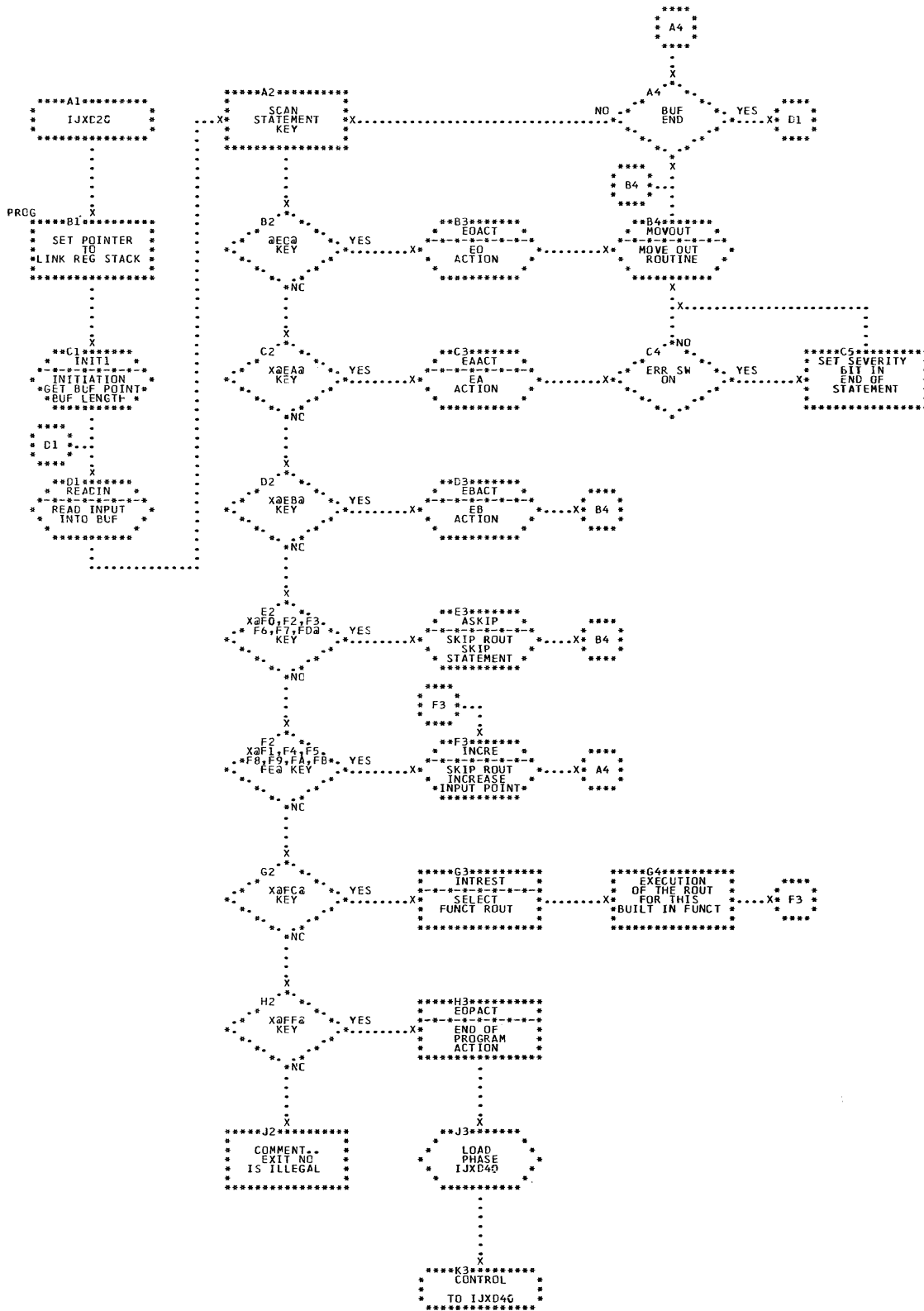


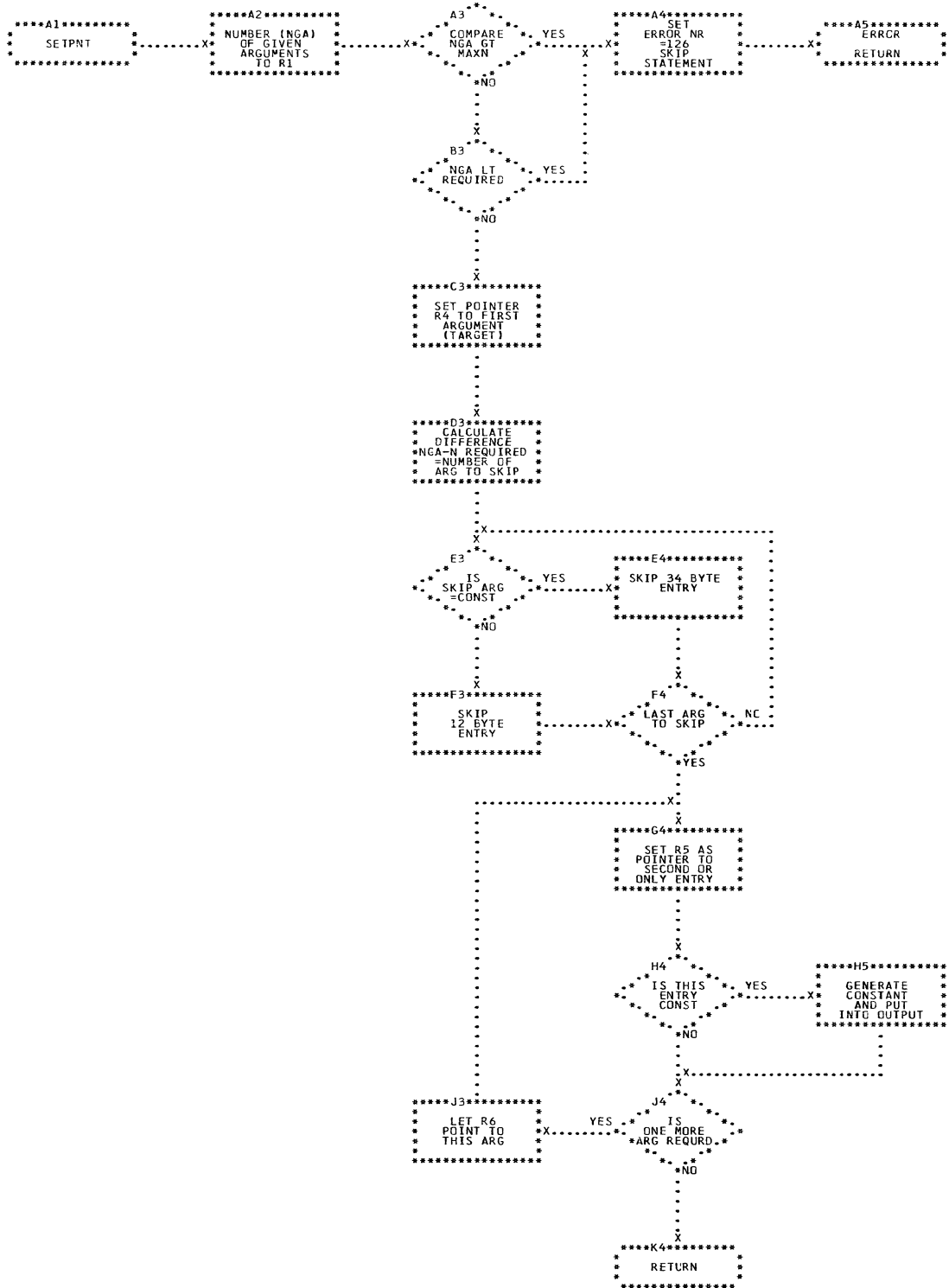


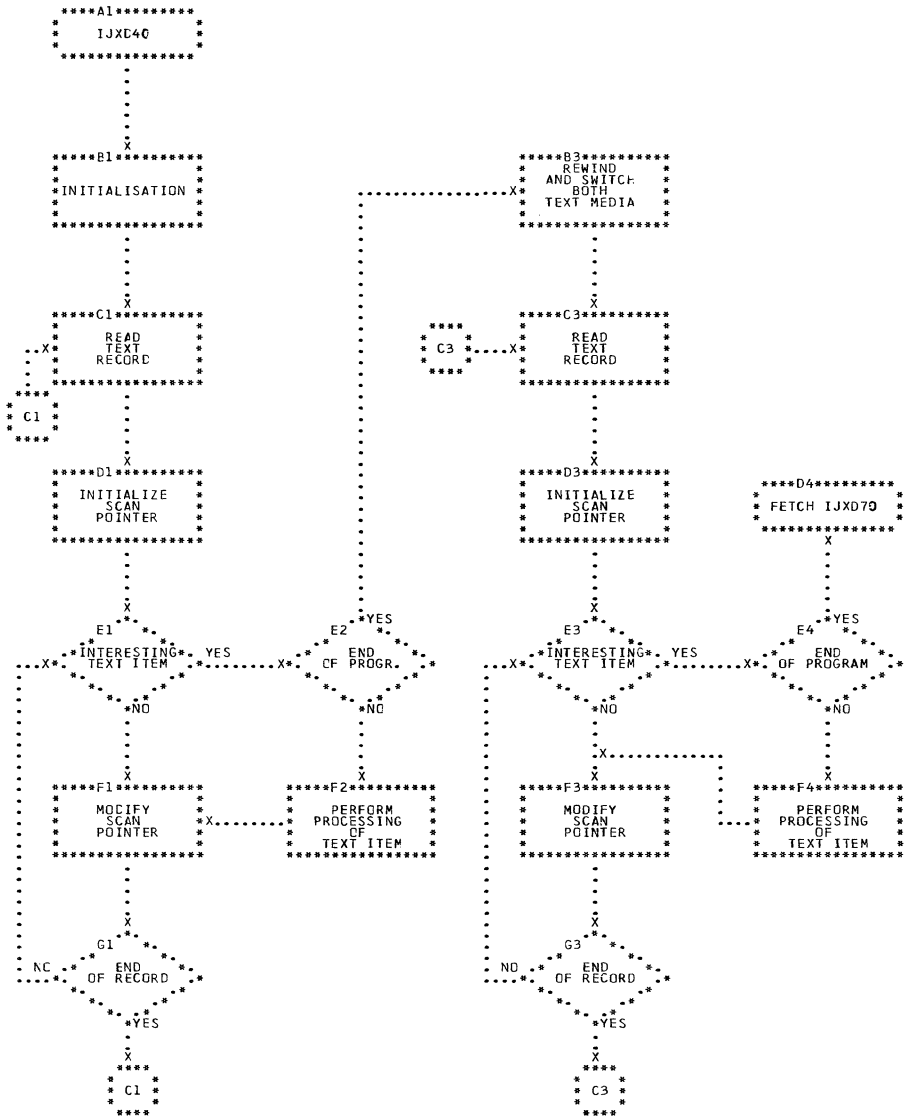












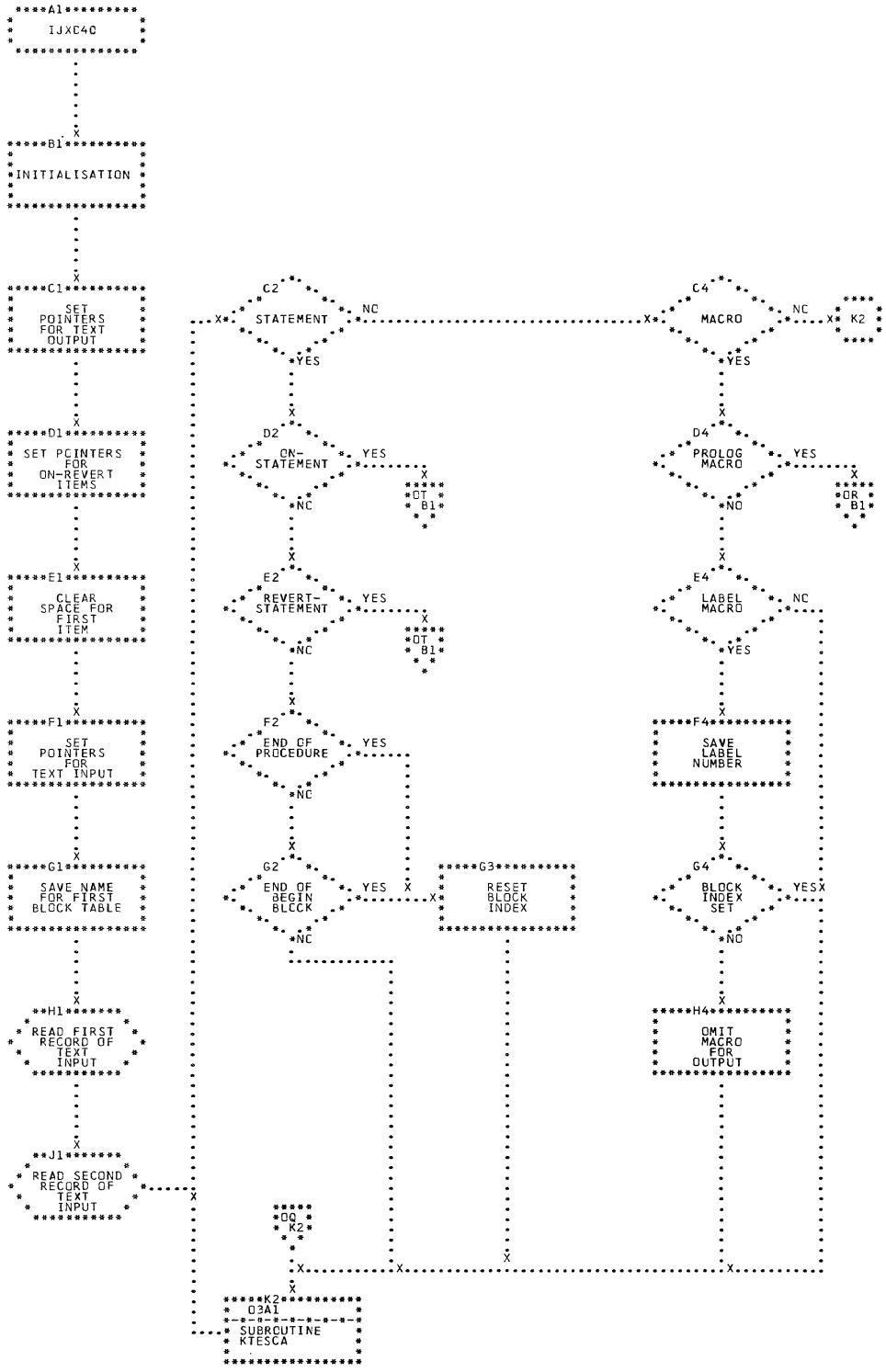
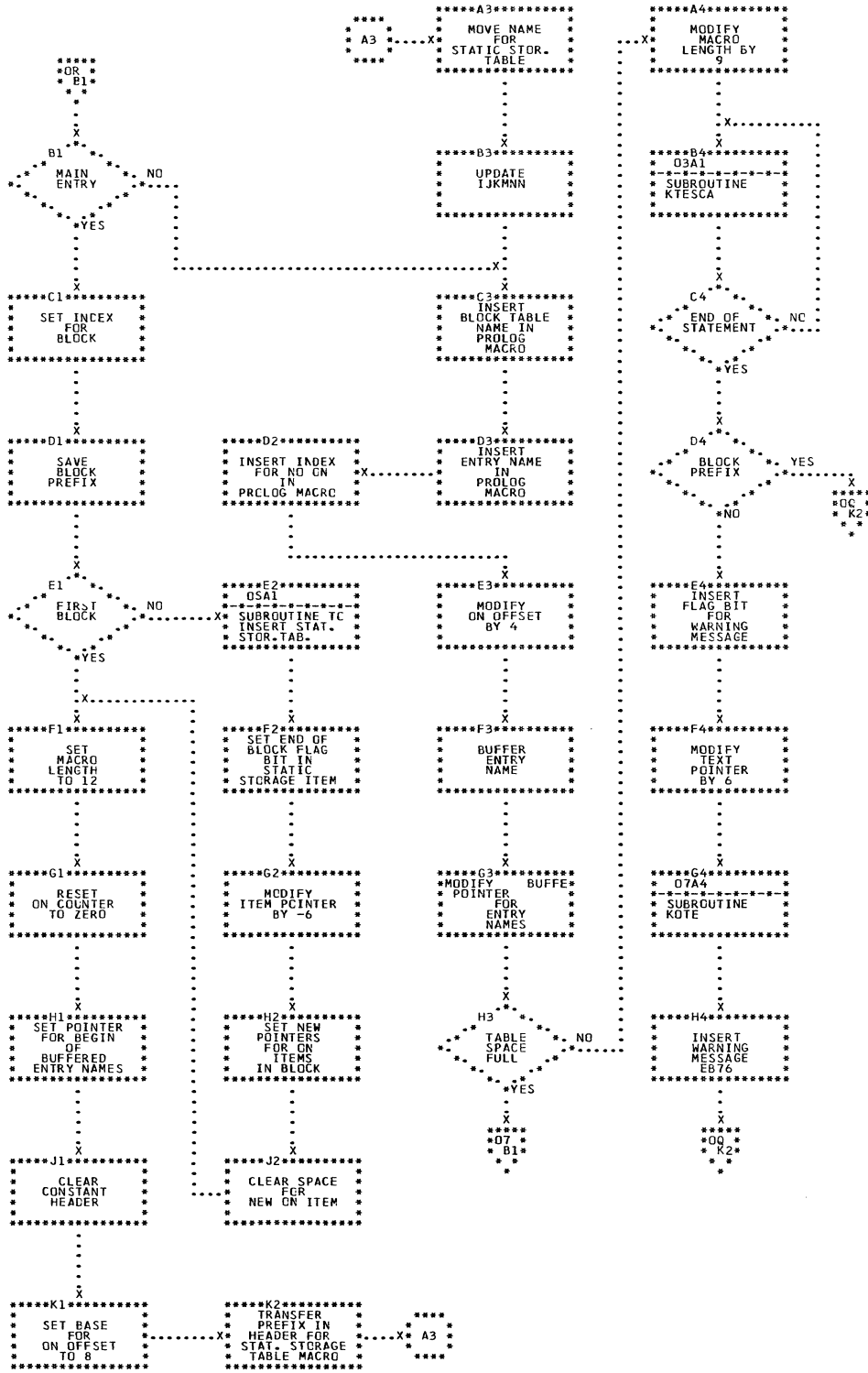
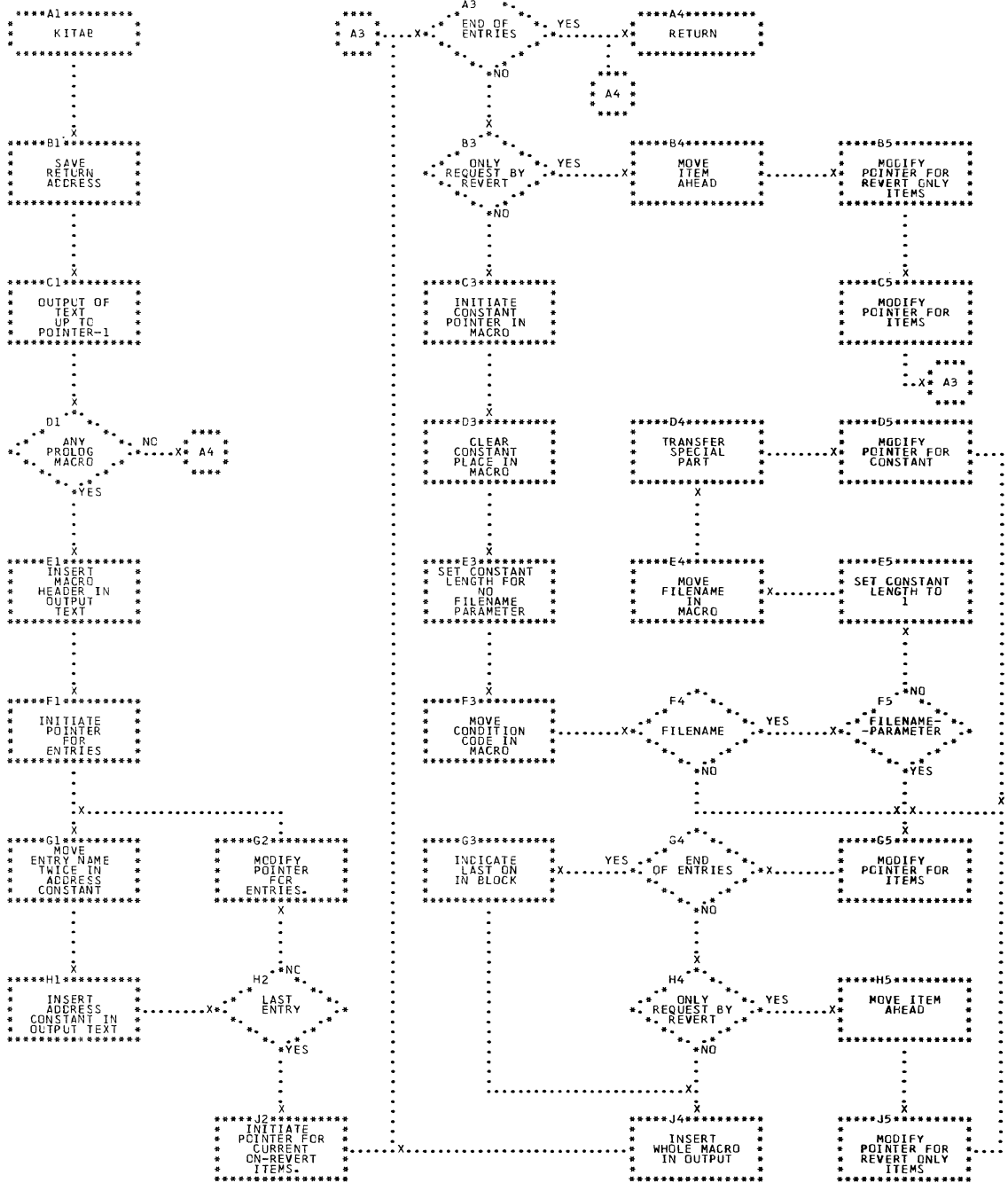
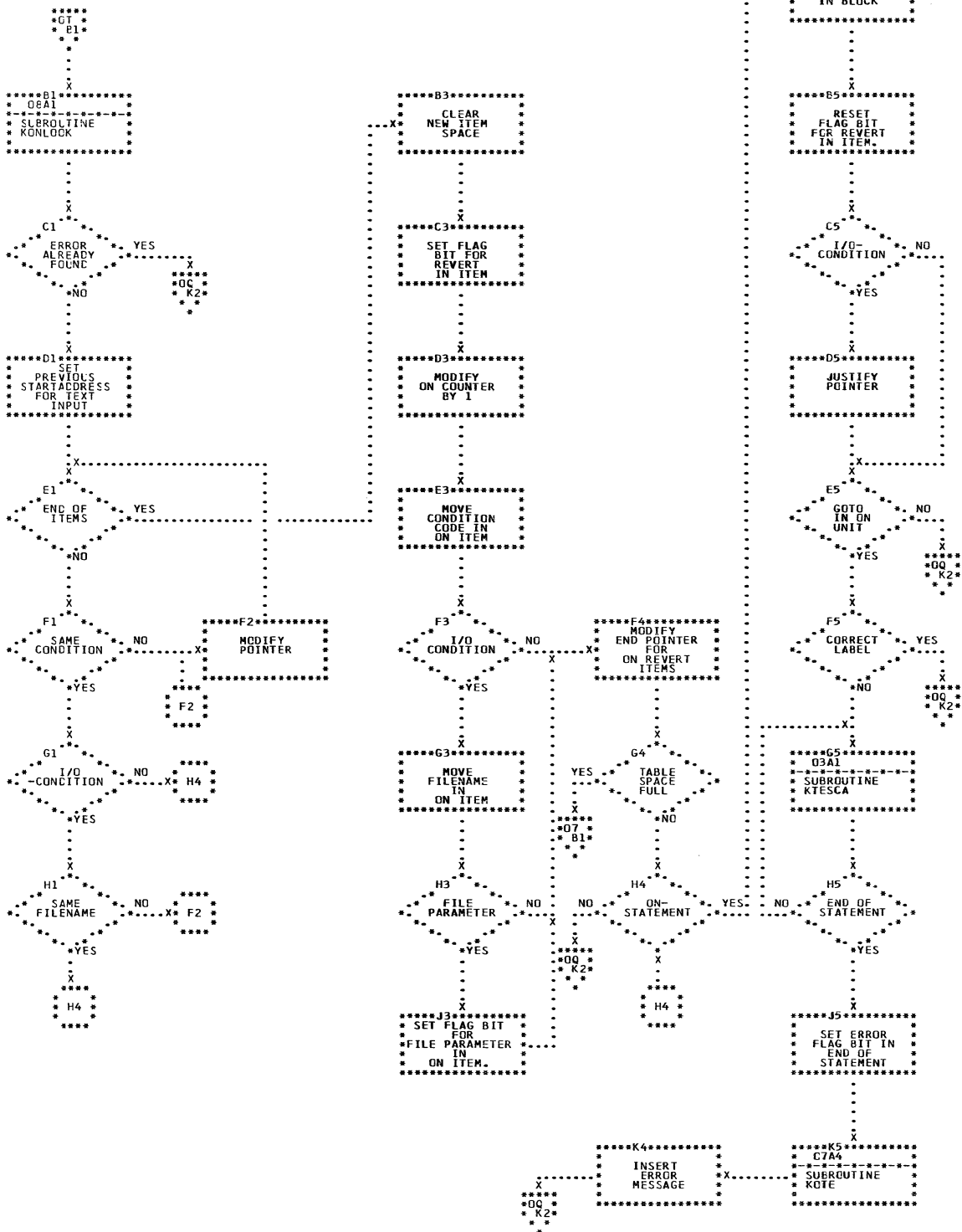


CHART 00. IJXD40 1. SCAN, MAIN PART







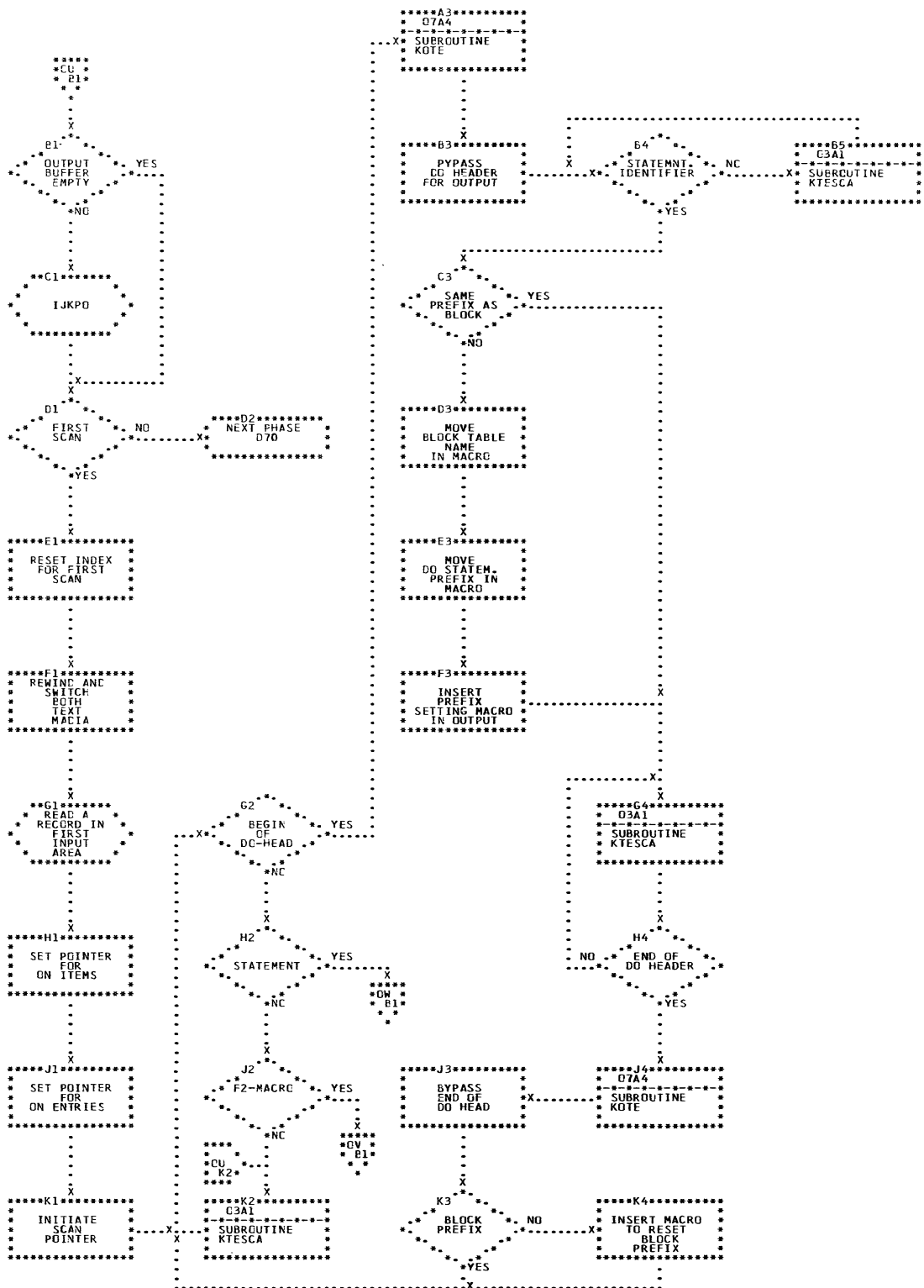
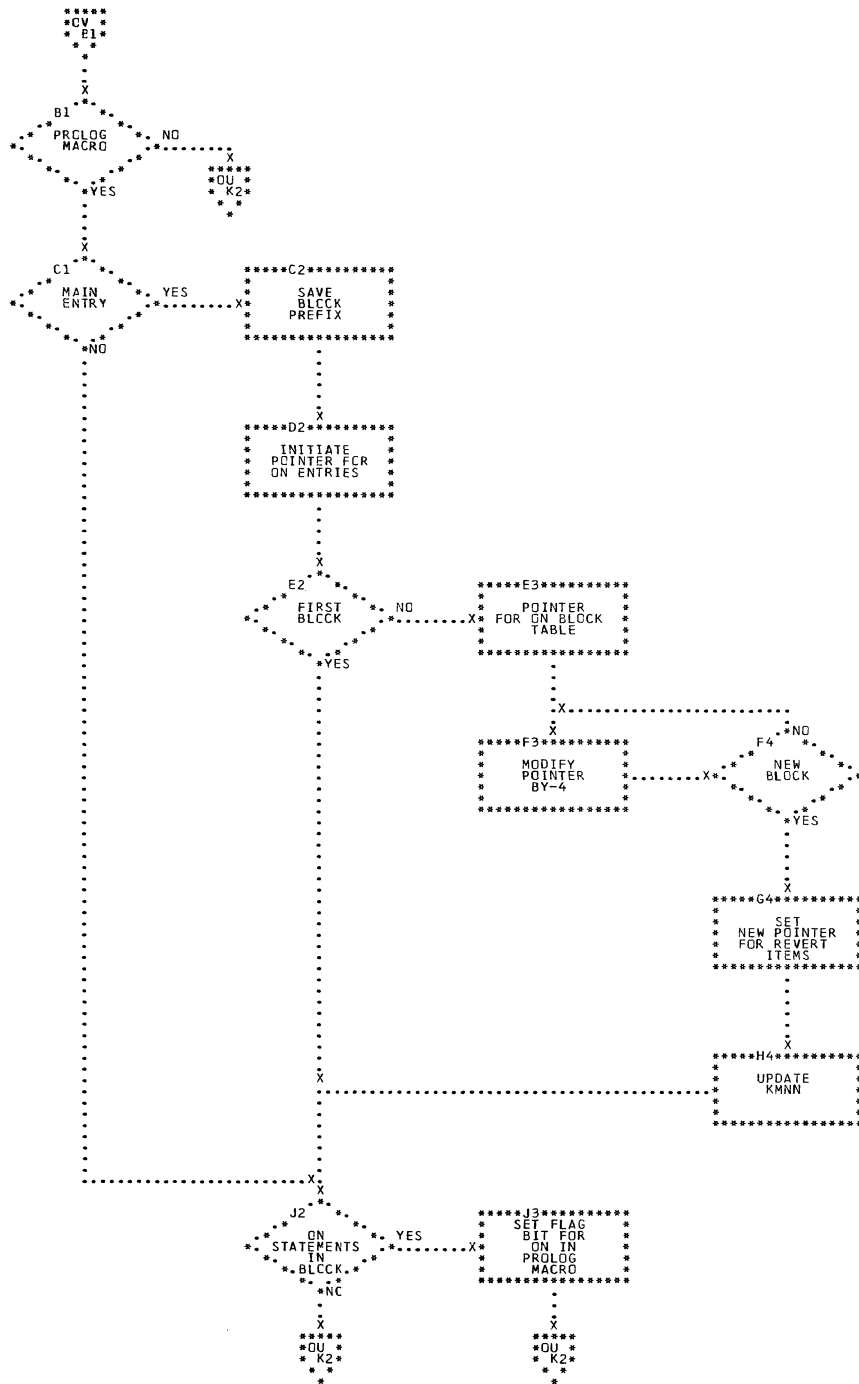
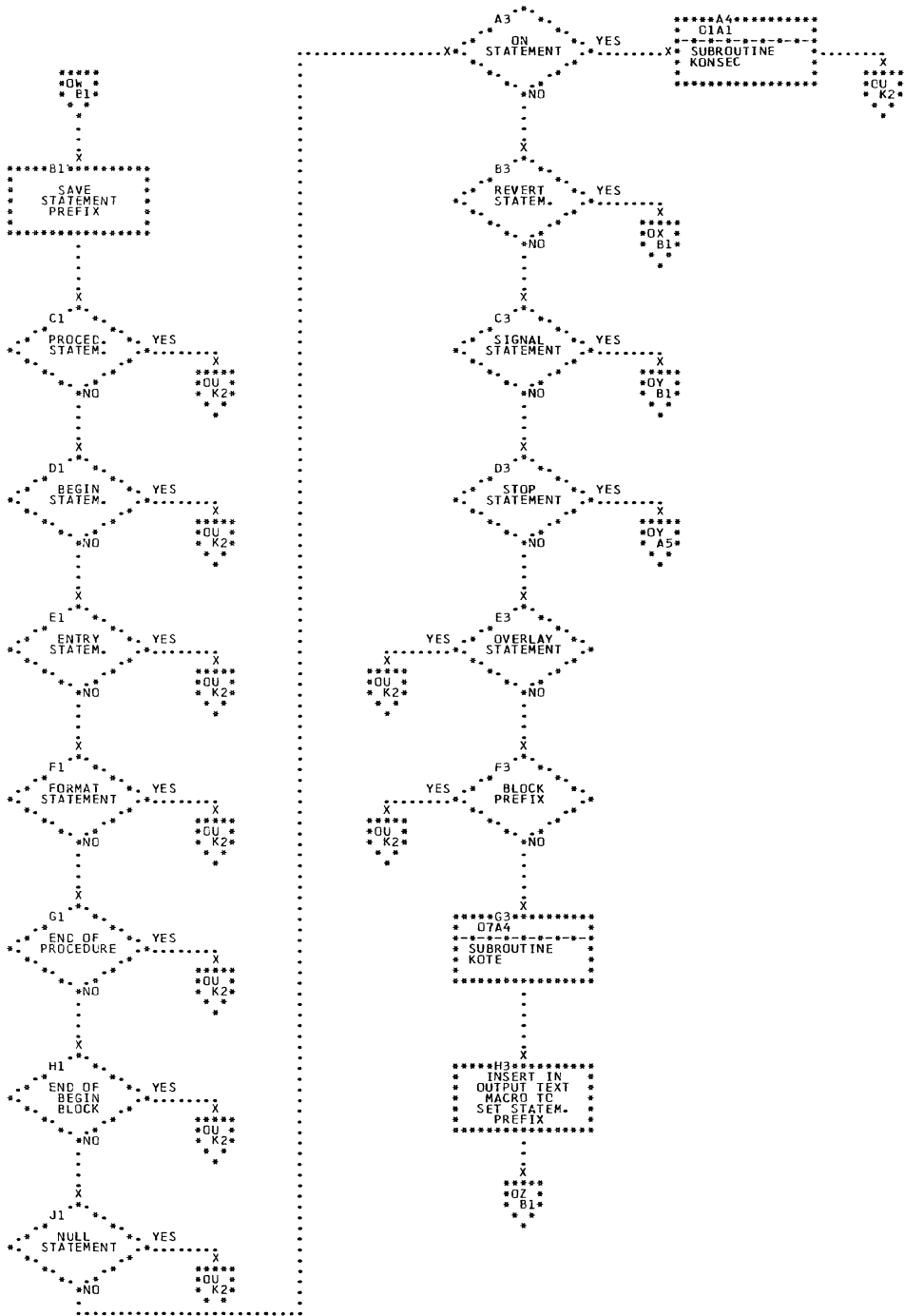
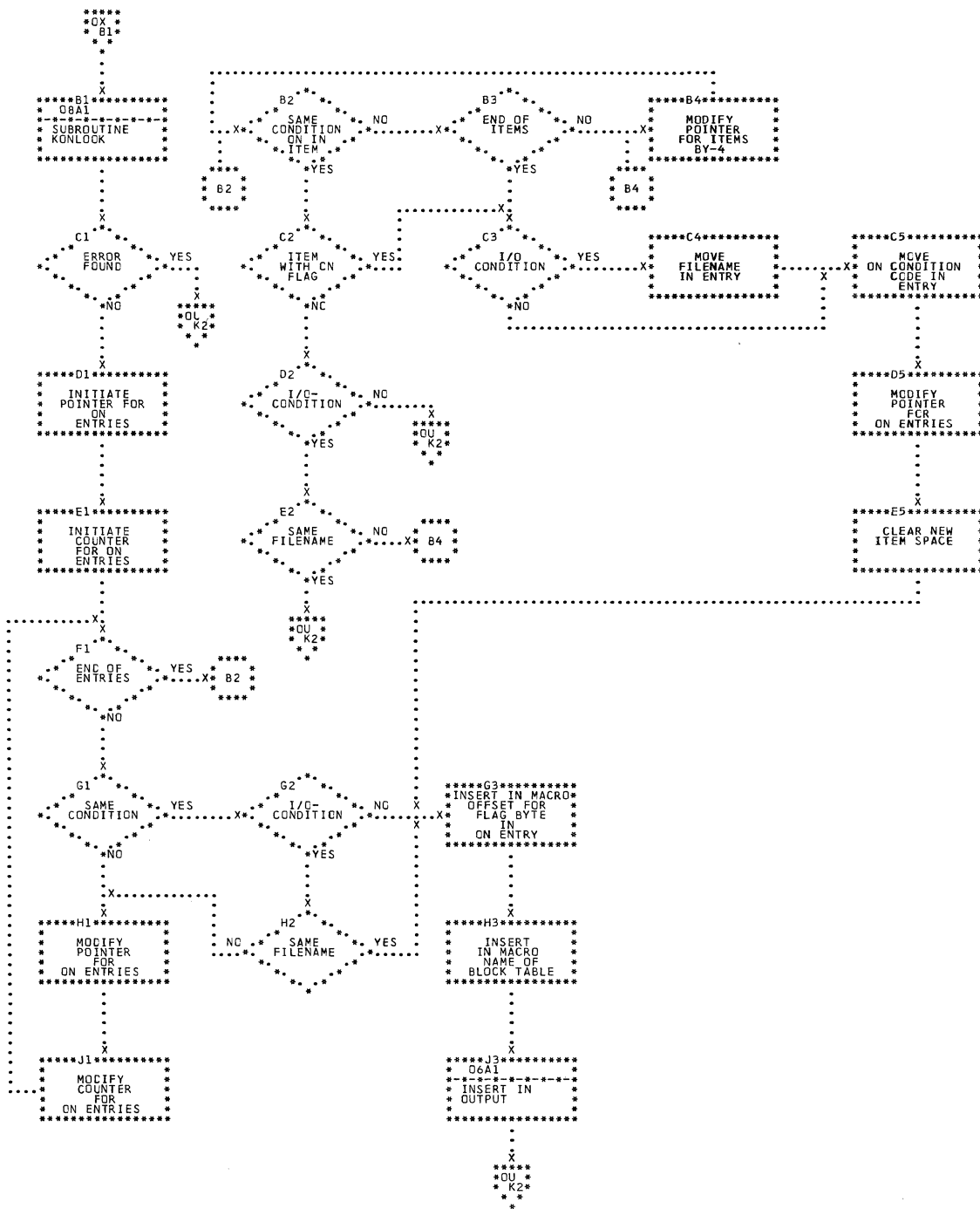


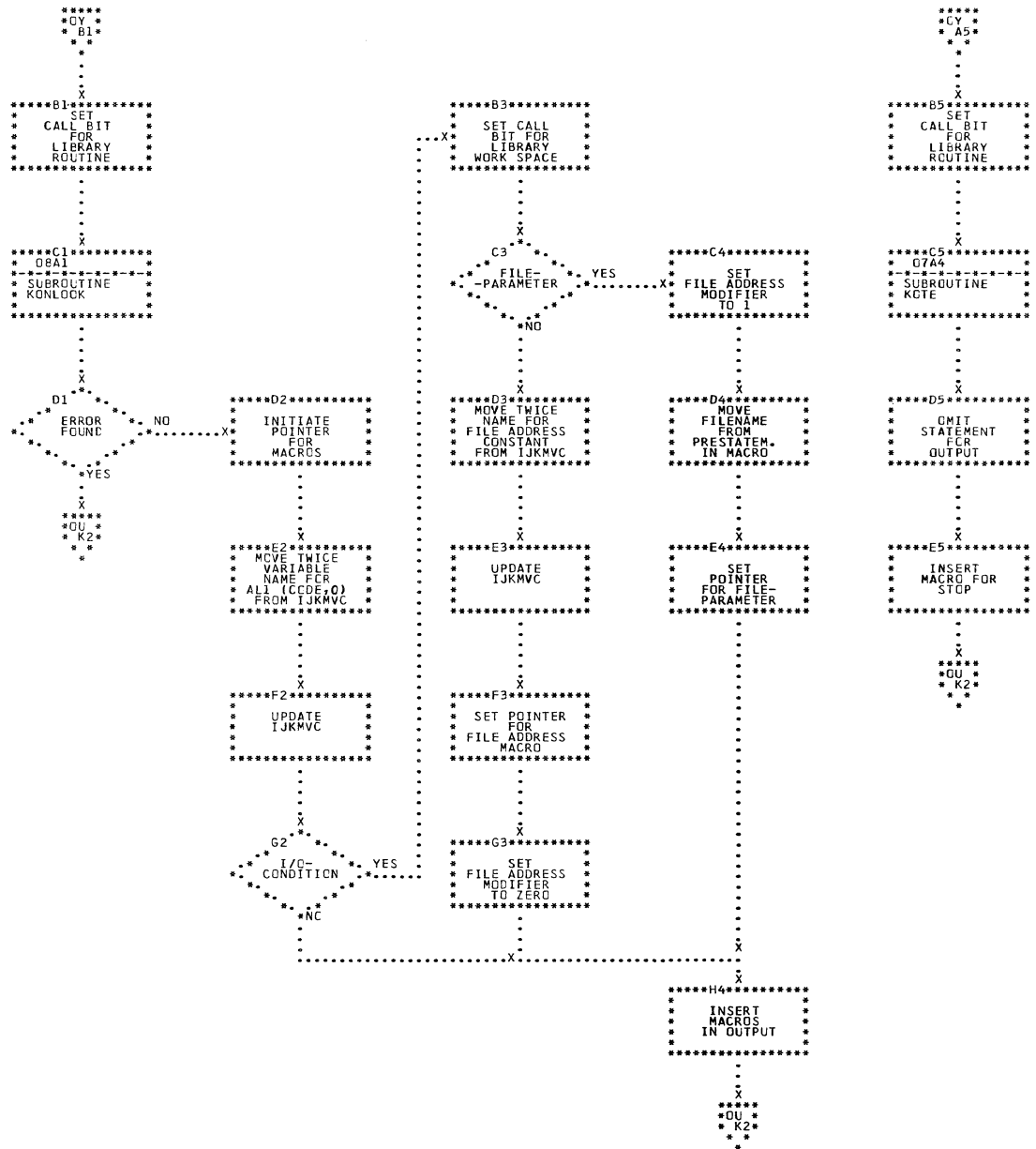
CHART 00. IJXD40

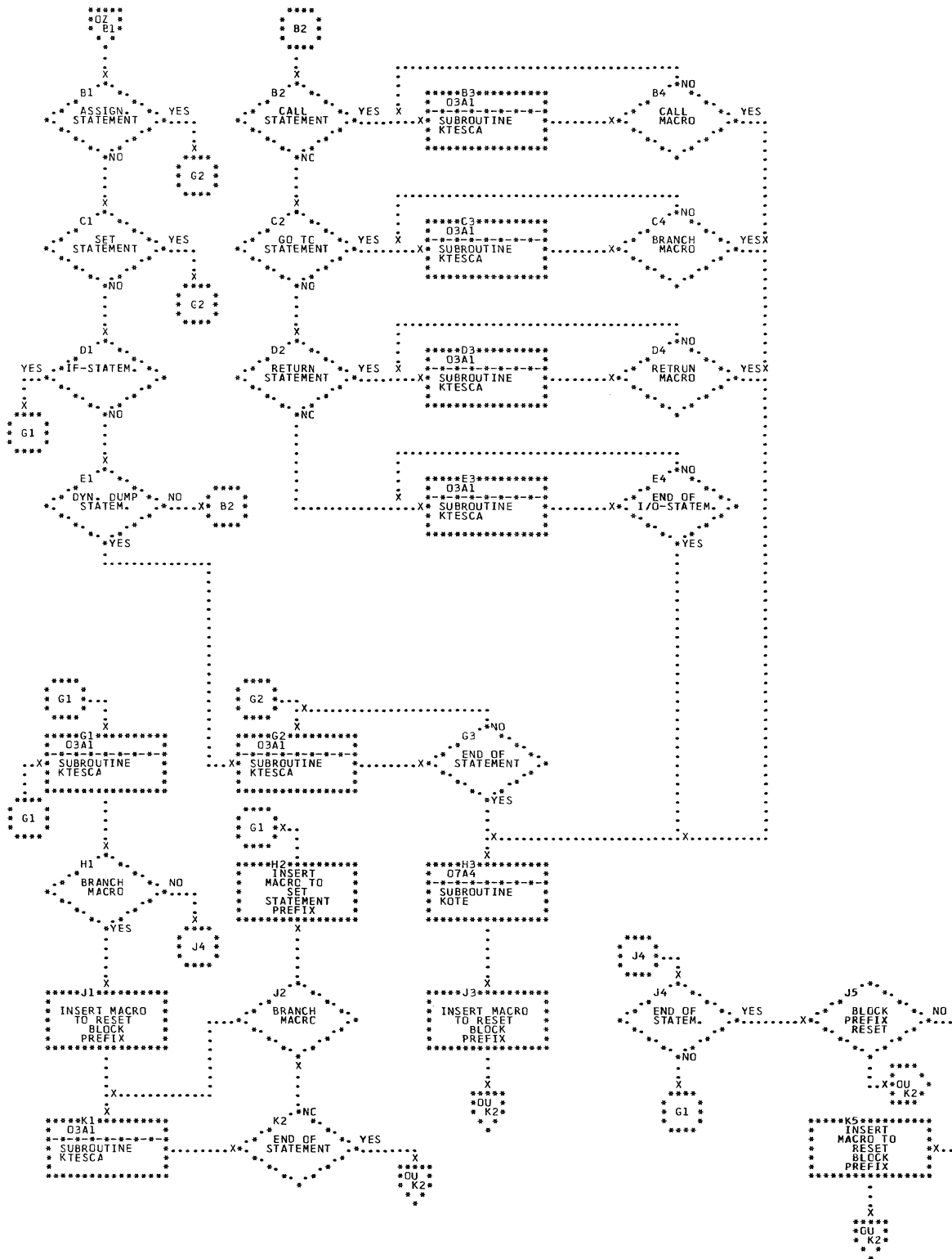
2. SCAN, MAIN PART











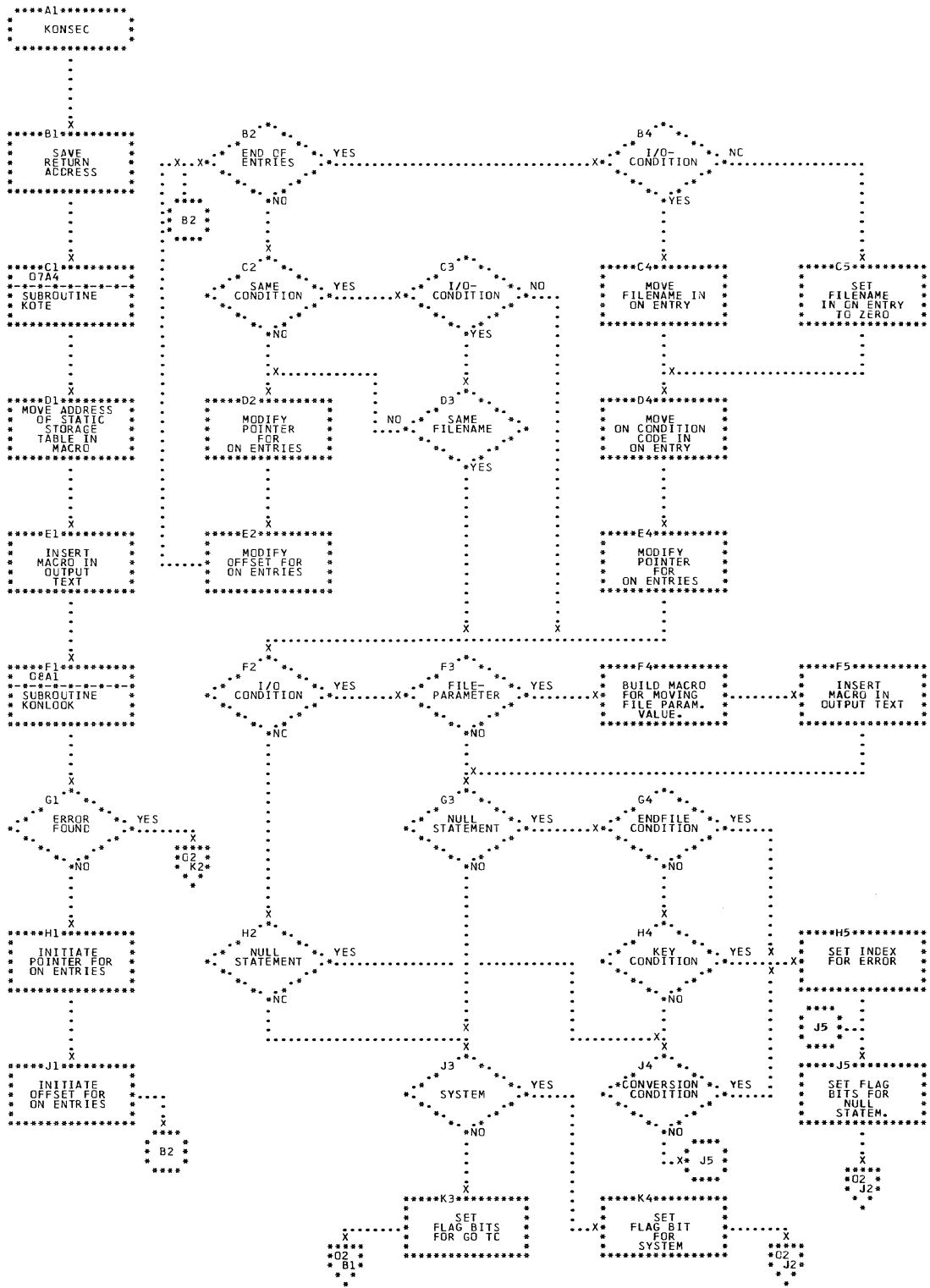
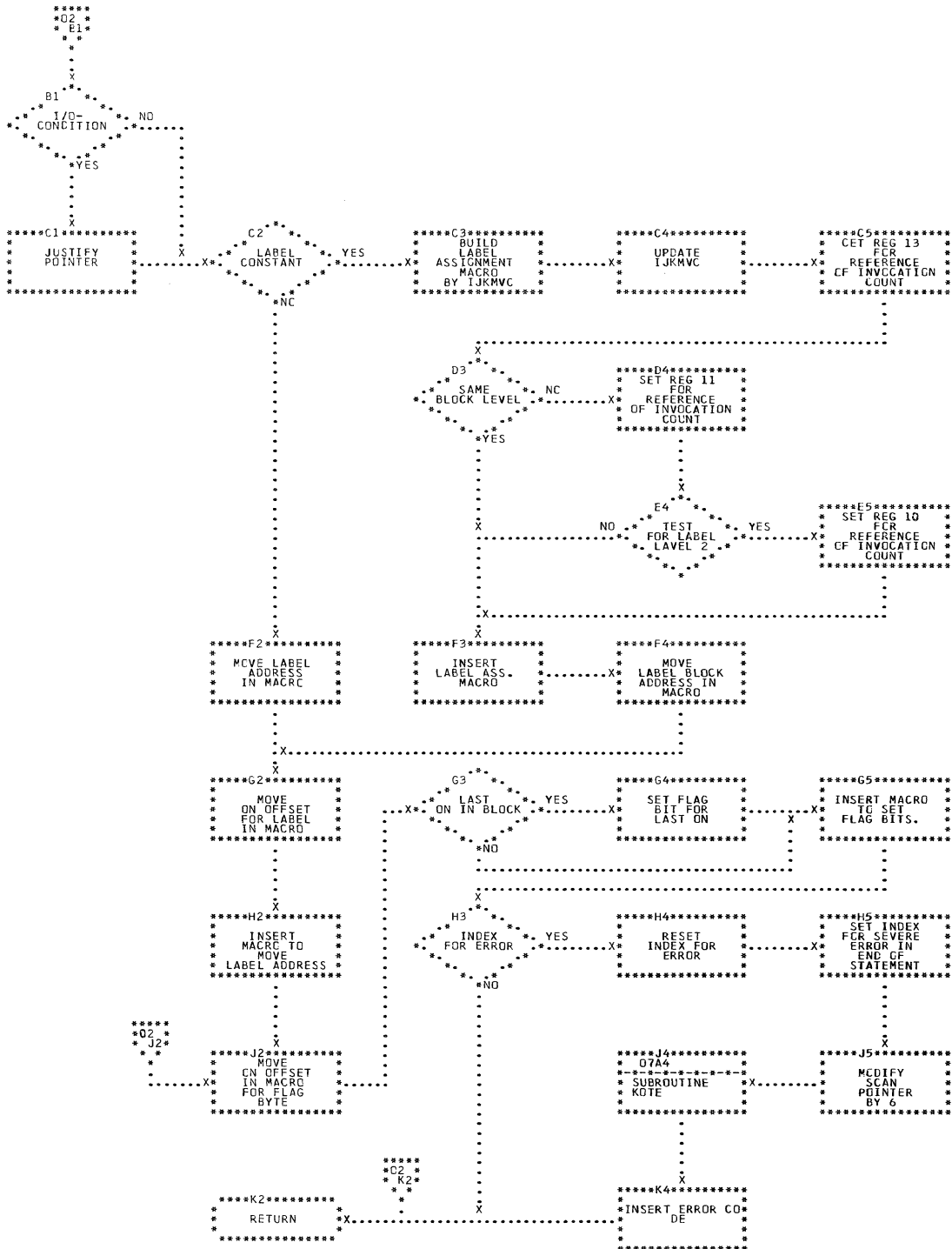


CHART 01. IJXD40

2. SCAN, ON STATEMENT (1)



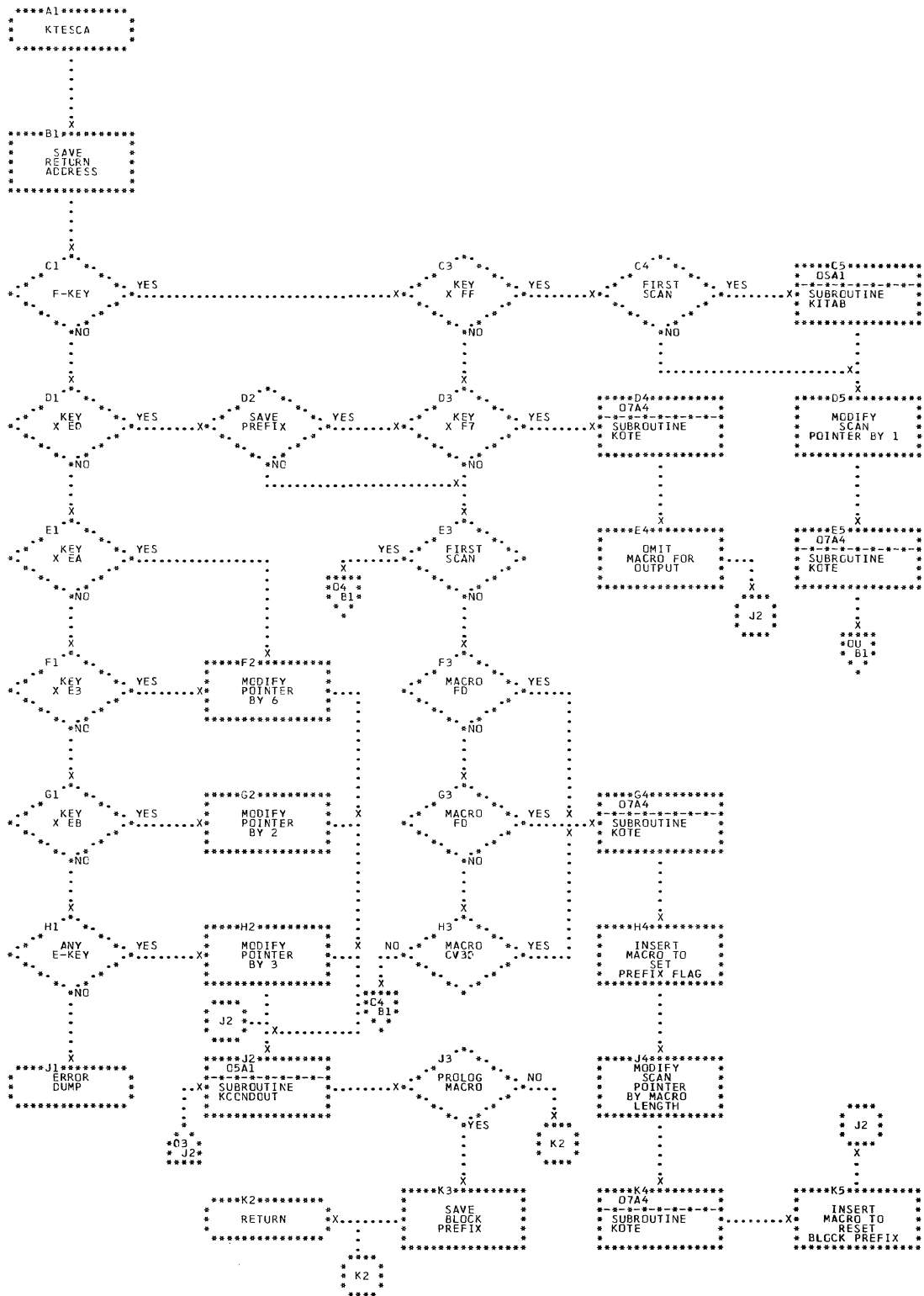
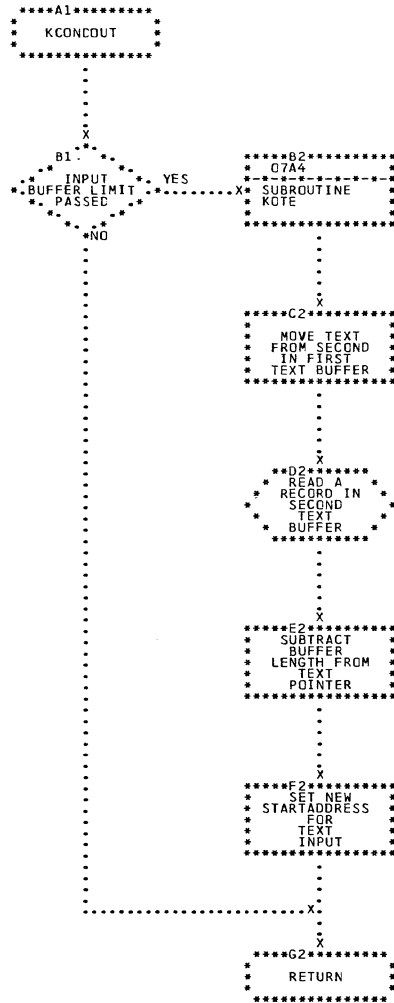


CHART 03. IJXD40

TEXT SCAN (1)



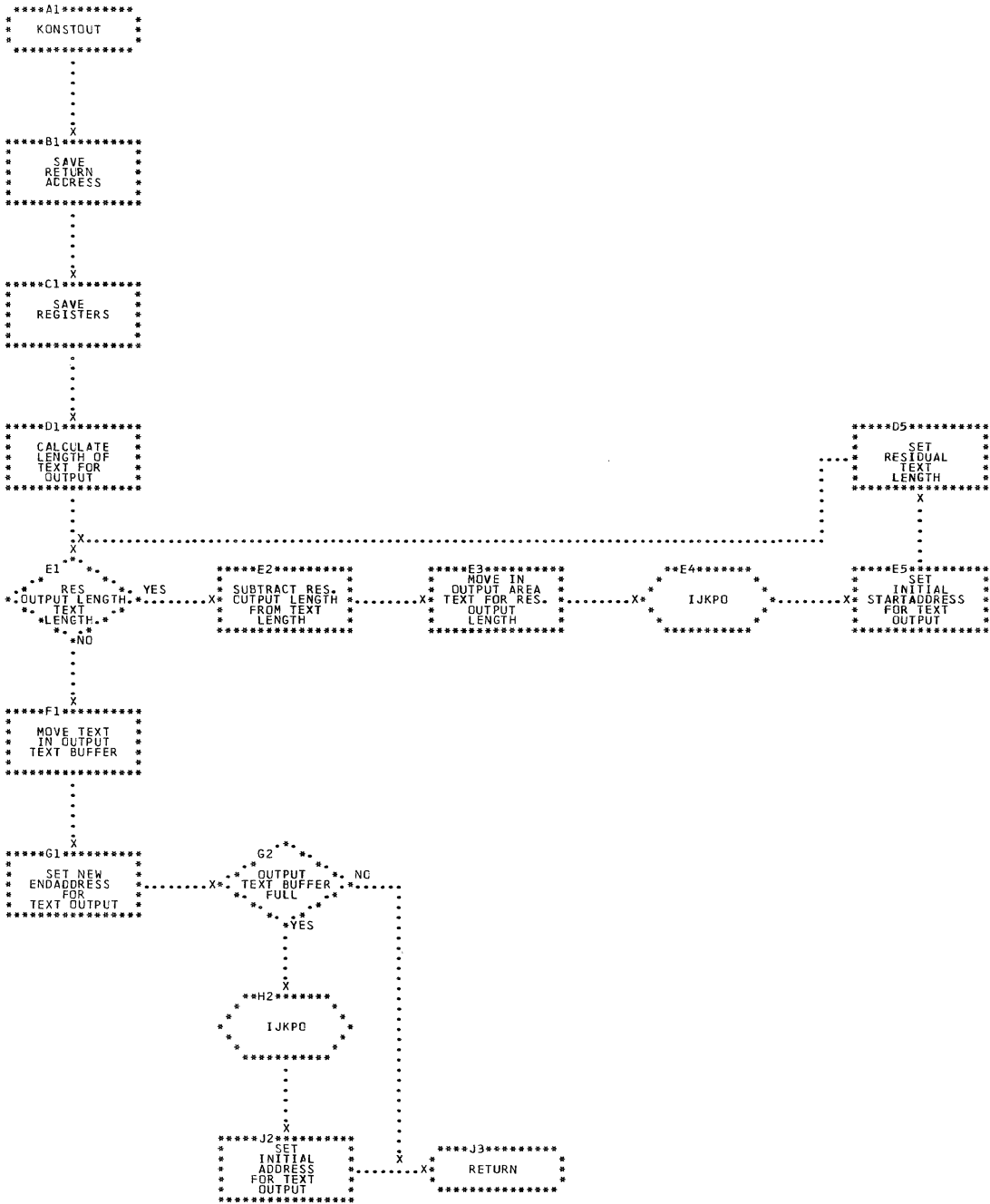


CHART 06. IJXD40 OUTPUT

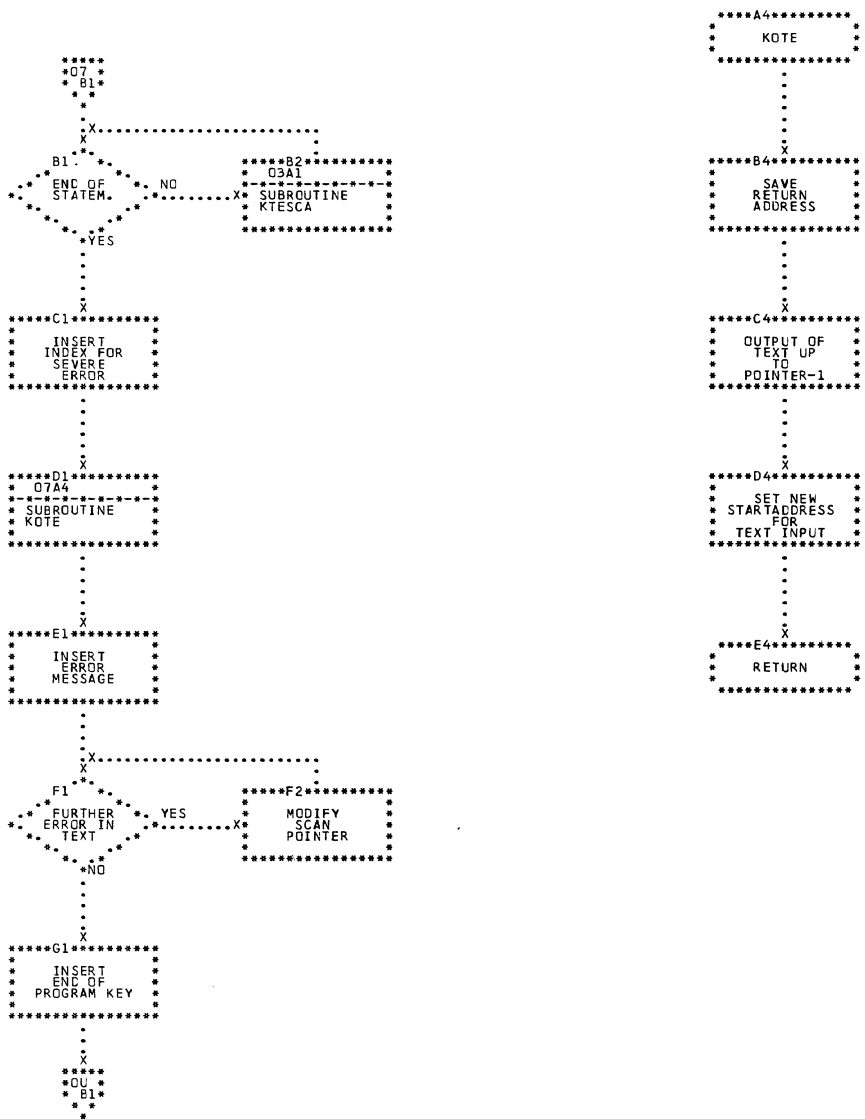
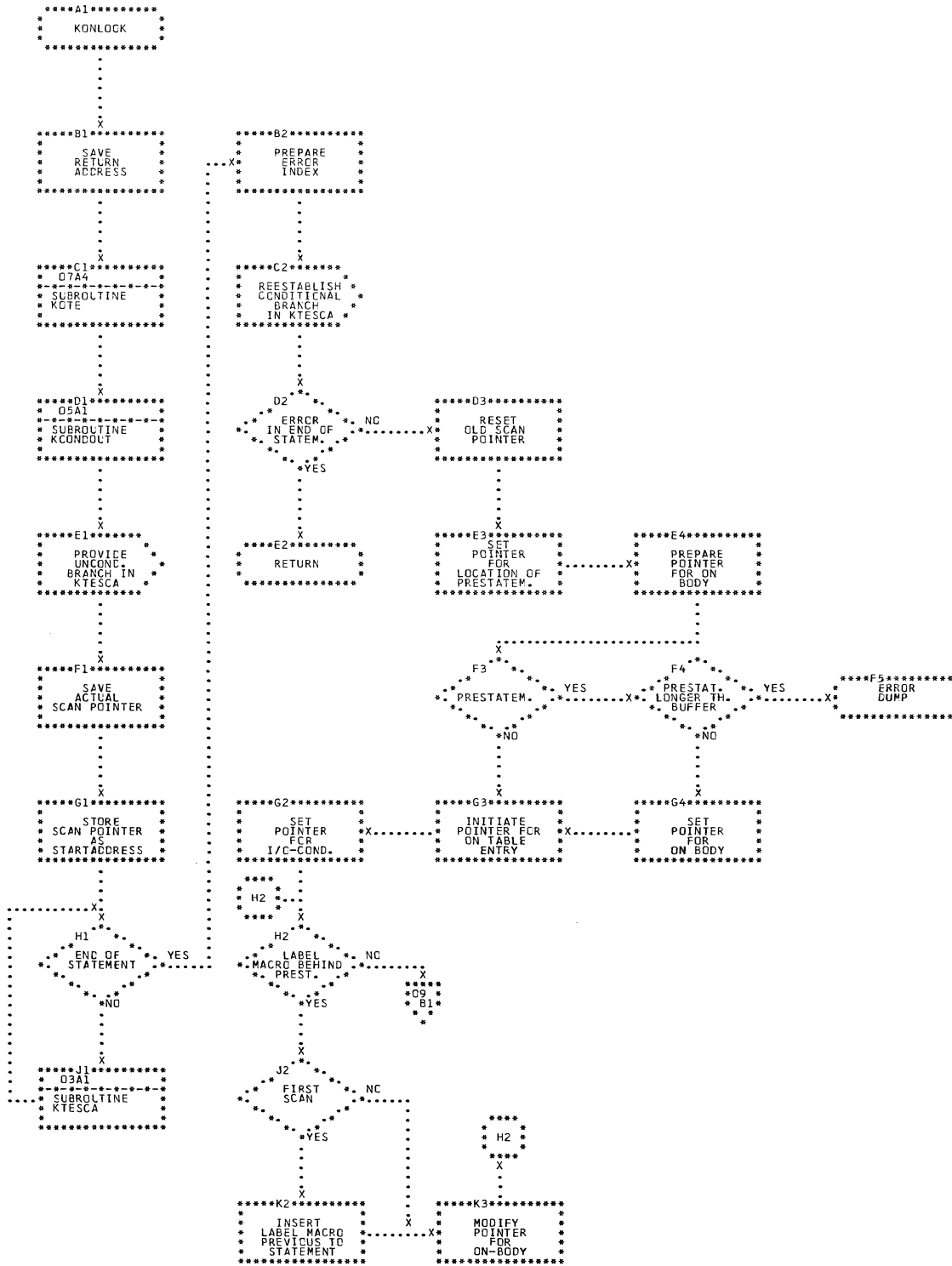
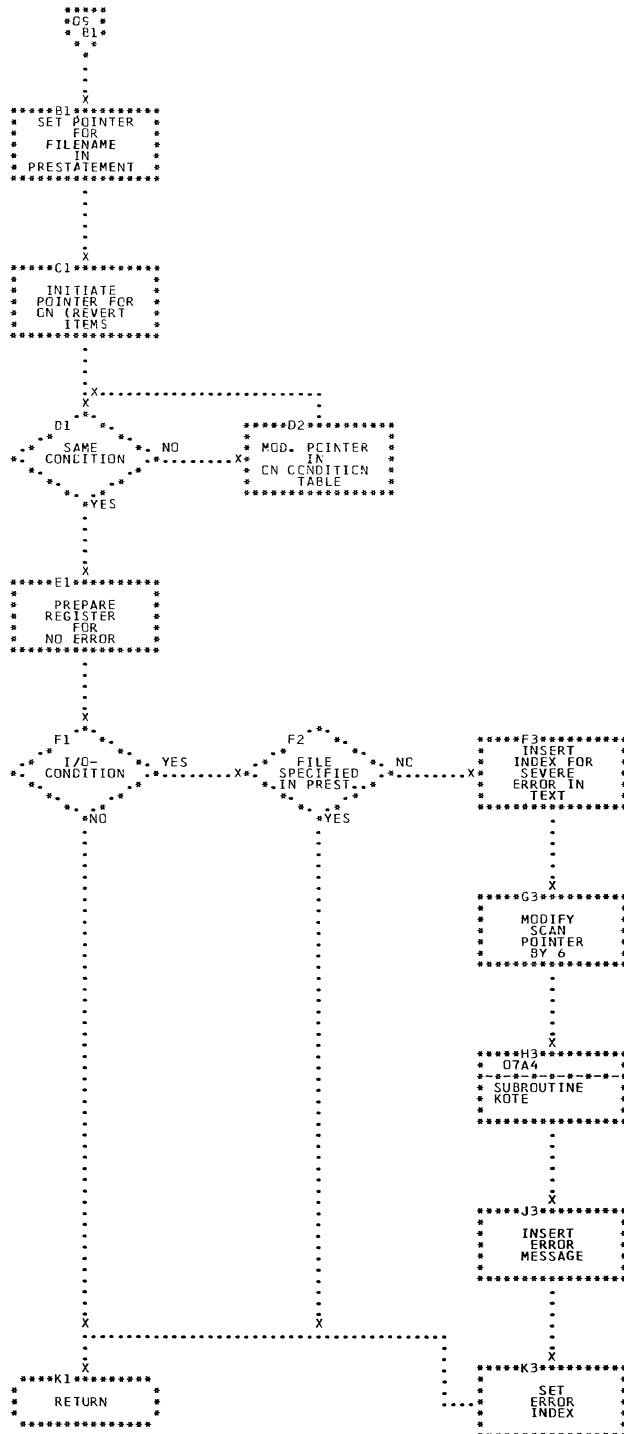
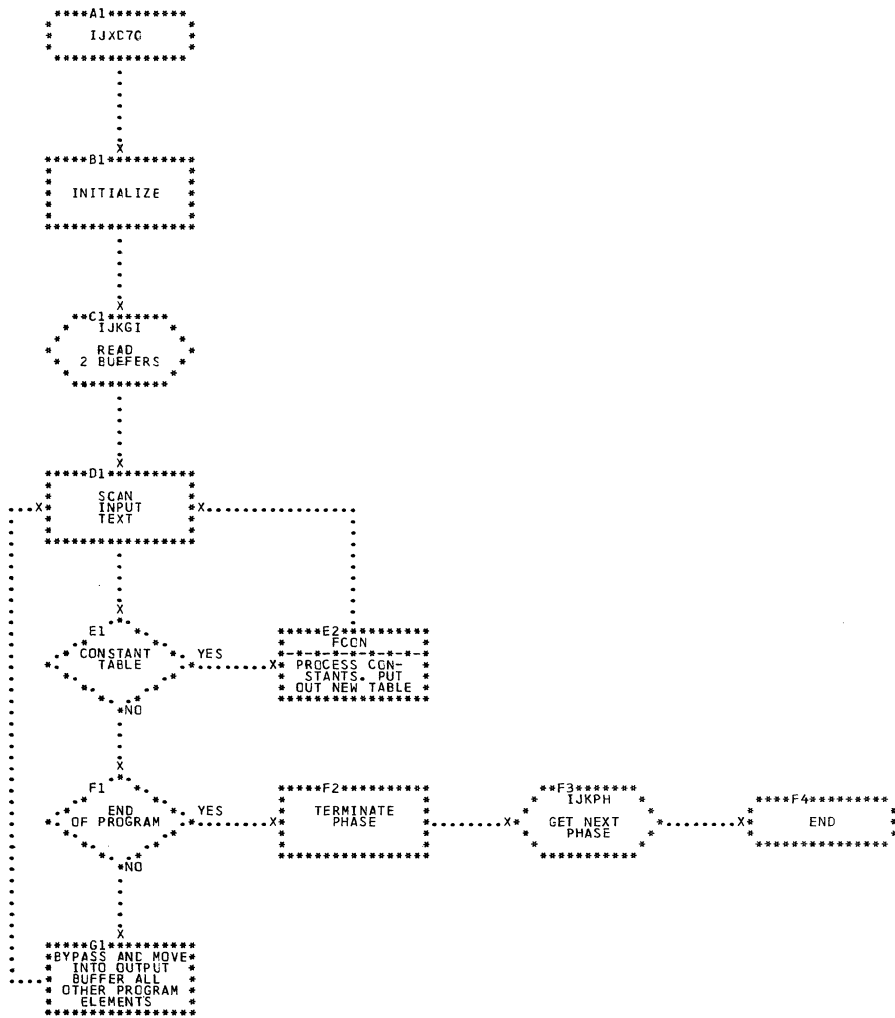


CHART 07. IJXD40

KINTER, KOTE







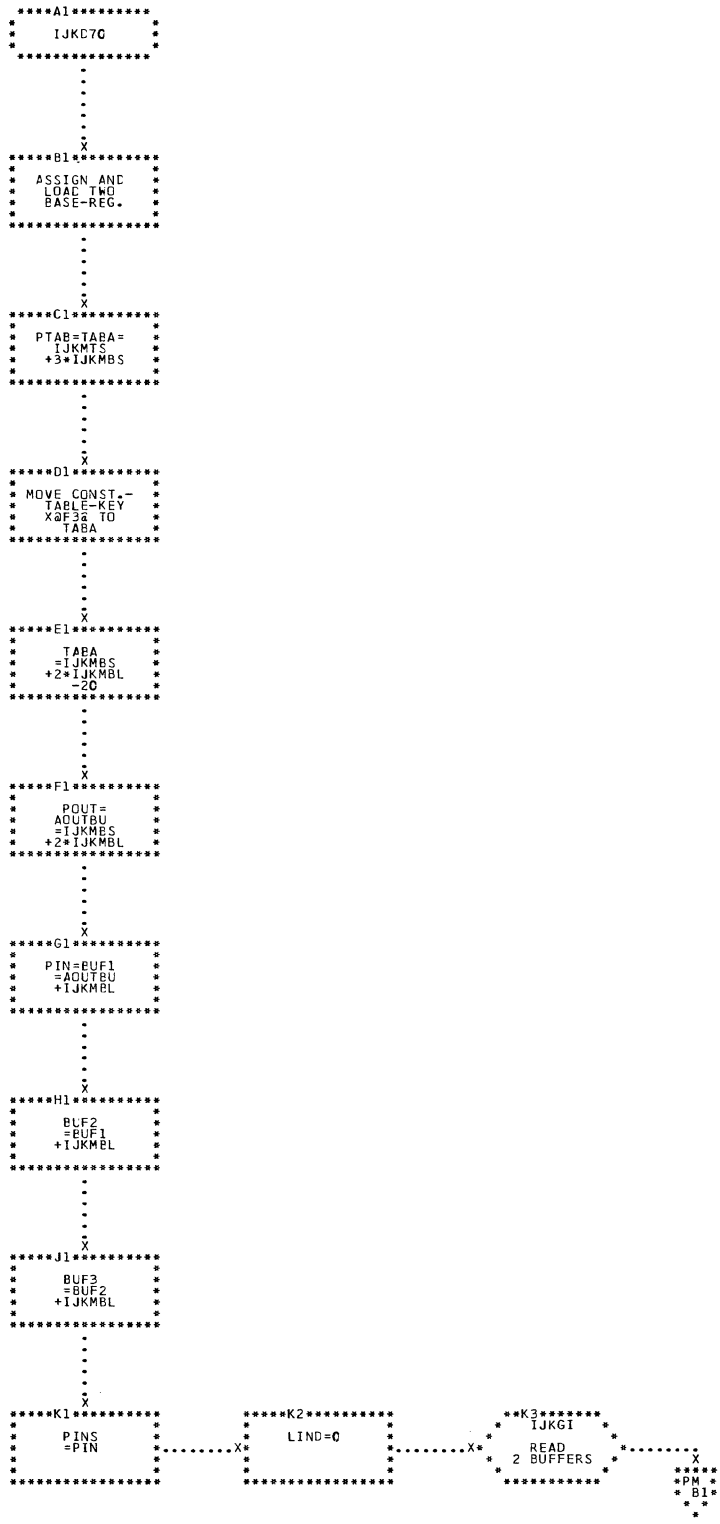


CHART PL. IJXD70

INITIALIZATION

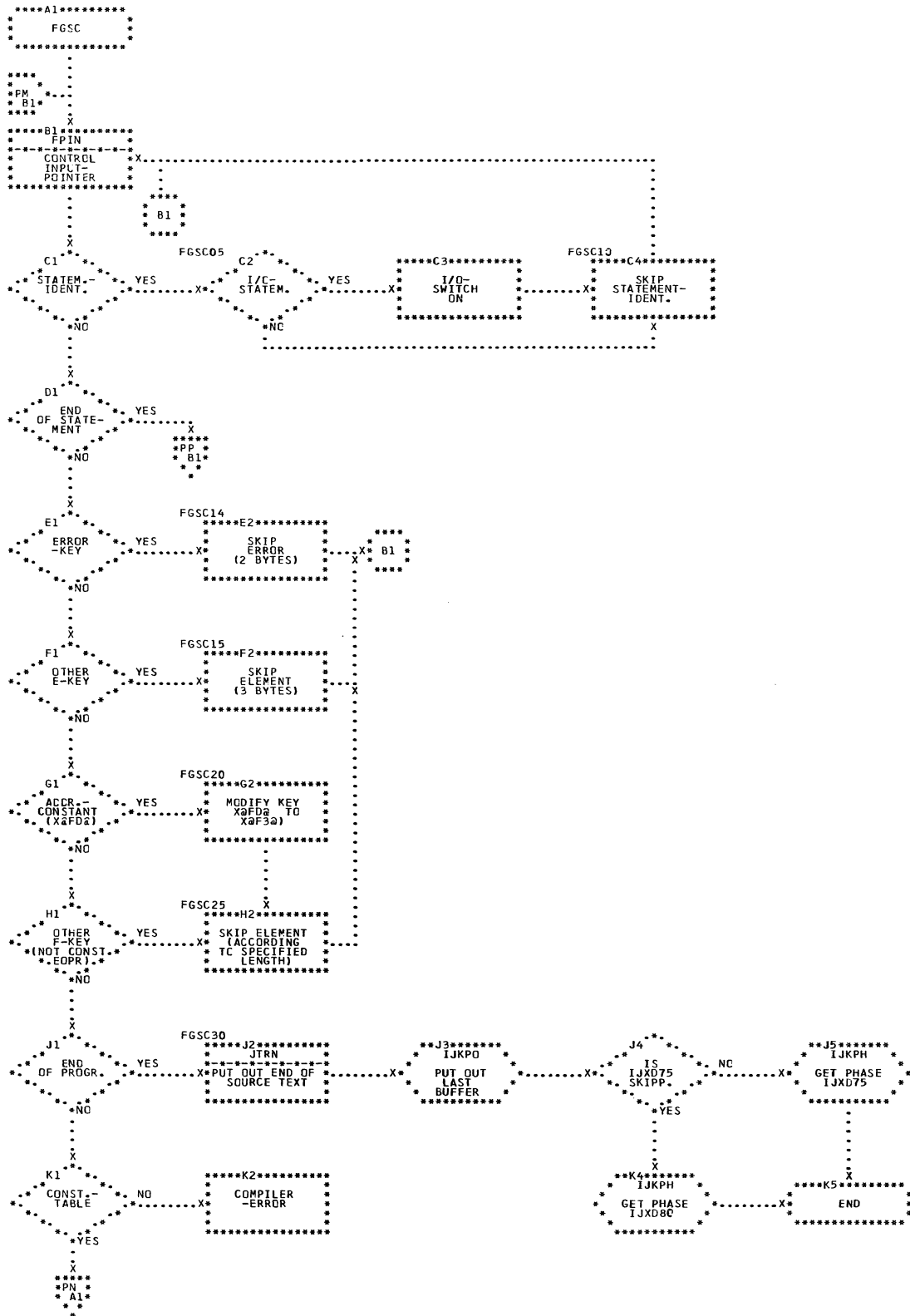
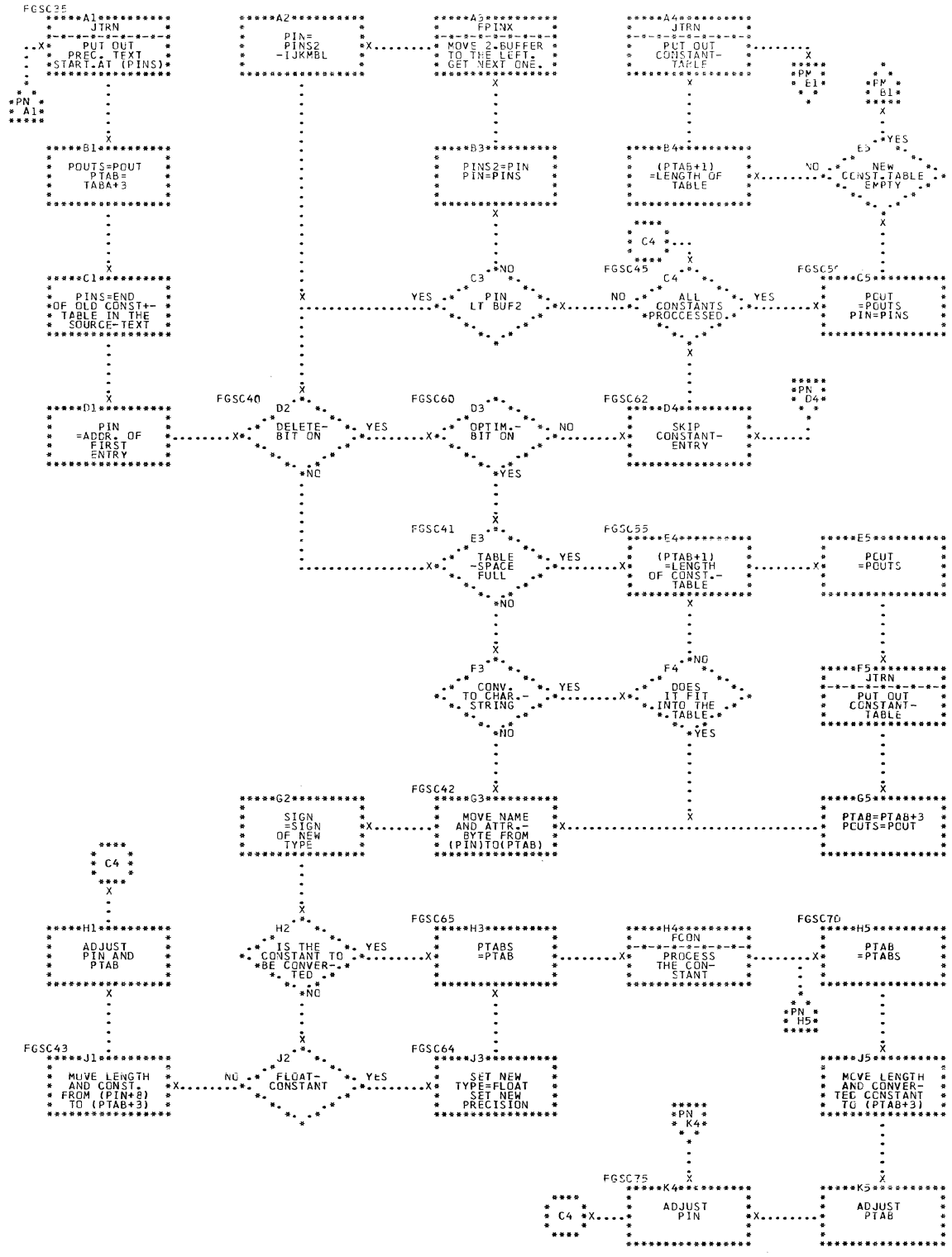


CHART PM. IJXD70 GENERAL SCAN



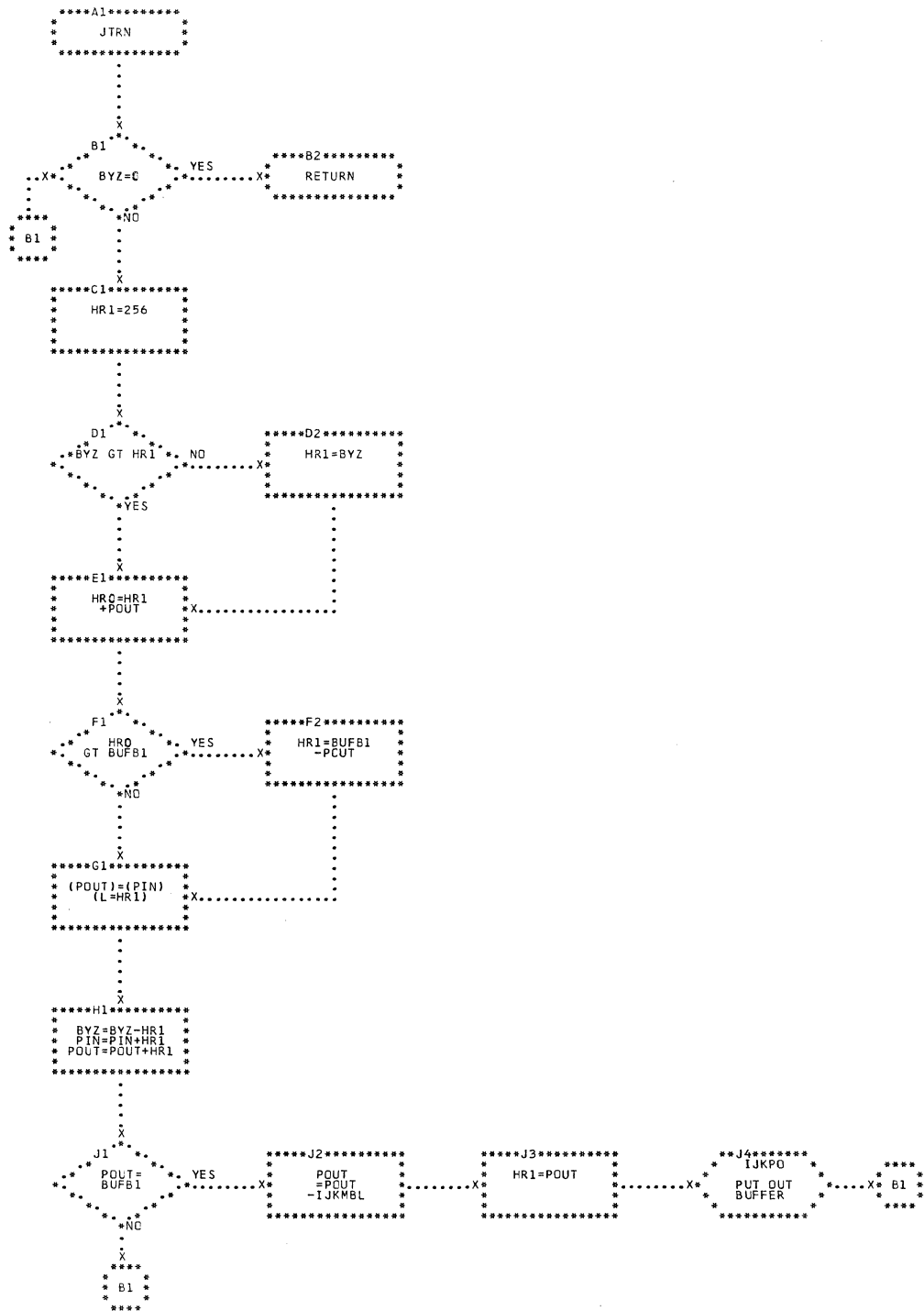
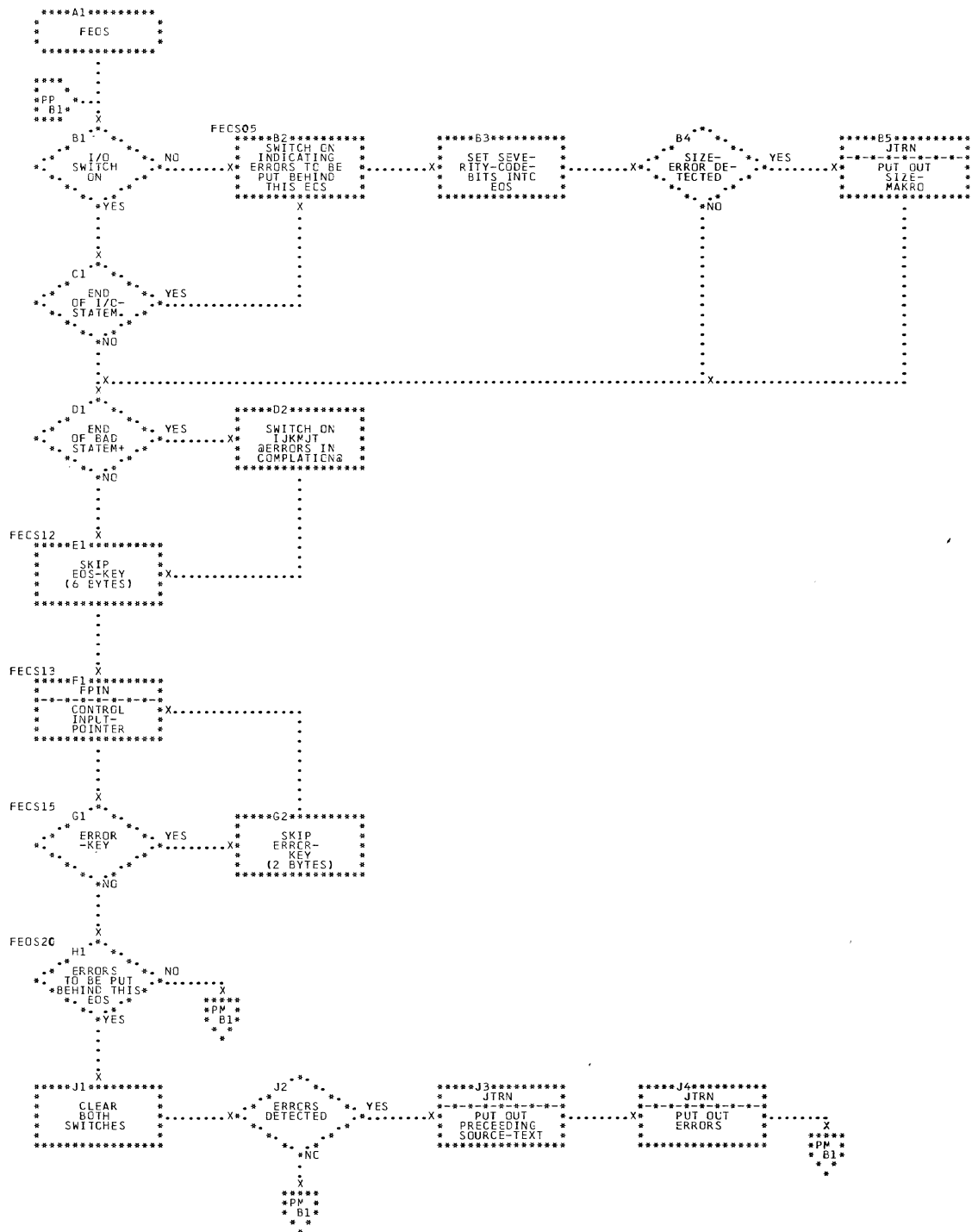
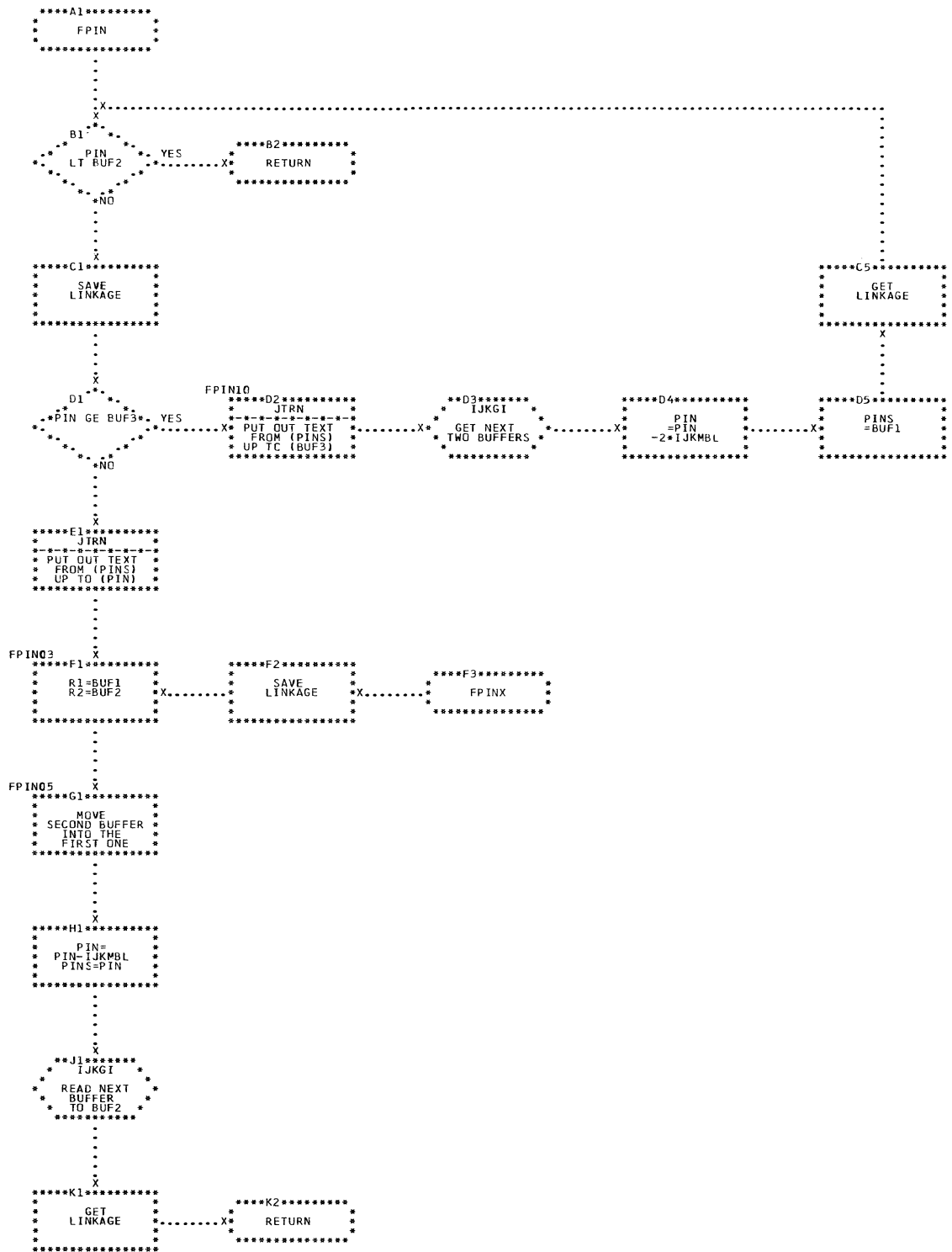
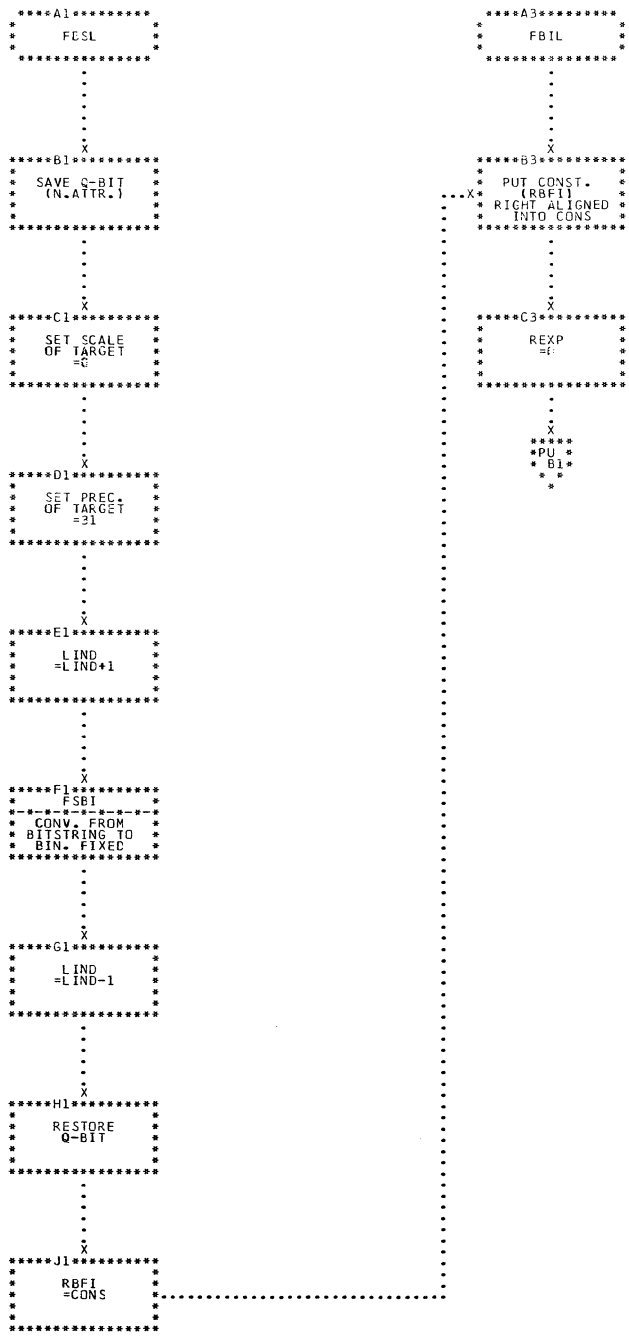


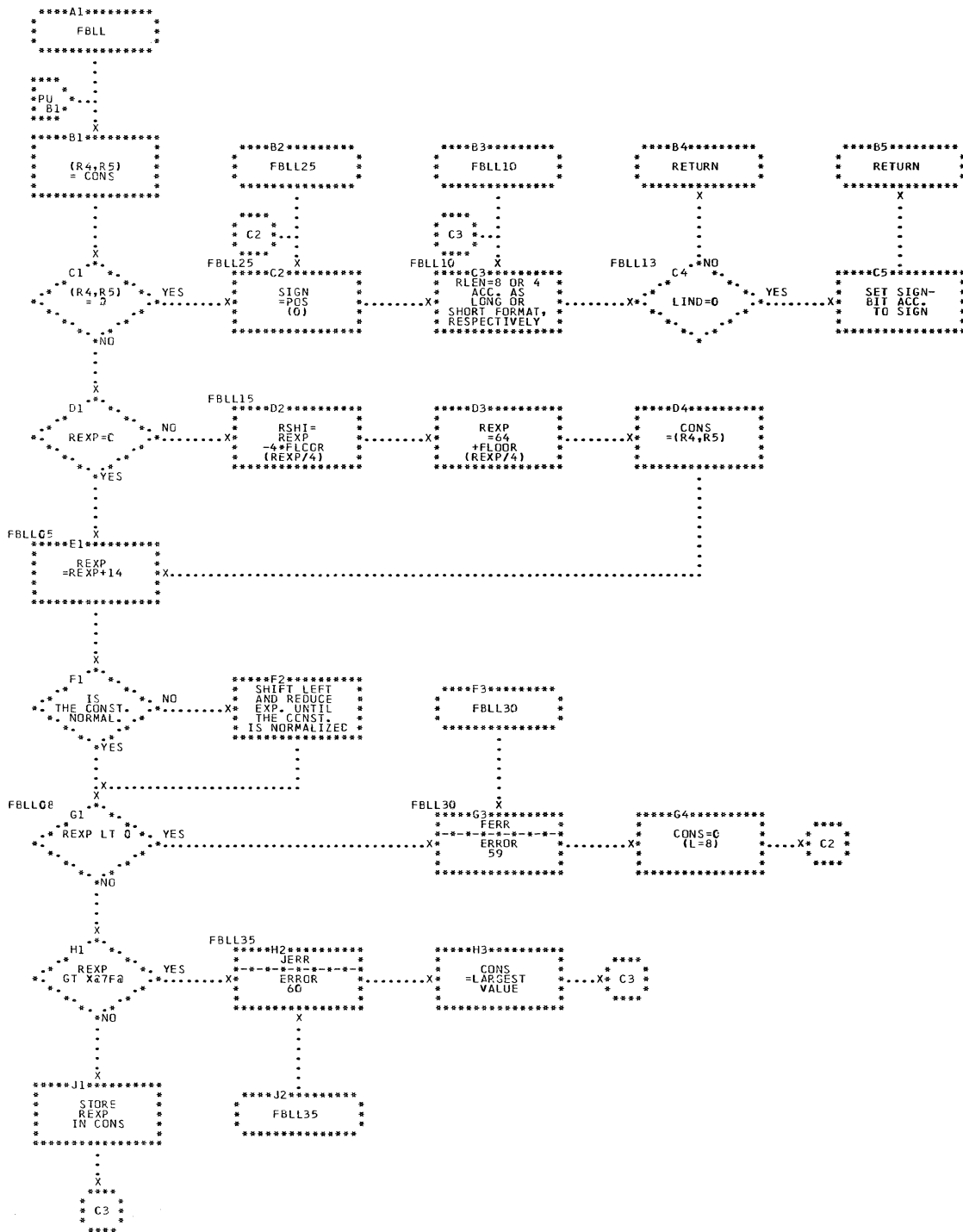
CHART PO. IJXD70

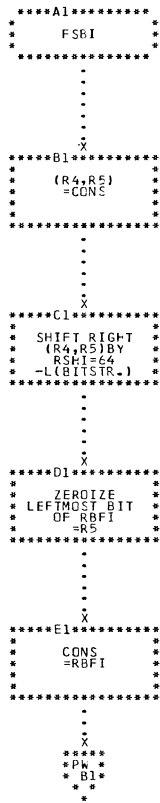
MOVE INTO THE OUTPUT-BUFFER

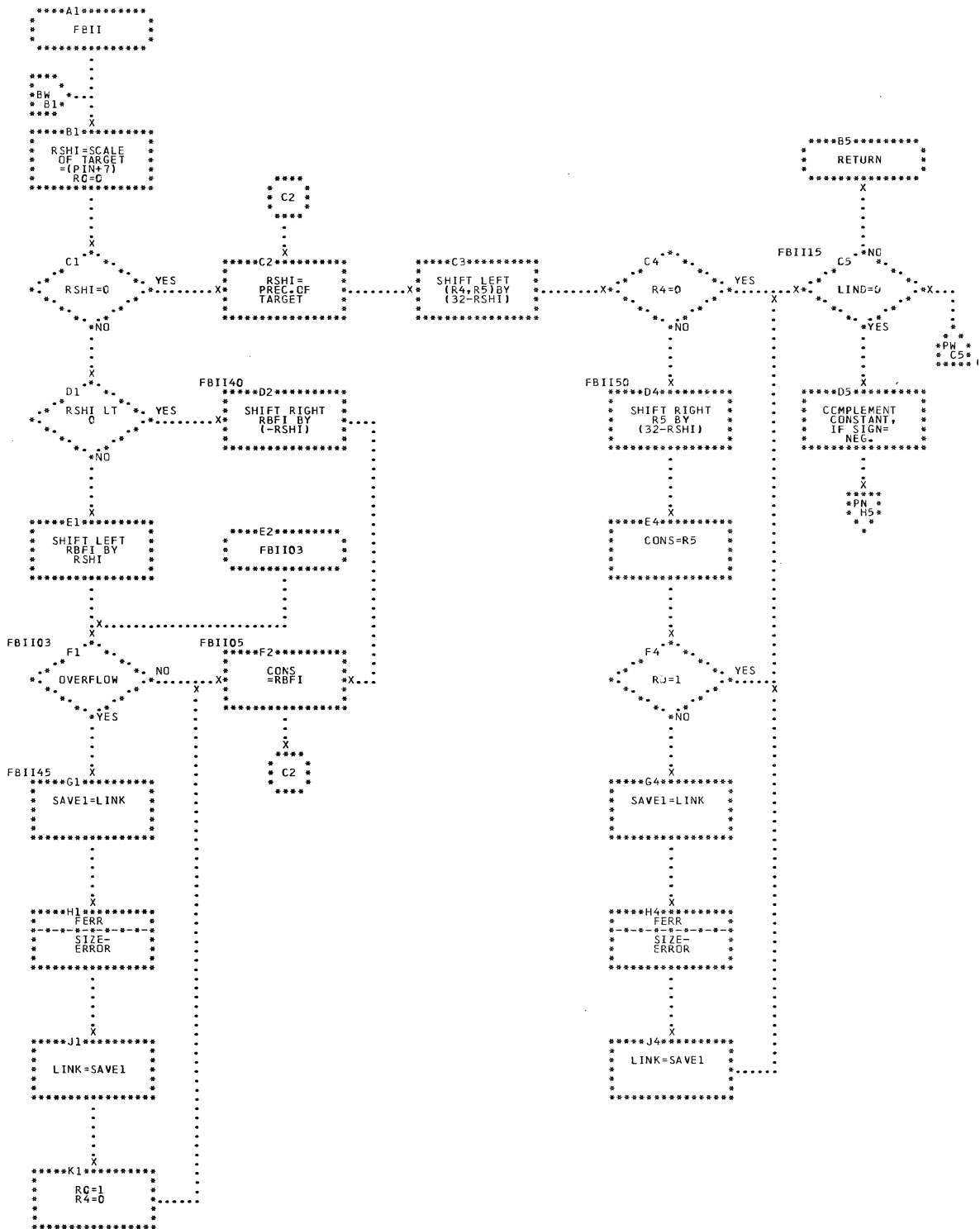


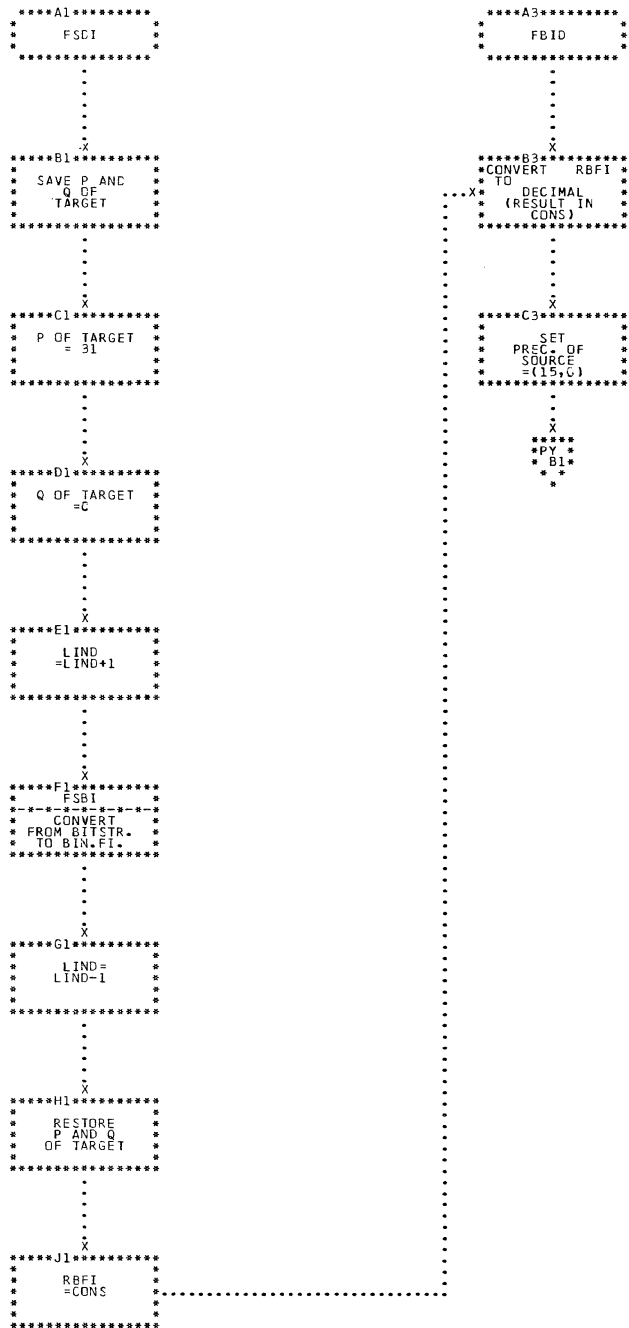












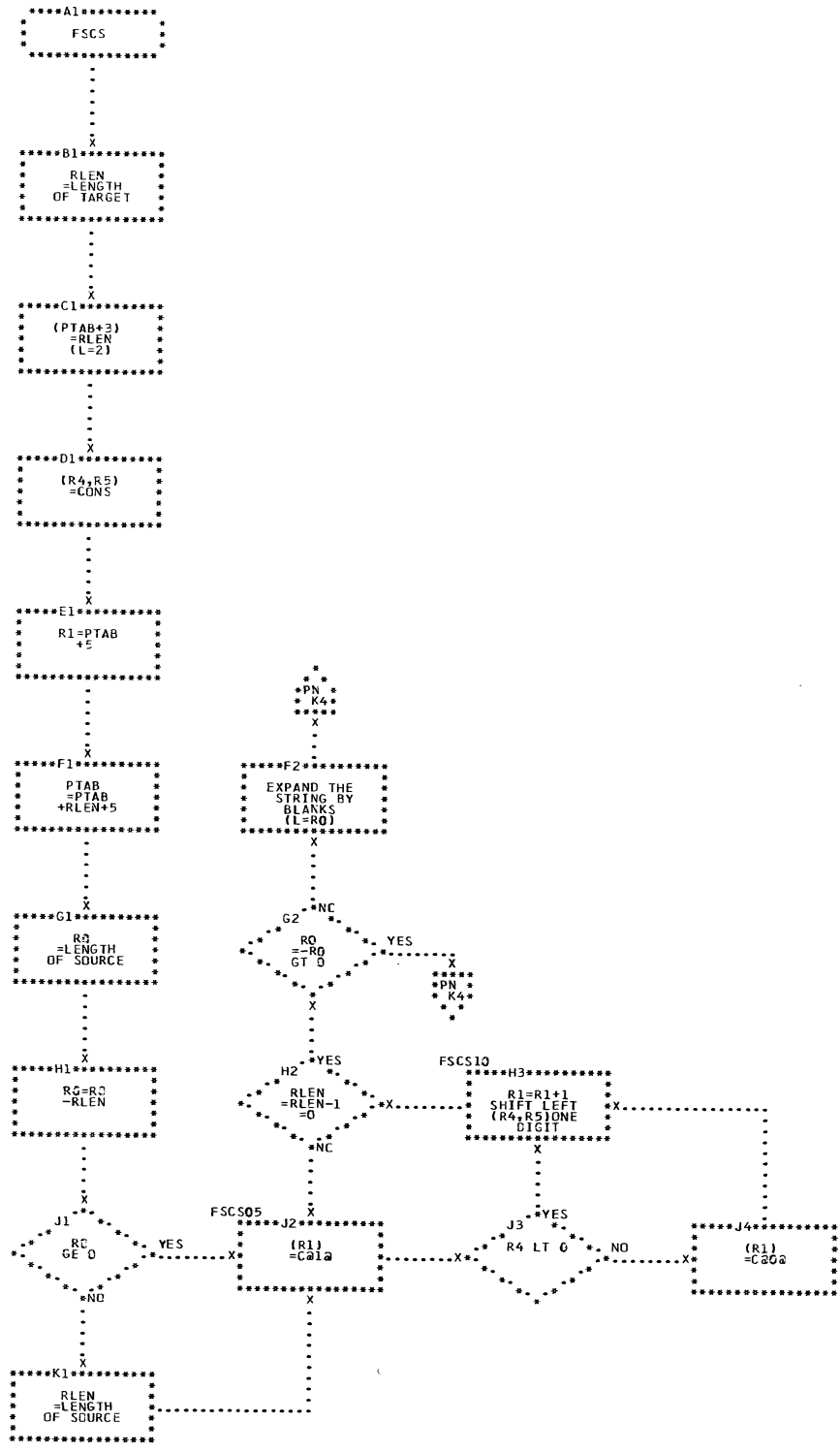
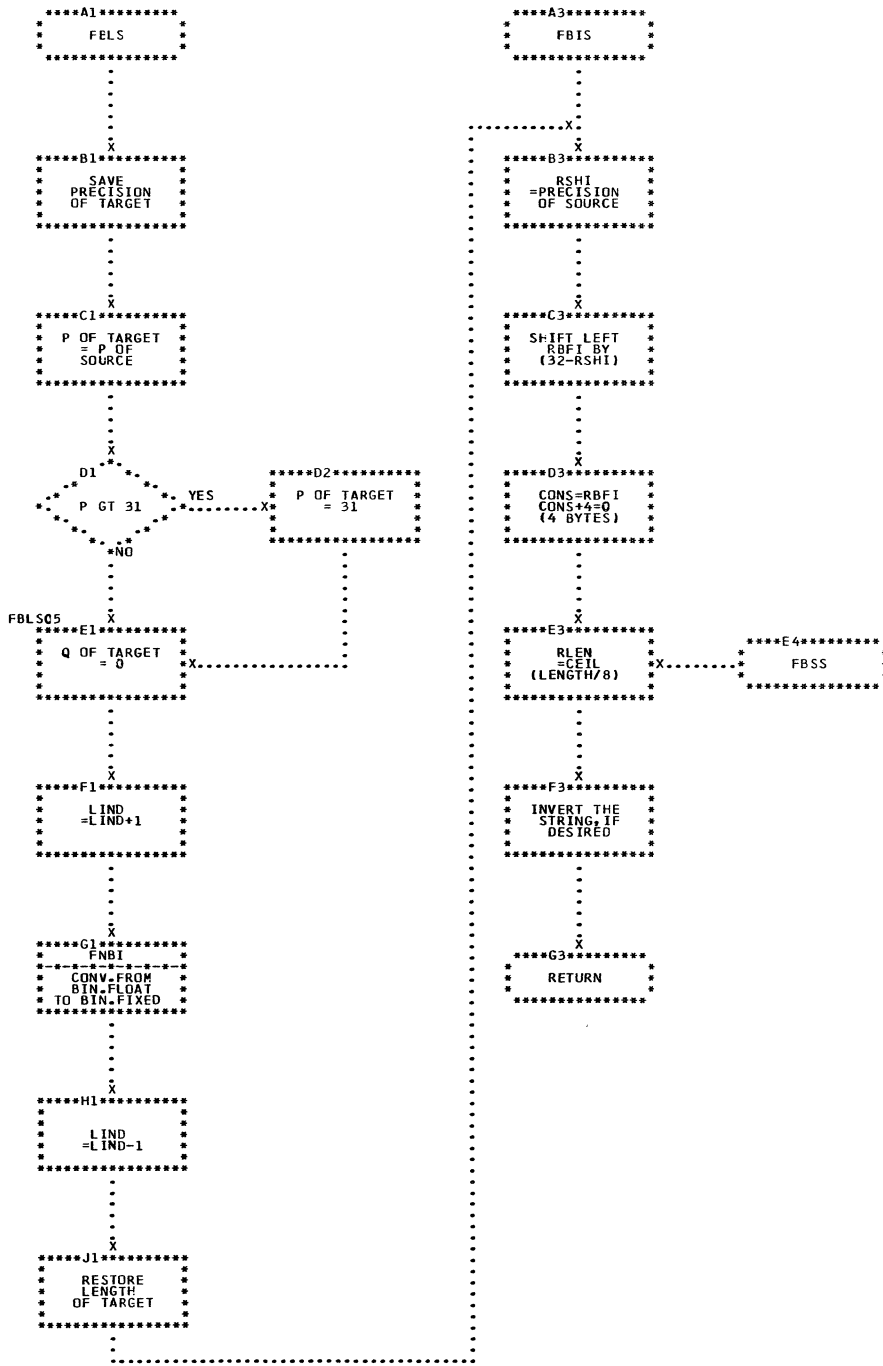


CHART PZ. IJXD70

BIT TO CHARACTER



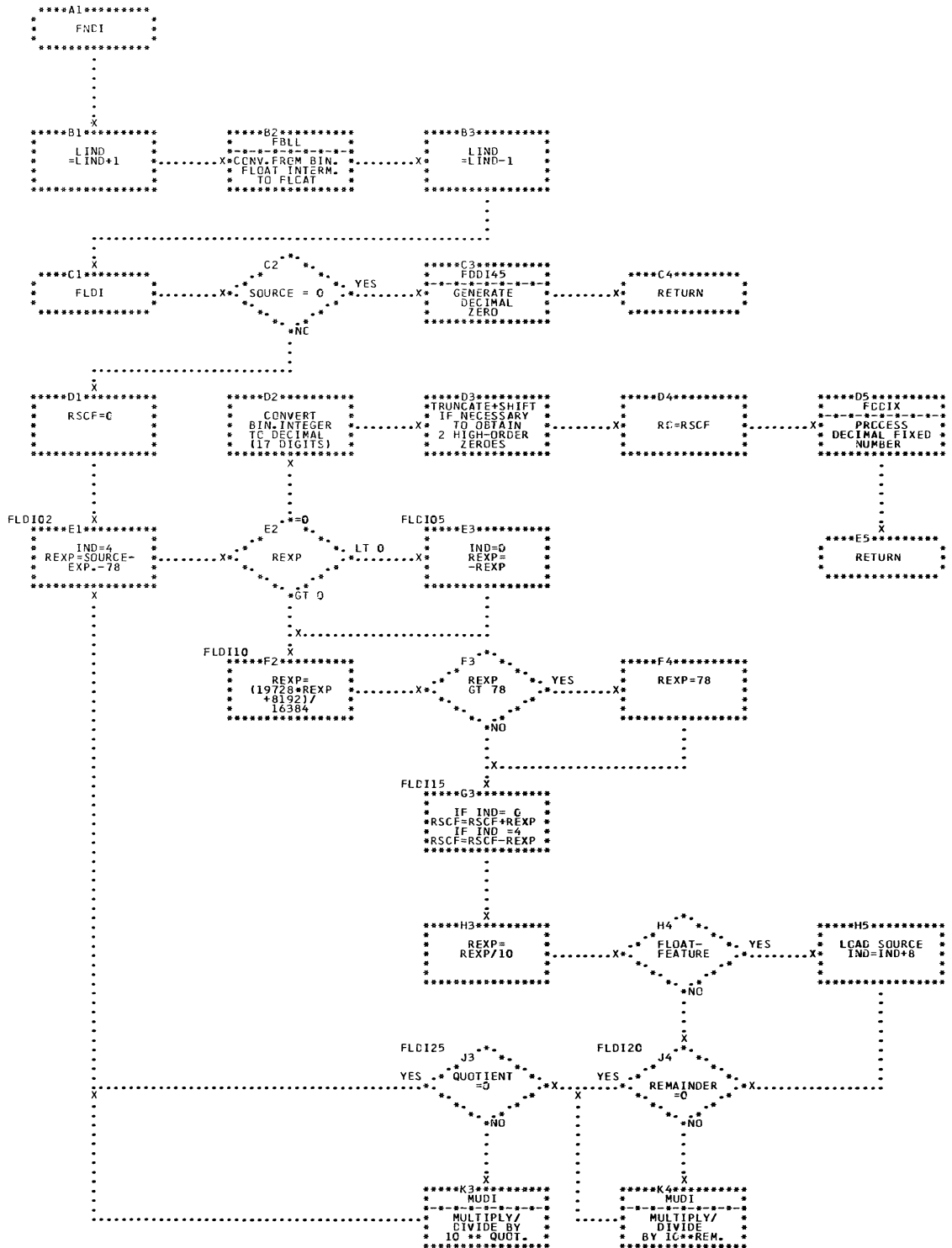


CHART QB. IJXD70

BINARY FLOAT TO DEC FIXED

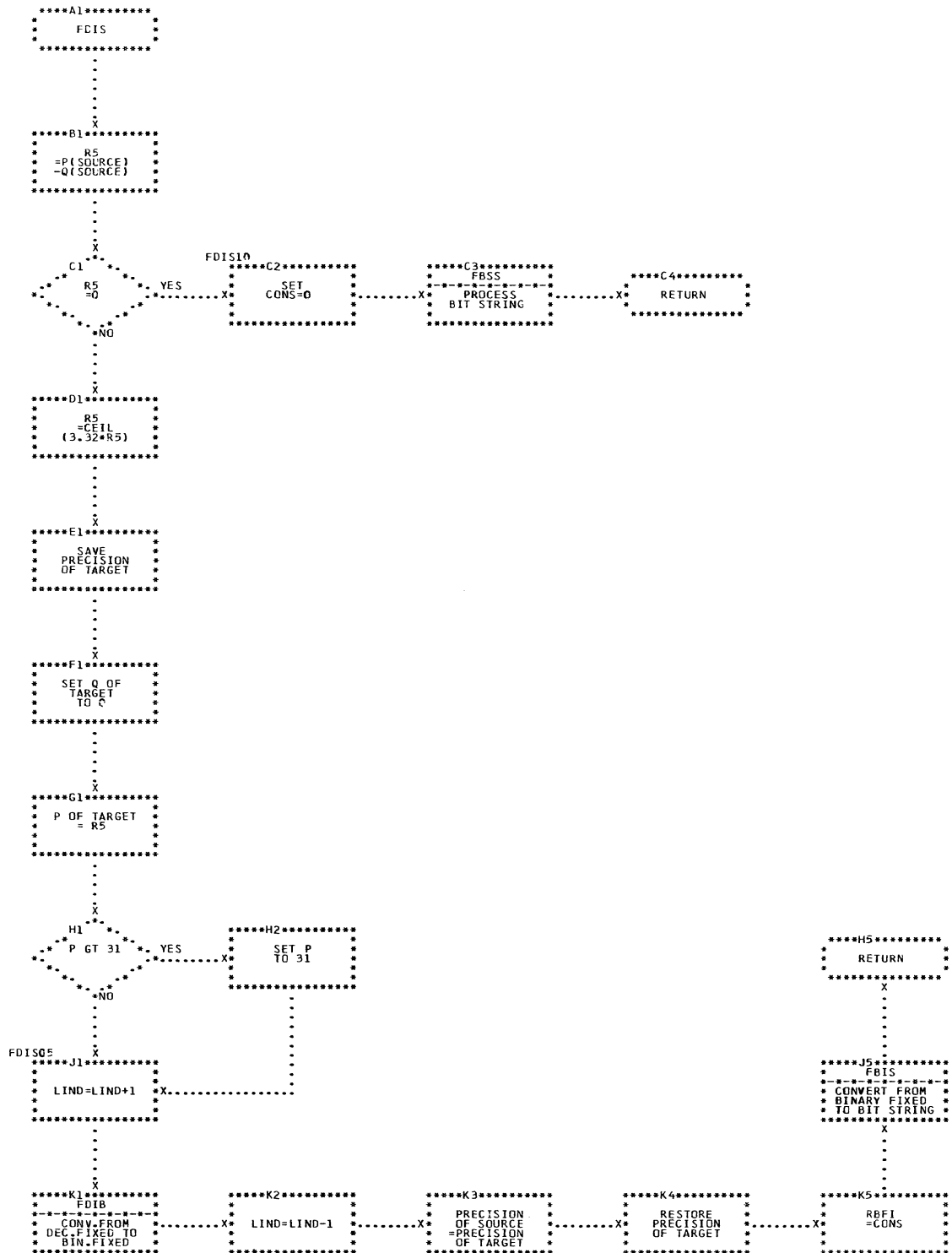


CHART QD. IJXD70

DECIMAL FIXED TO BIT

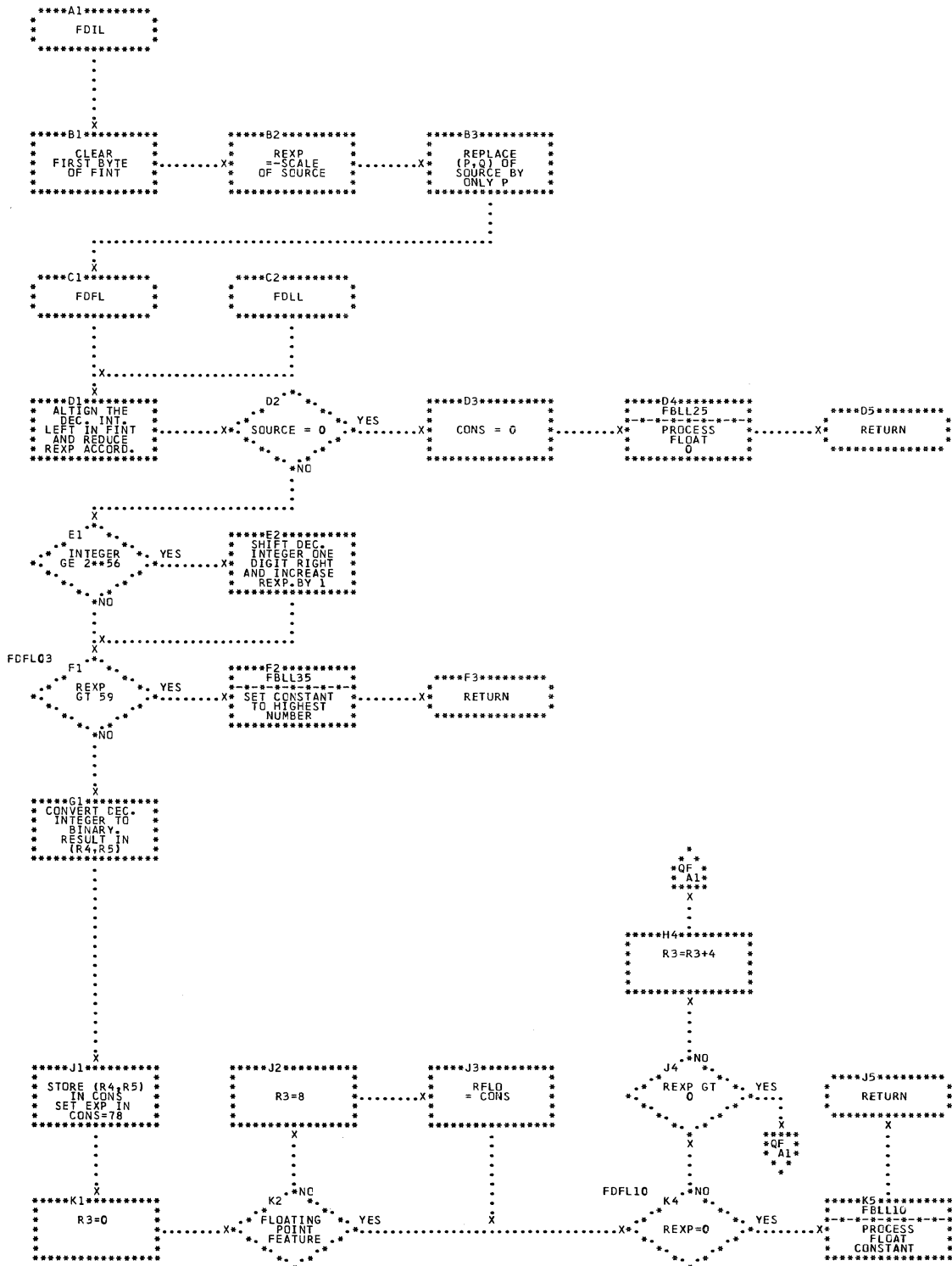
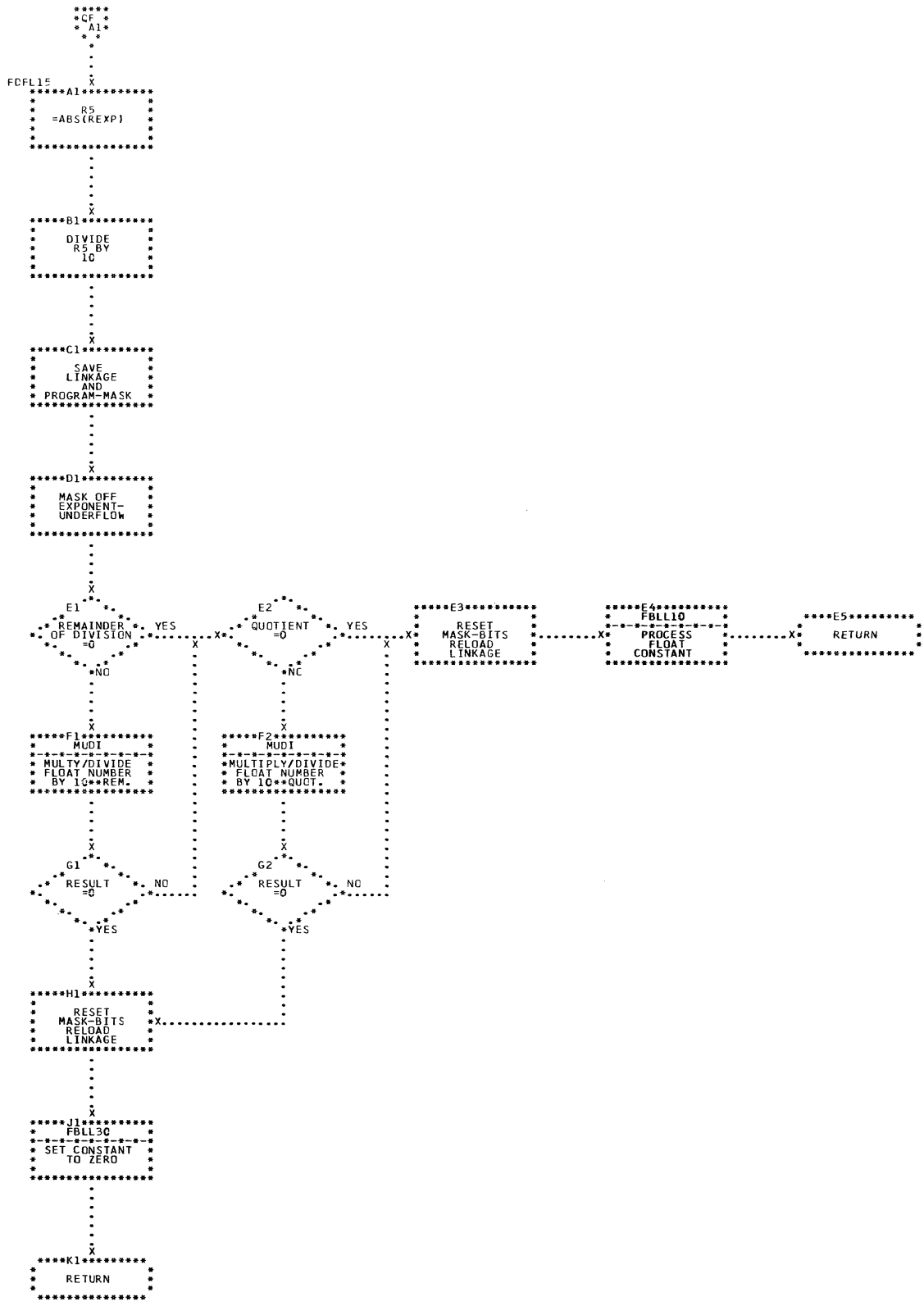


CHART QE. IJXD70 DECIMAL FIXED TO FLOAT



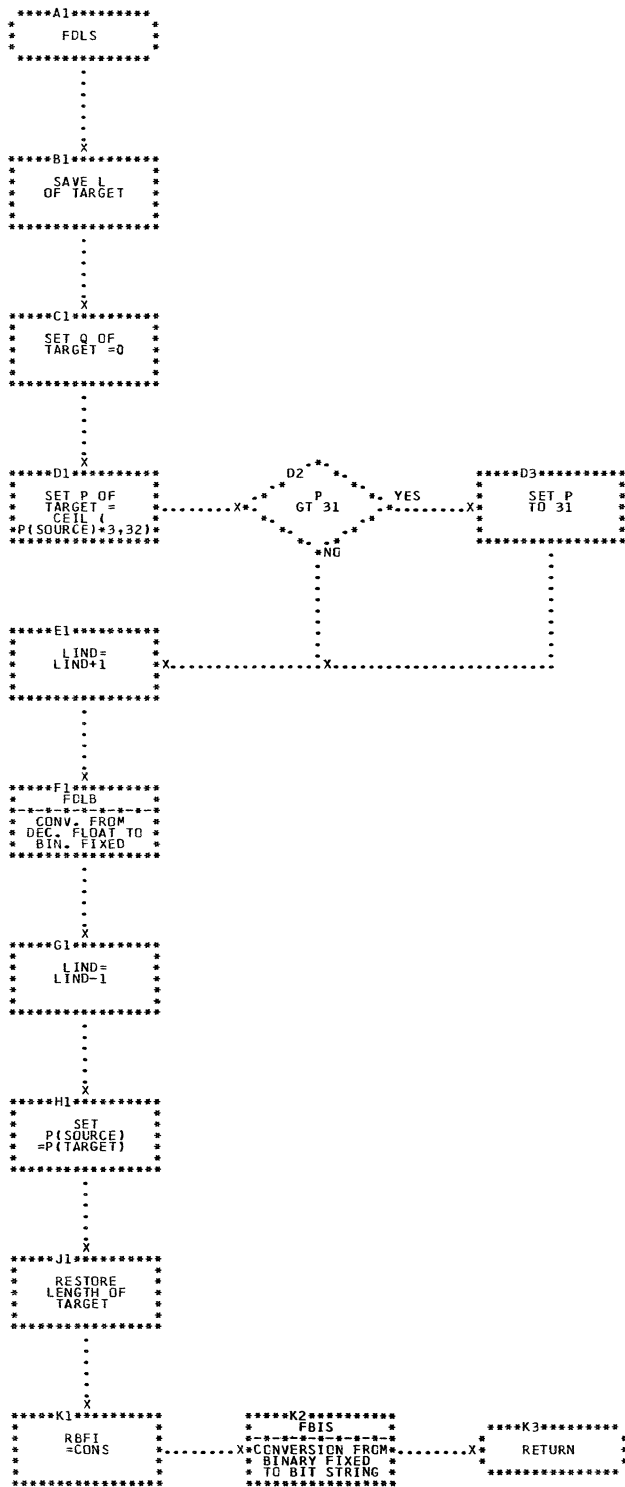
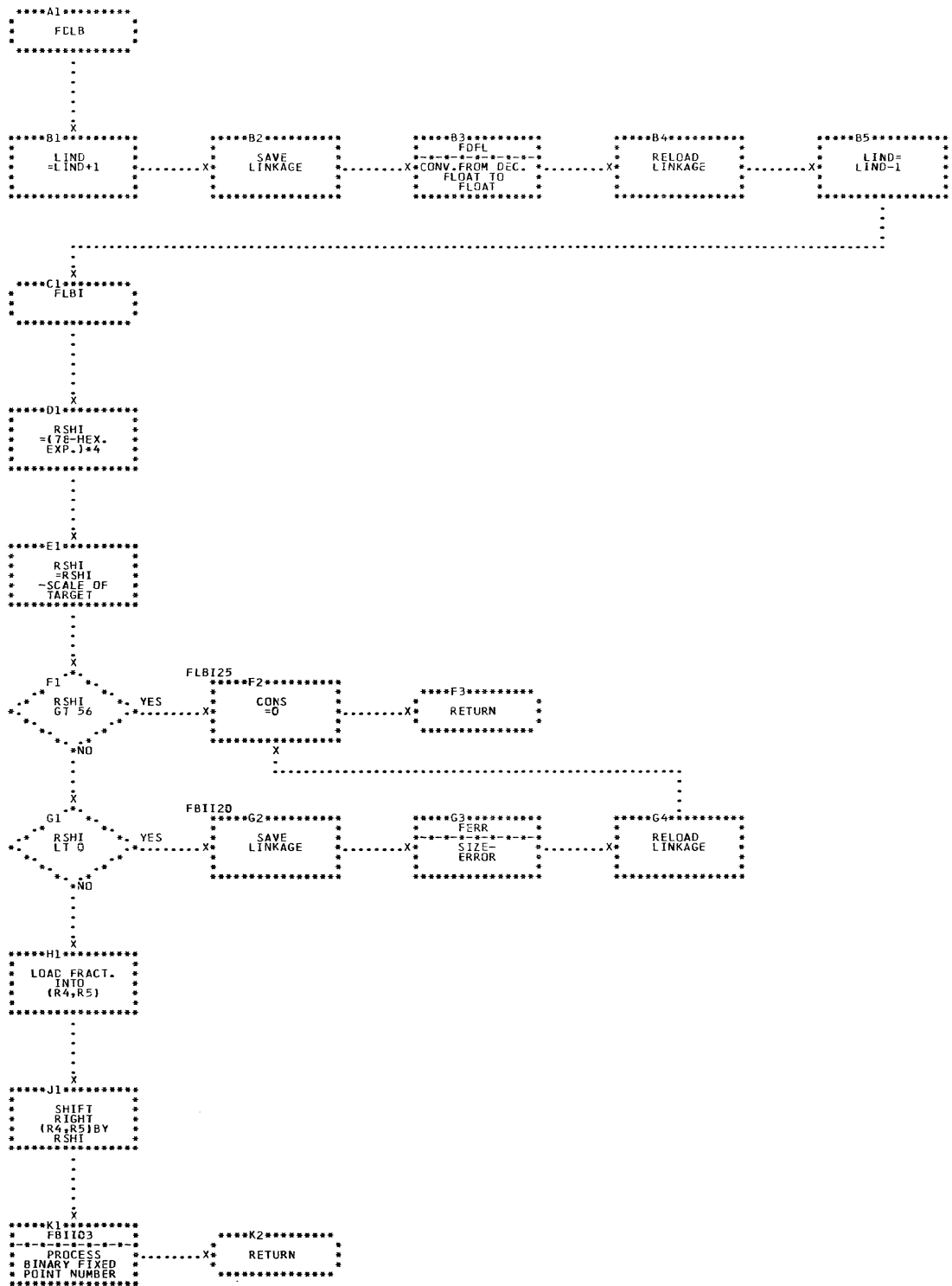
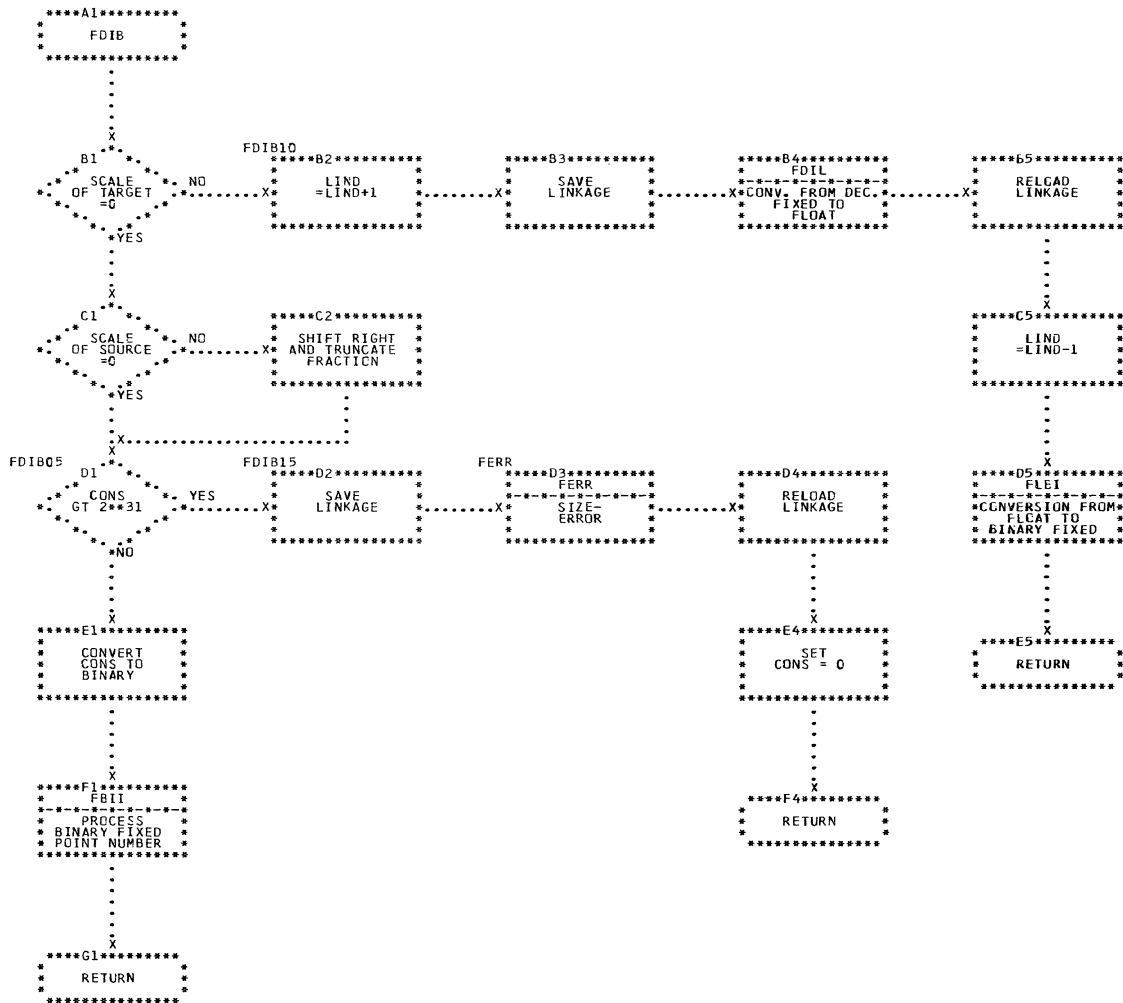
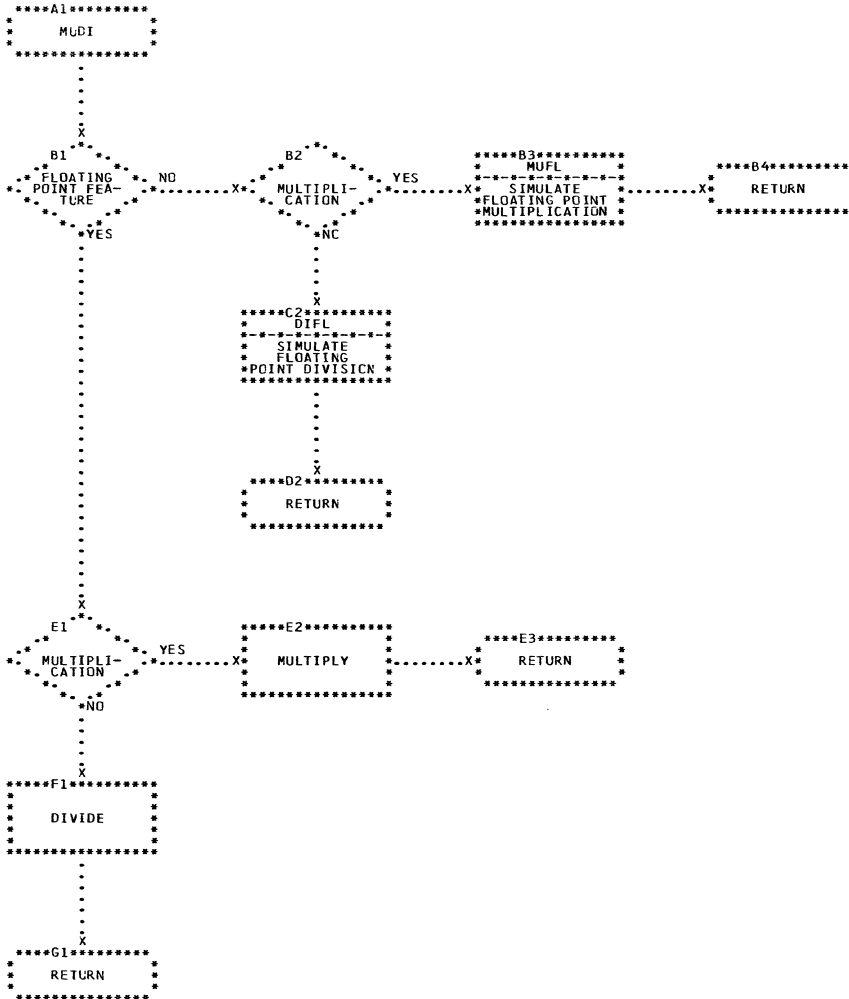


CHART QG. DOS/TOS PL/I DECIMAL FLOAT TO BIT







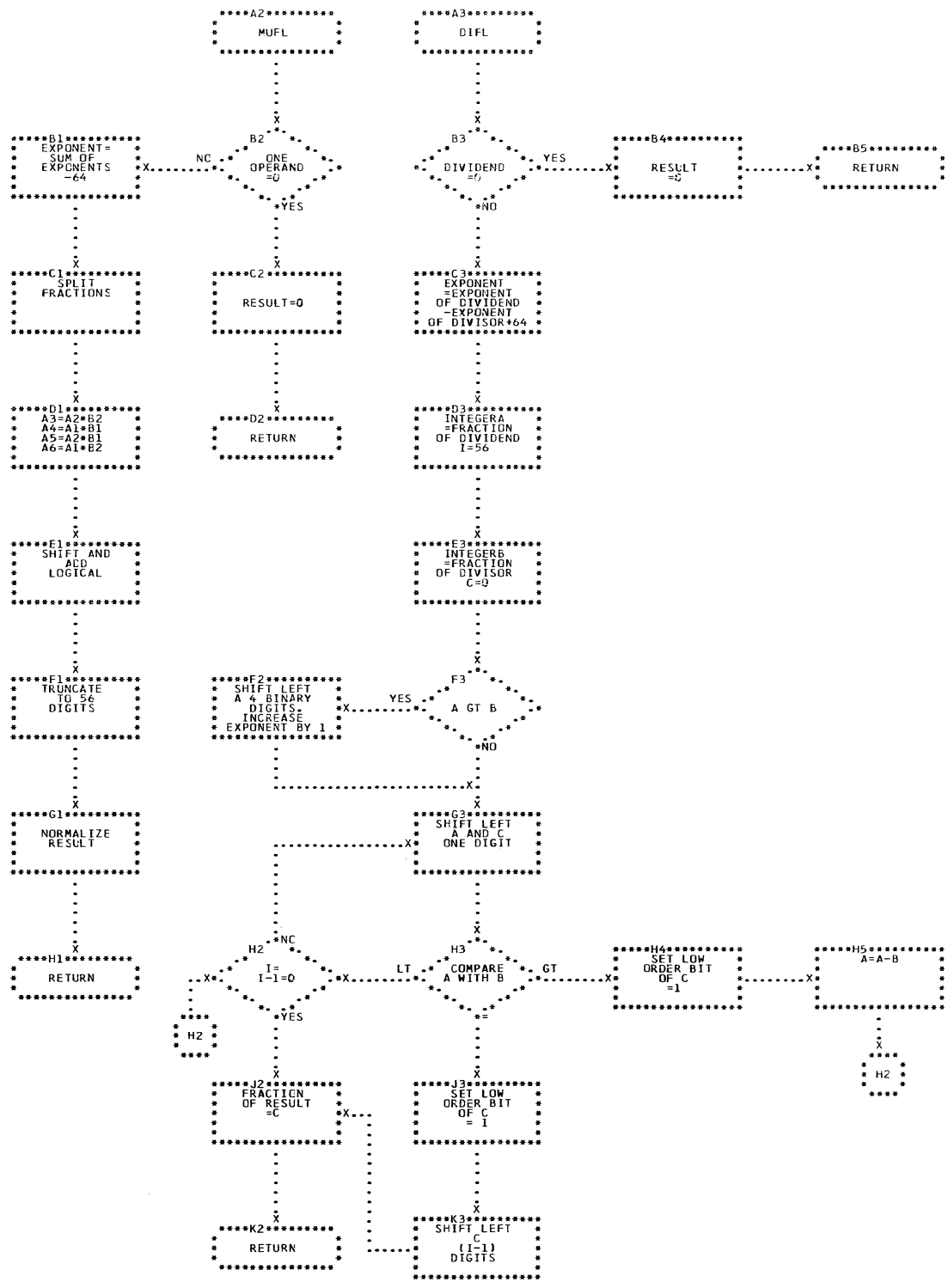
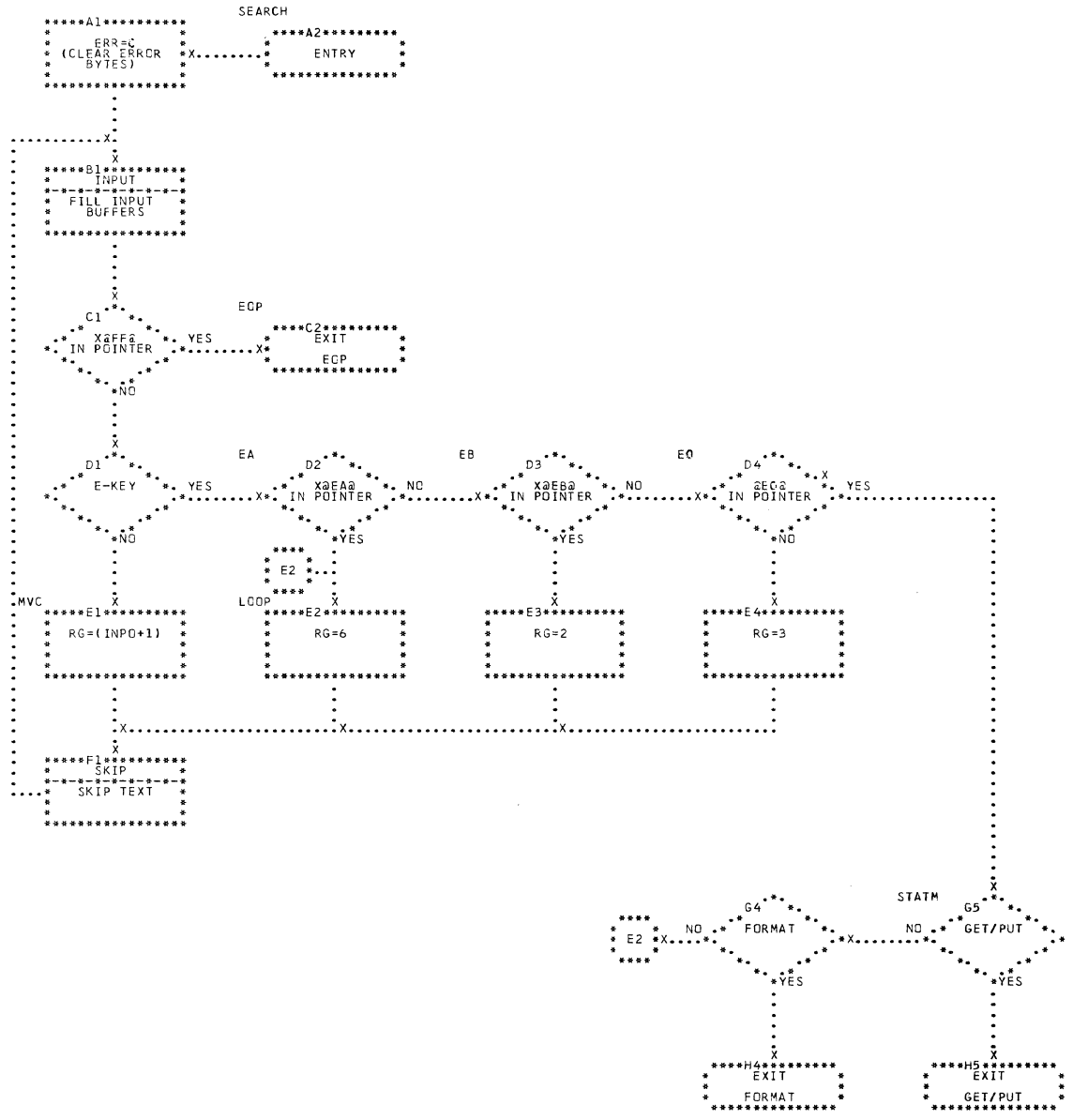
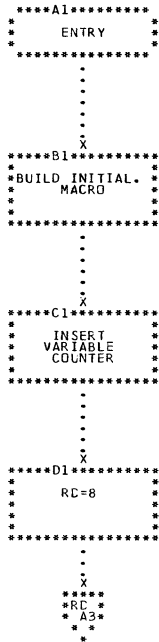


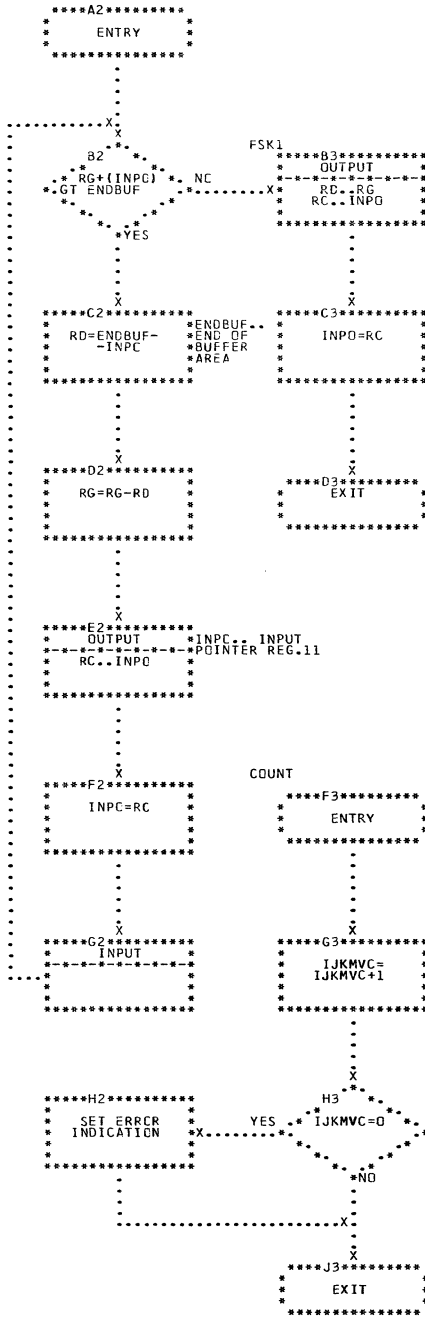
CHART QL. IJXD70 SIMULATION OF MULTIPLICATION



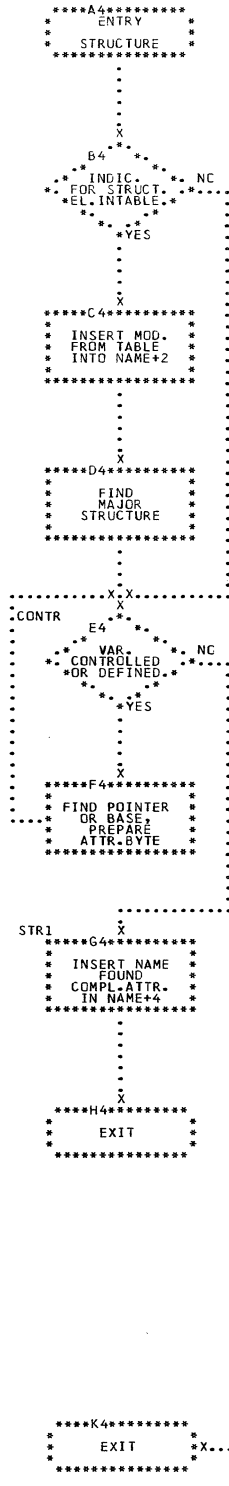
GENIN



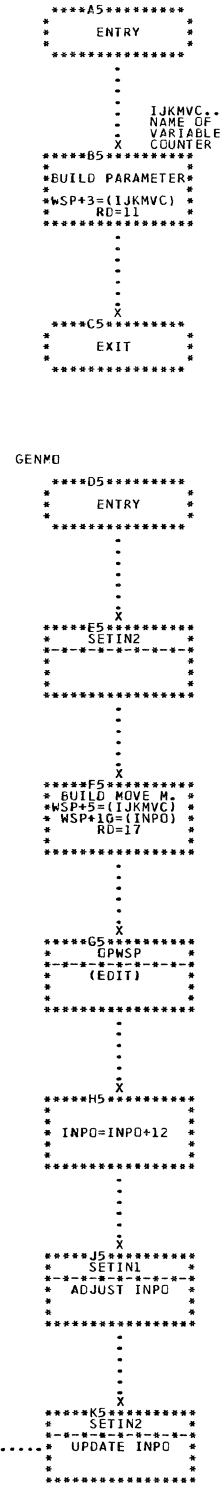
SKIP

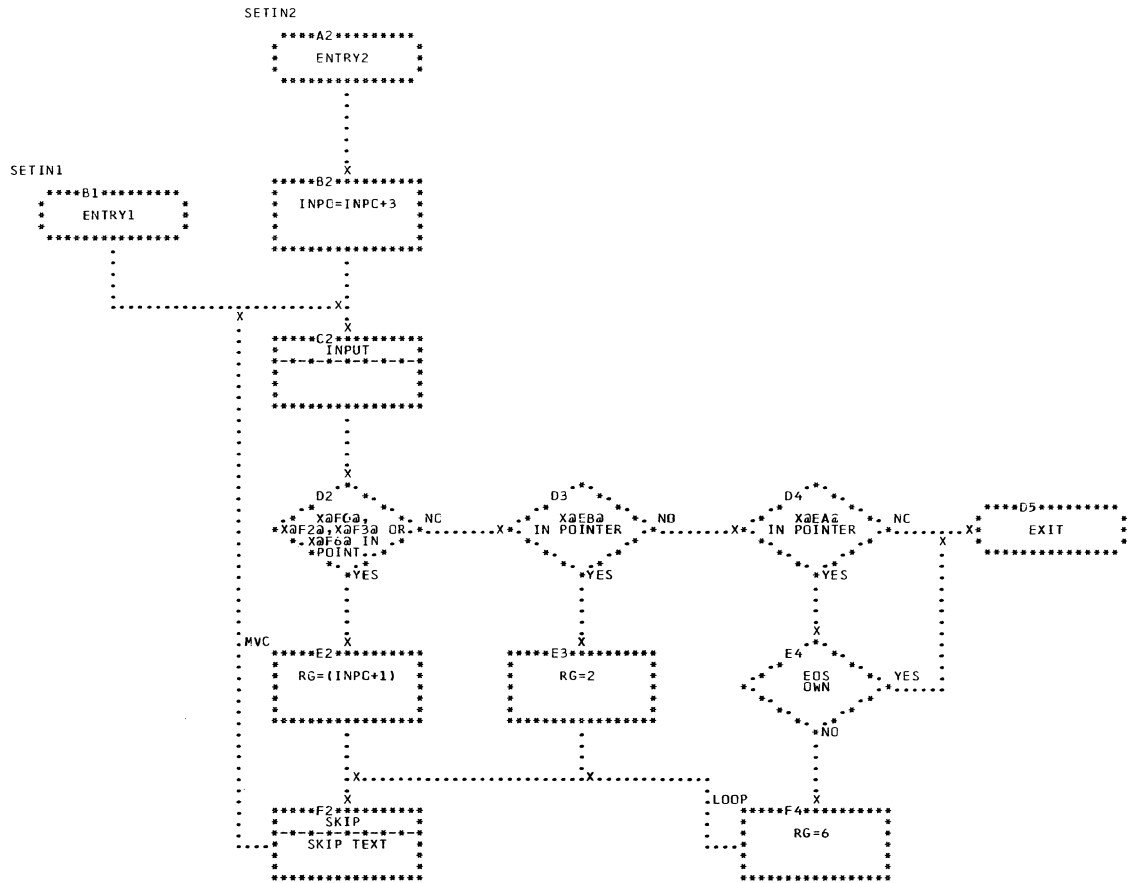


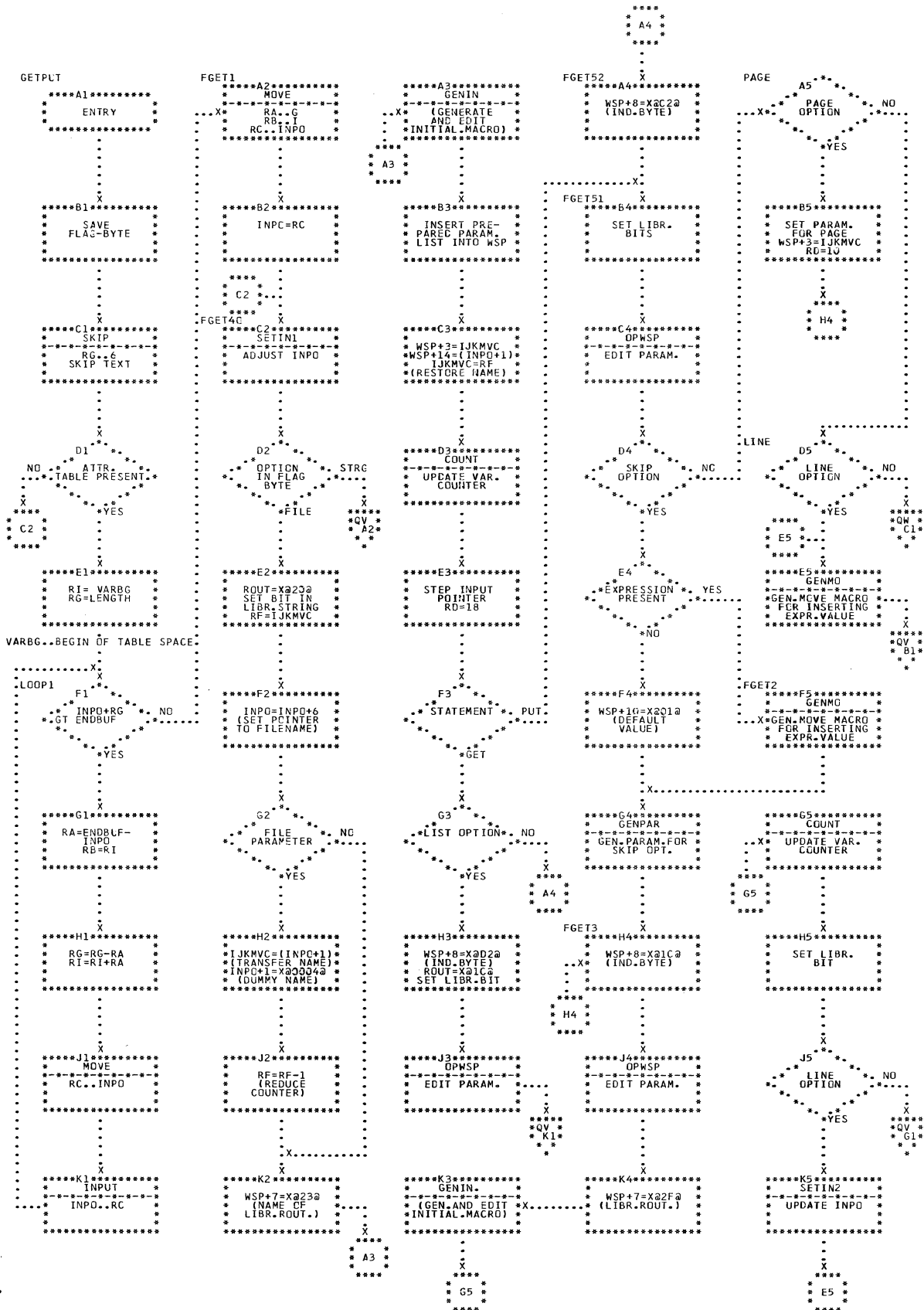
STR



GENPAR







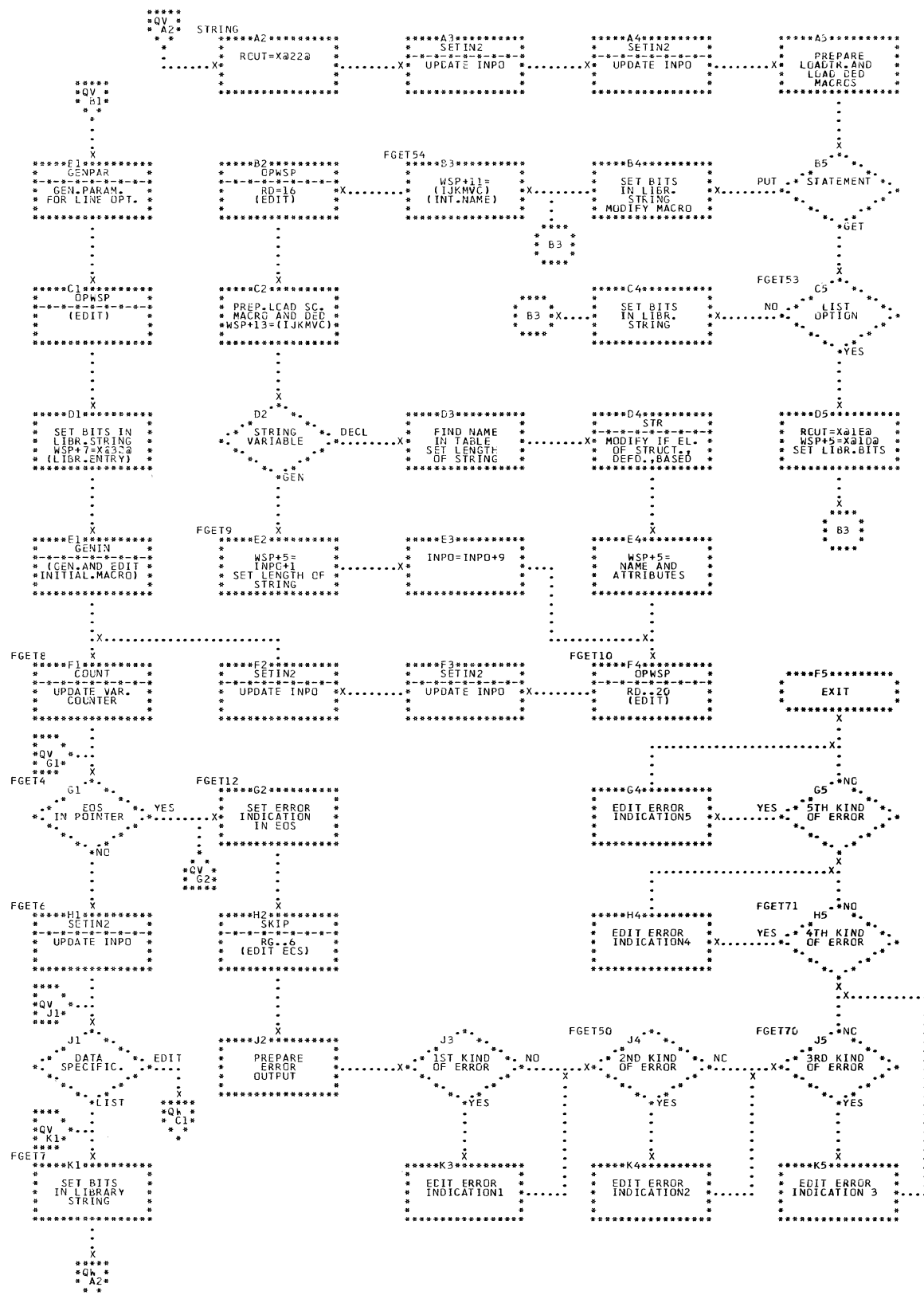


CHART QV. IJXD75

GET/PUT2

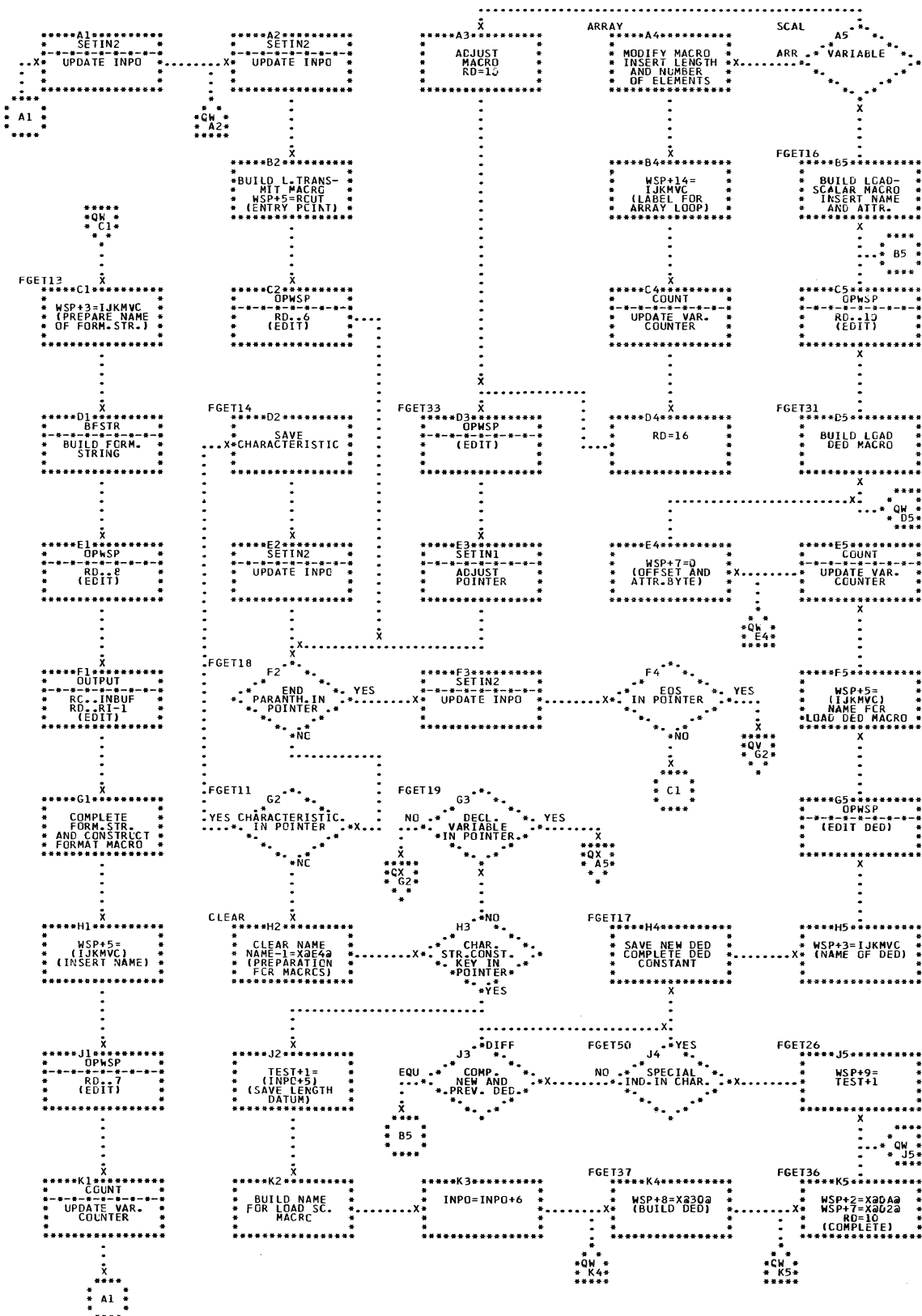


CHART QW. IJXD75 GET/PUT3

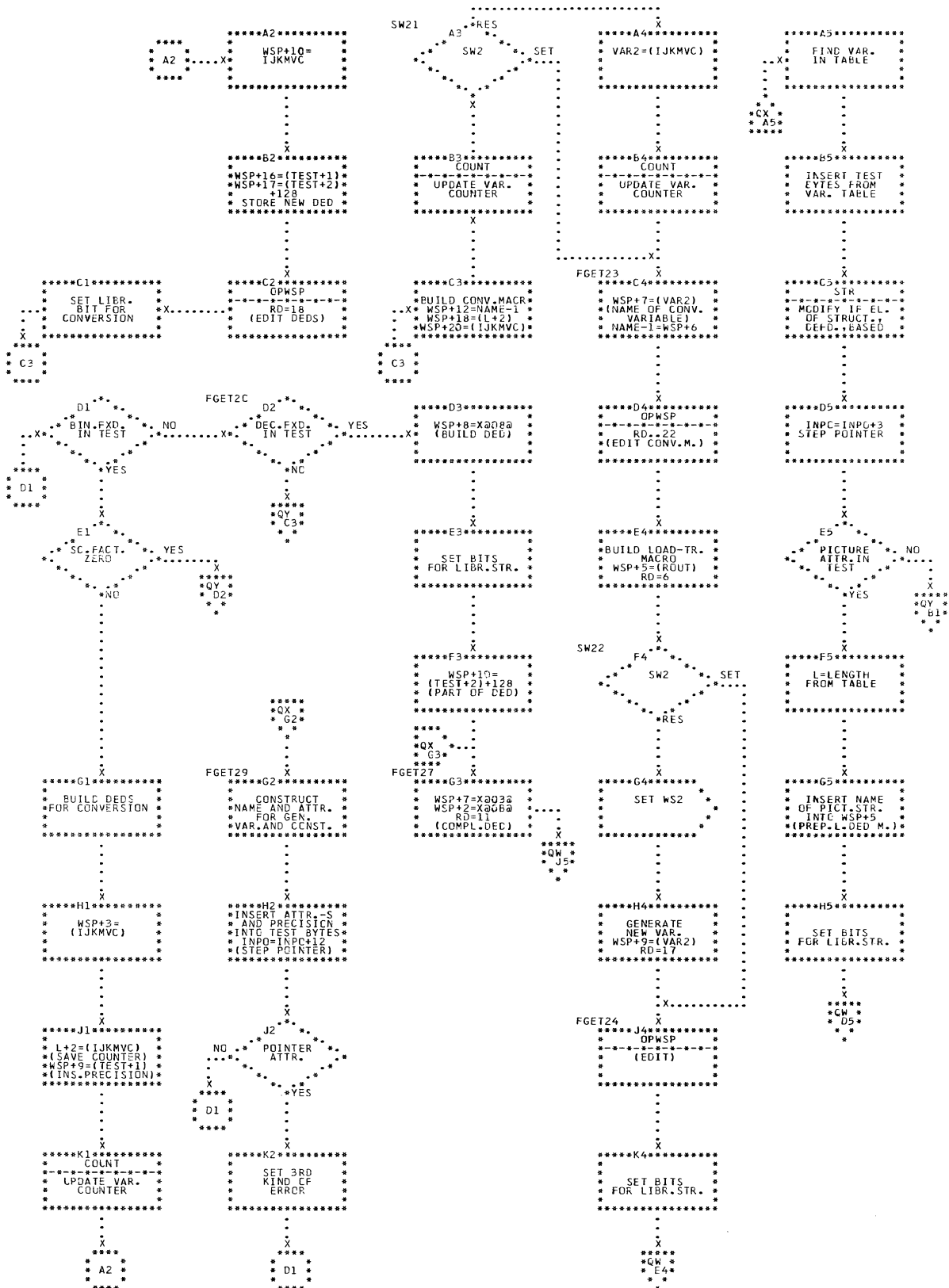
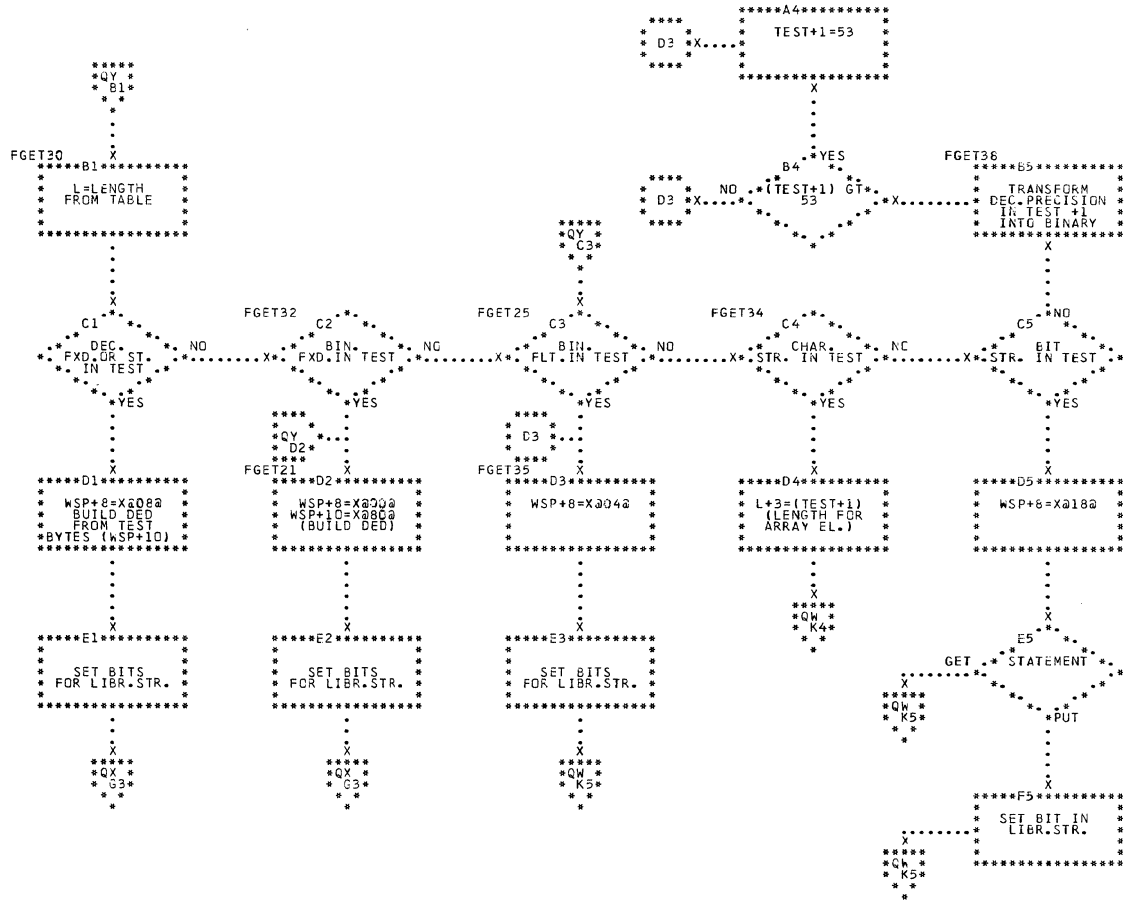
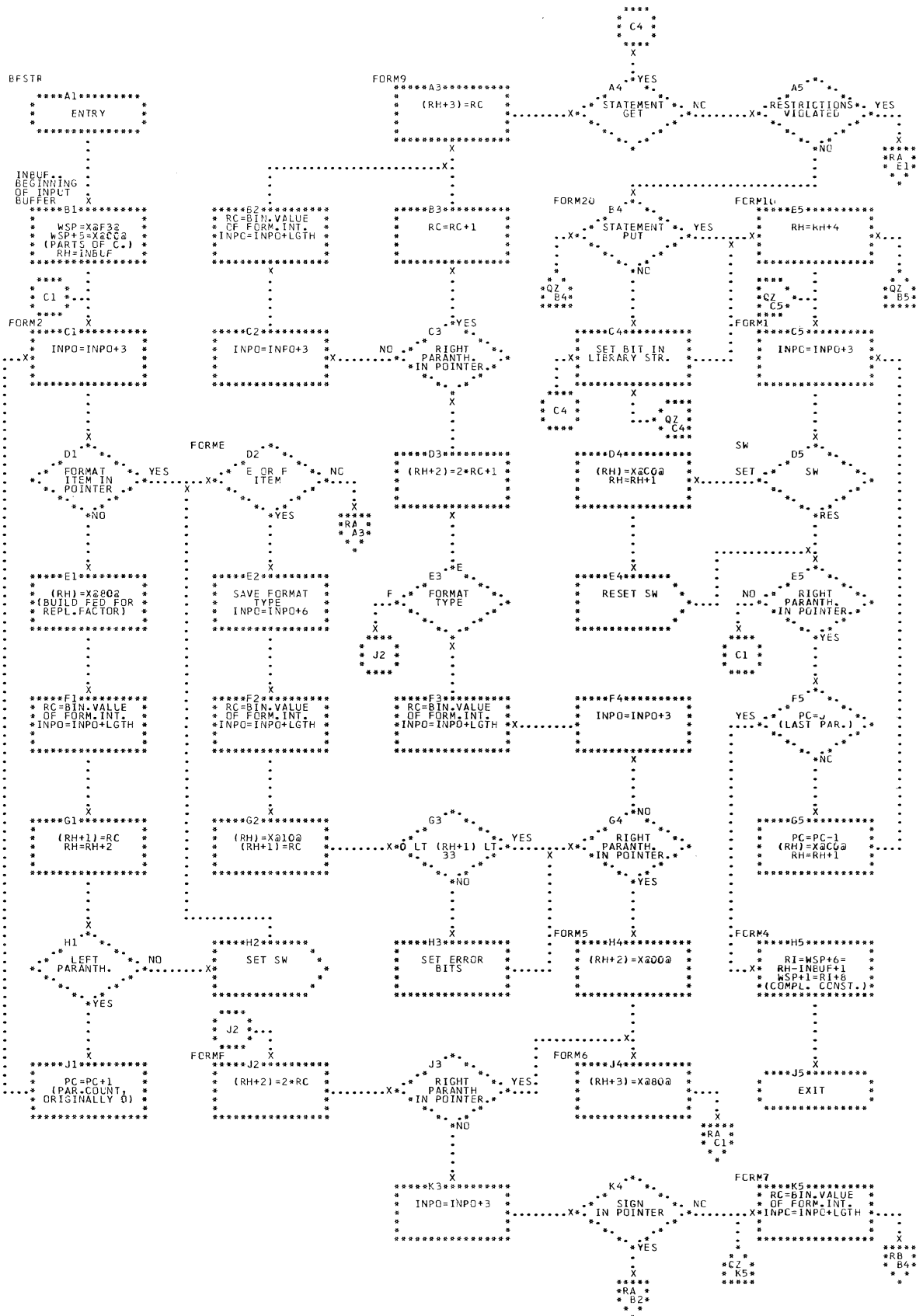


CHART QX. IJXD75

GET/PUT4





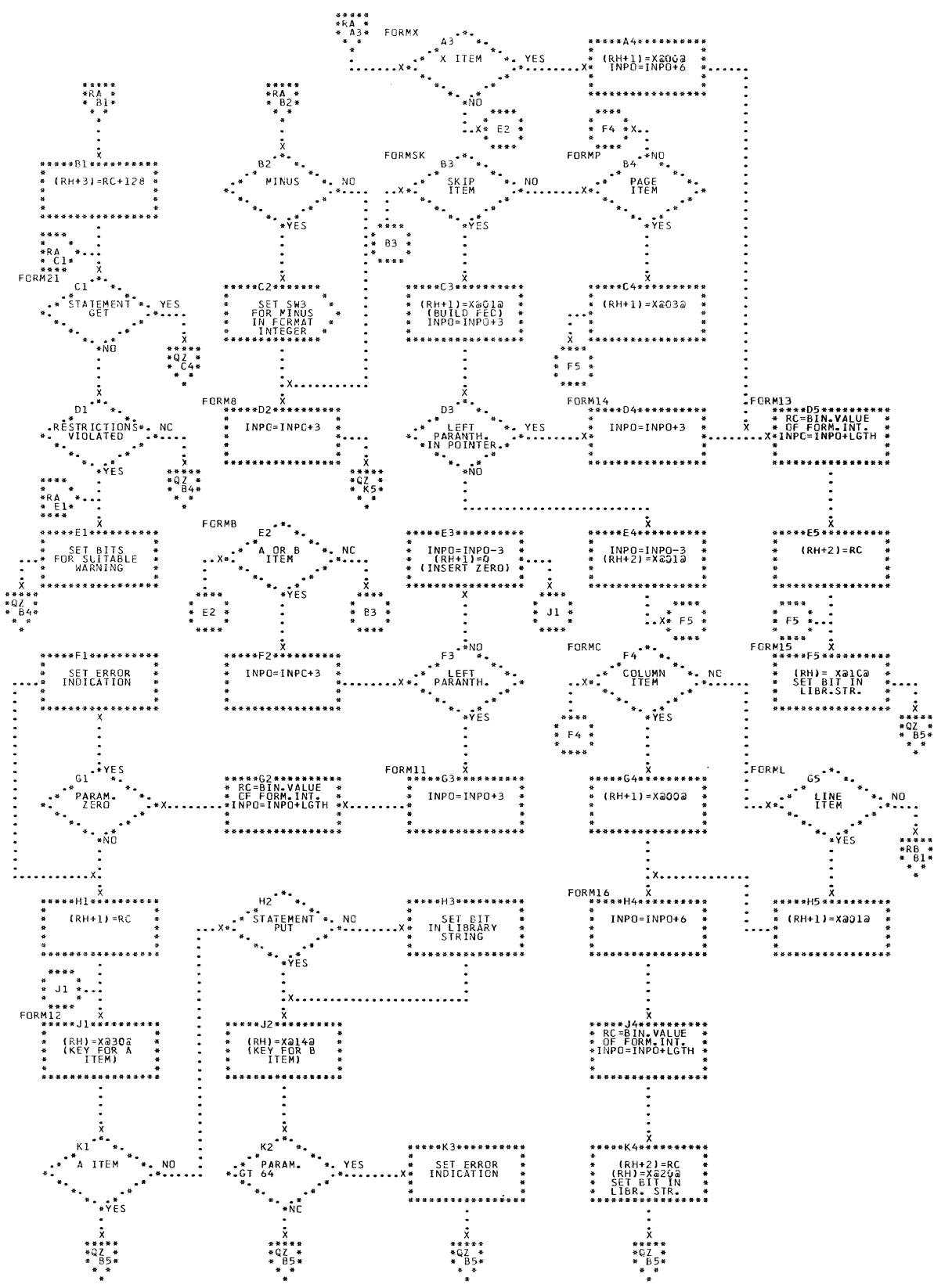


CHART RA. IJXD75 BUILD FORMAT STRING 2

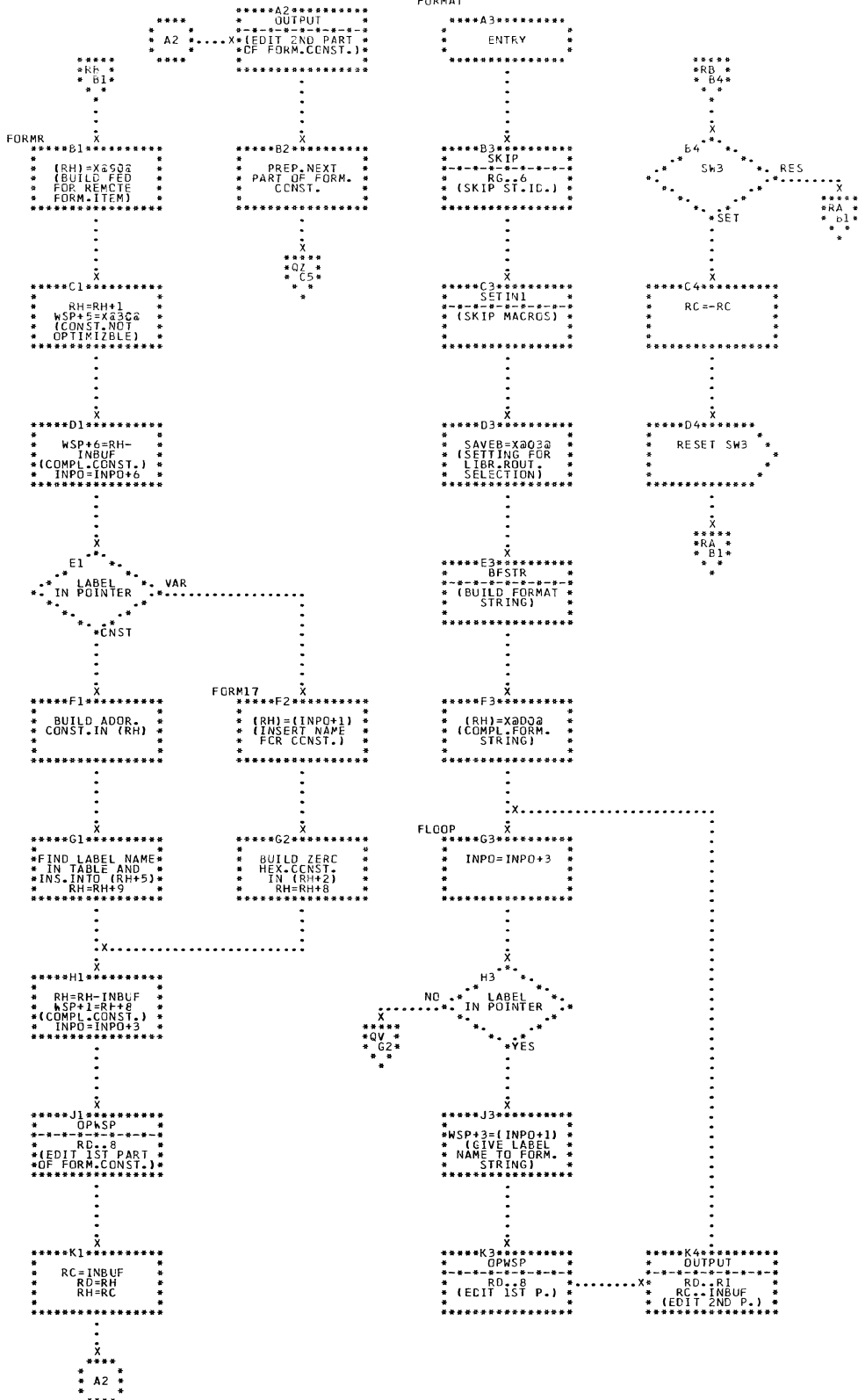
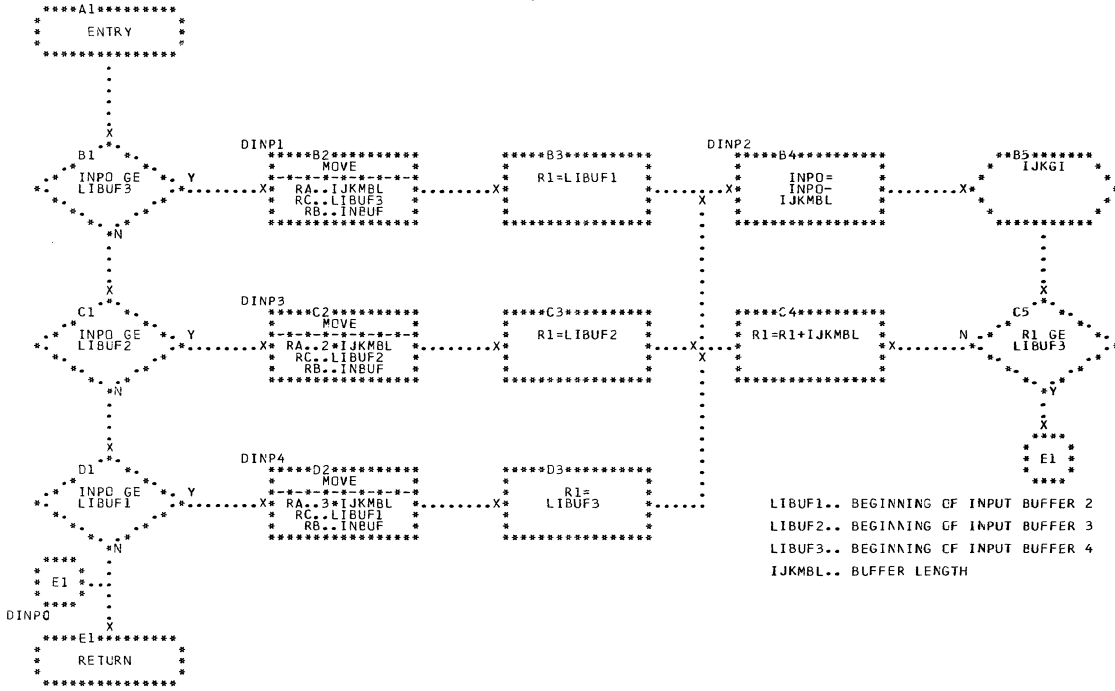
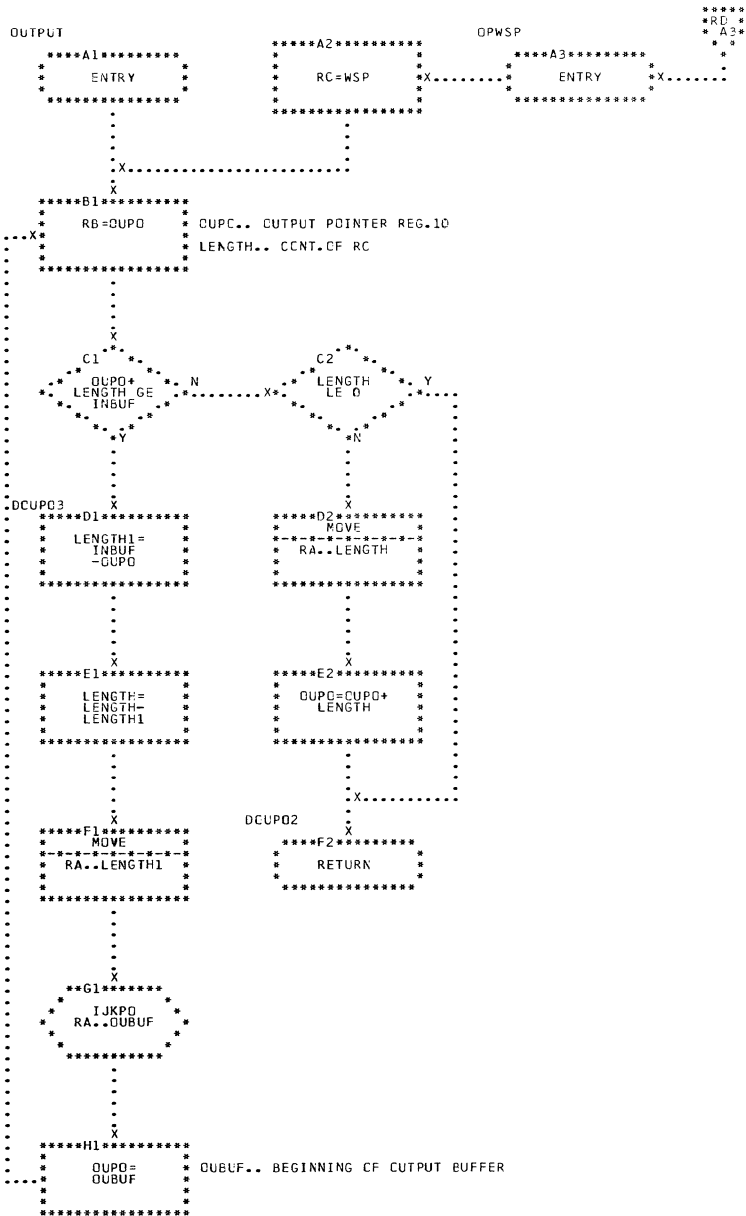


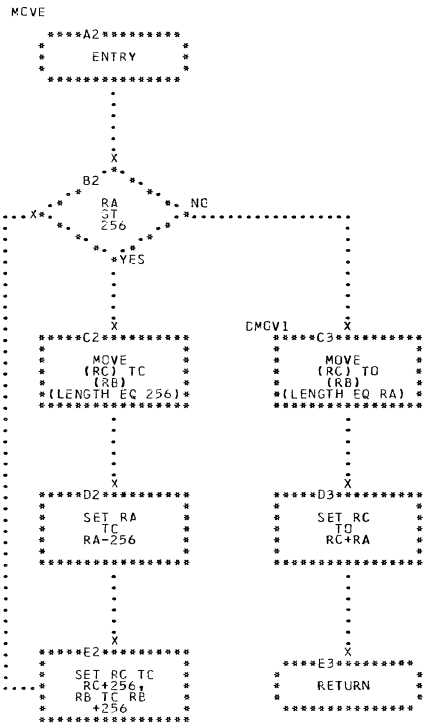
CHART RB. IJXD75

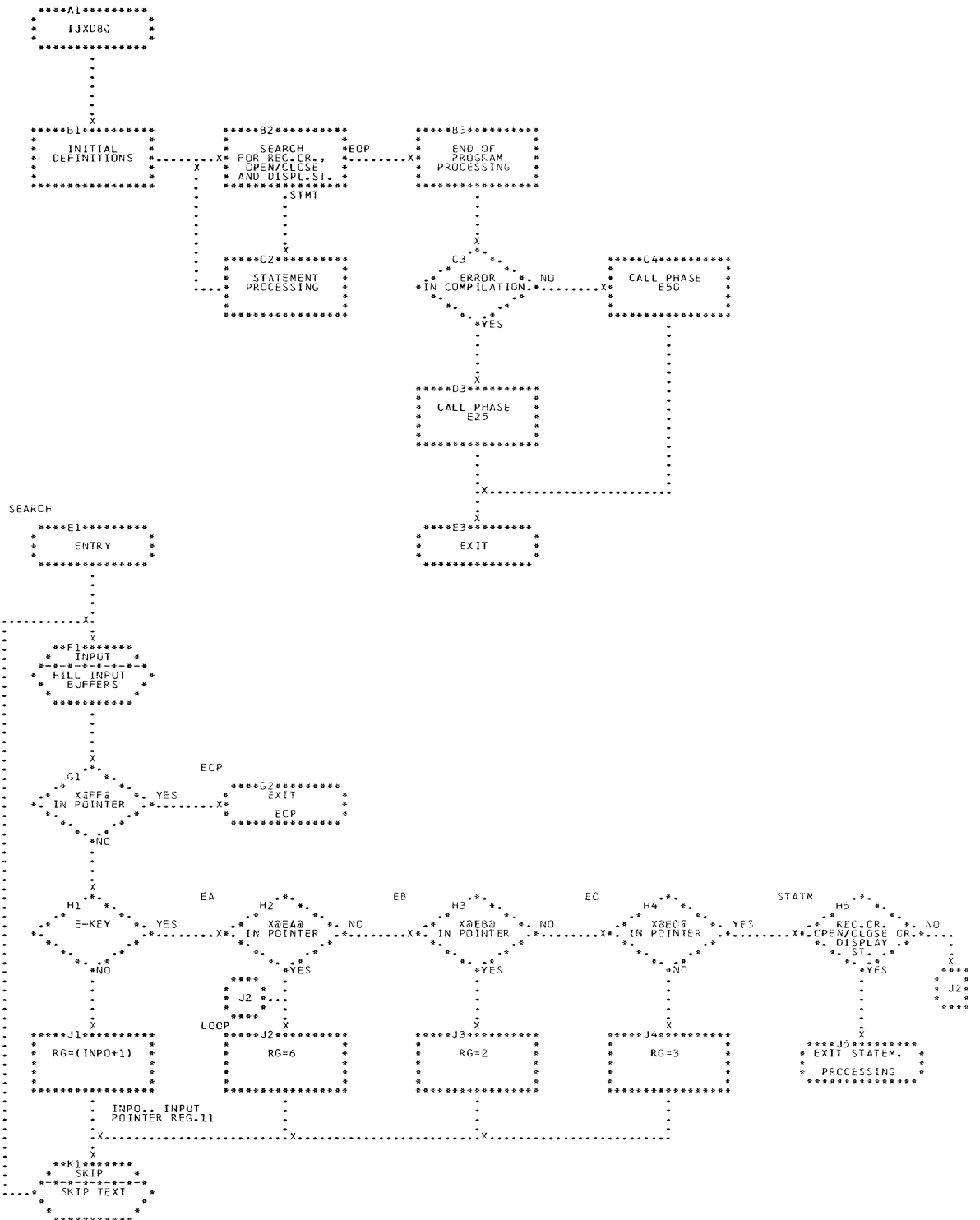
BUILD FORMAT STRING 3,FORMAT

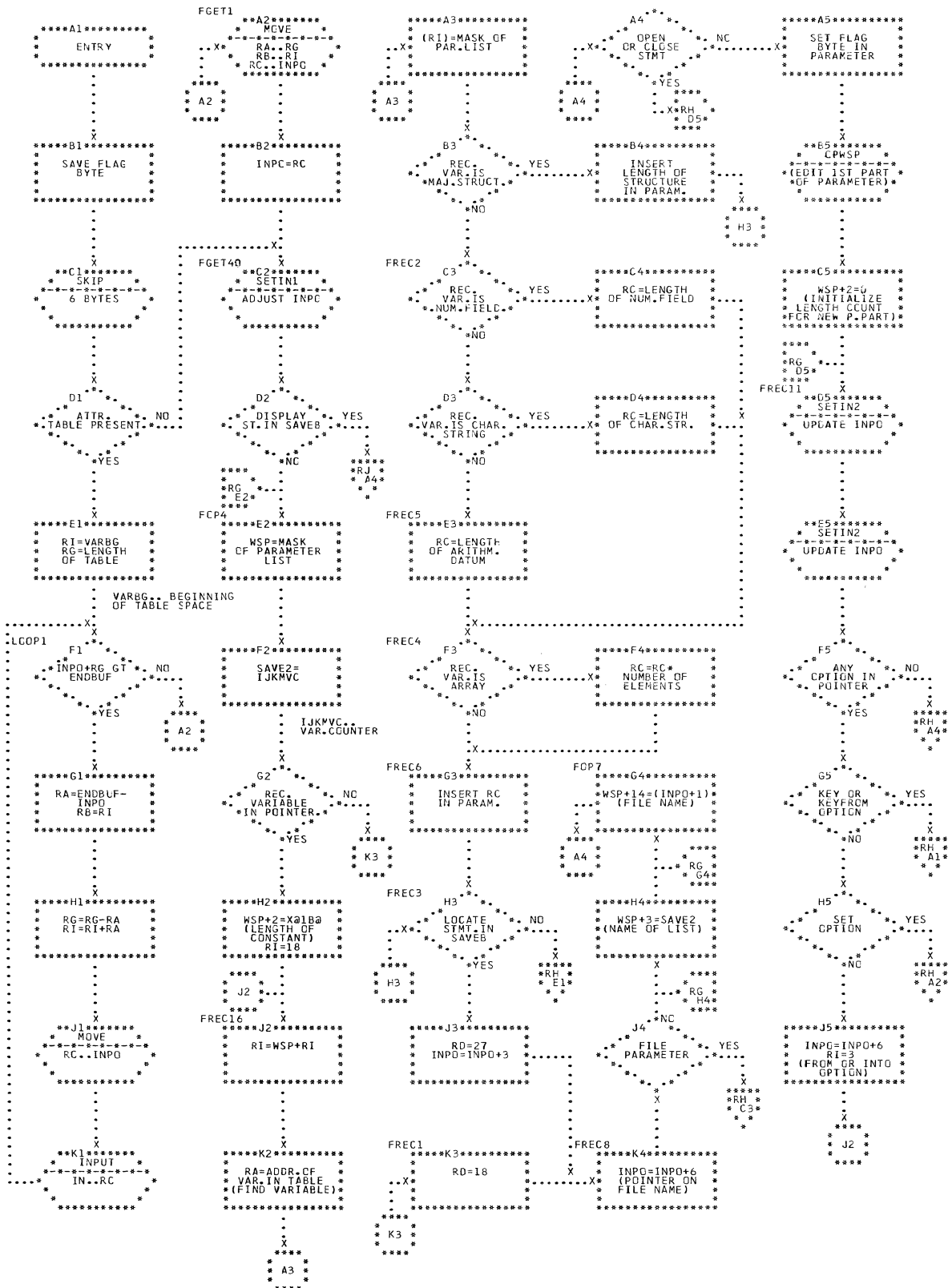
INPUT

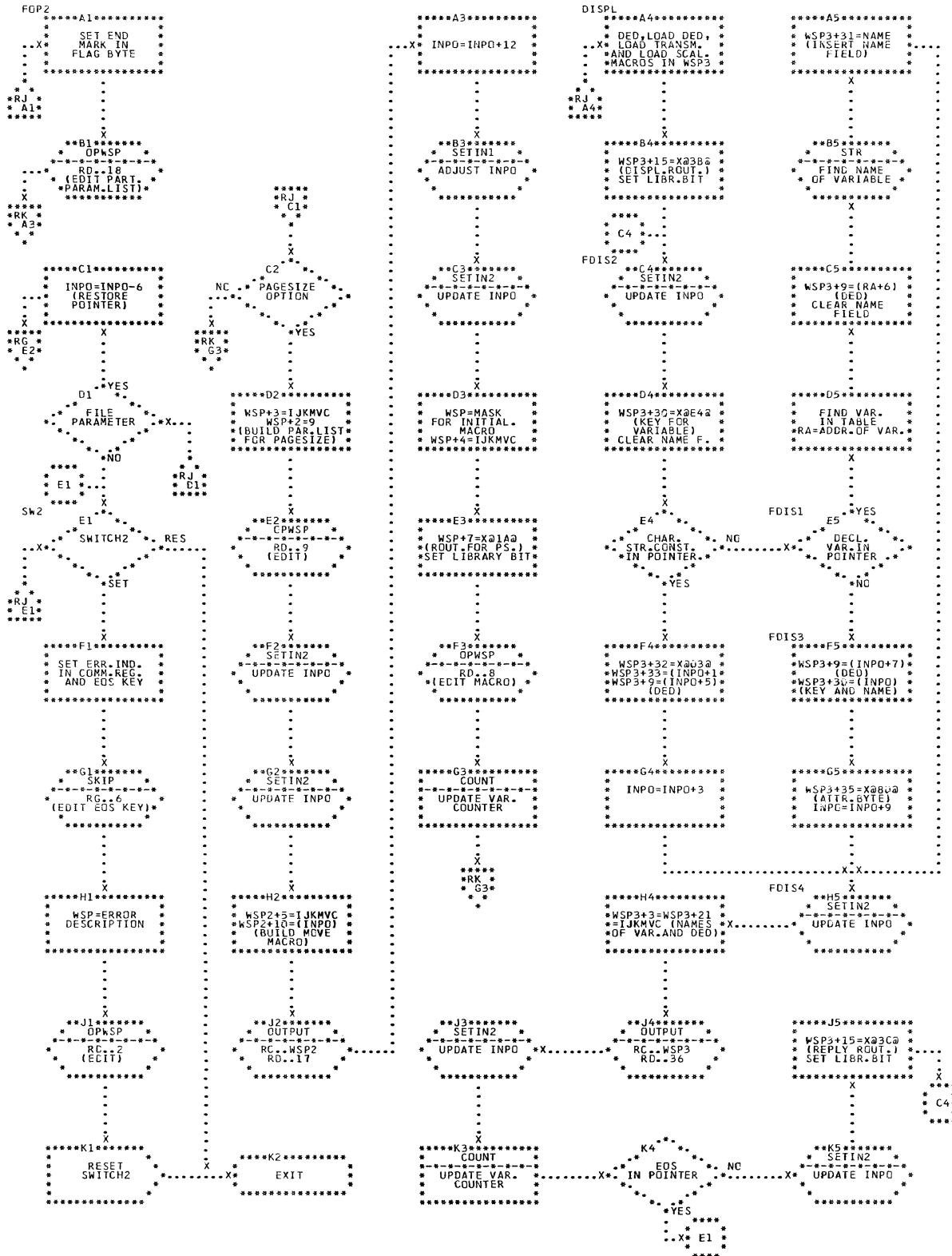












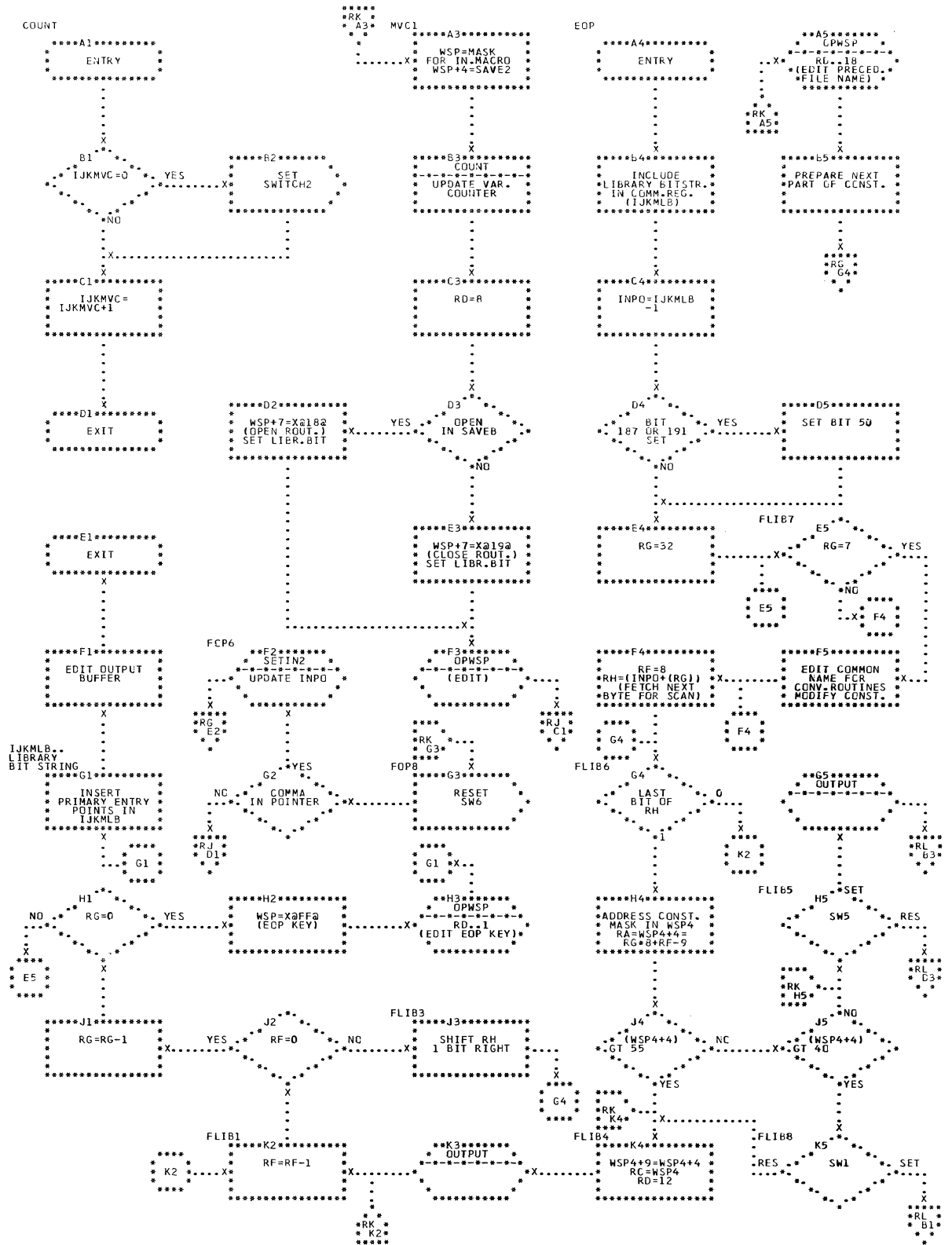
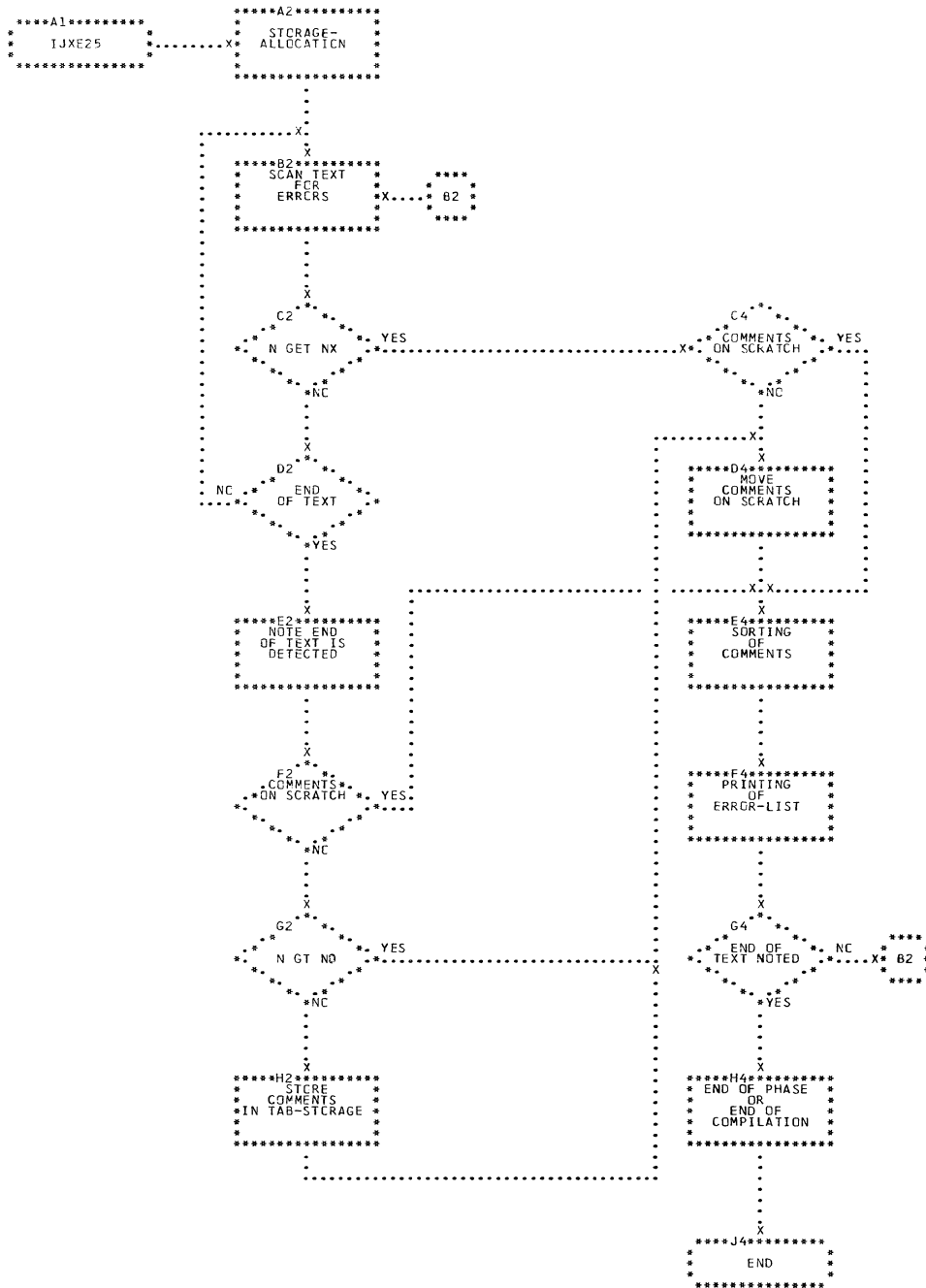
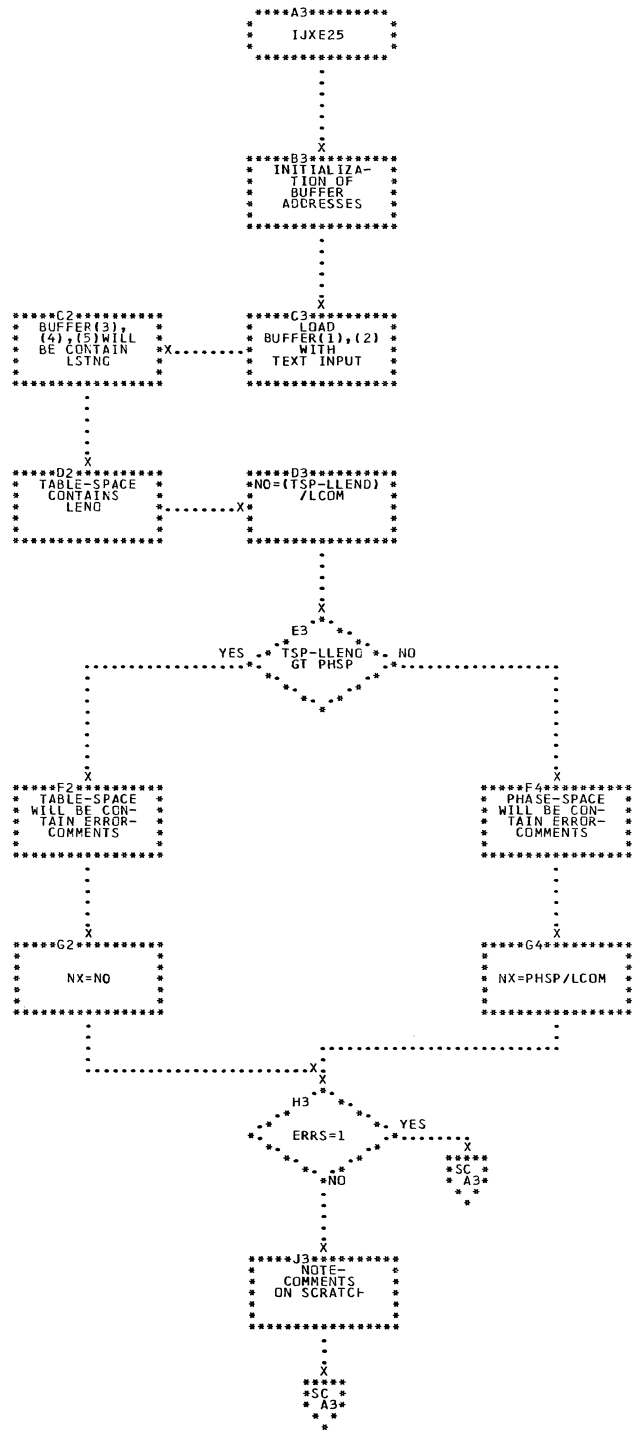
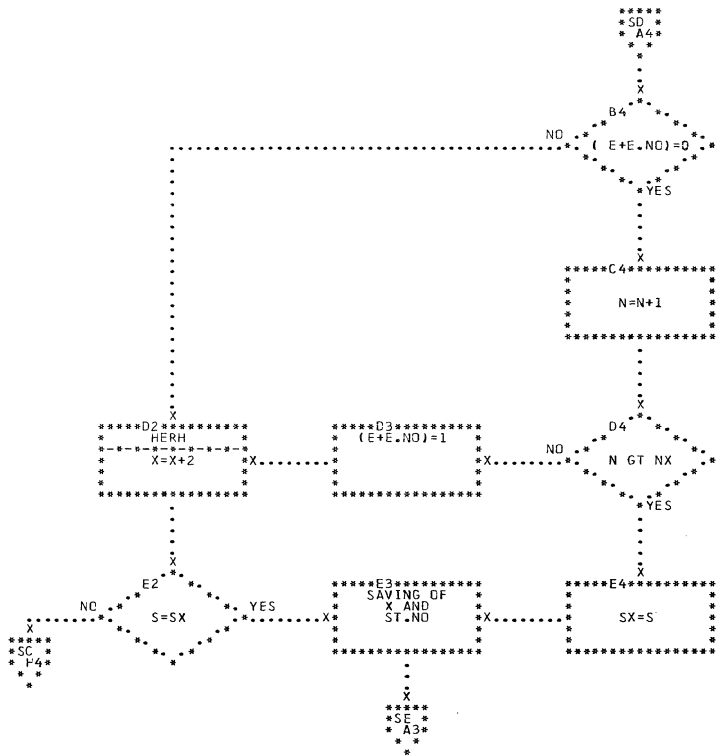


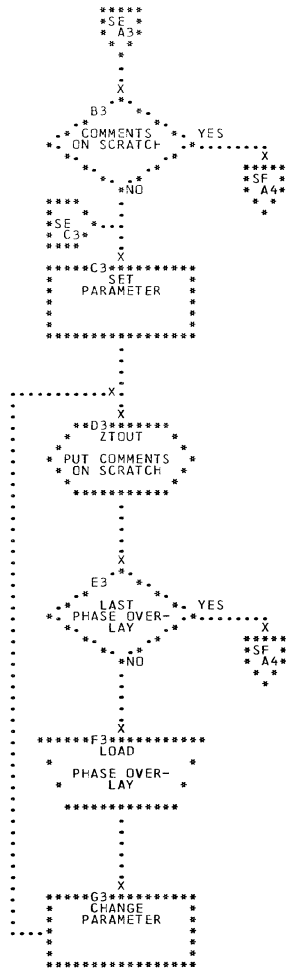
CHART RK. IJXD80

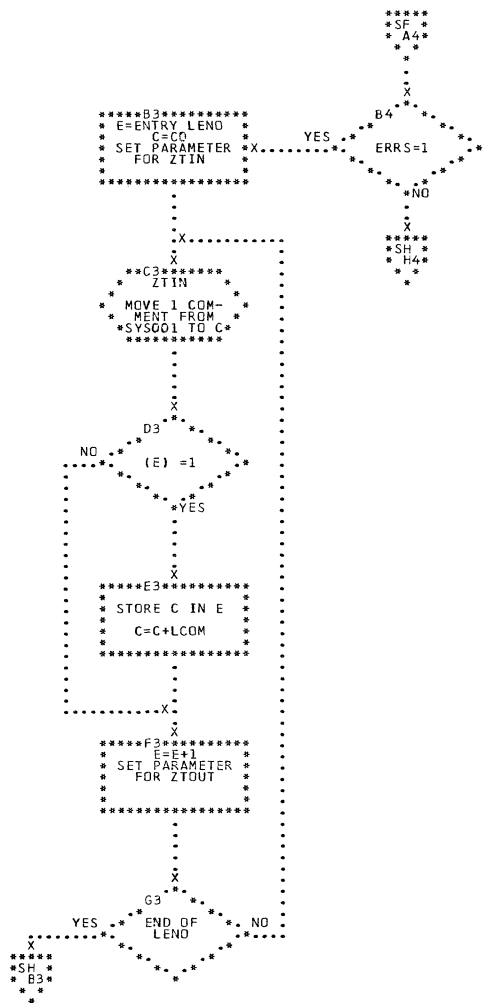
COUNT, ST.PROC.4, EOP PROC.1

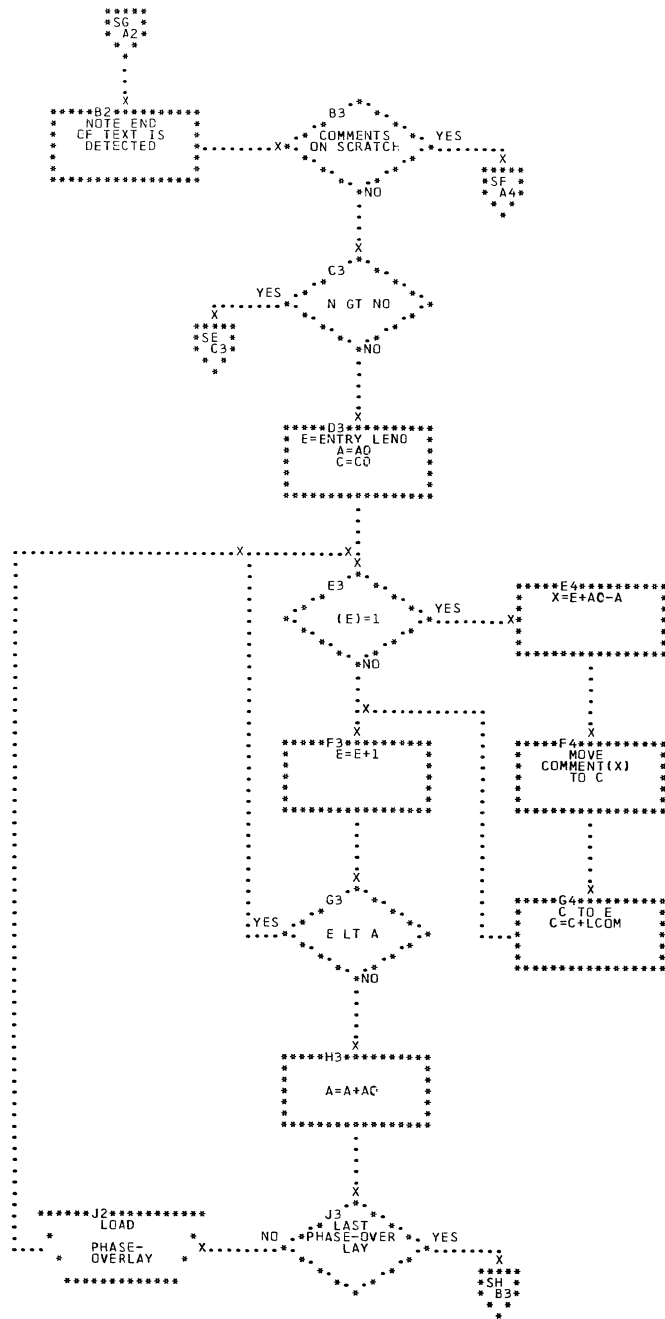


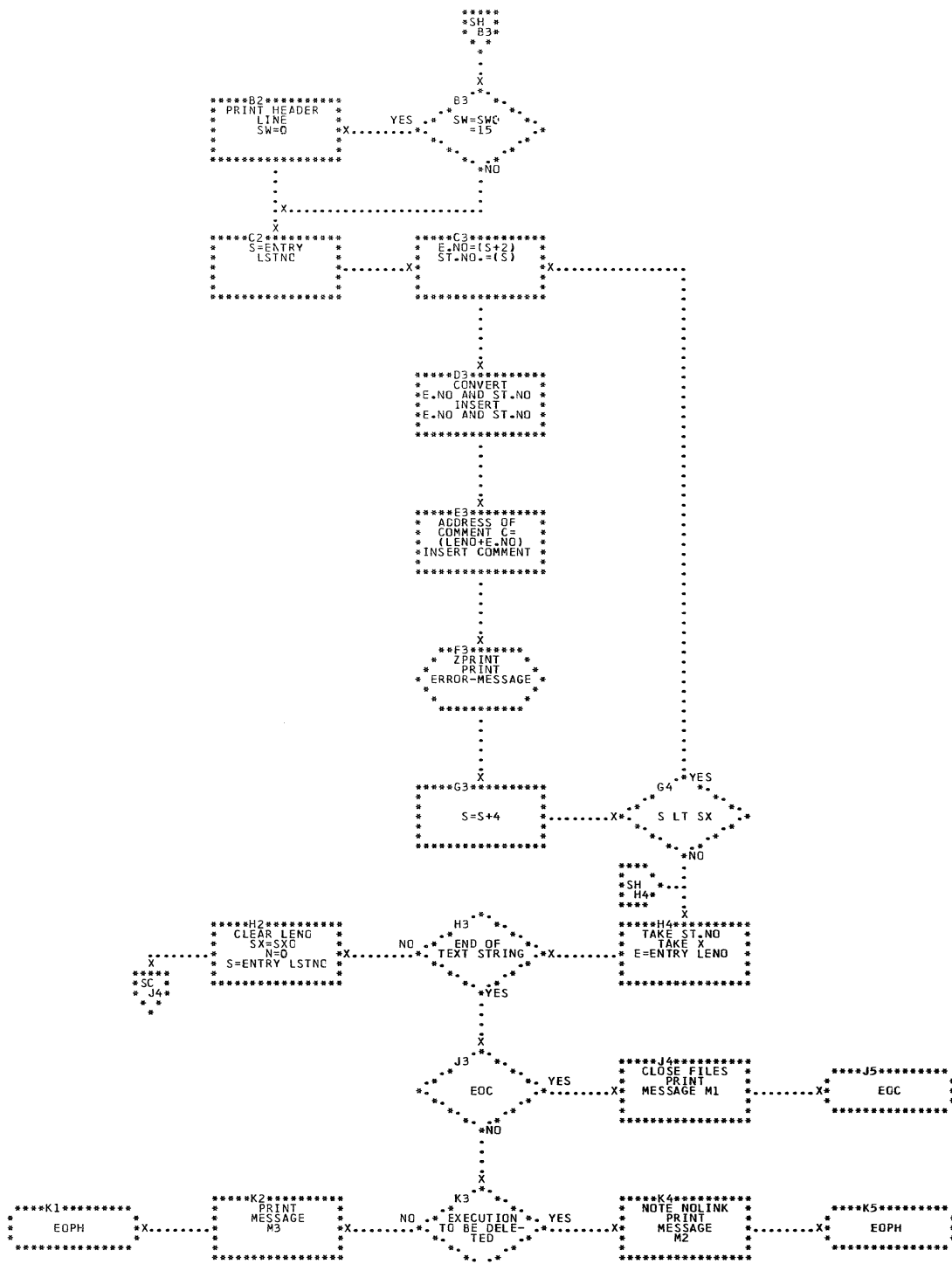


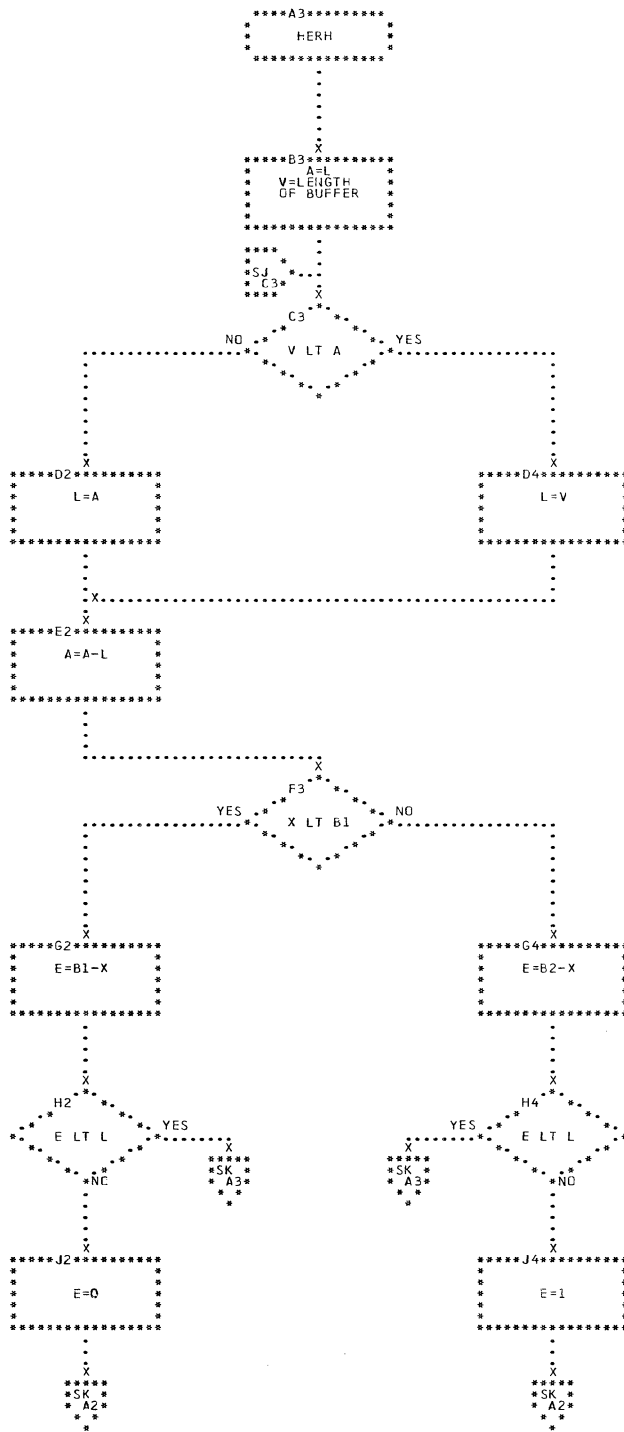












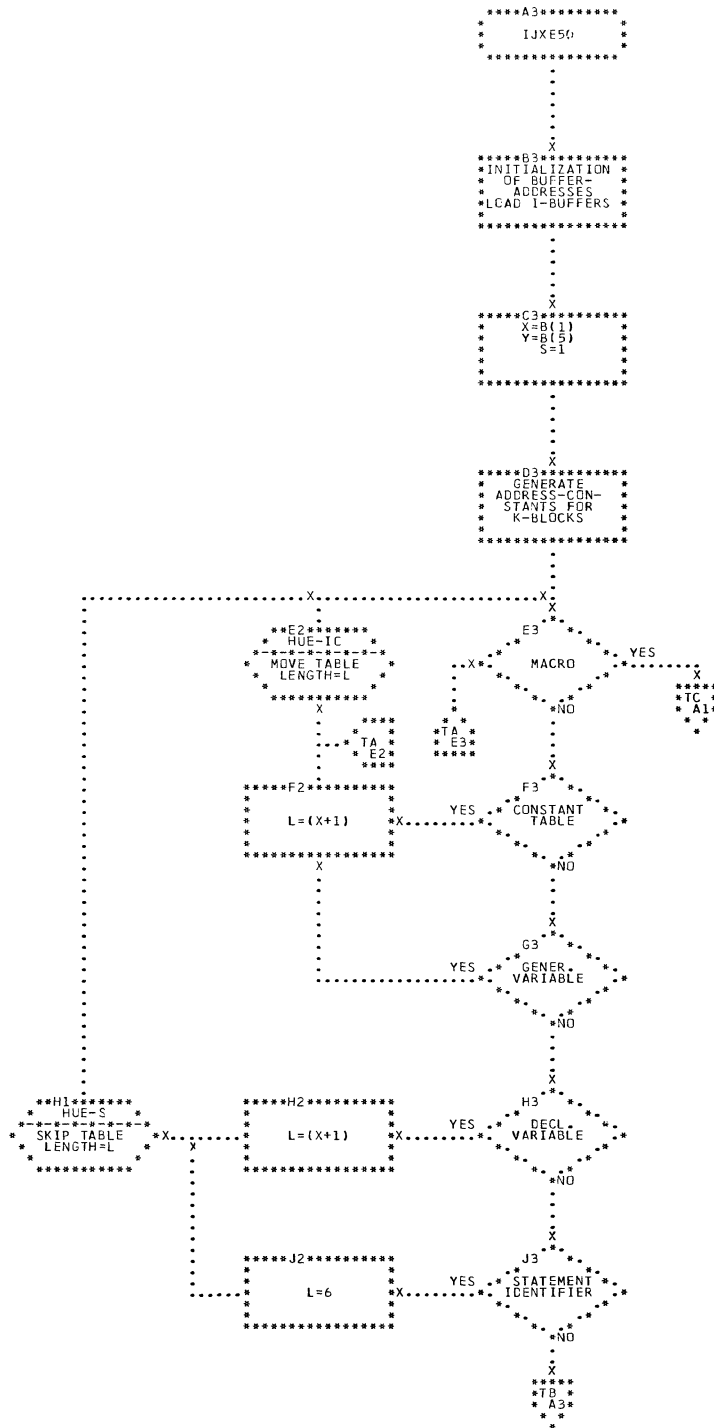
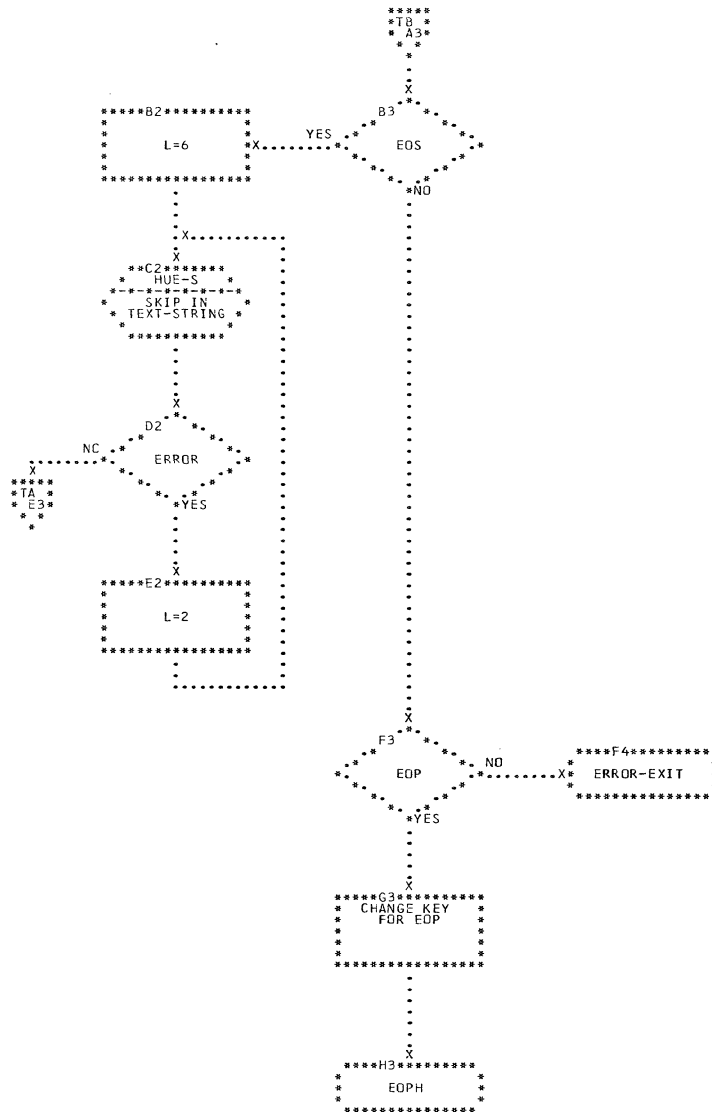
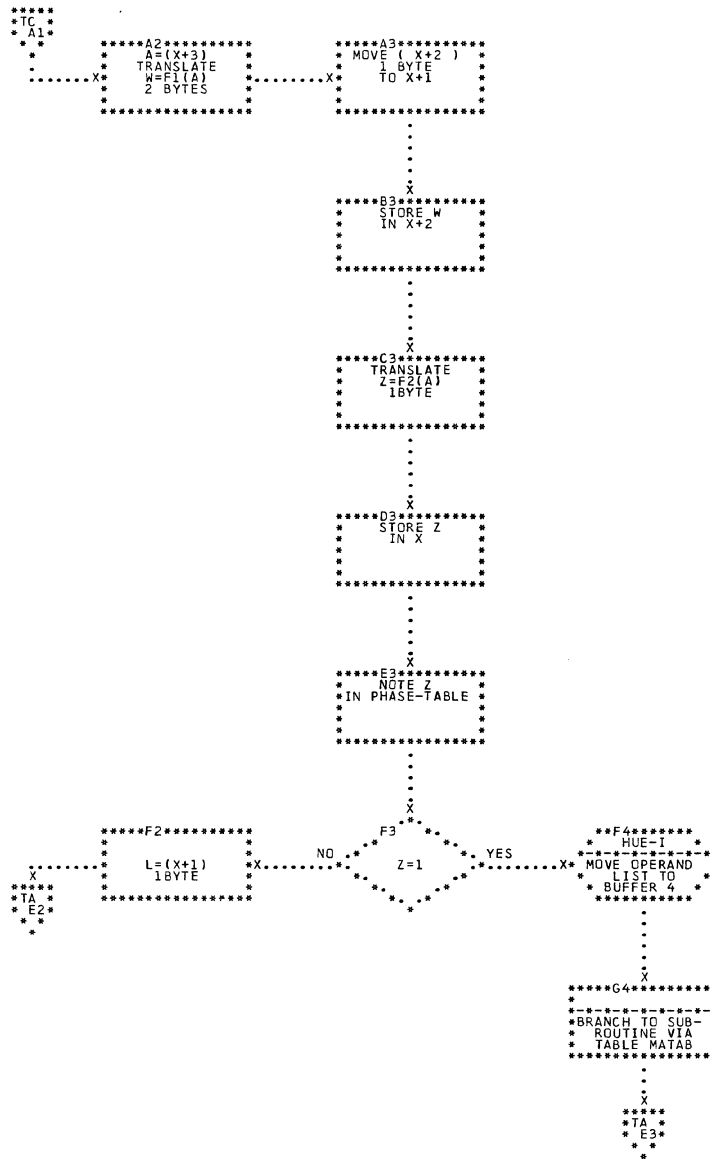
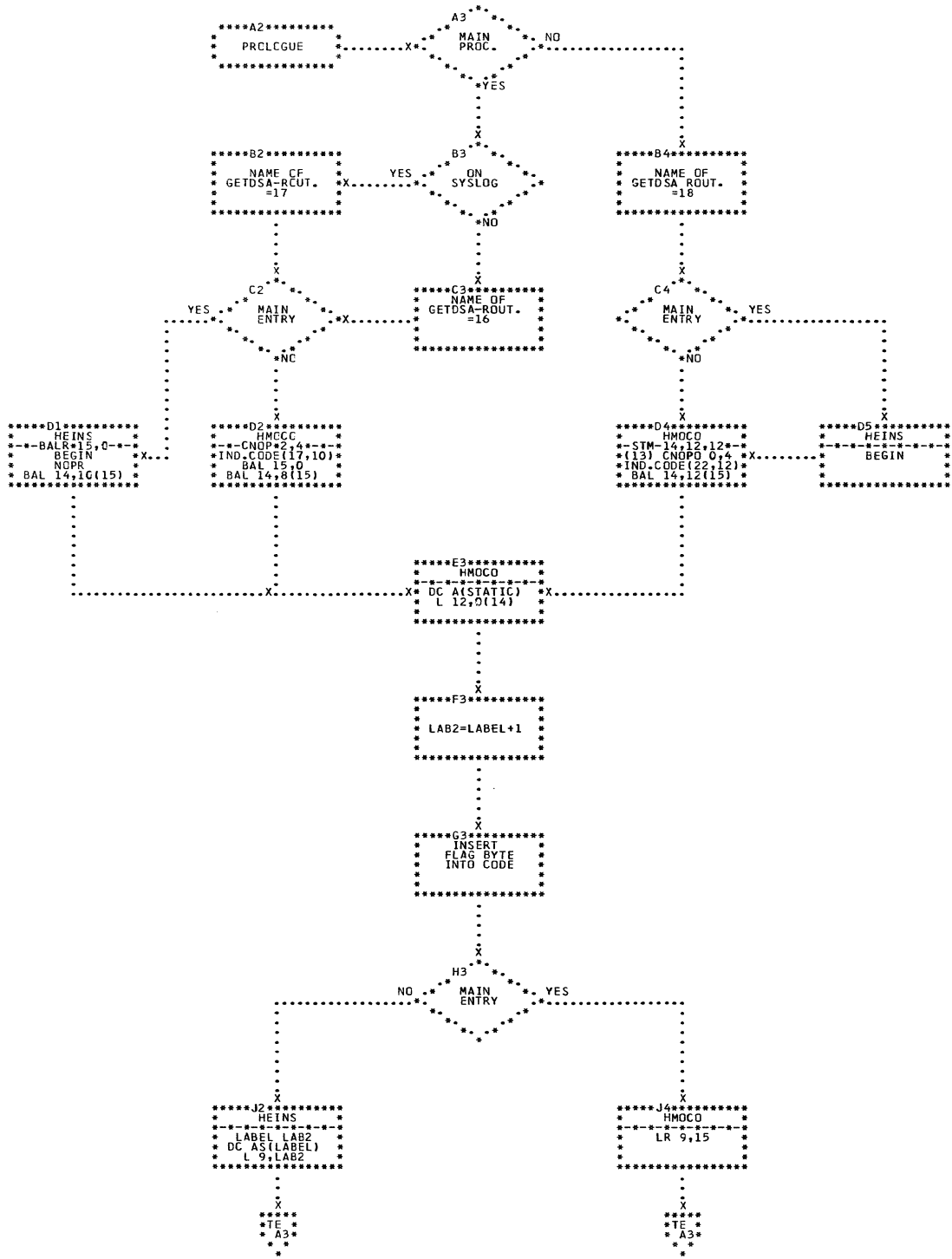


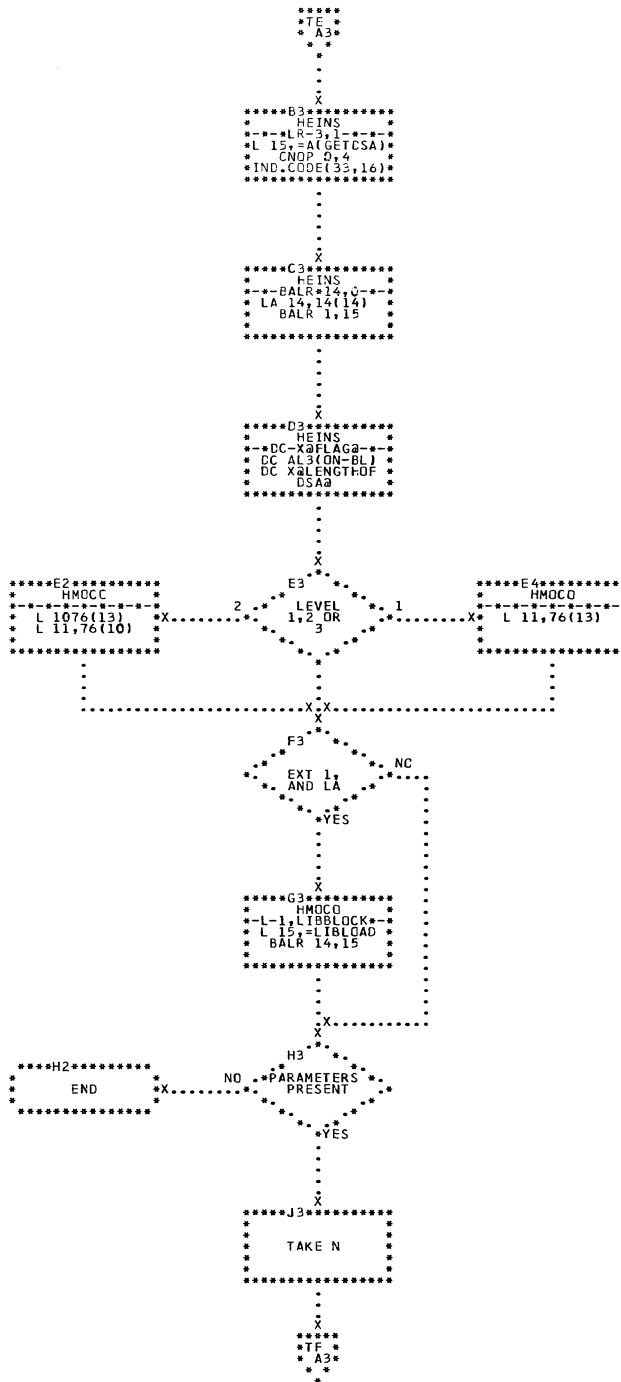
CHART TA. IJXE50

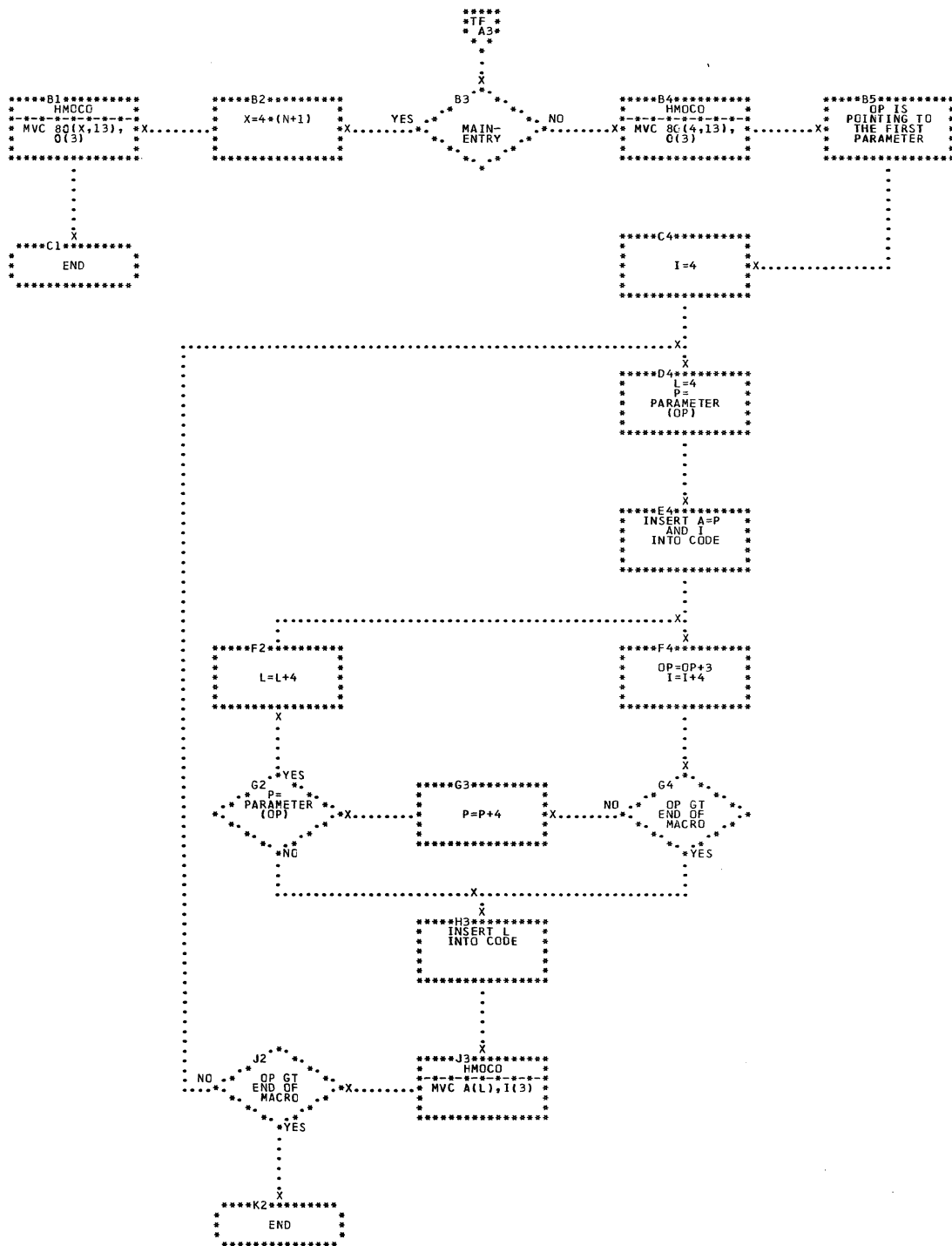
GENERAL FLOW











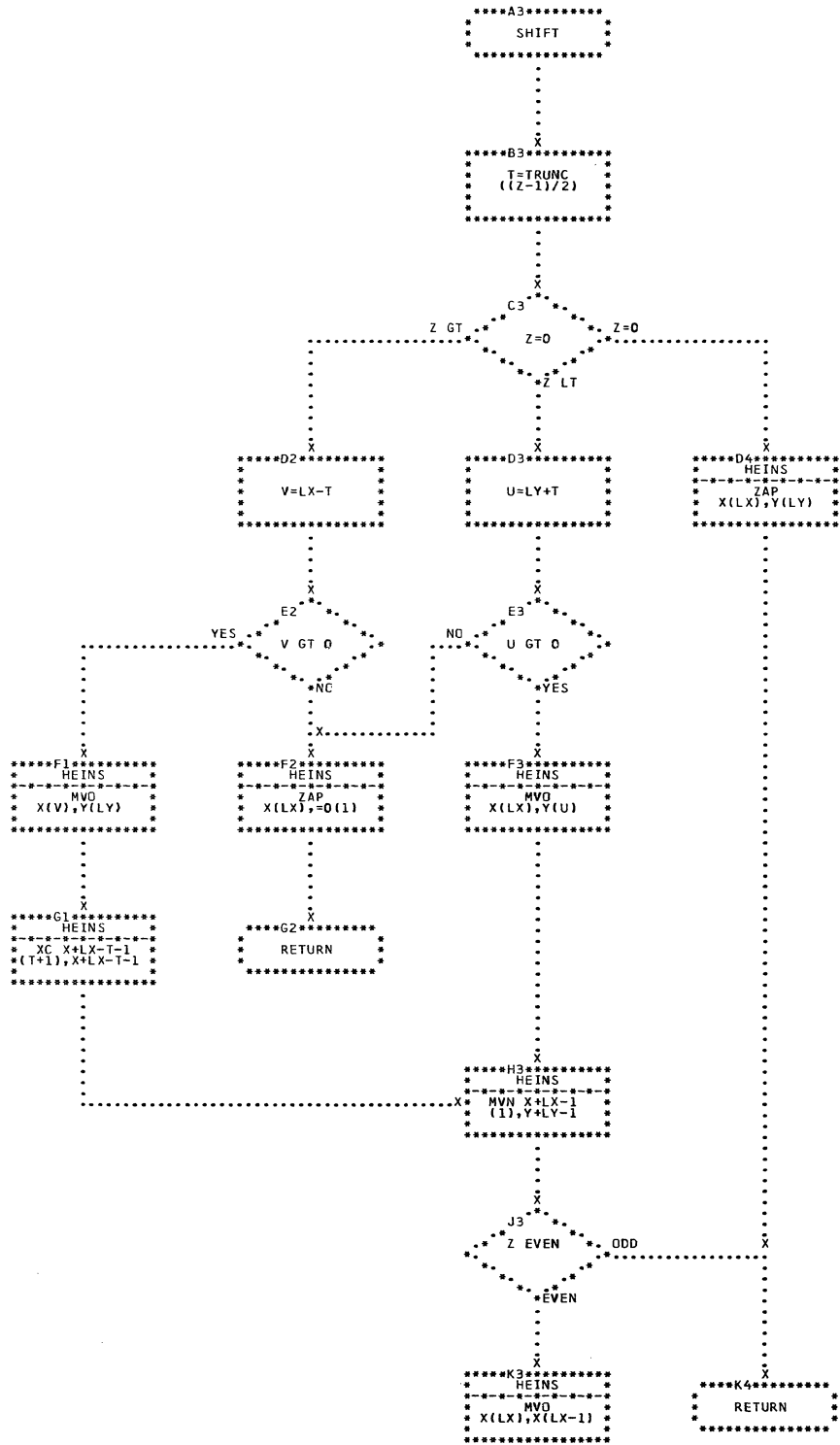
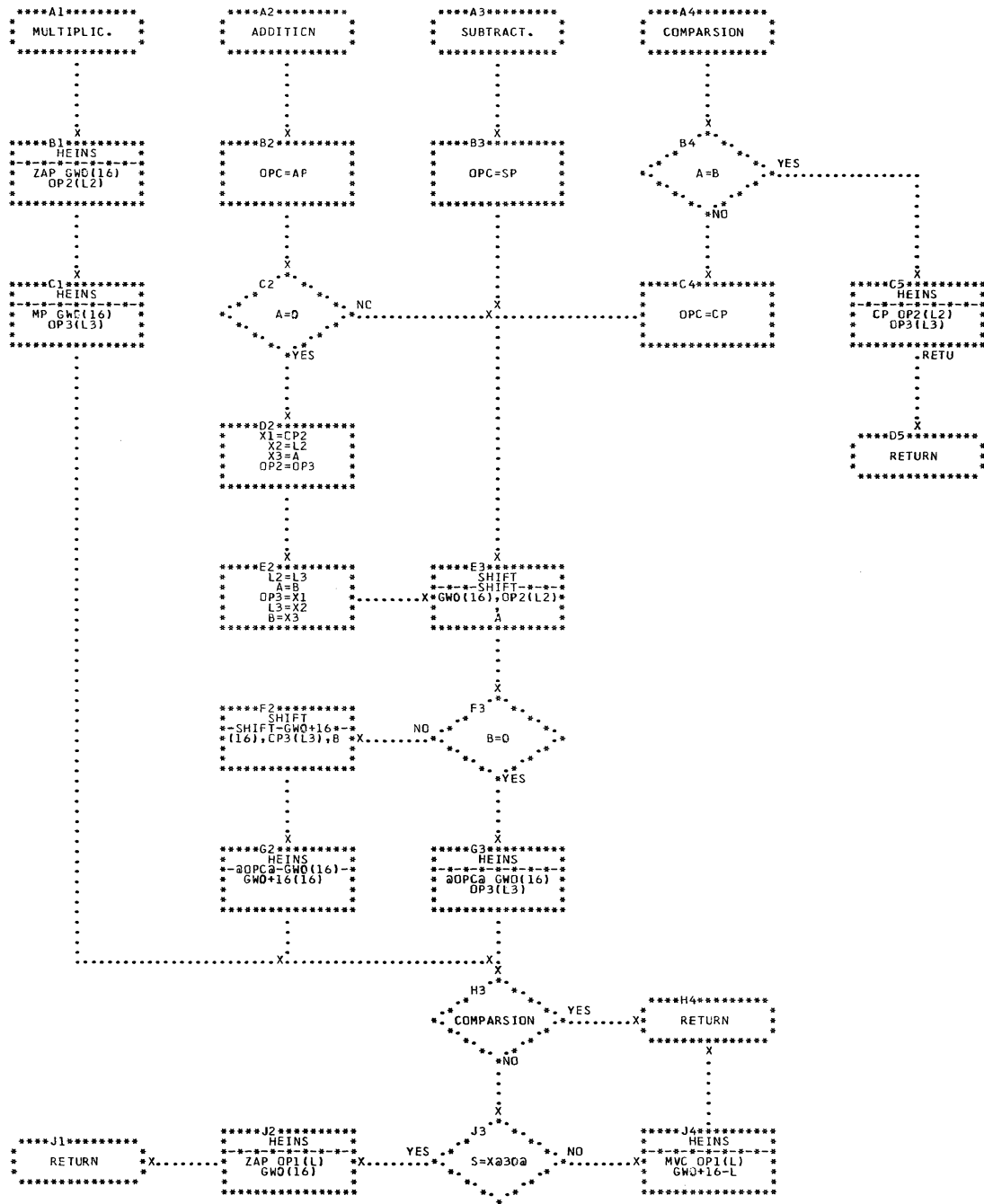
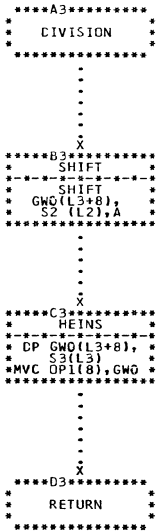
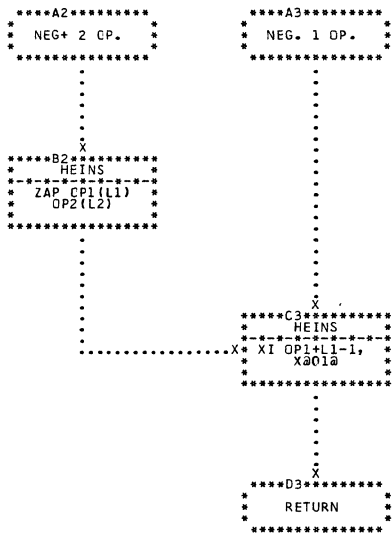


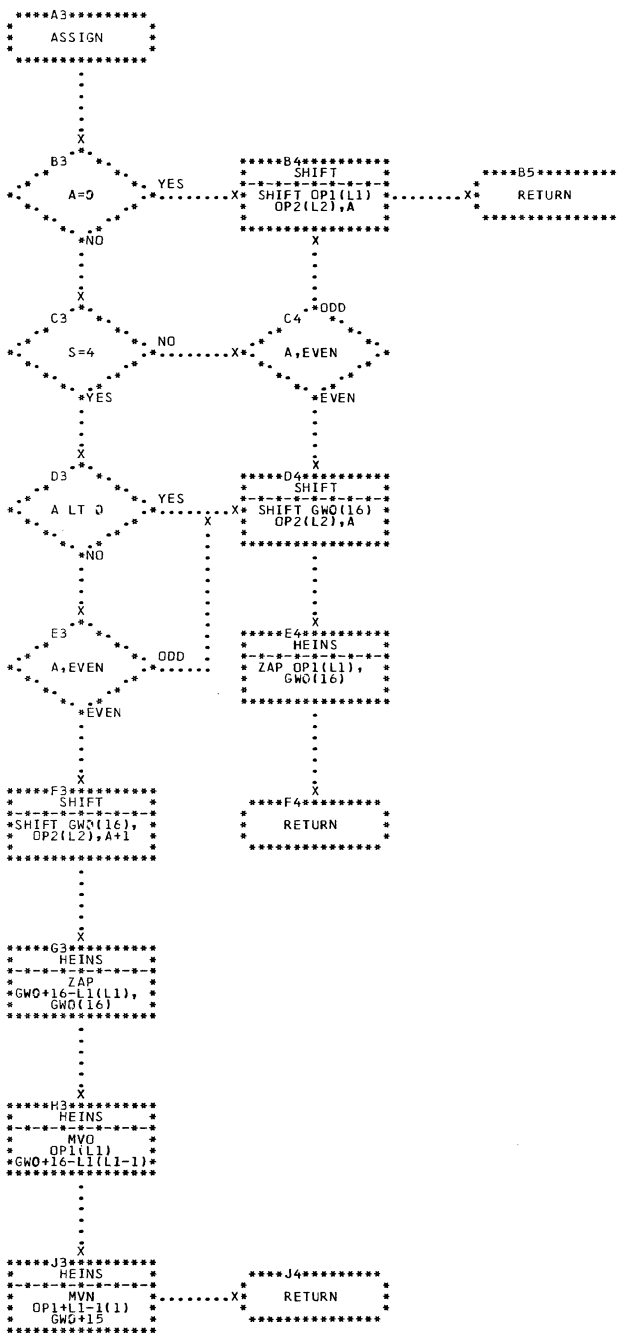
CHART TG. IJXE50

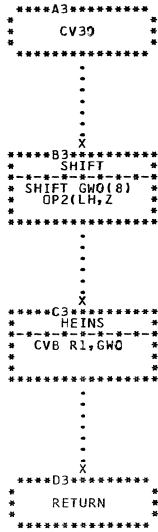
DECIMAL SHIFT

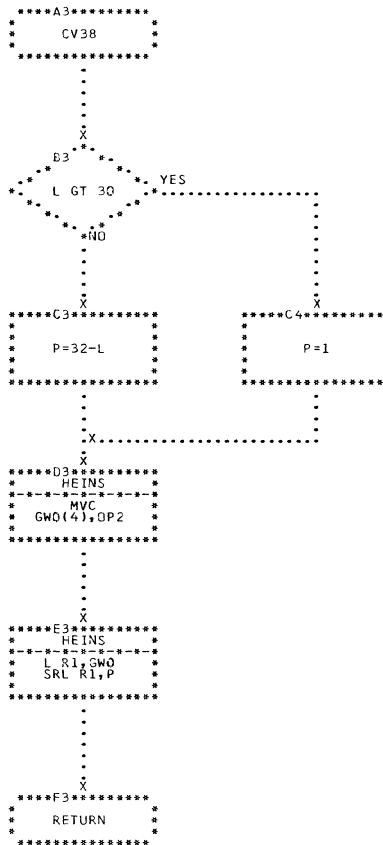


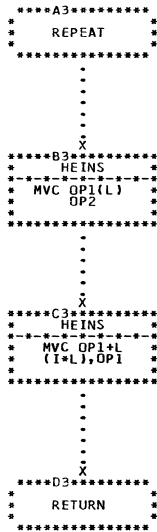


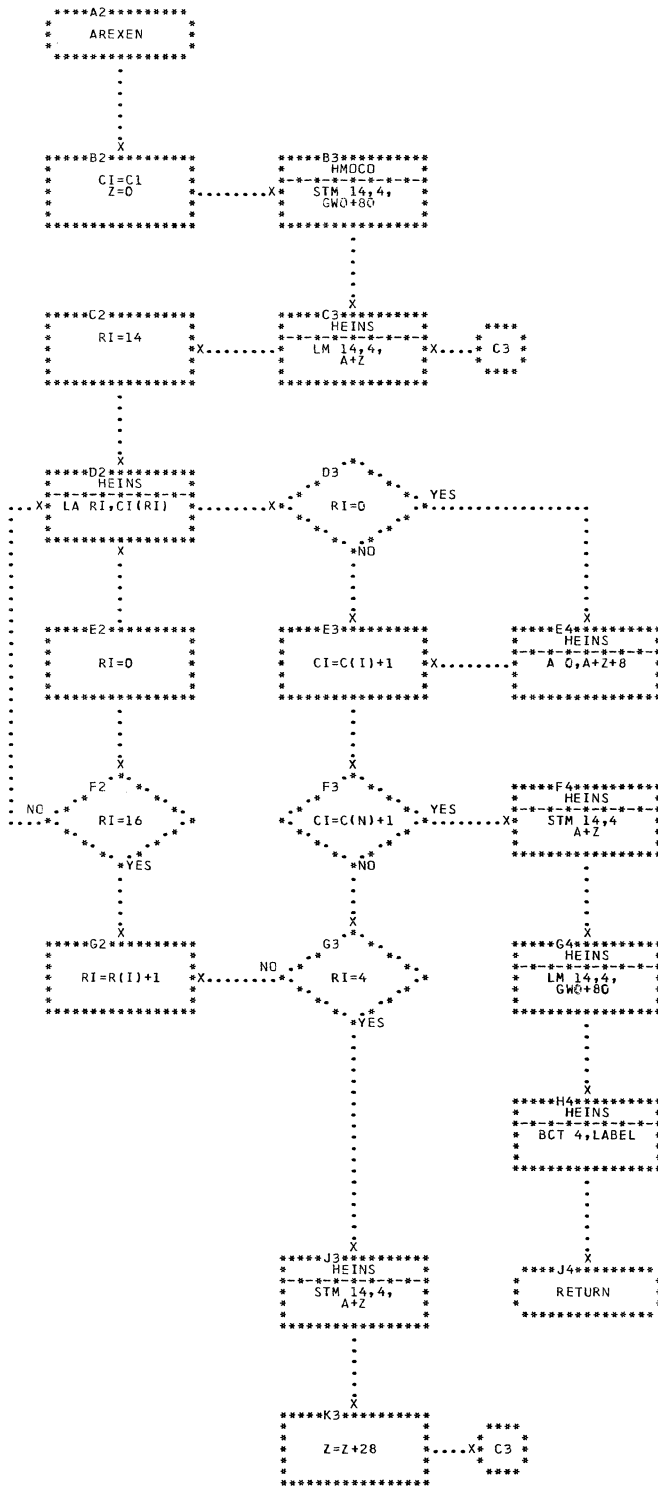


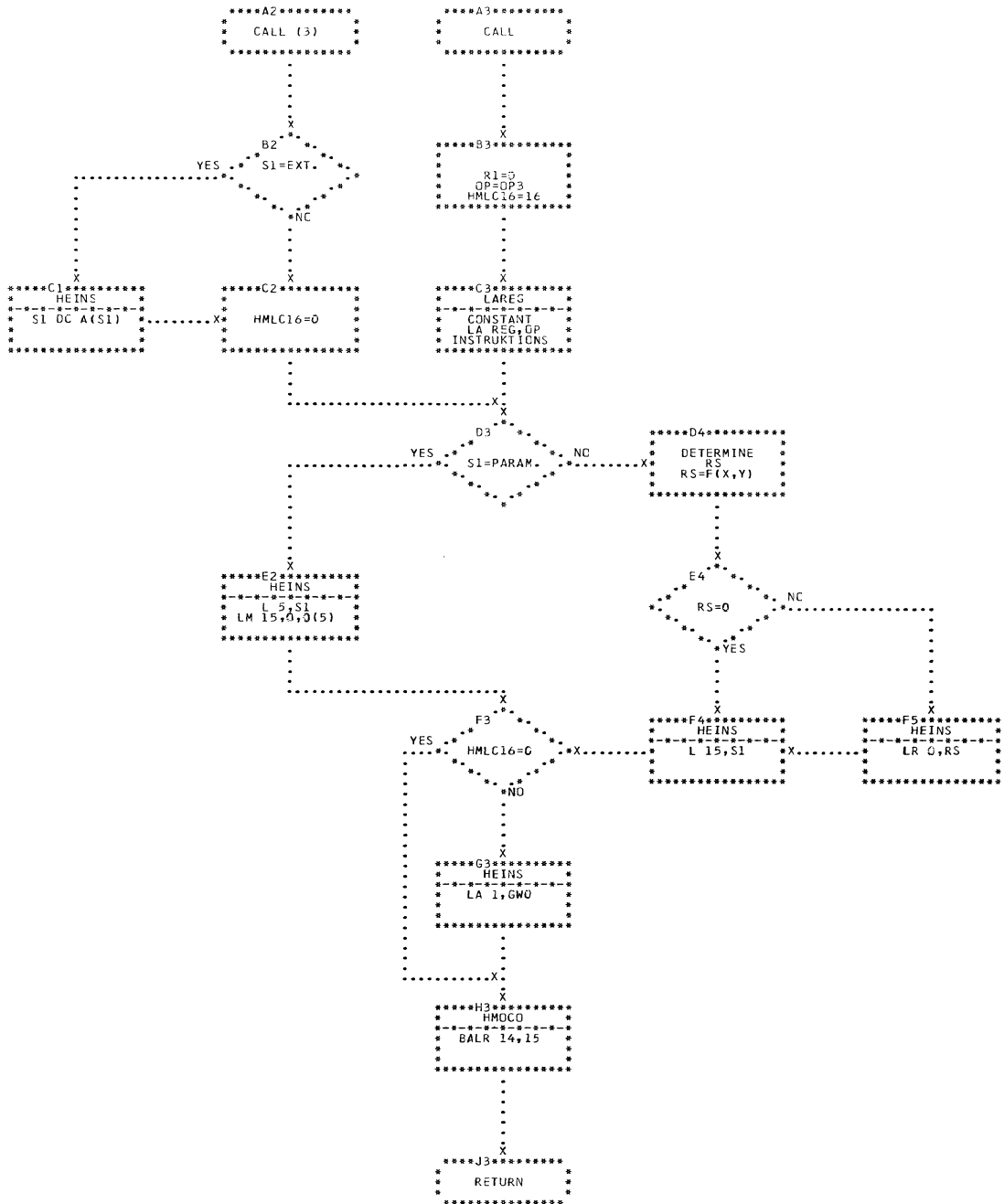


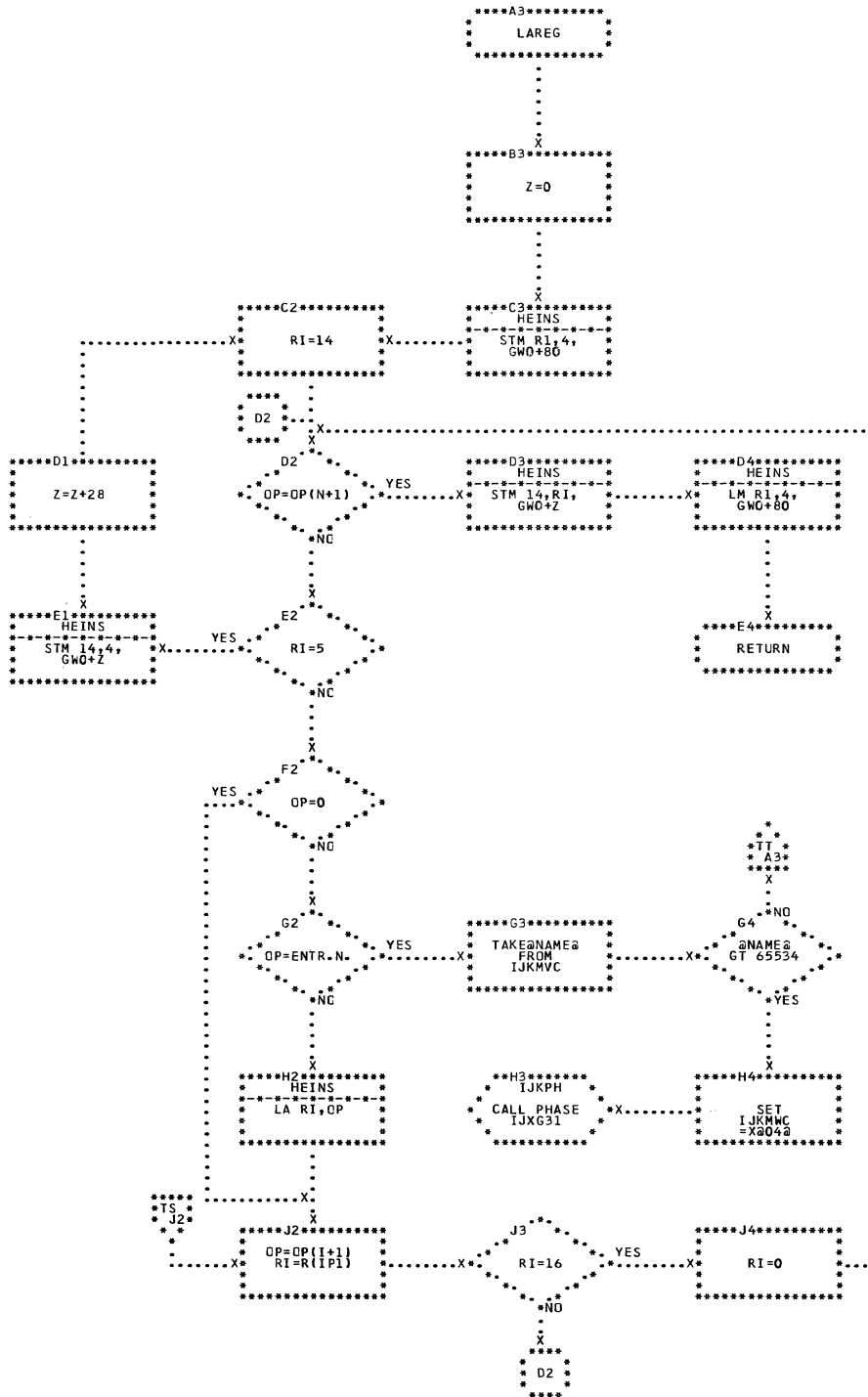


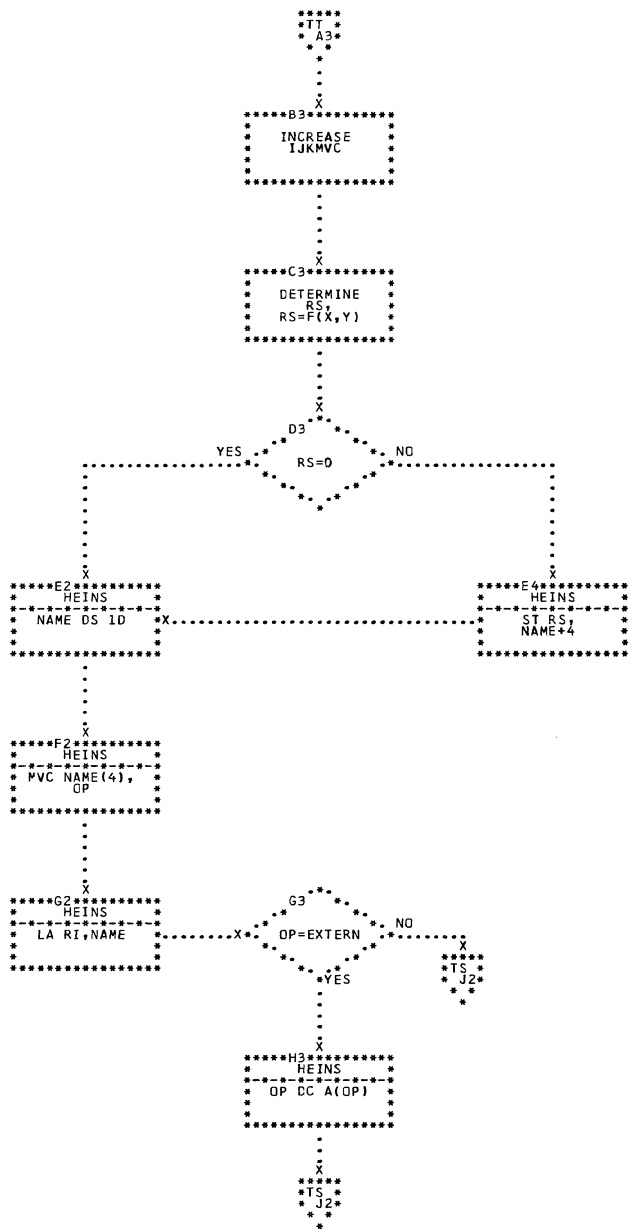


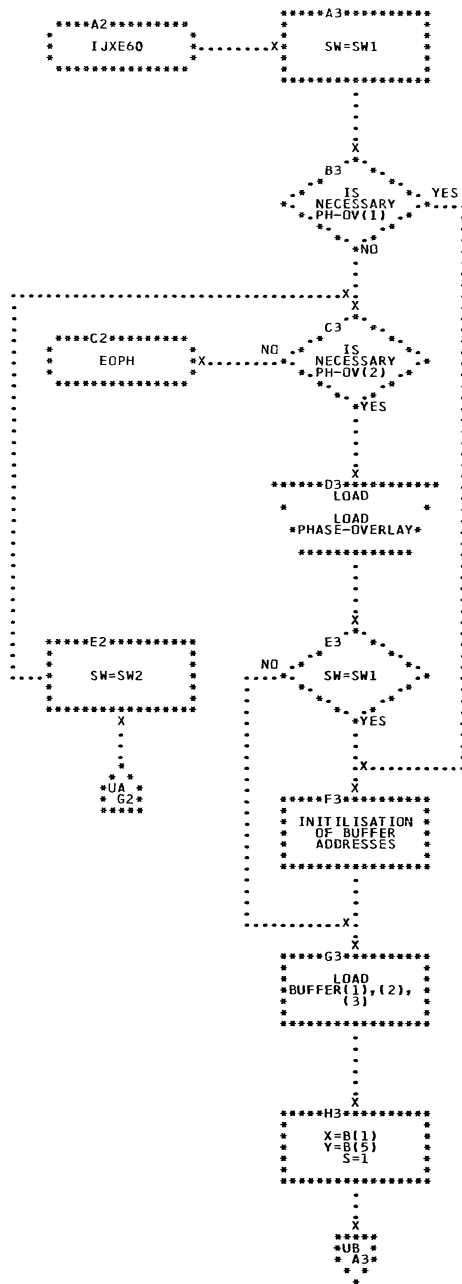


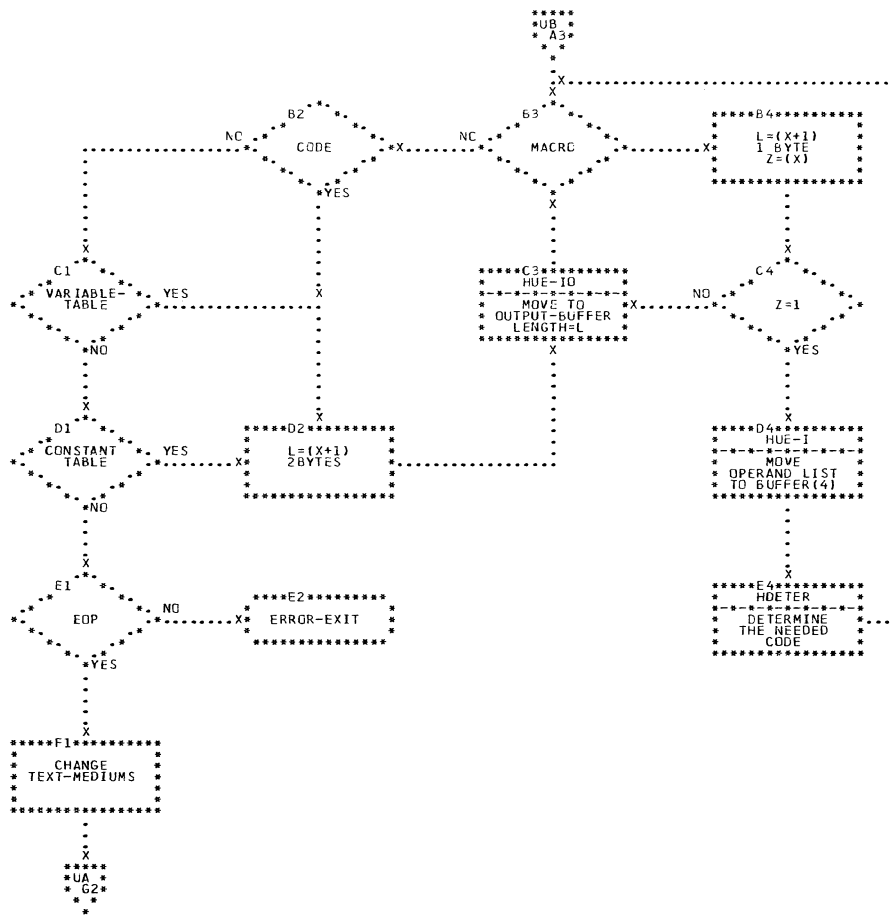












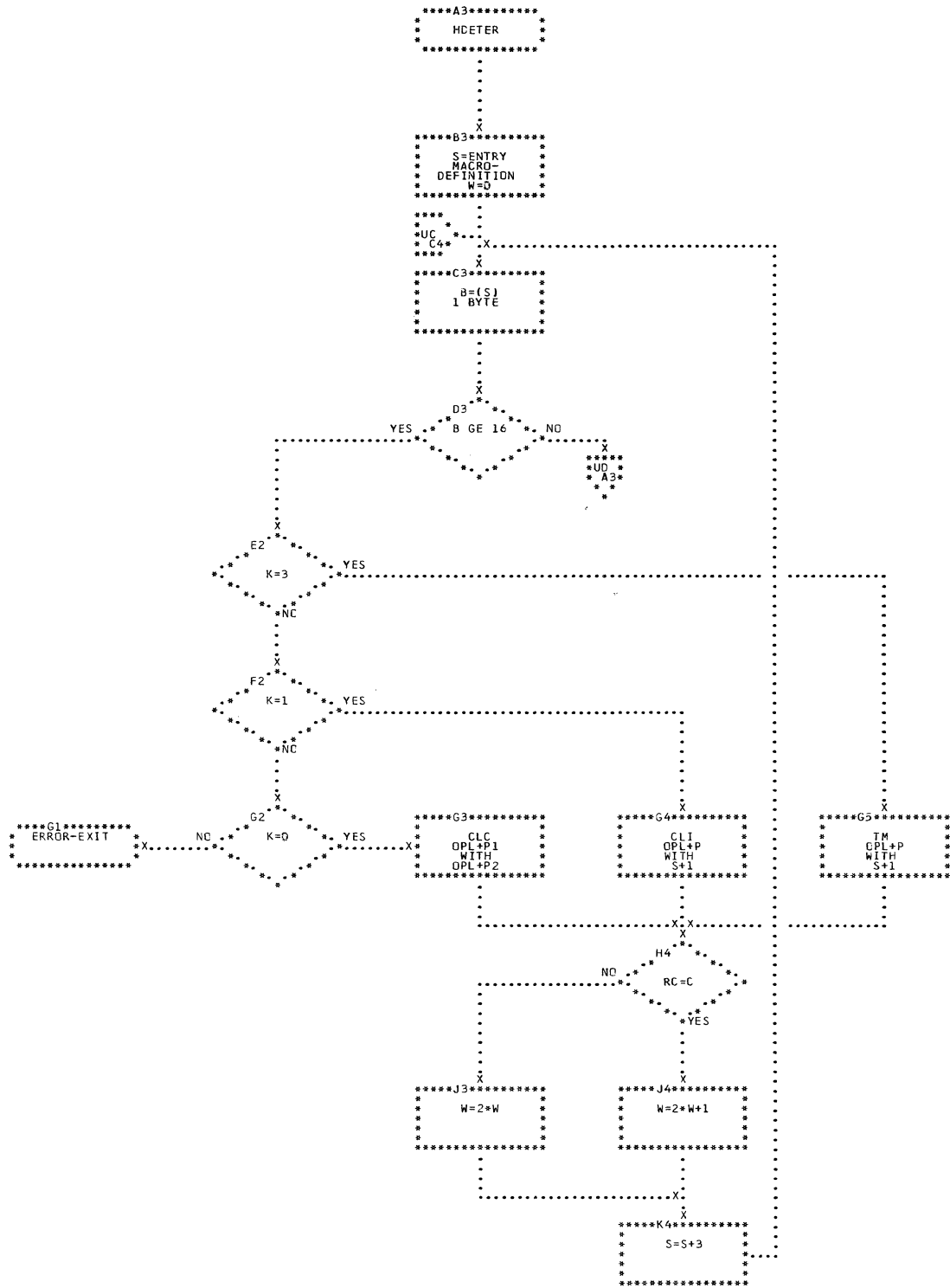
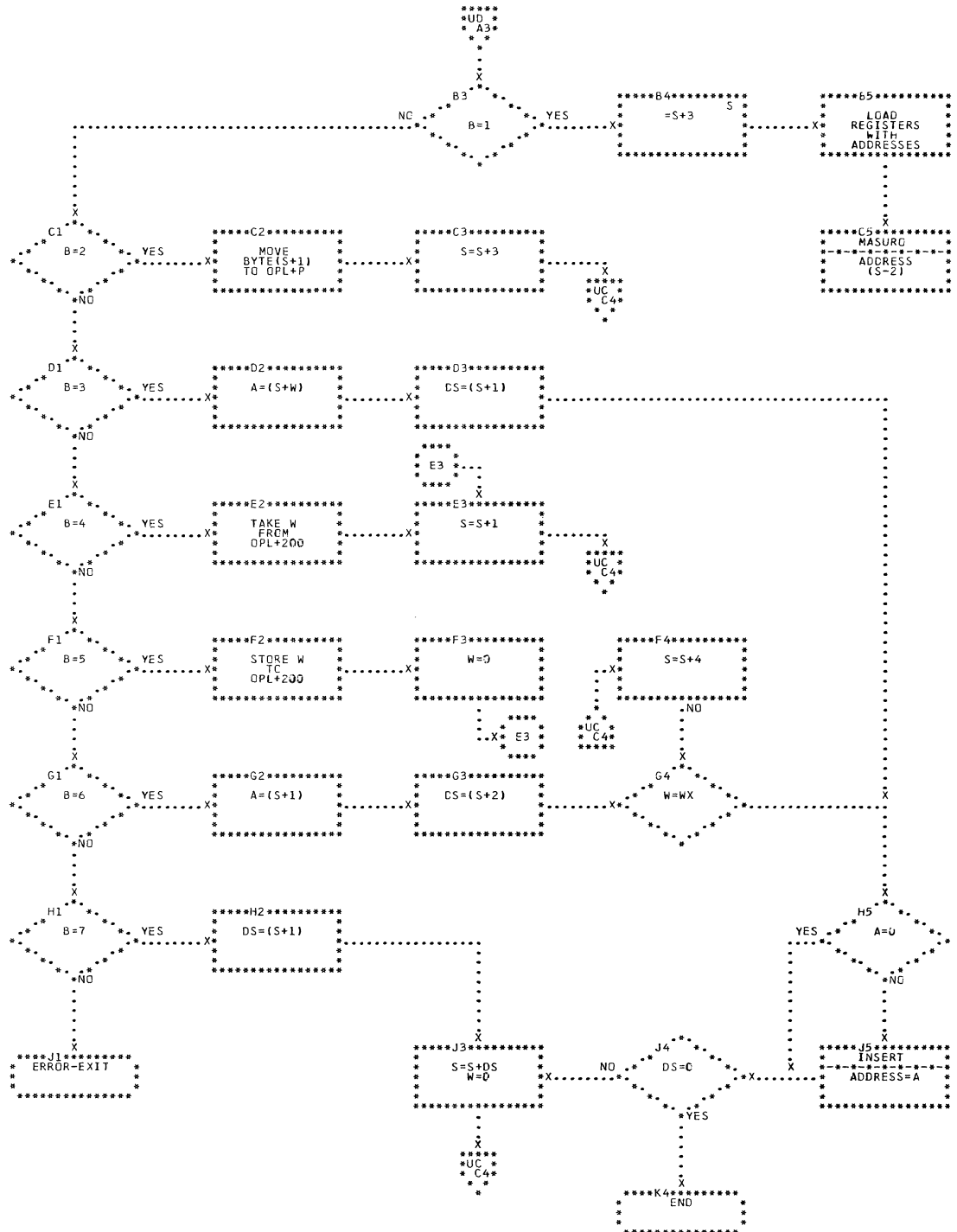
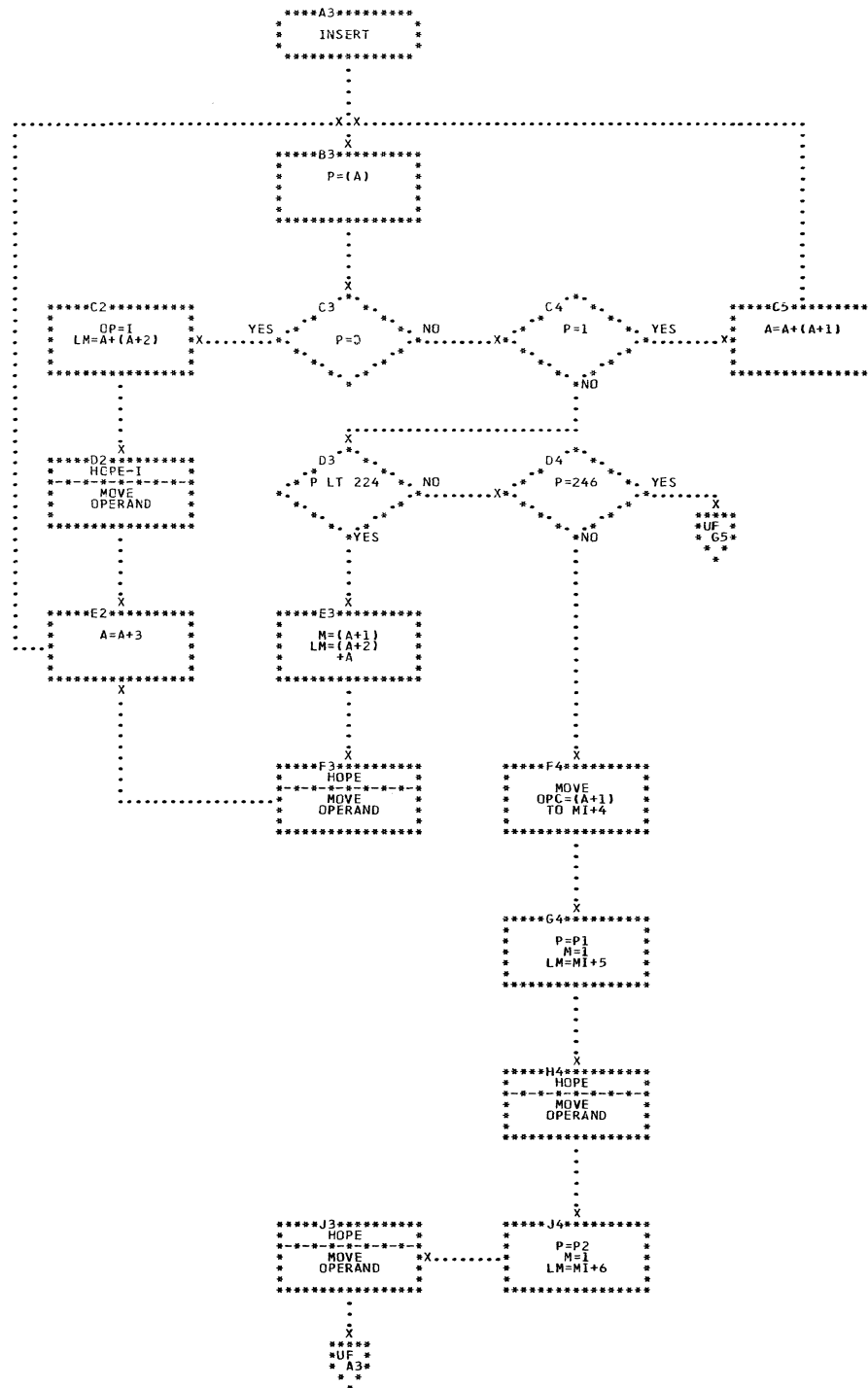
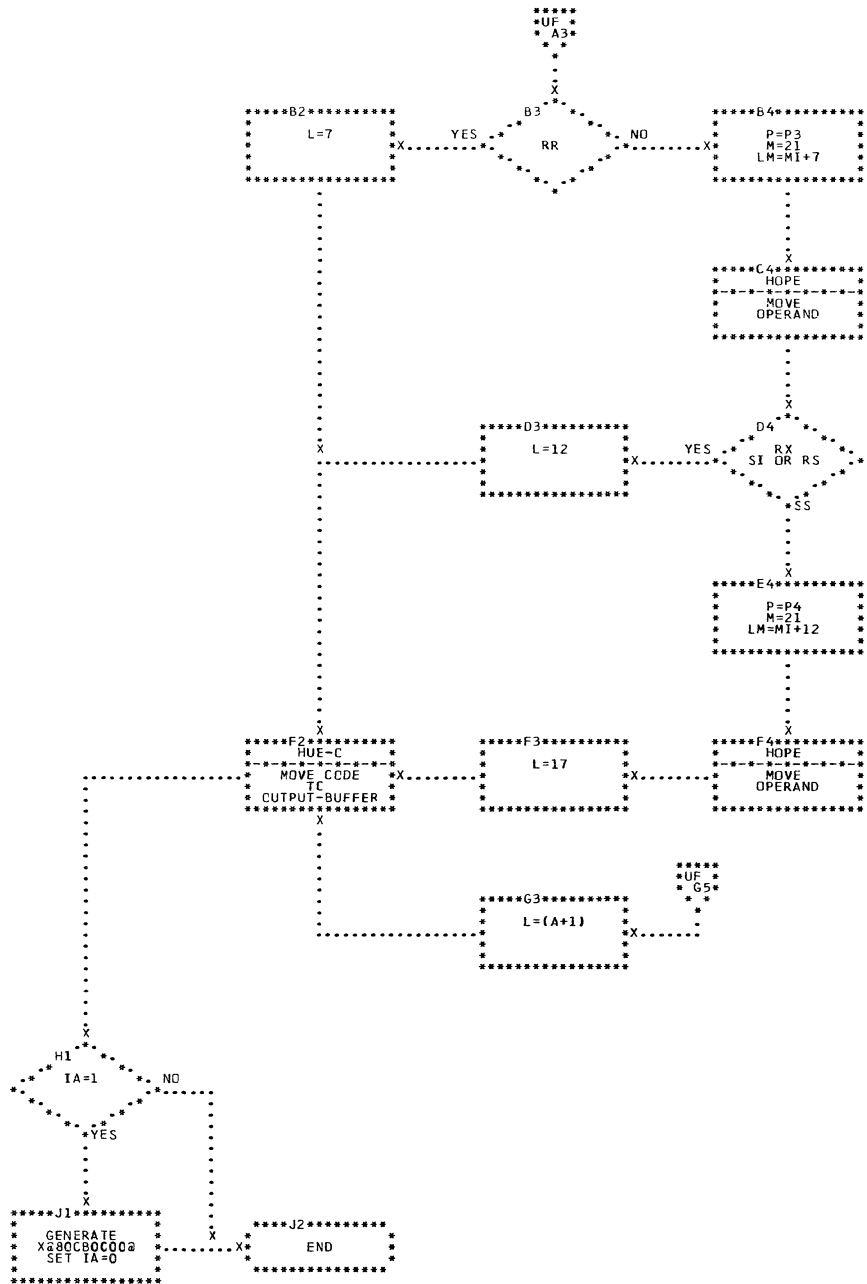
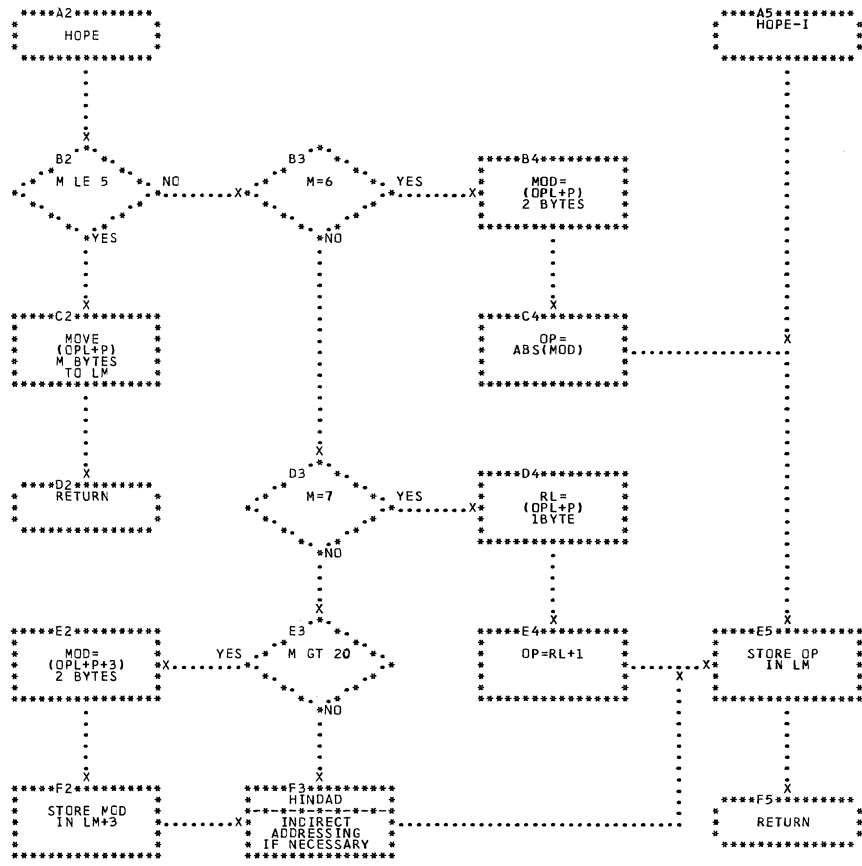


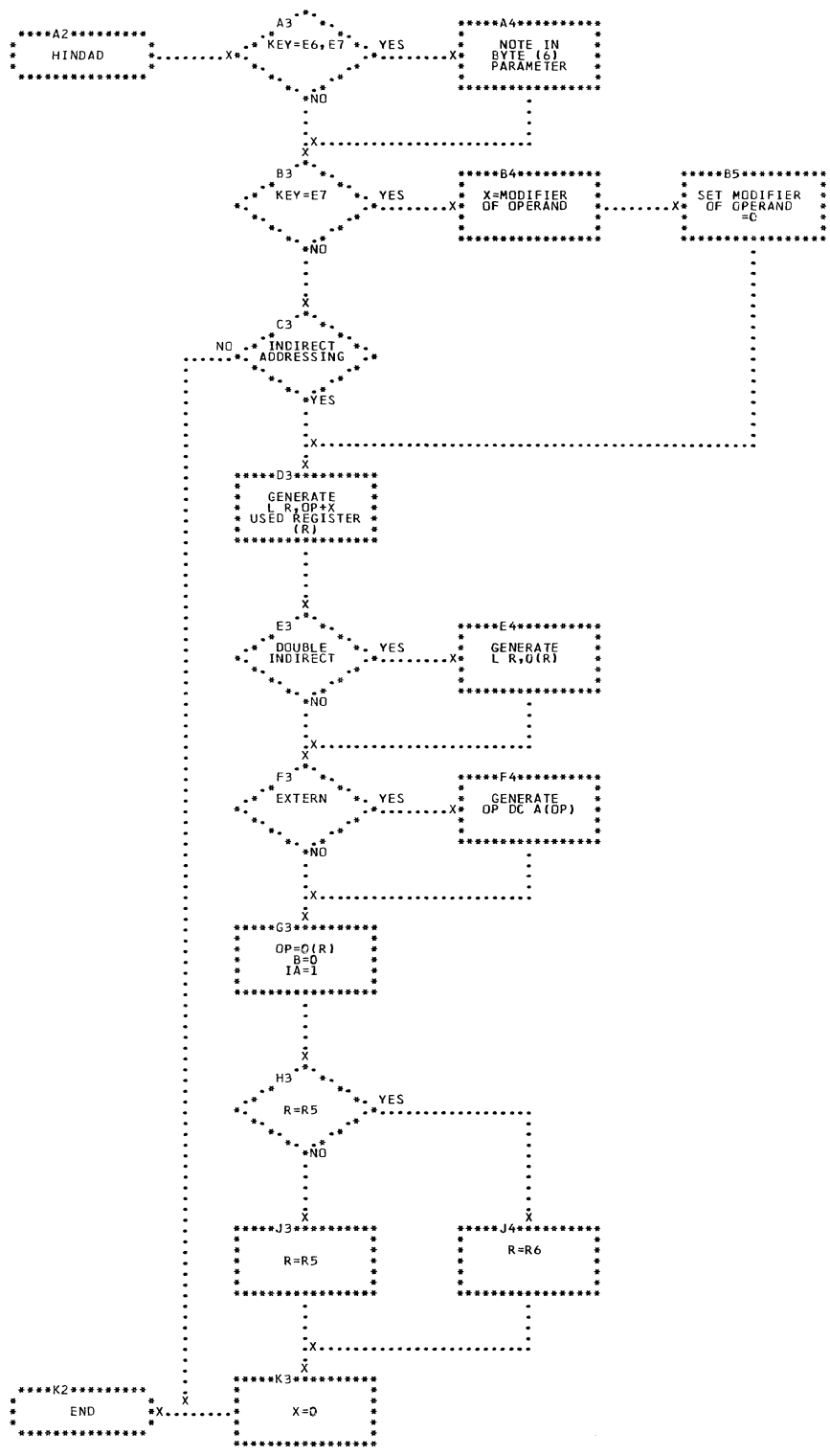
CHART UC. IJXE60 DETERMINATION OF CODE

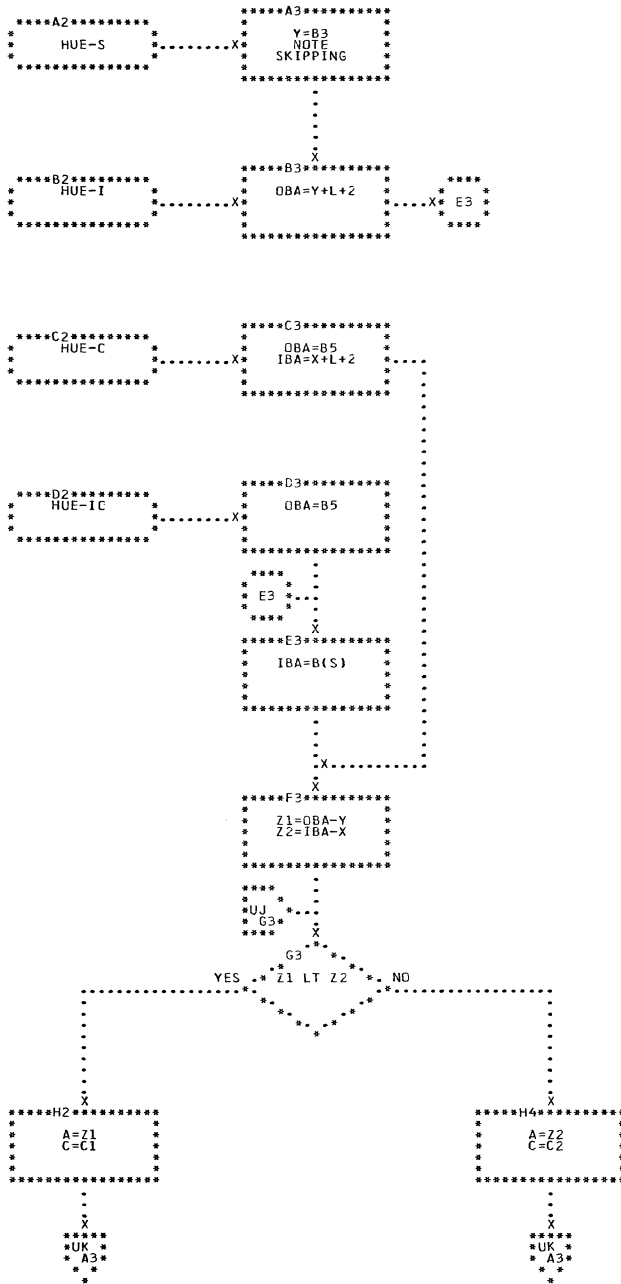


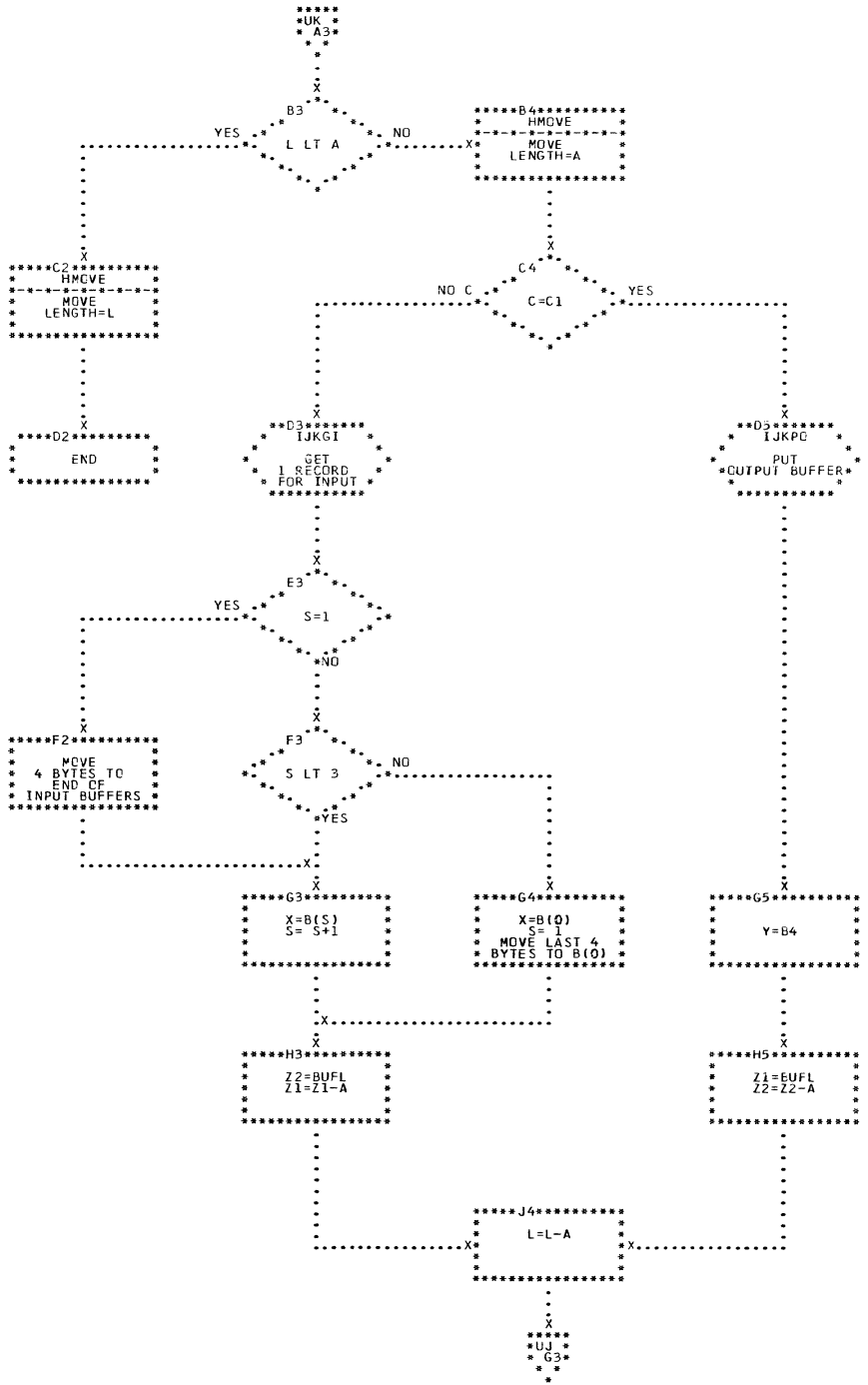


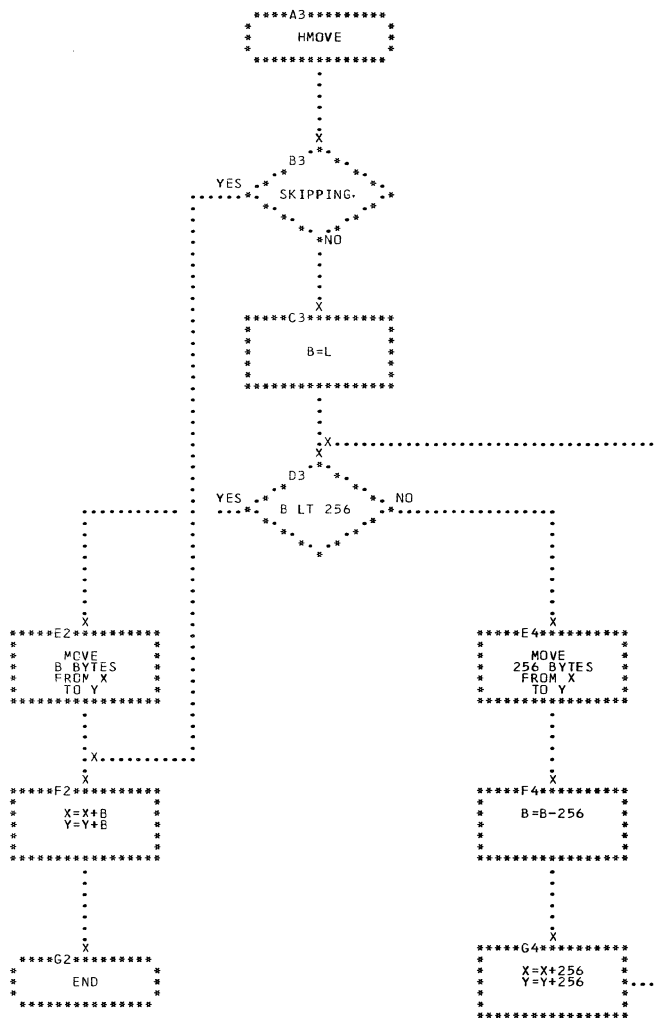


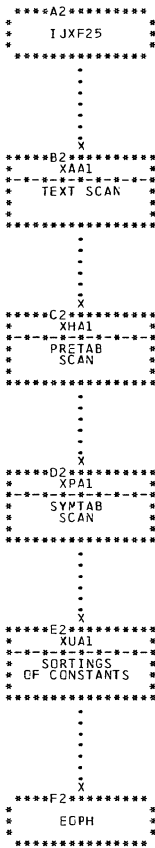


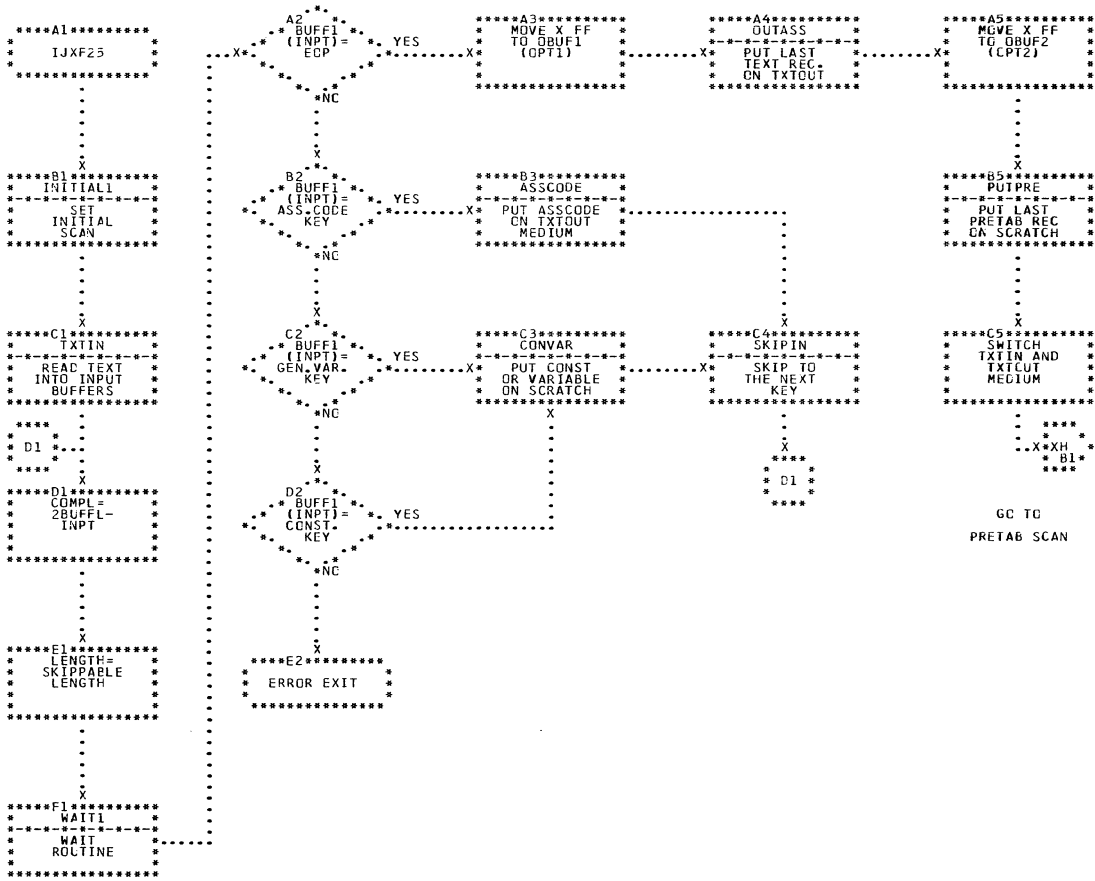












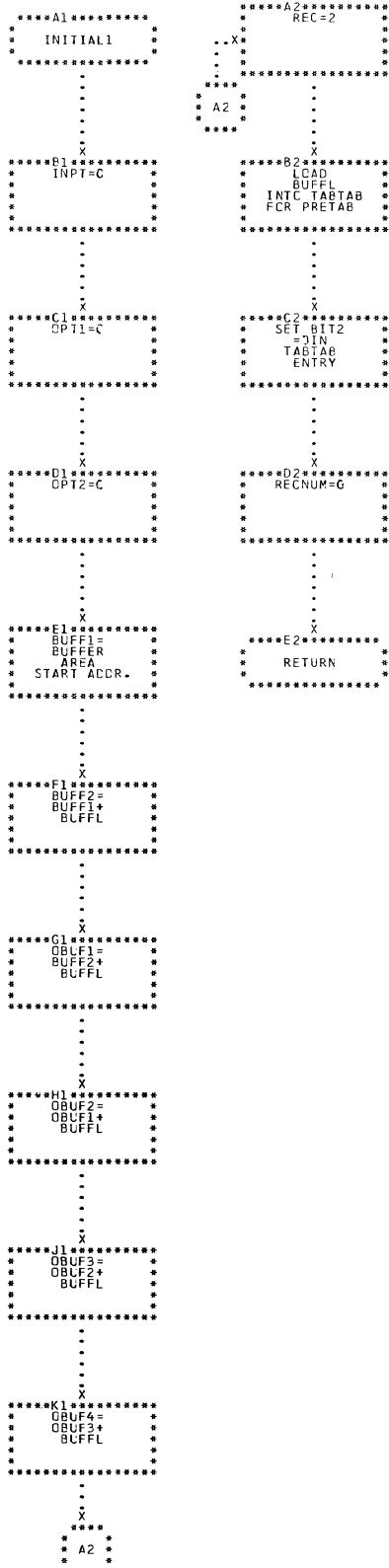
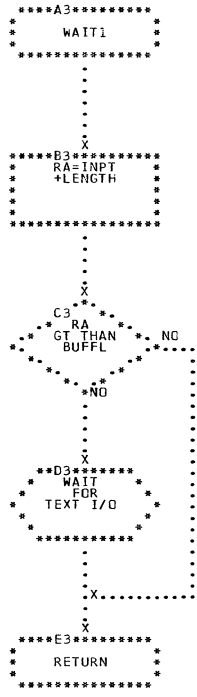
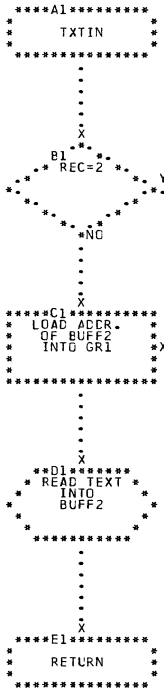


CHART XB. IJXF25

INITIAL 1



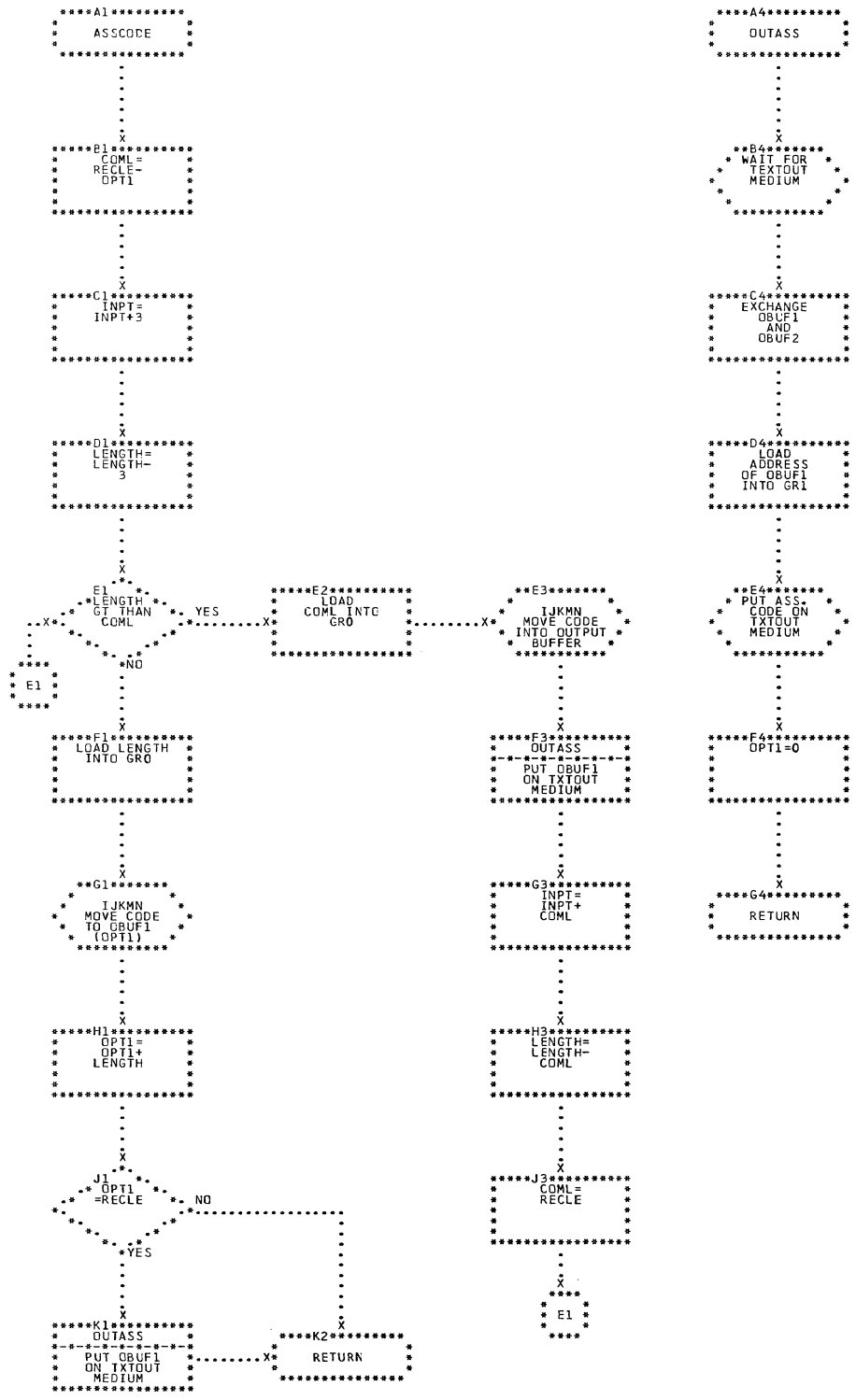
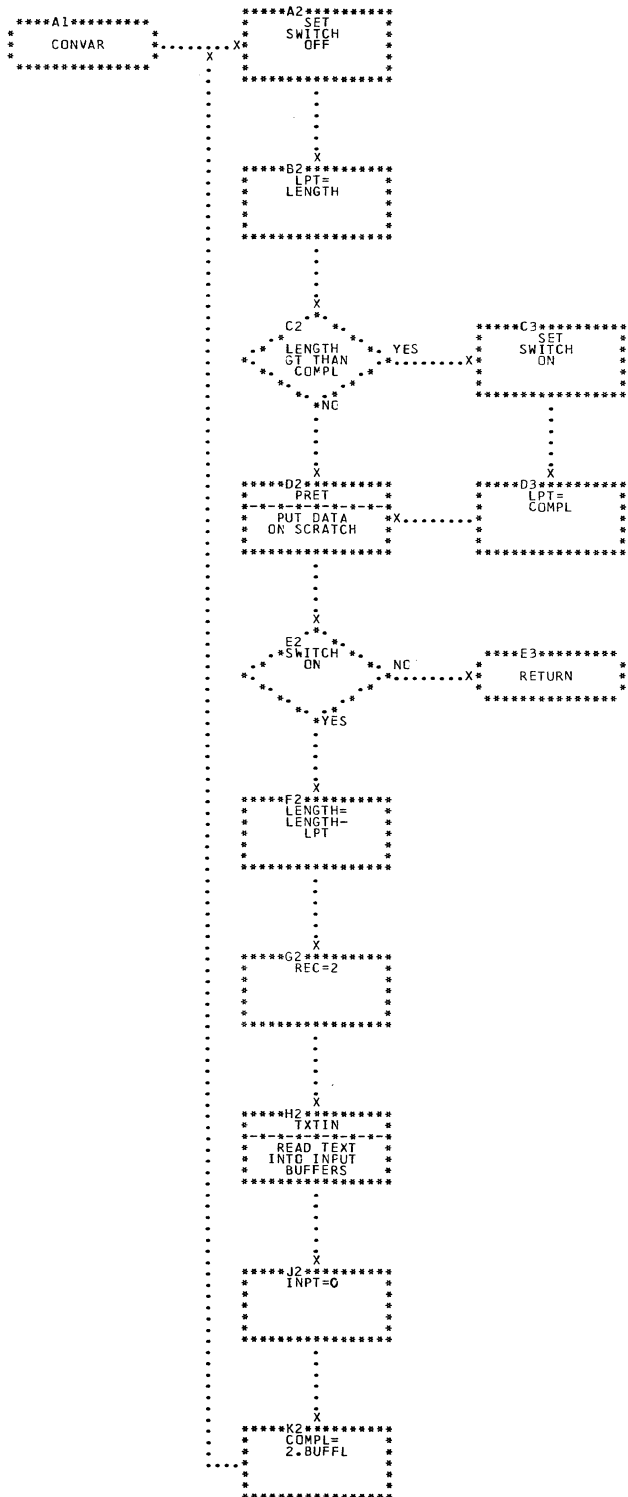


CHART XD. IJXF25

ASSCODE,OUTASS



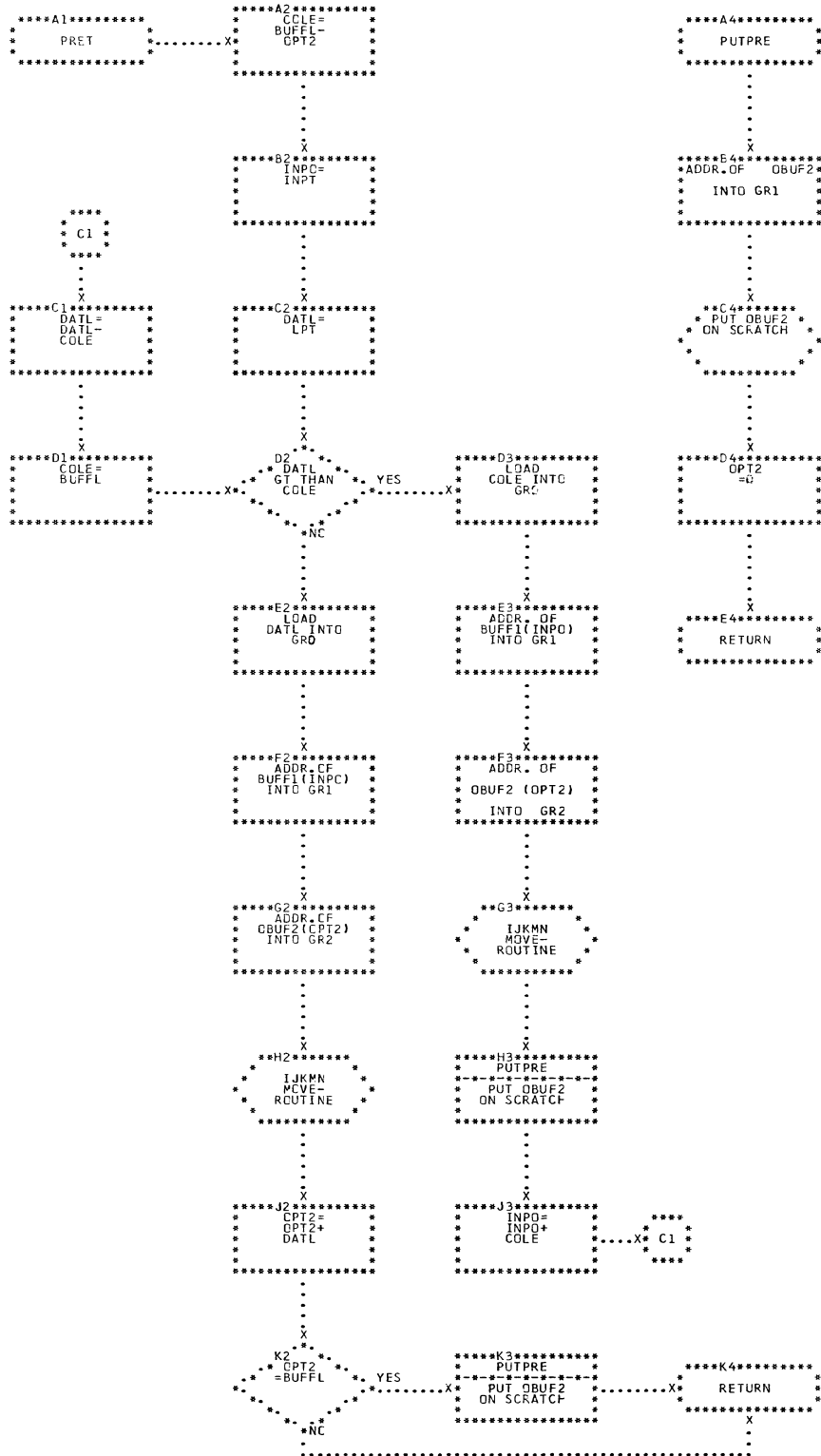
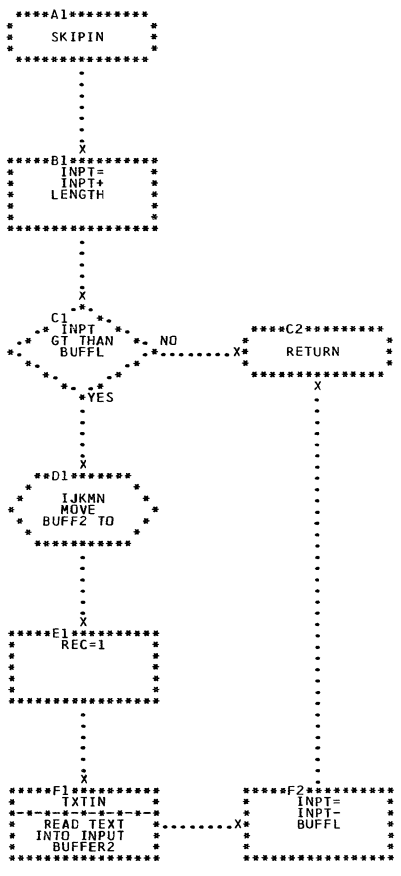
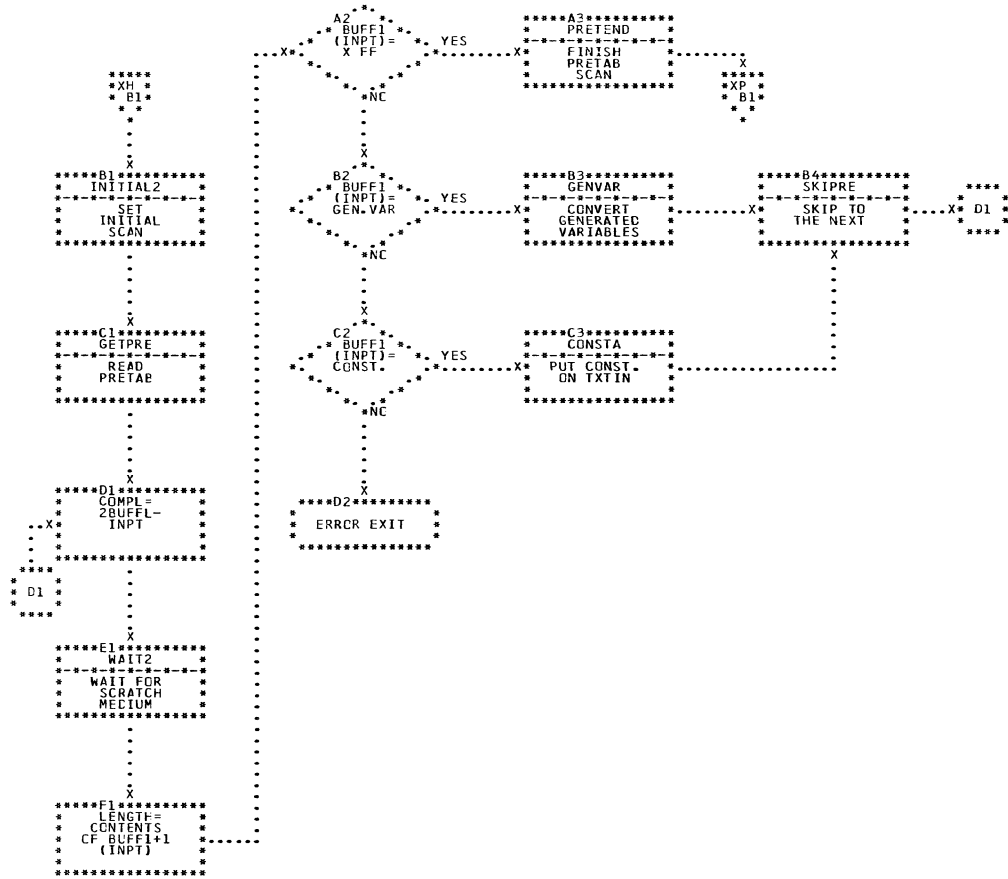
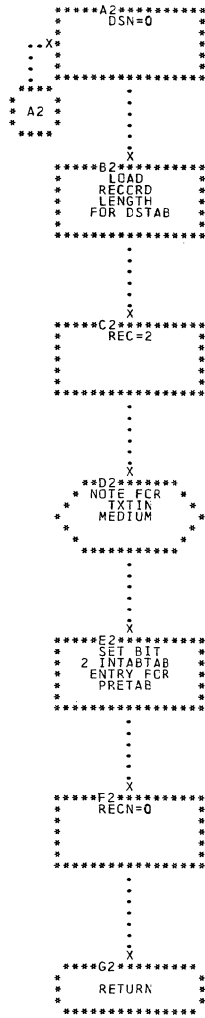
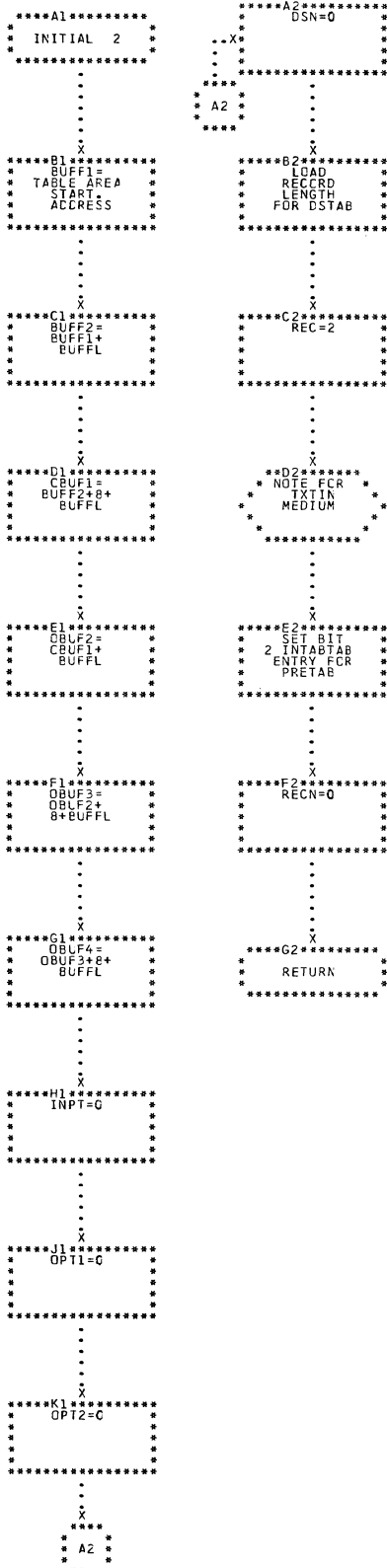


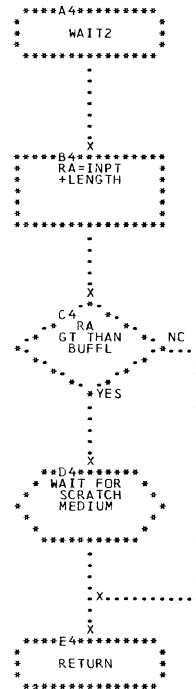
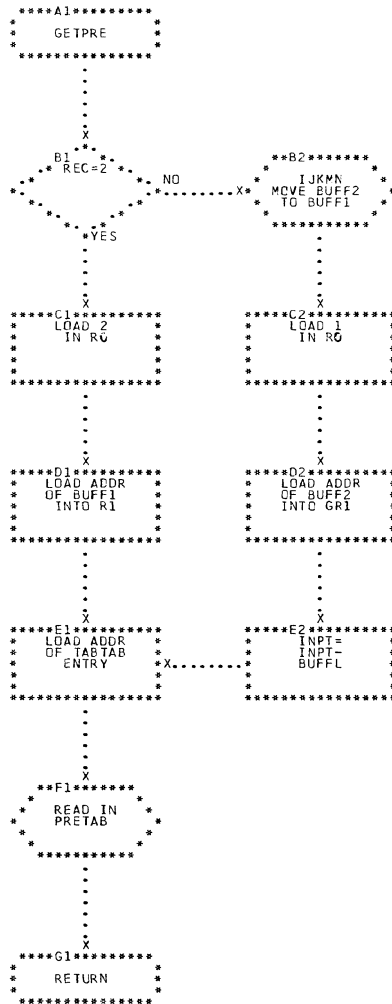
CHART XF. IJXF25

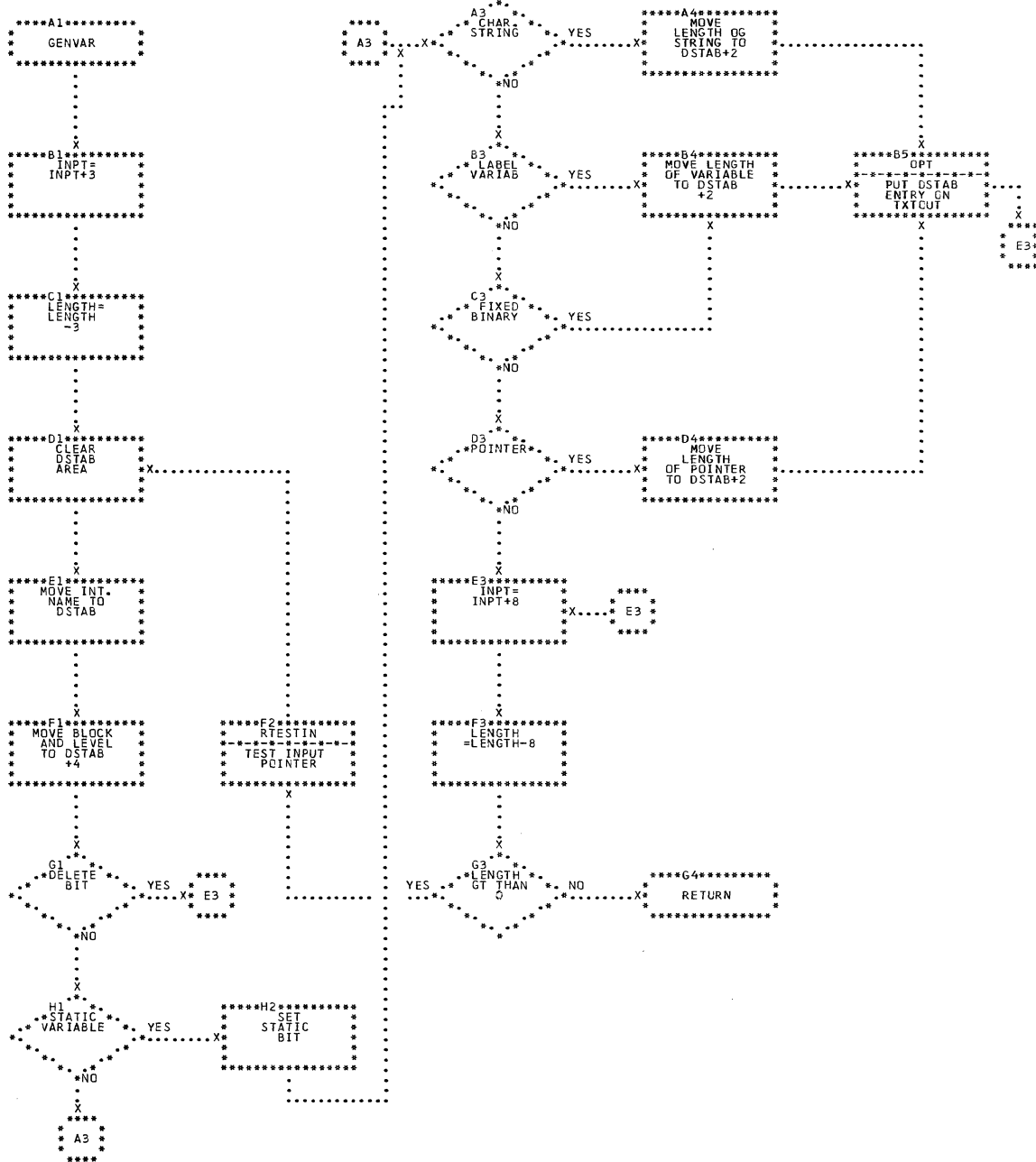
PRET,PUTPRE

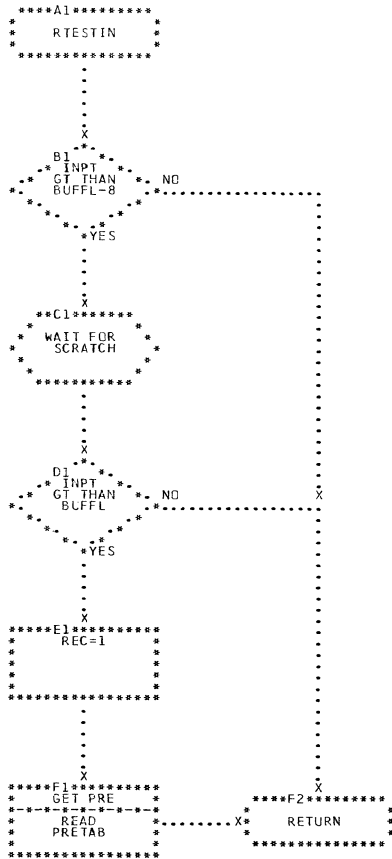


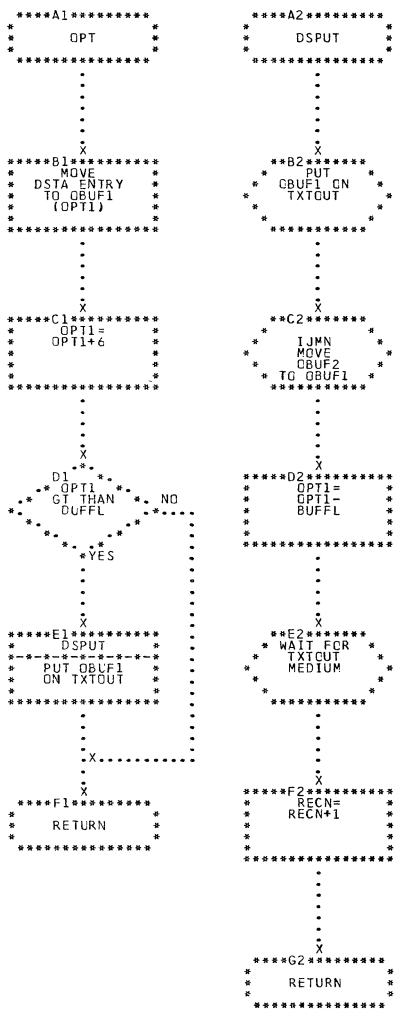


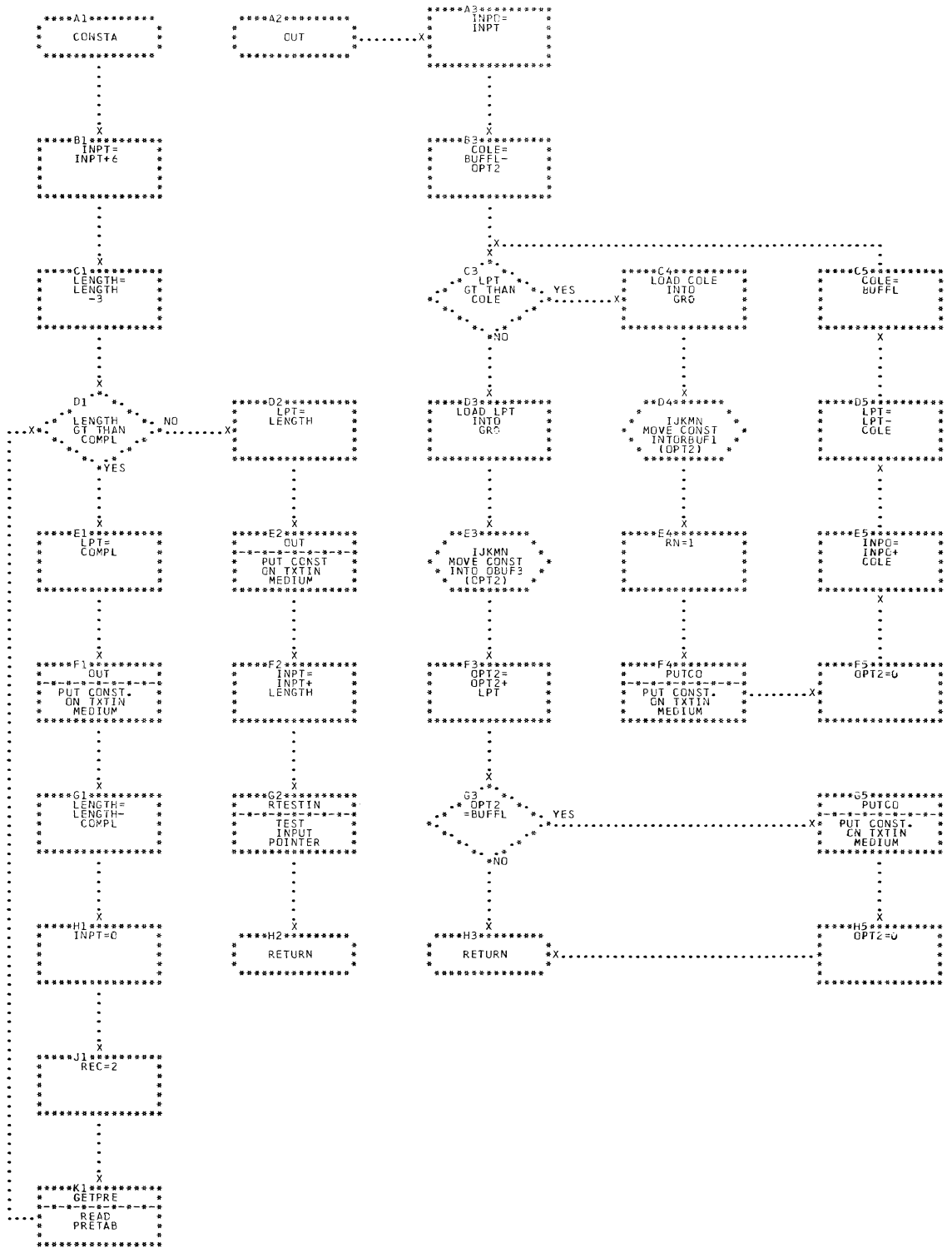


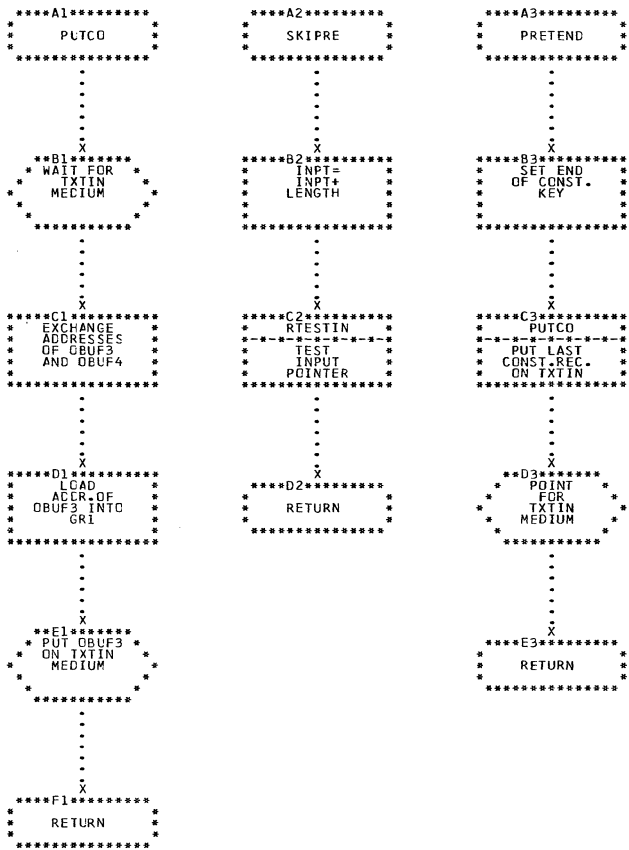


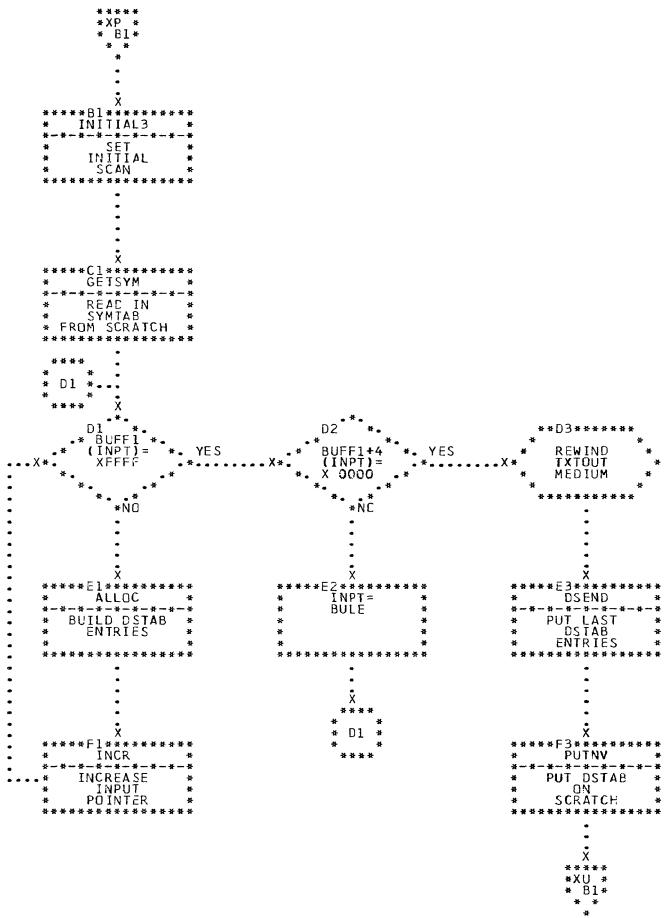


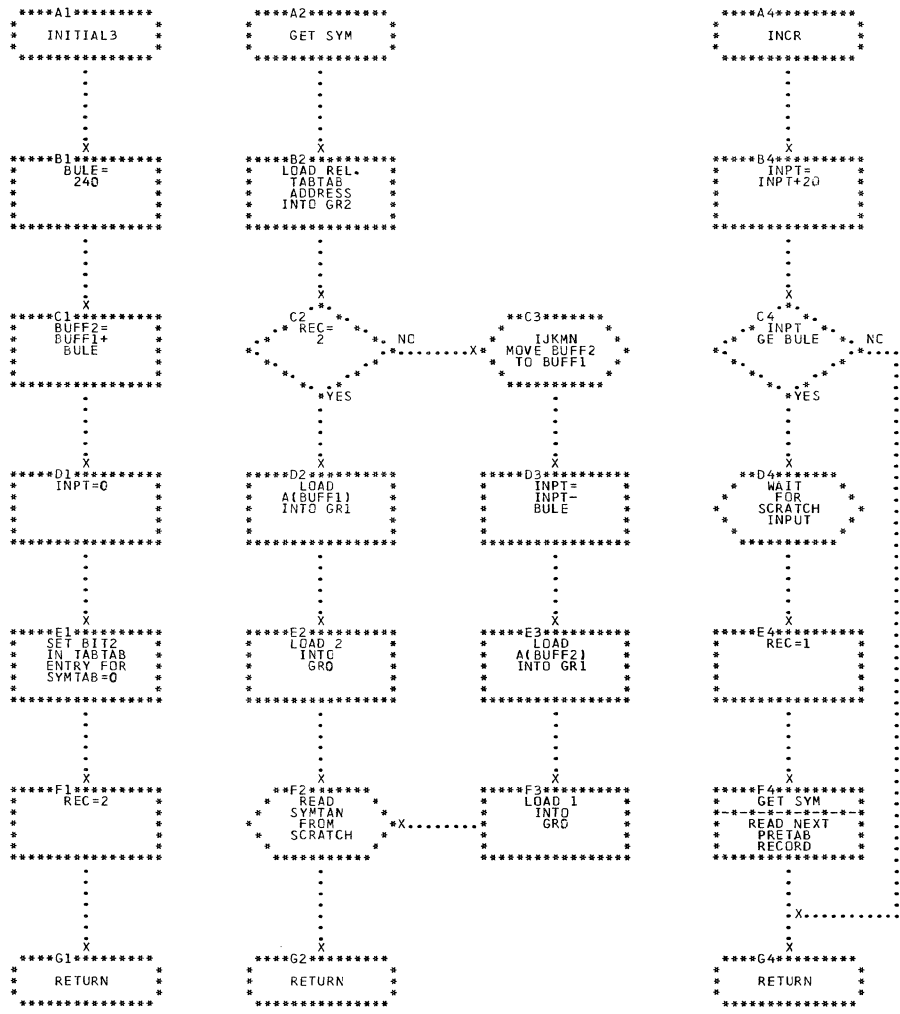












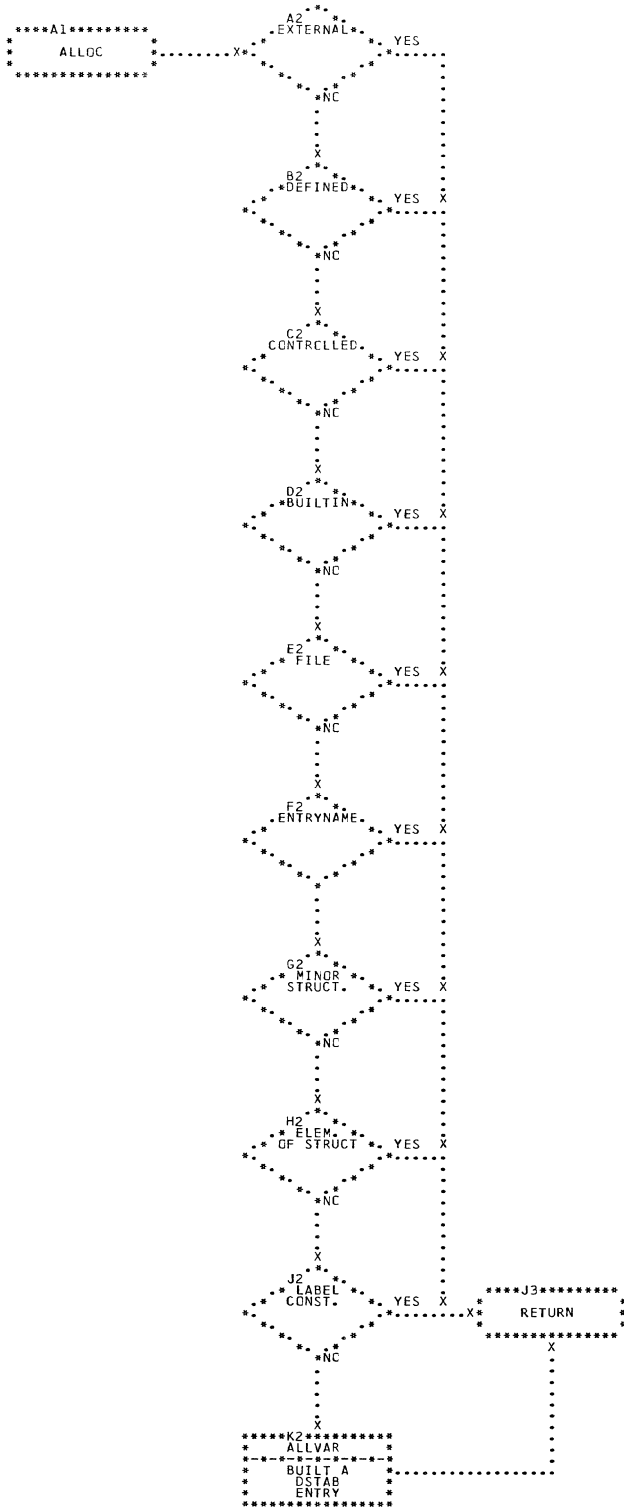
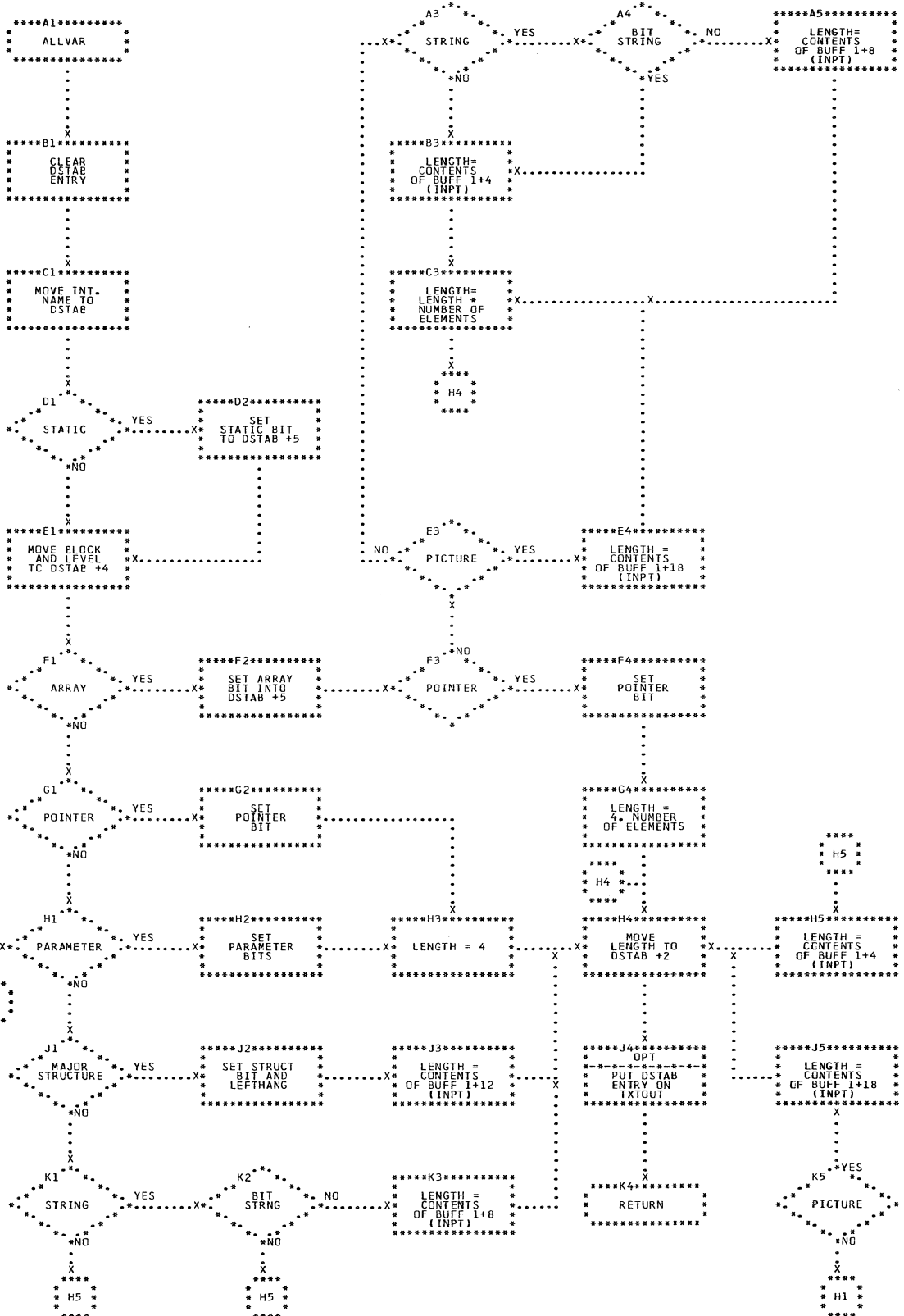
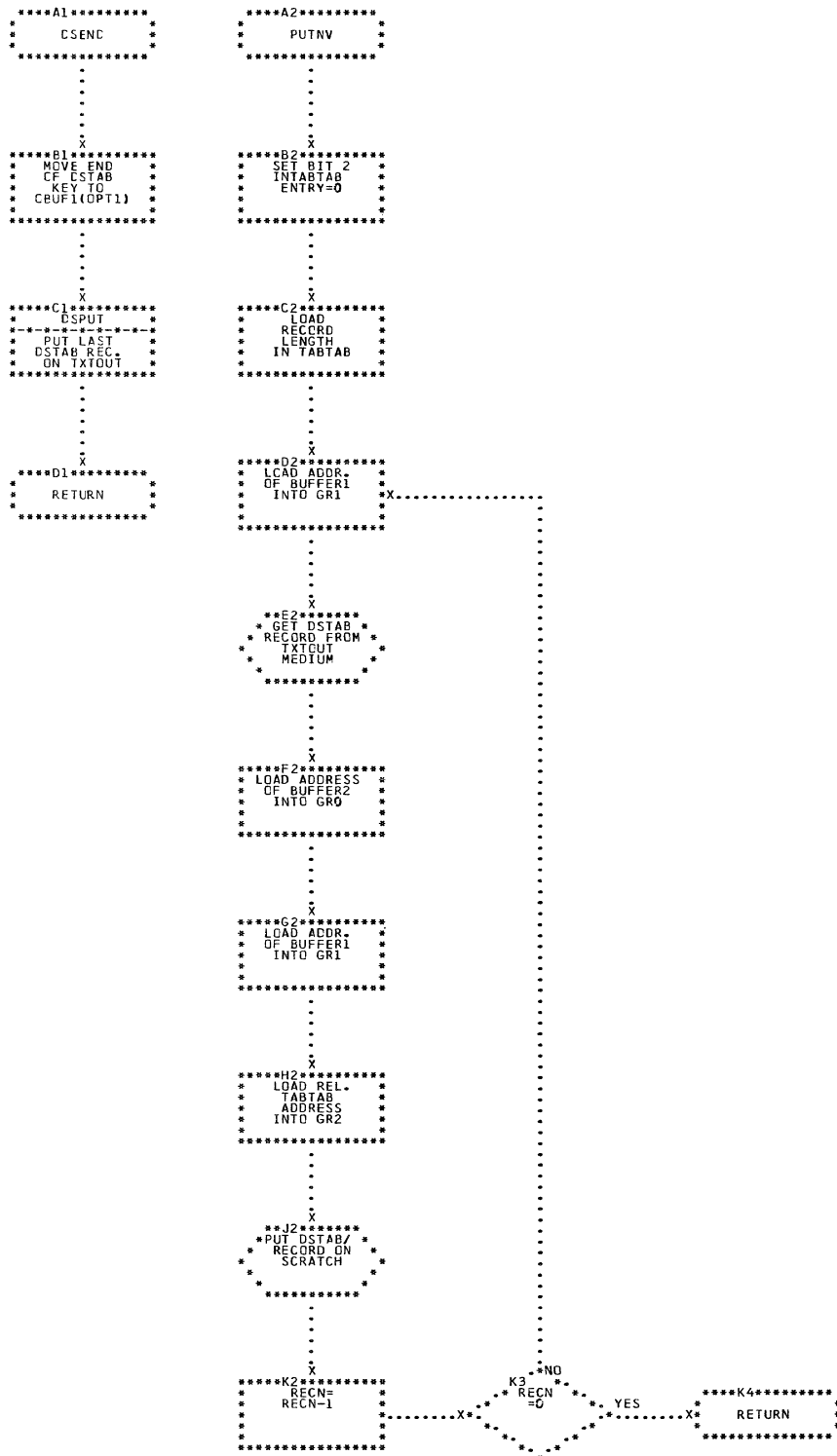


CHART XR. IJXF25 ALLOC





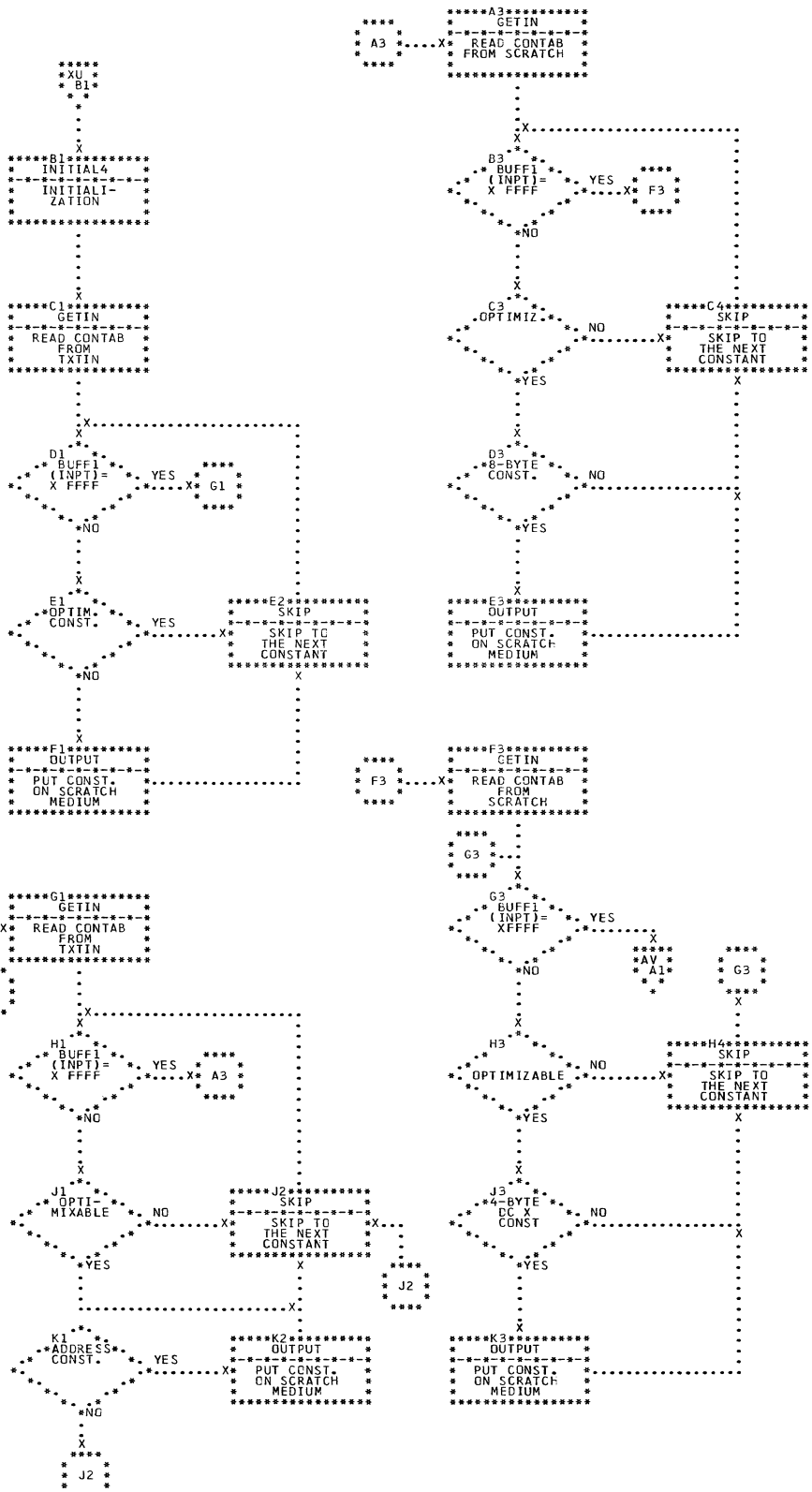


CHART XU. IJXF25 SORTING OF CONSTANTS

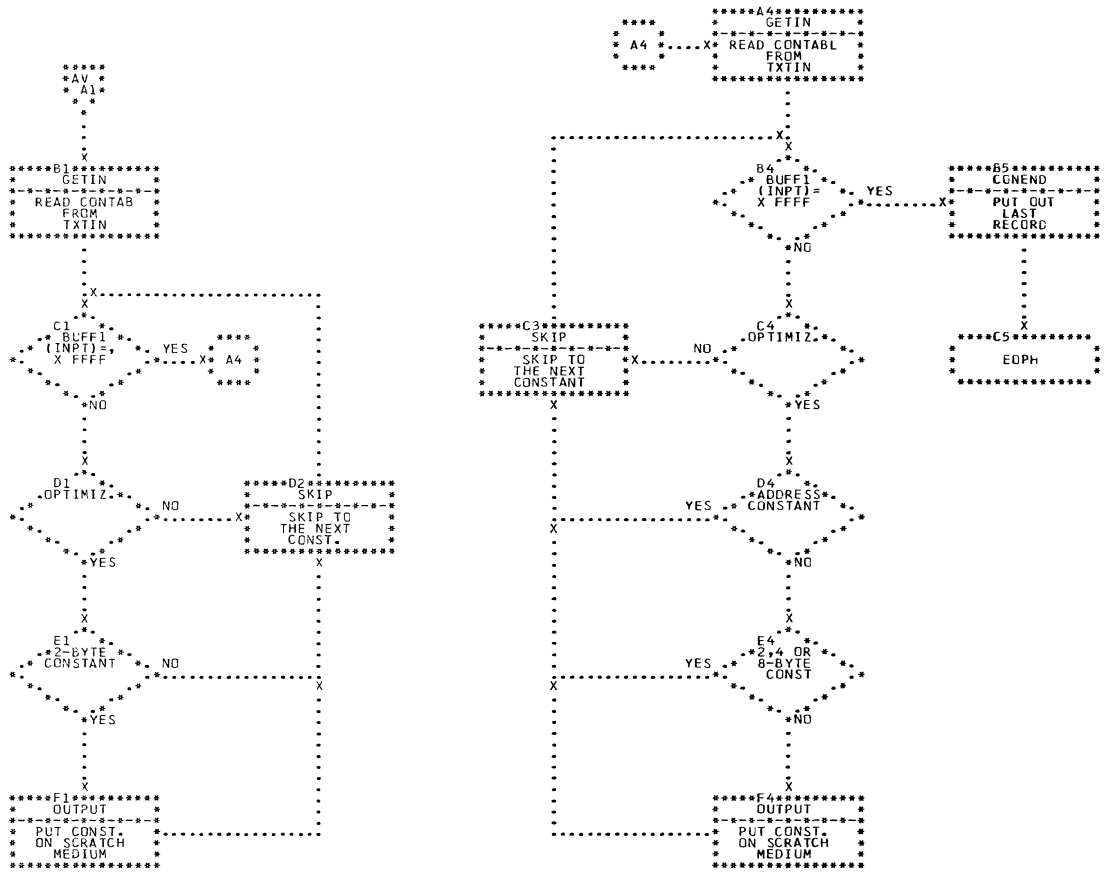
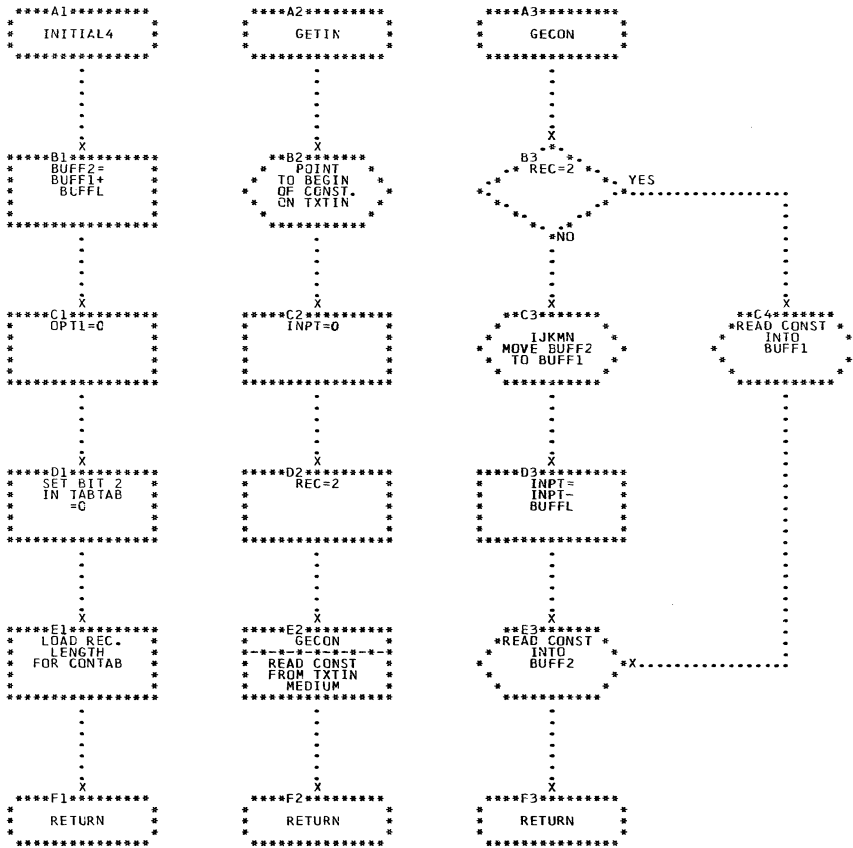


CHART XV. IJXF25

SORTING OF CONSTANTS



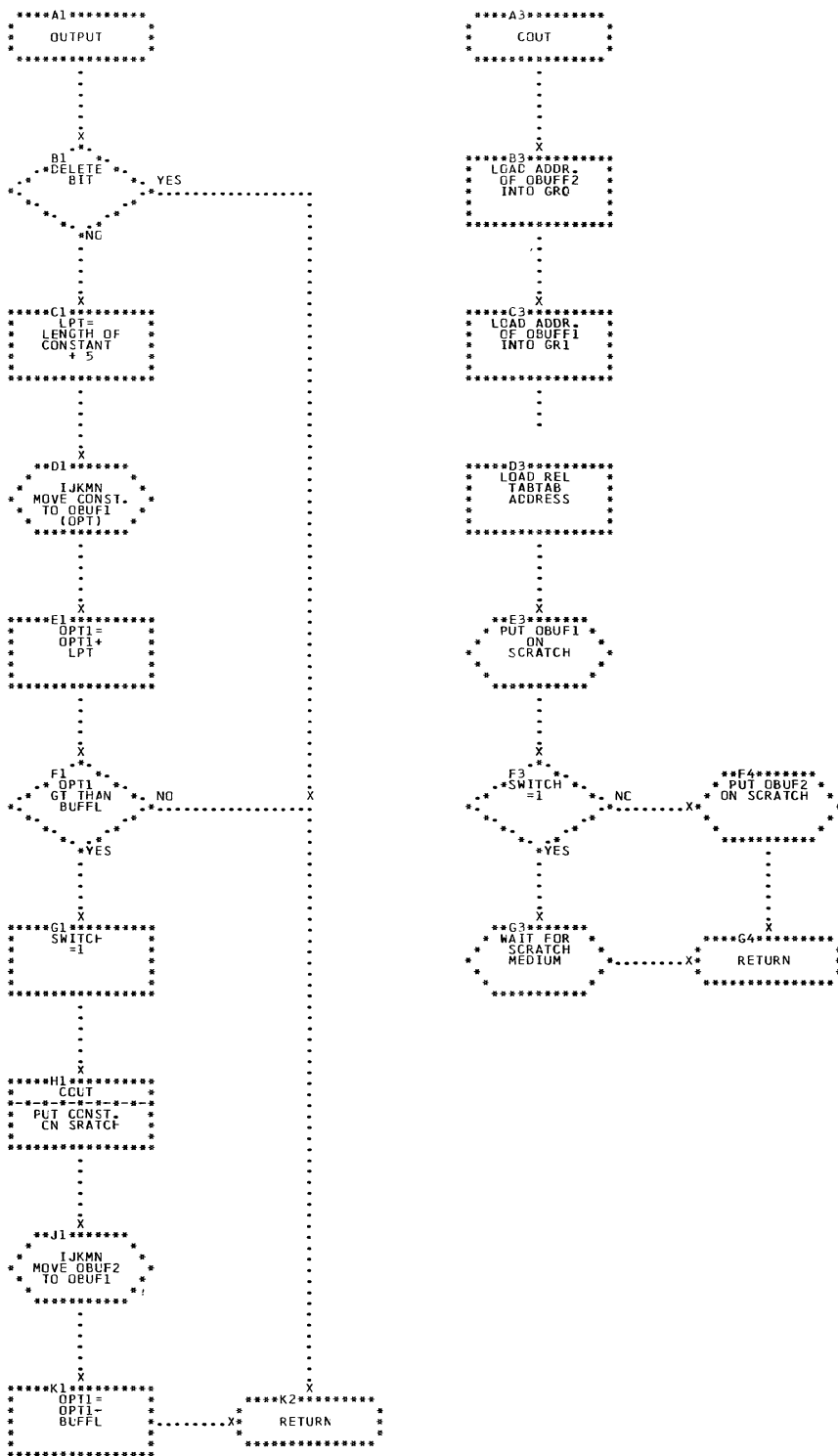


CHART XX. IJXF25

OUTPUT,COUT

*****A1*****
SKIP

.....
X
*****B1*****
INPT=
LENGTH OF
CONSTANT

.....
X
*****C1*****
INPT
GT THAN
BUFFL-20

NO
*****C2*****
RETURN

YES
.....
X
*****D1*****
WAITER FOR
IXTIN
MECIUM

.....
X
*****E1*****
INPT
GT THAN
BUFFL

NO
.....
X
*****F1*****
REC=1

YES
.....
X
*****G1*****
GECON
READ NEXT
REC FROM
IXTIN

.....
X
*****H1*****
RETURN

*****A3*****
CCNEND

.....
X
*****B3*****
MOVE
END OF
CONTAB KEY
TO OBUF1(GPT)

.....
X
*****C3*****
OPT1=
OPT1+2

.....
X
*****D3*****
OPT1
GT THAN
BUFFL

YES
*****D4*****
SWITCH
=2

NO
.....
X
*****E3*****
SWITCH
=1

*****E4*****
COUT
PUT LAST
RECORDS ON
SCRATCH

.....
X
*****F4*****
RETURN

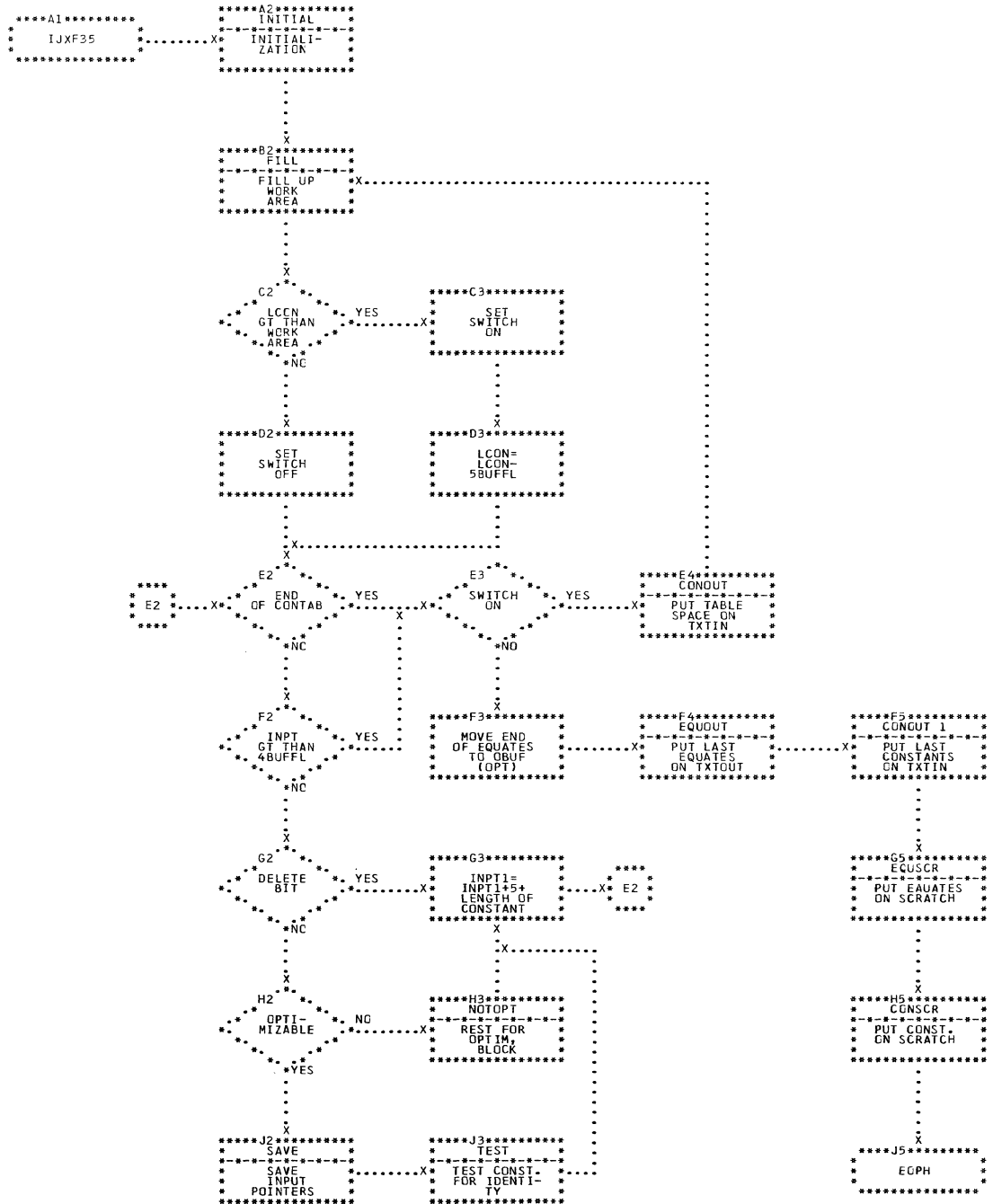
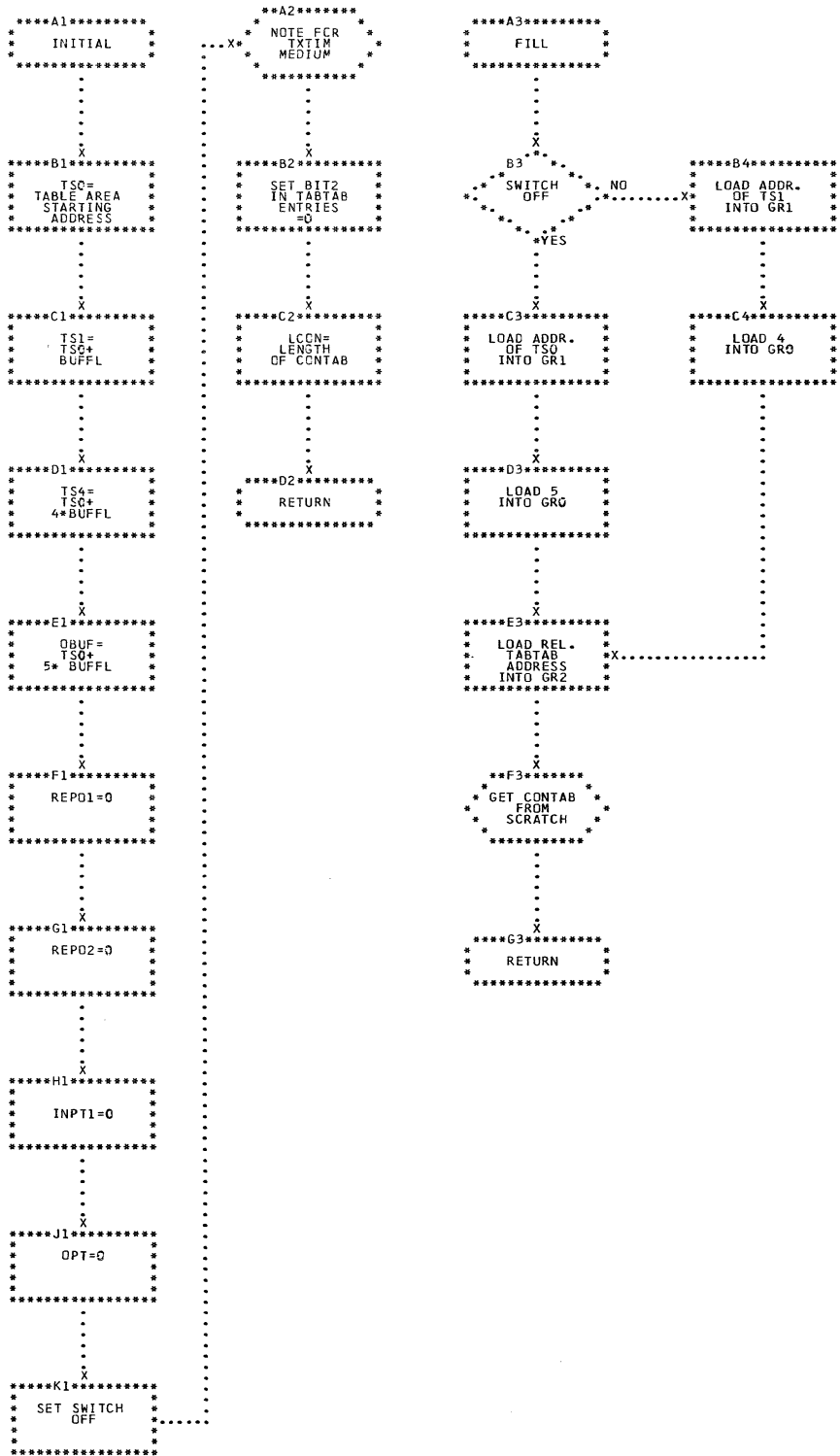
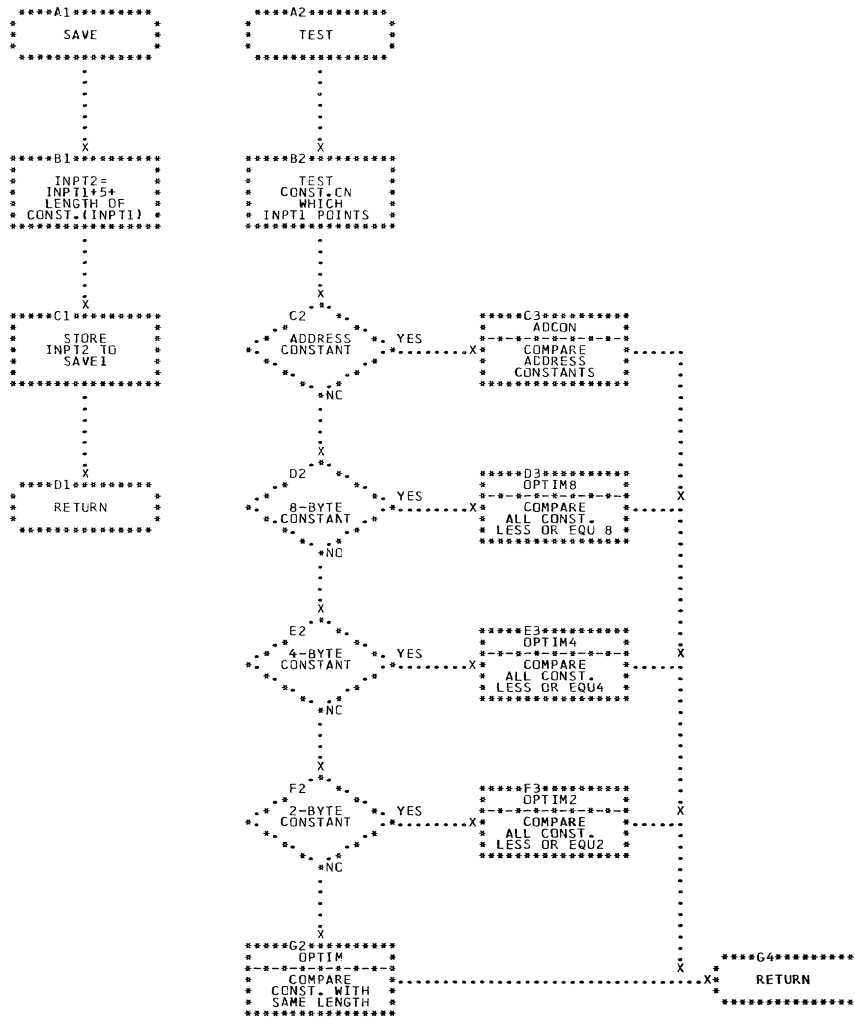


CHART YA. IJXF35

OPTIMIZATION OF CONSTANTS





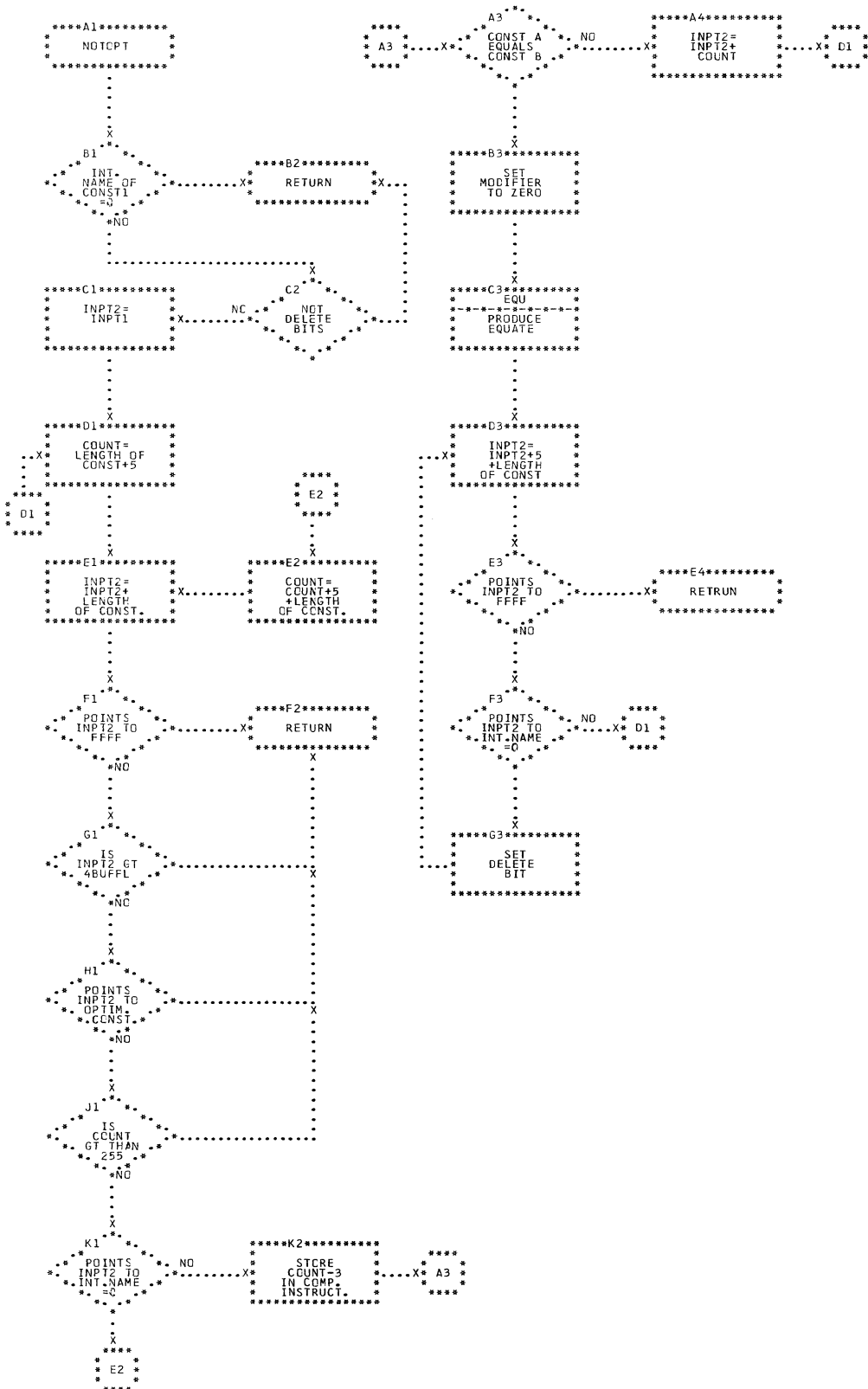
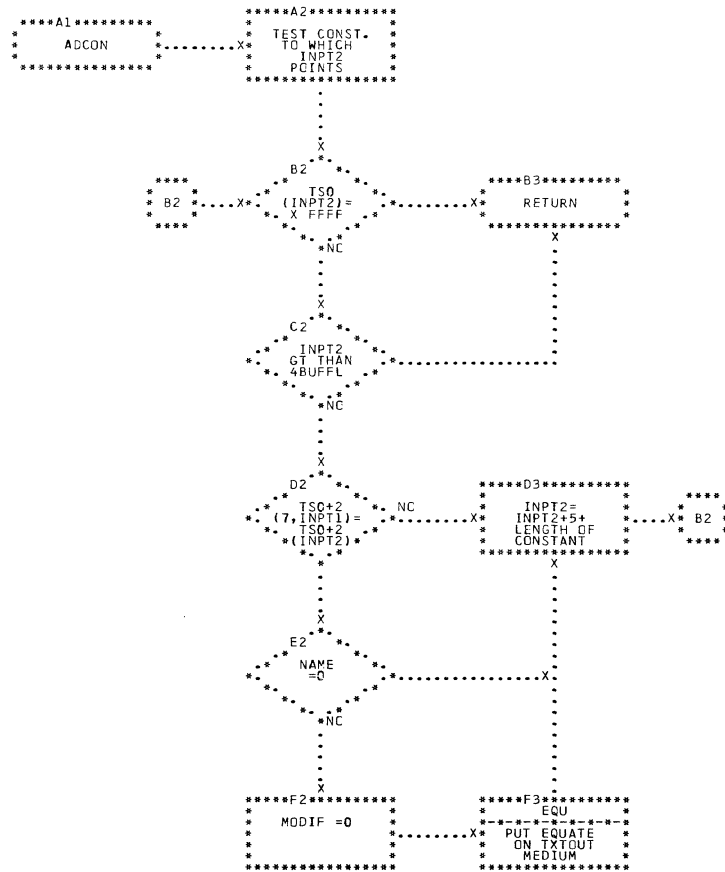
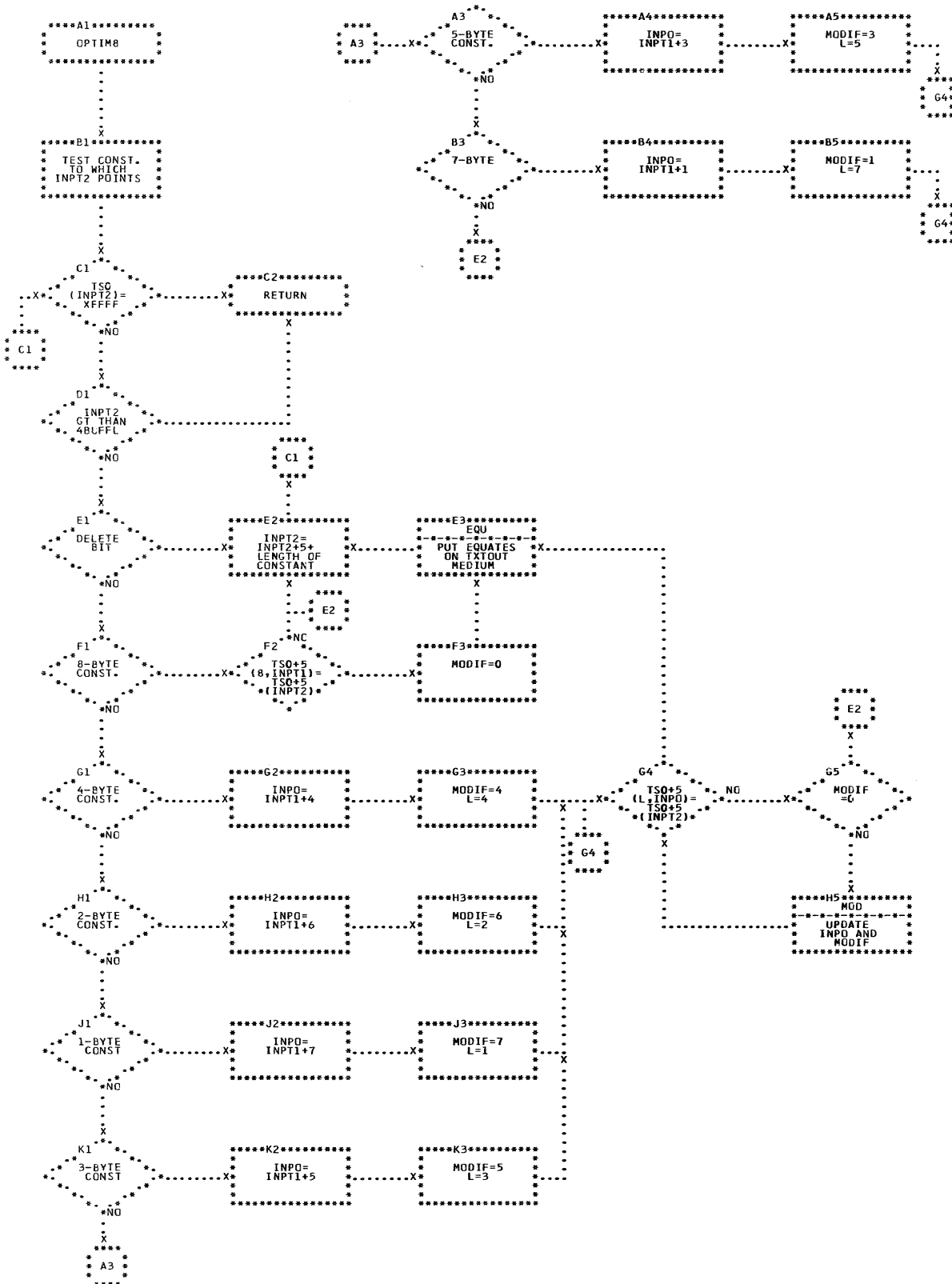
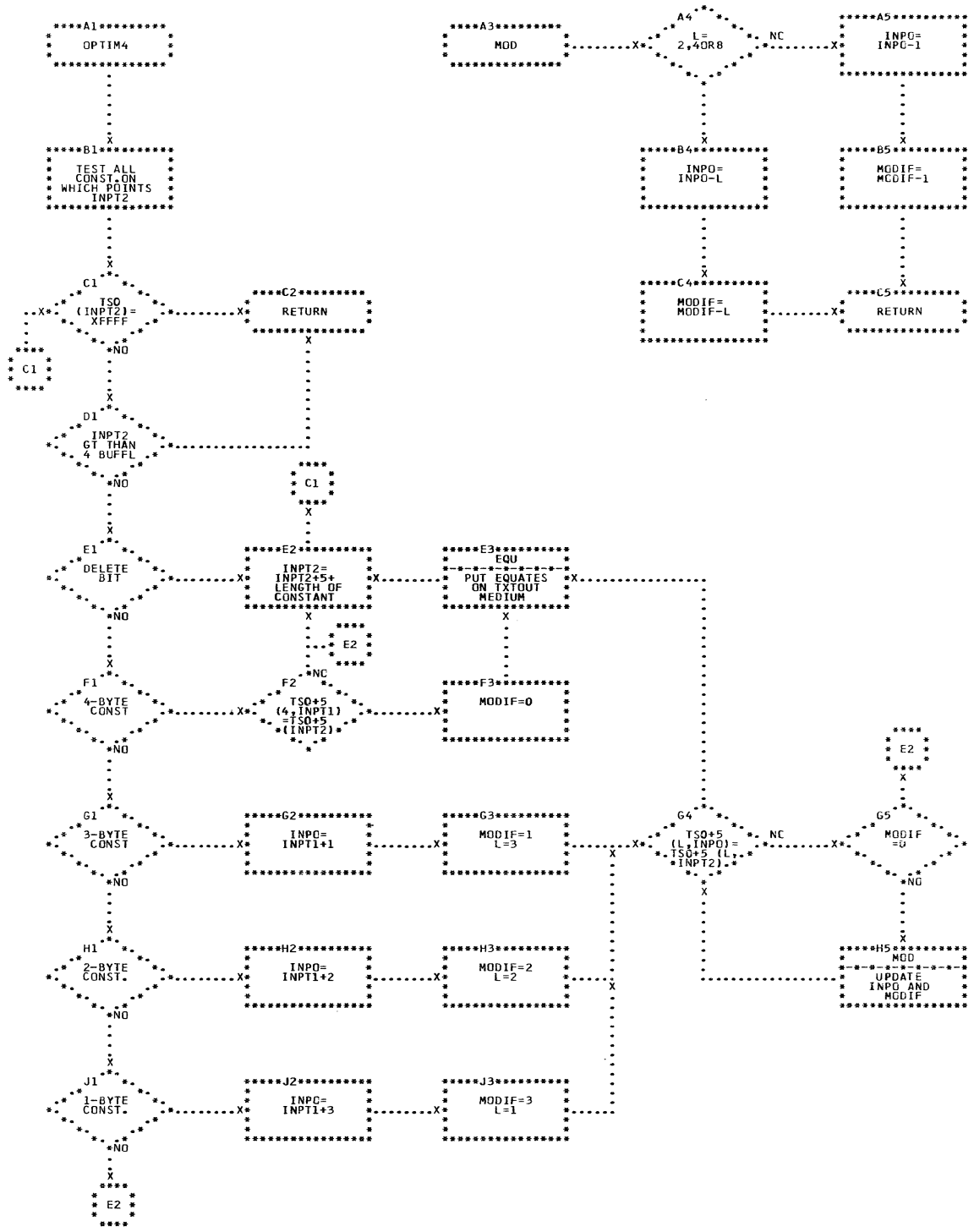
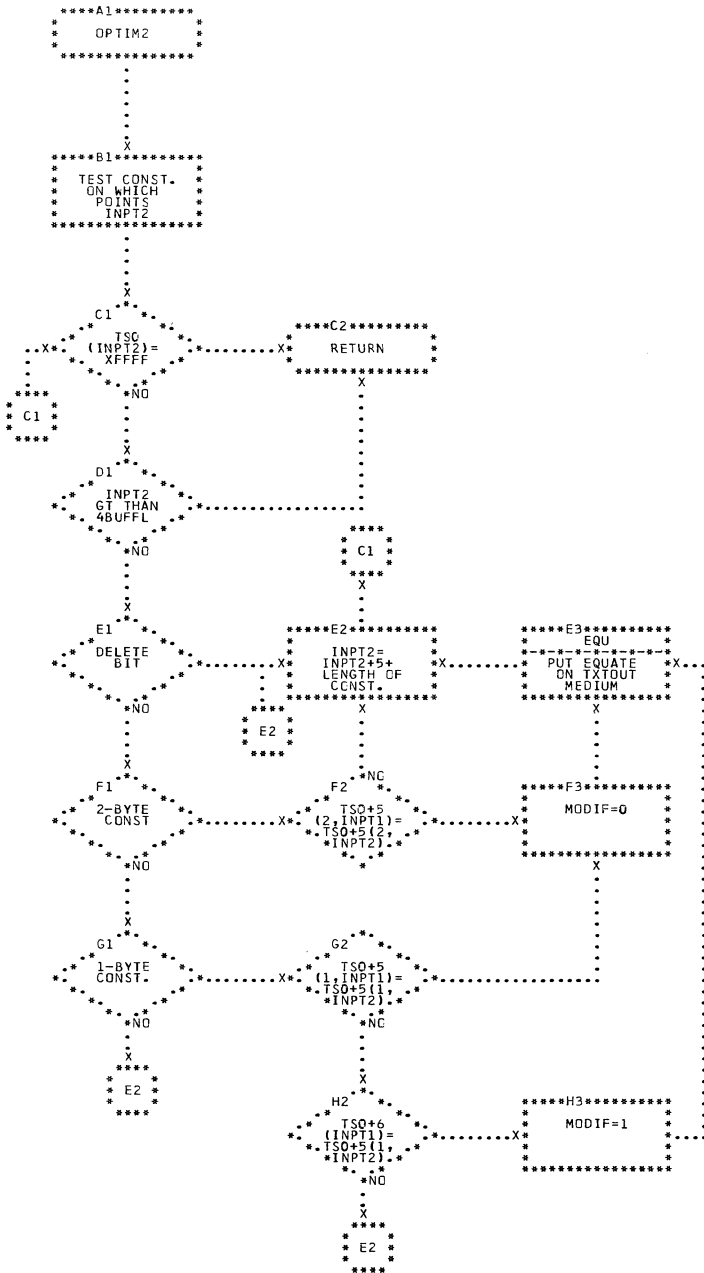


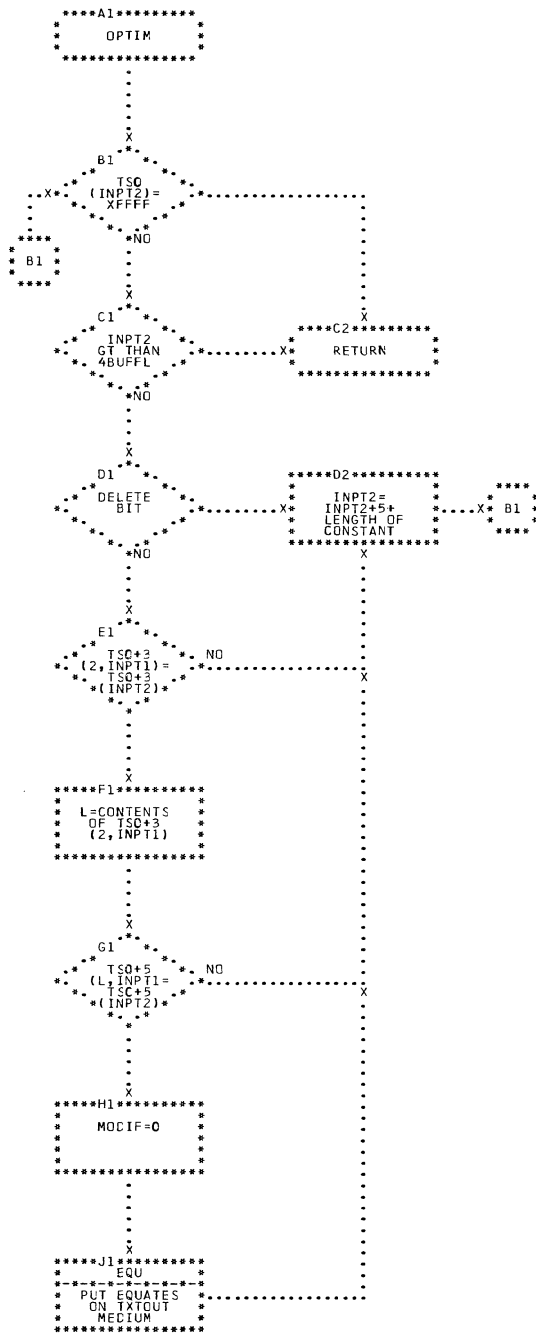
CHART YD. IJXF35 NOTOPT

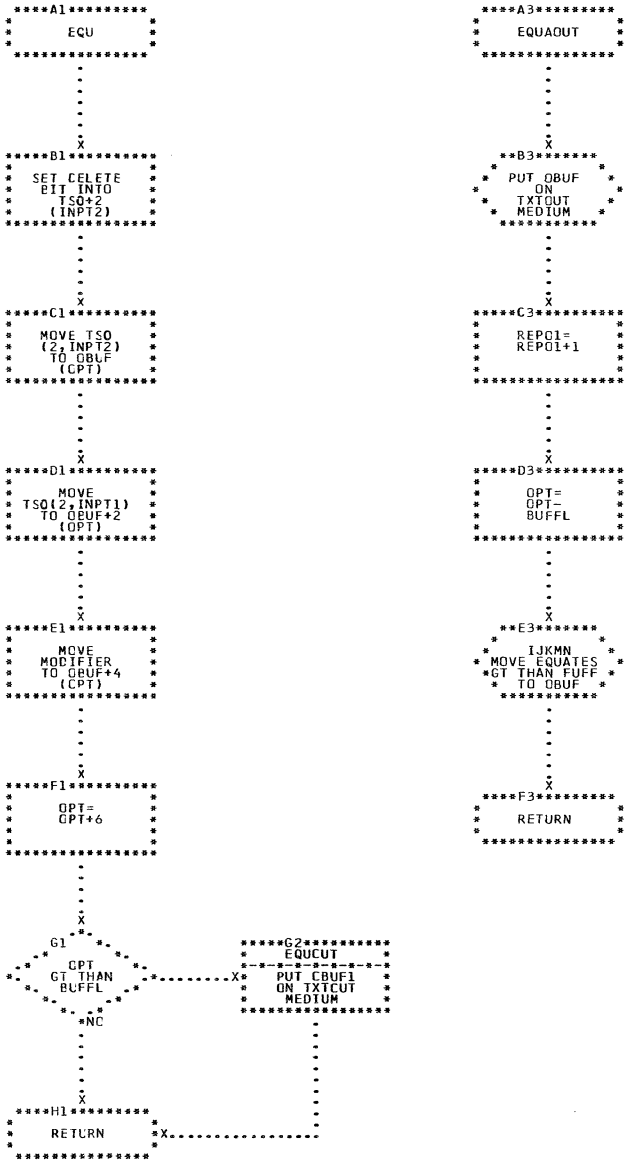


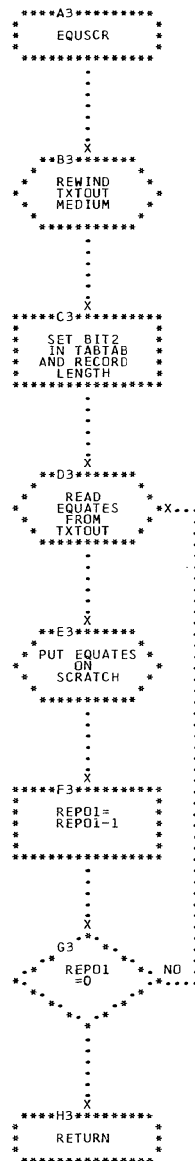
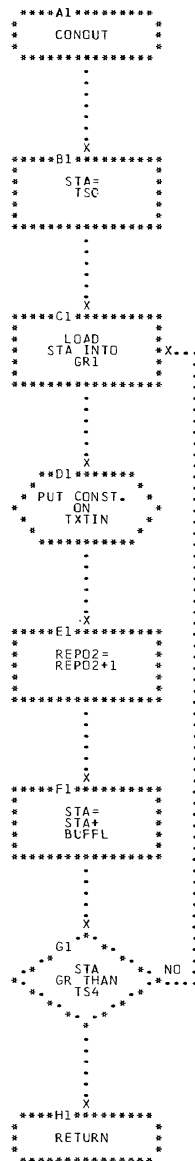


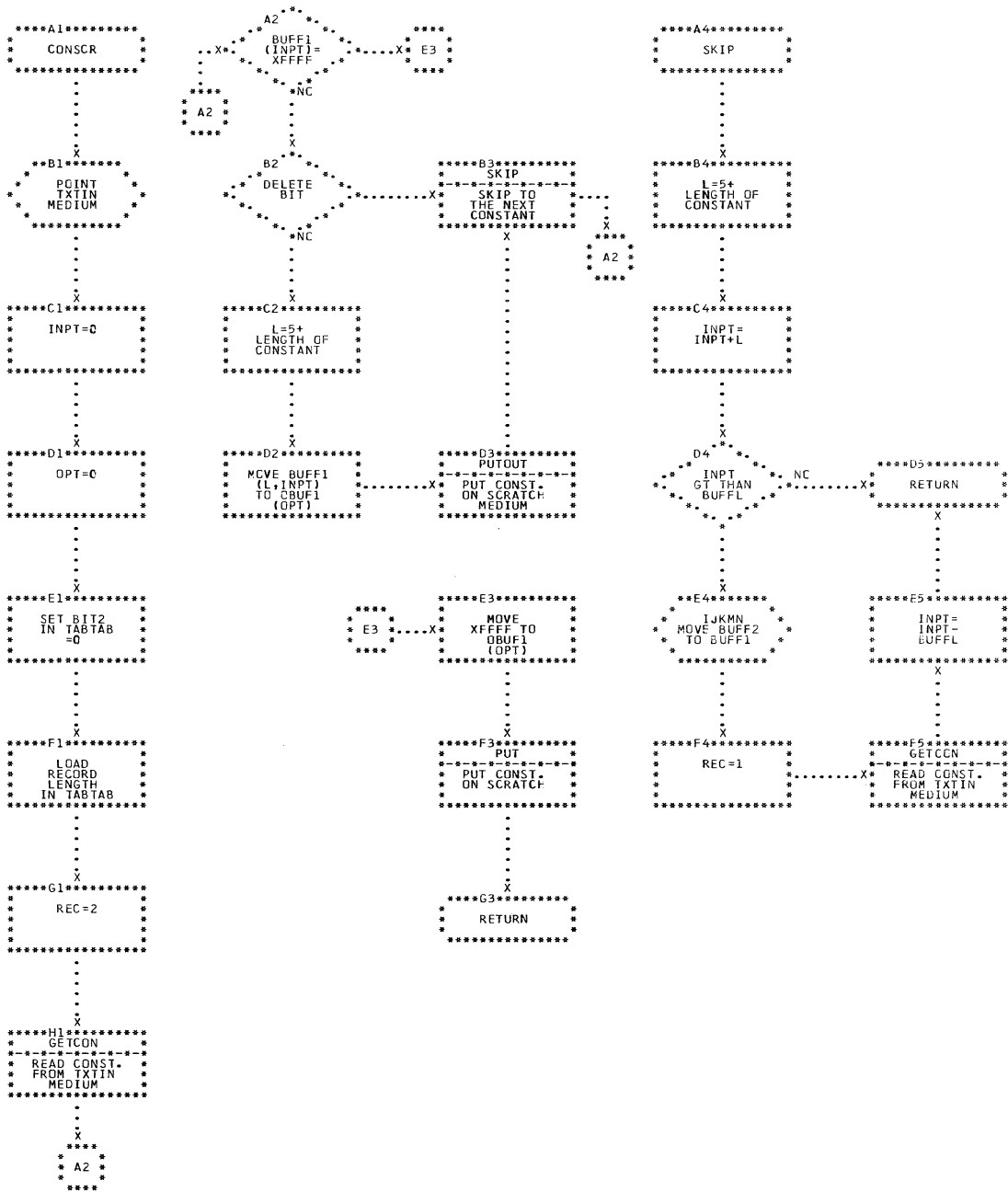


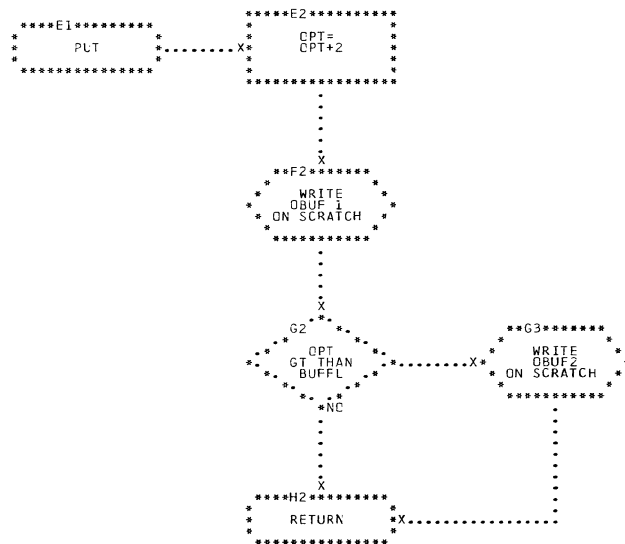
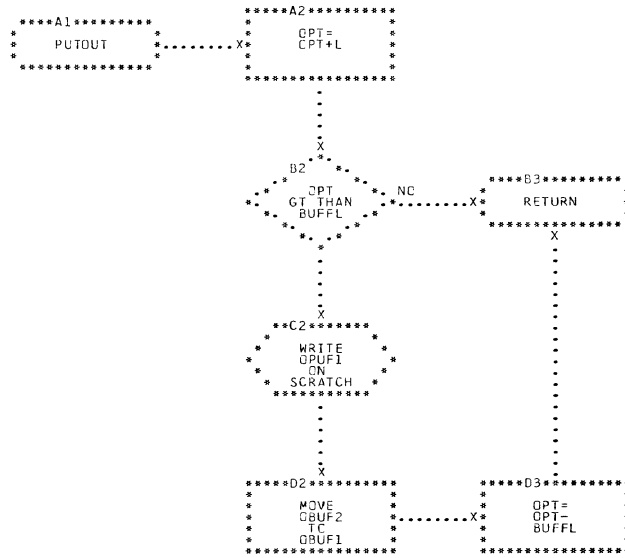


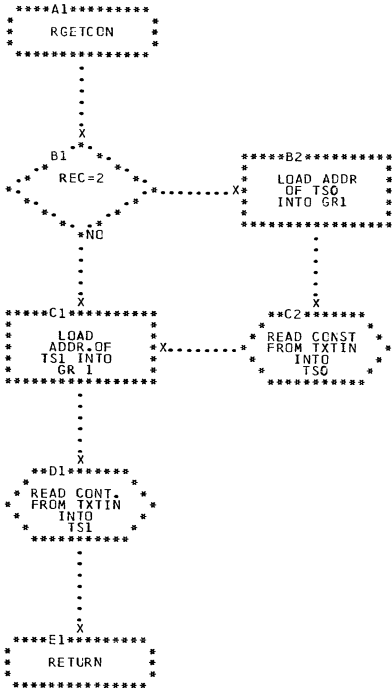


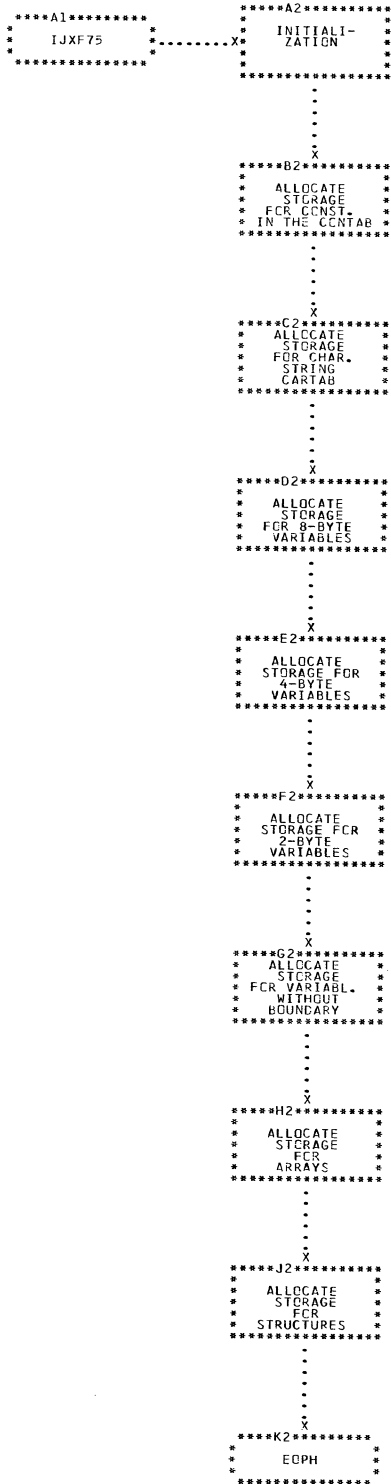


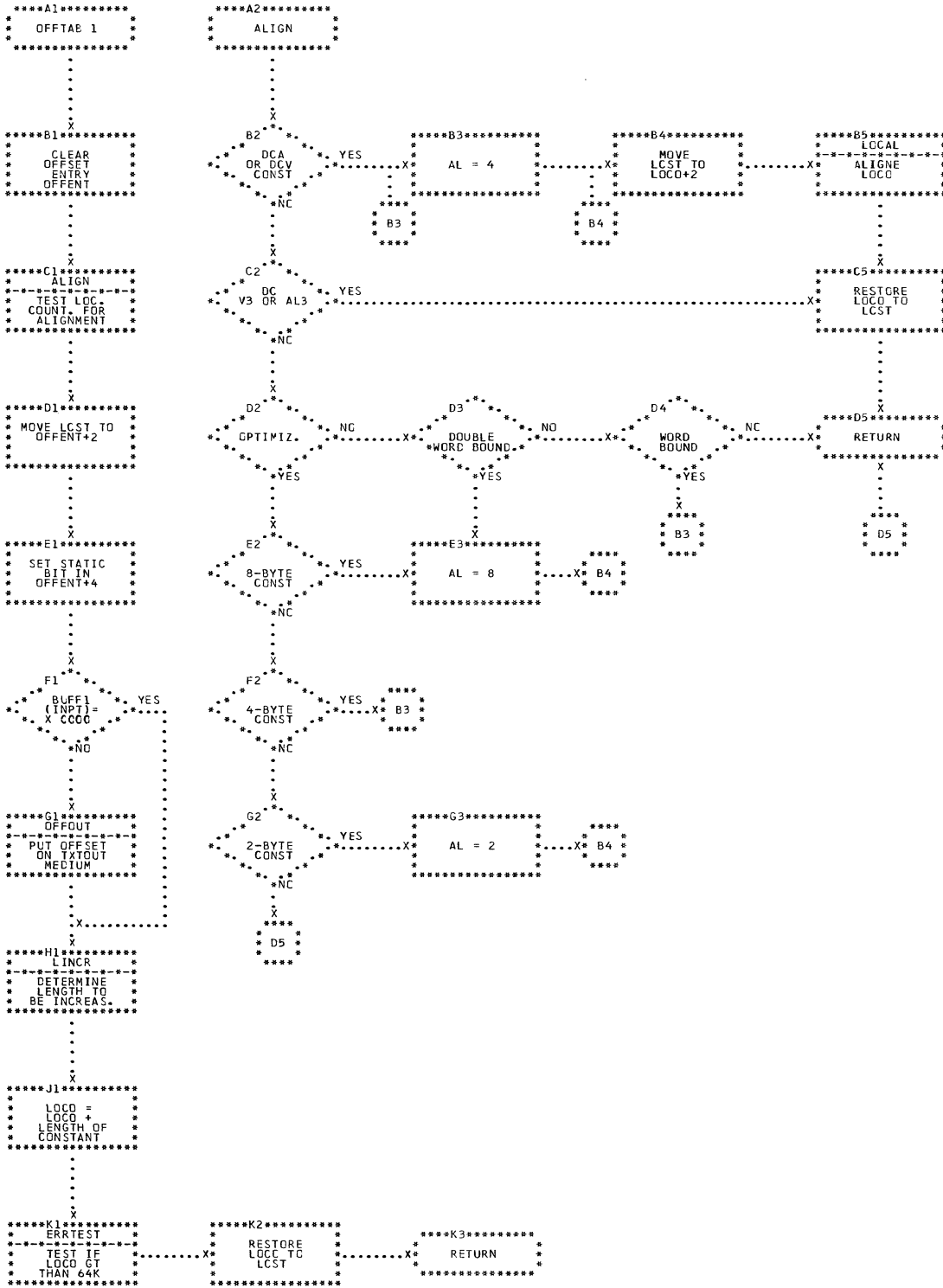


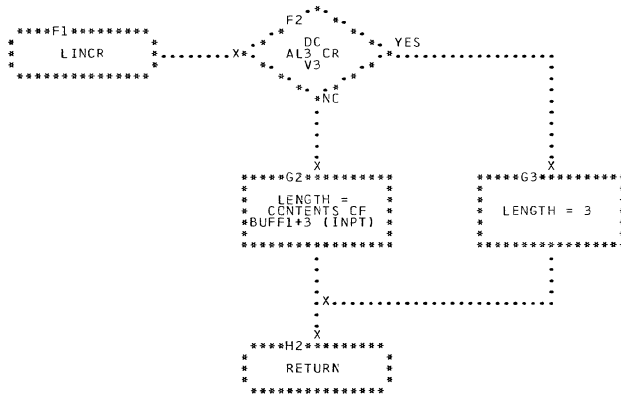
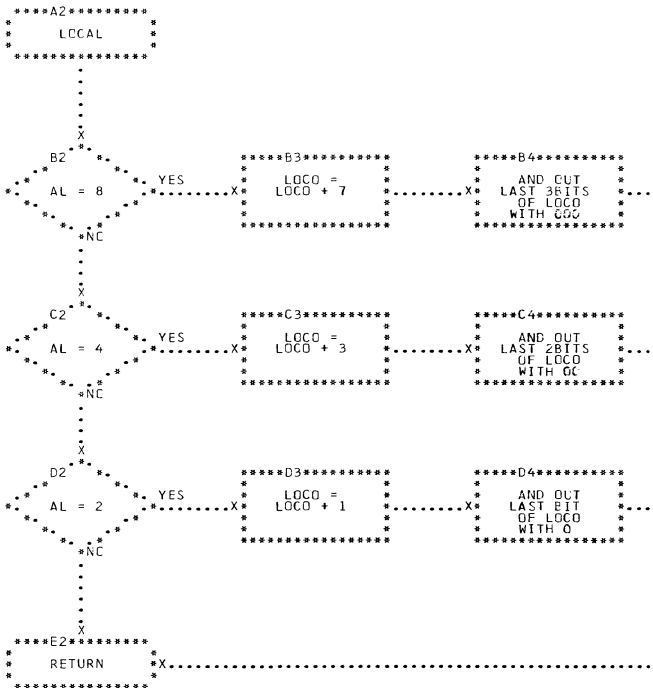


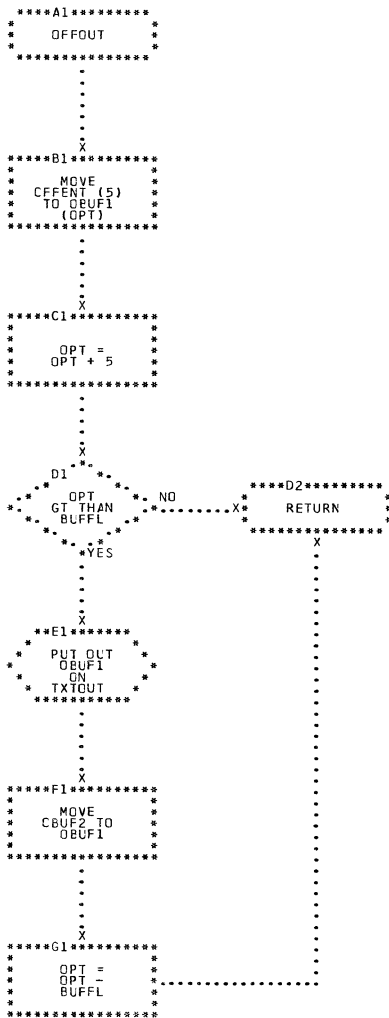


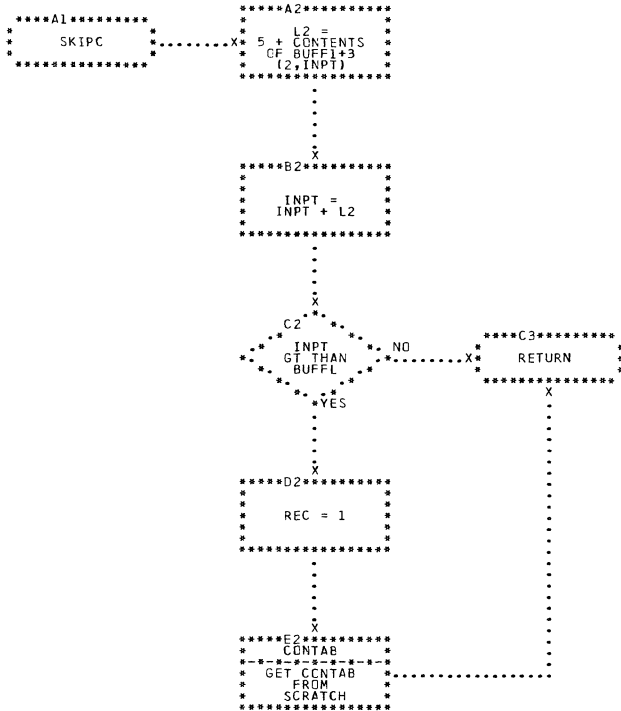


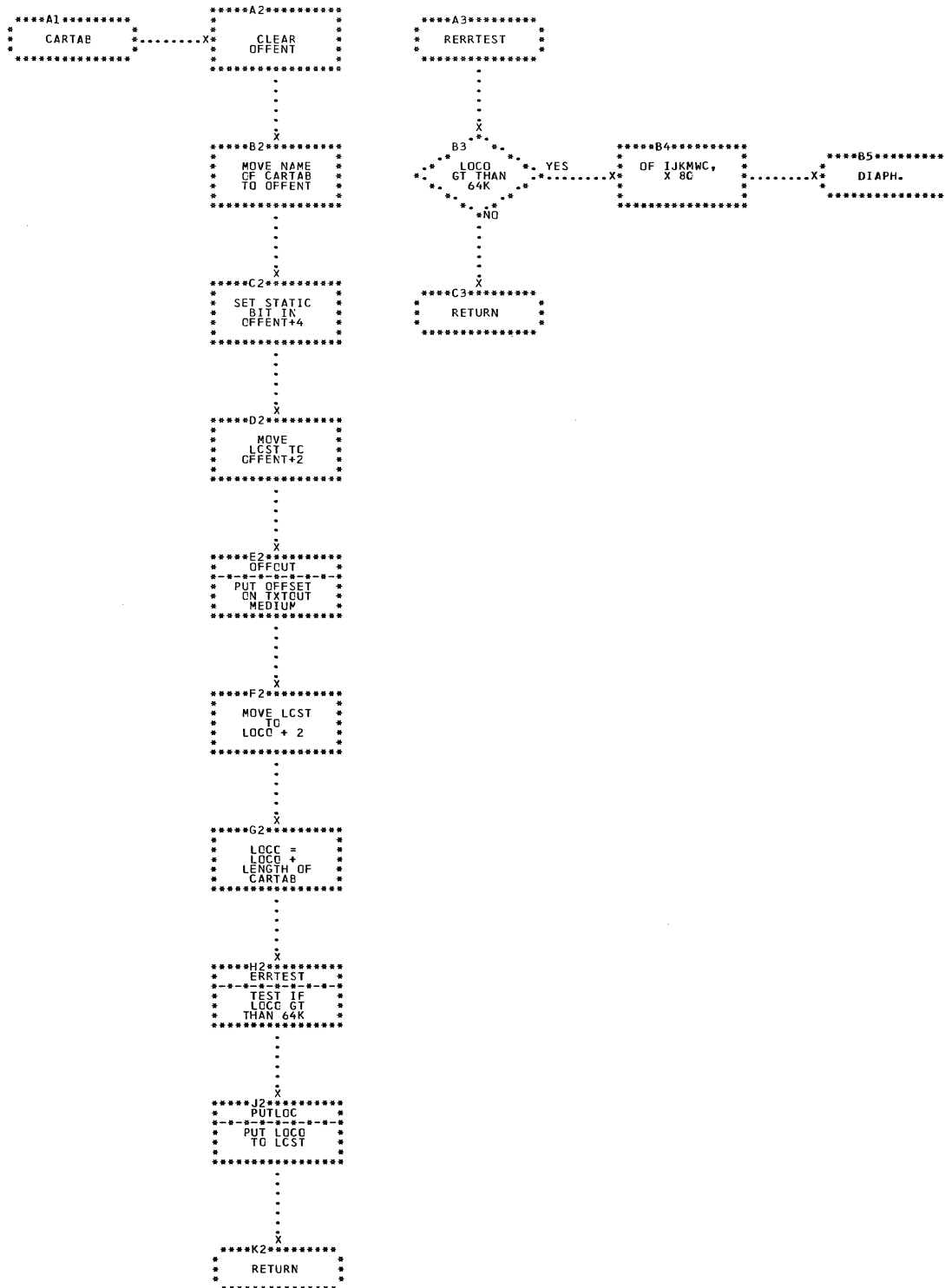












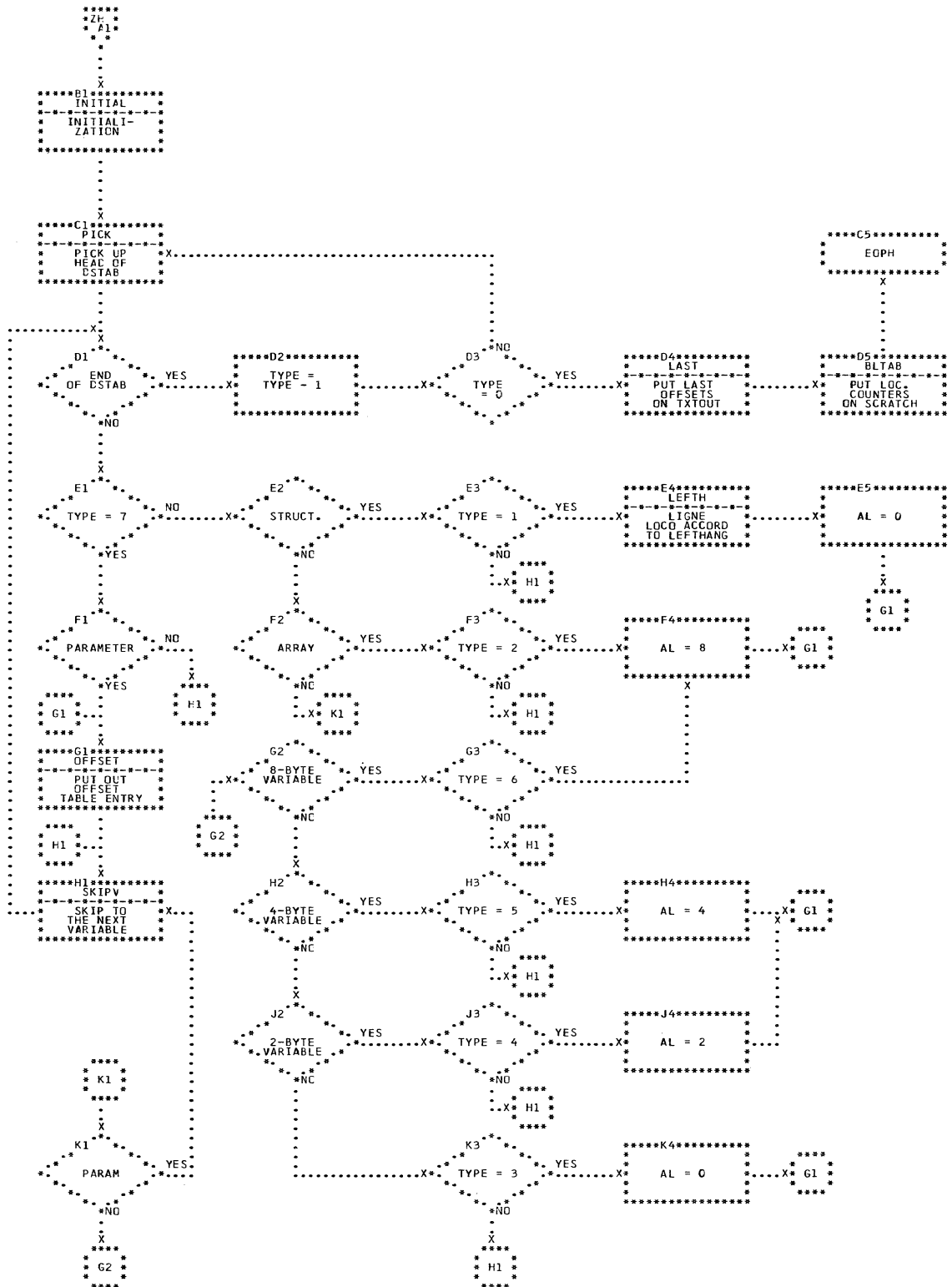
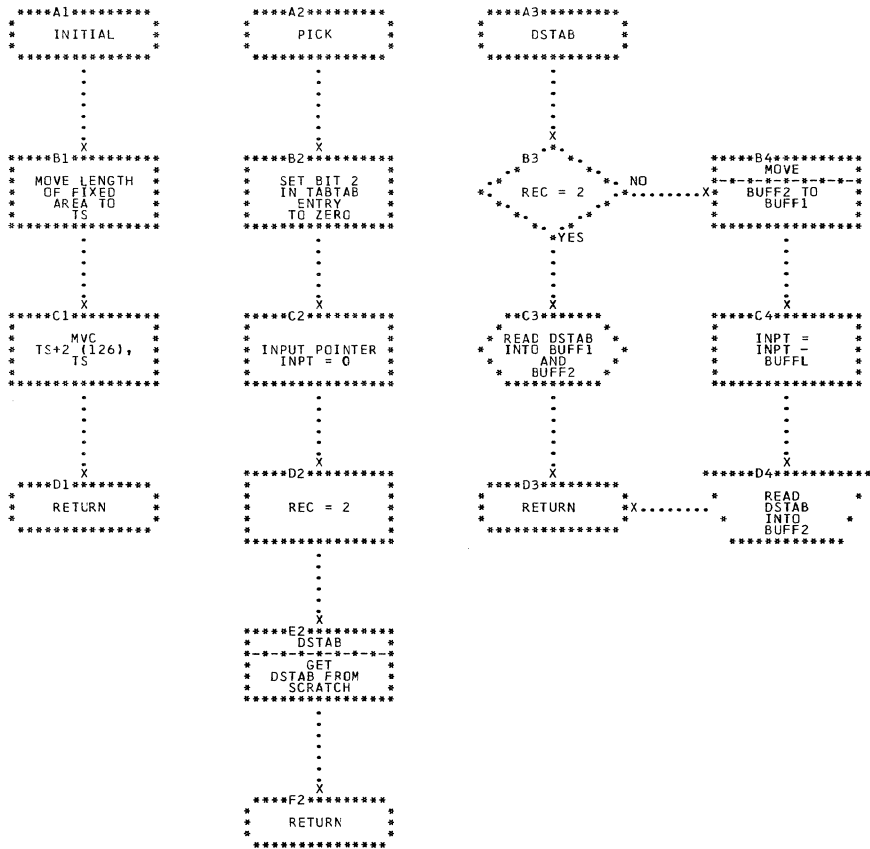


CHART ZH. IJXF75

ALLOCATION OF VARIABLES



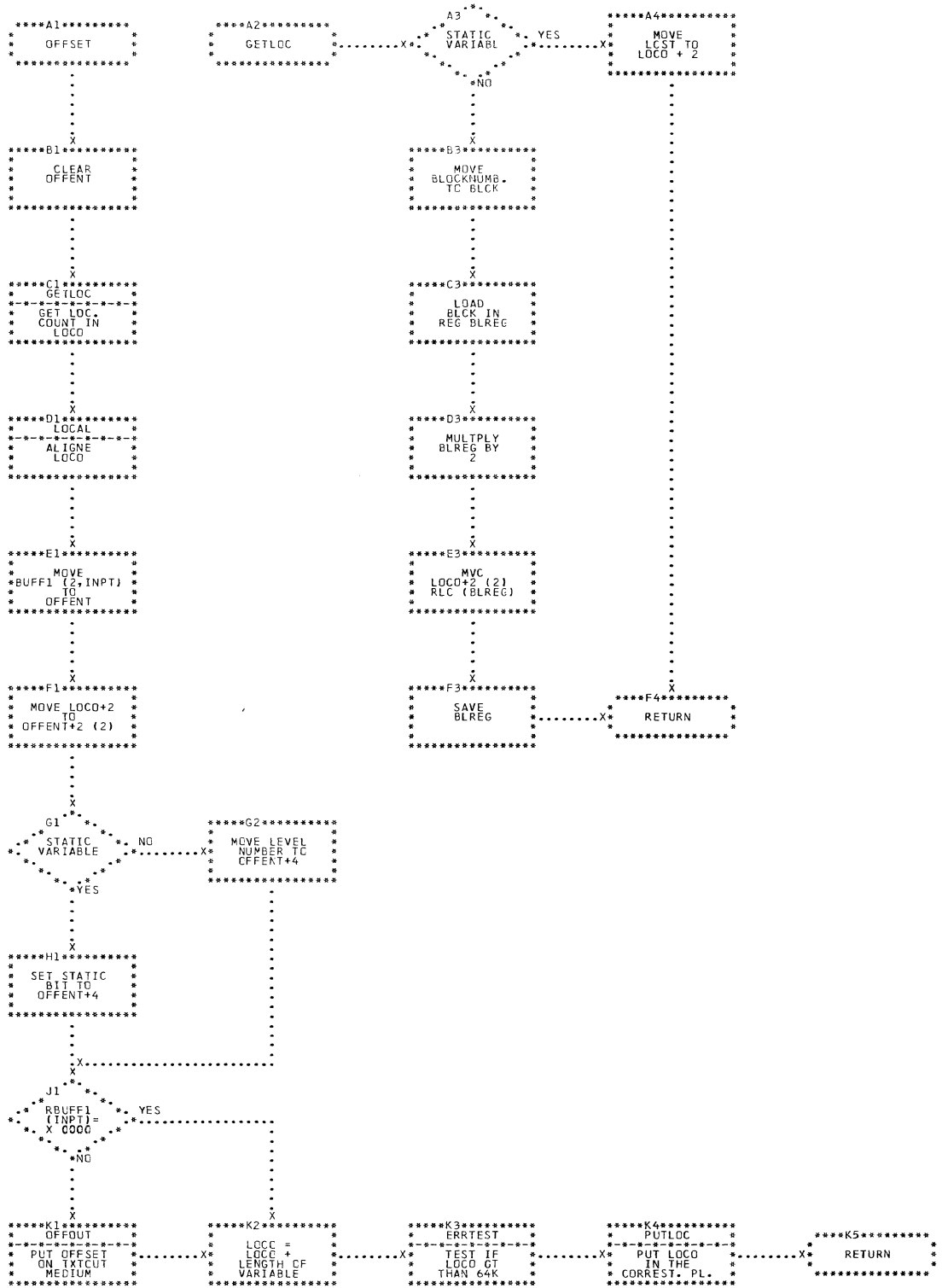
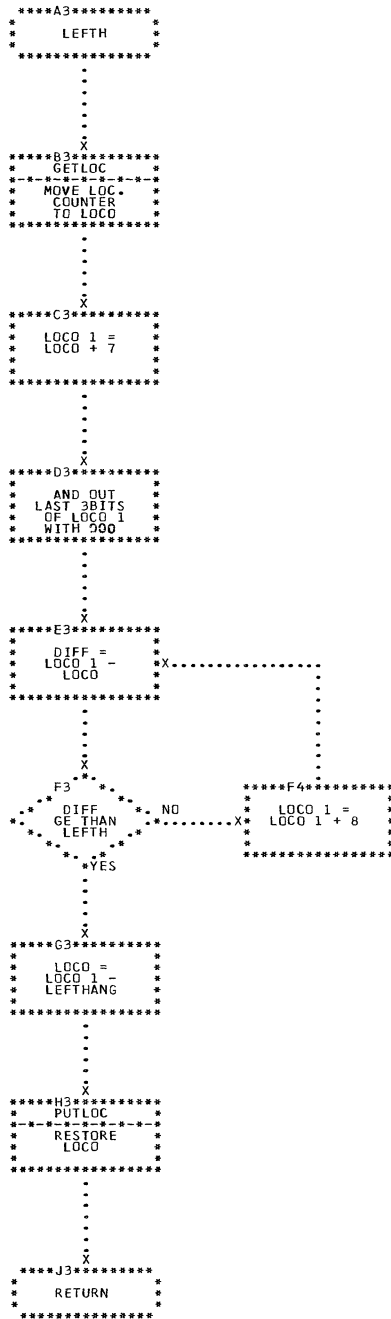
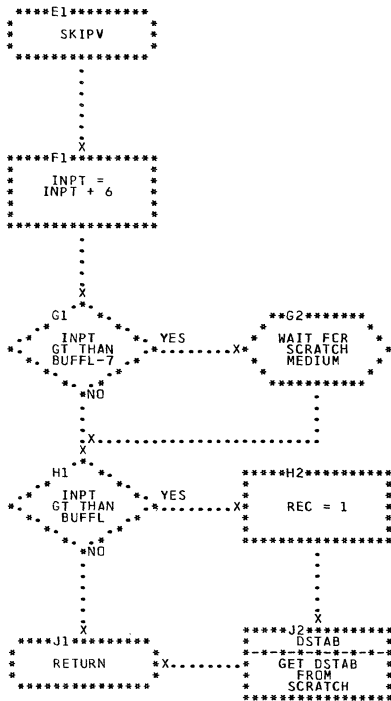
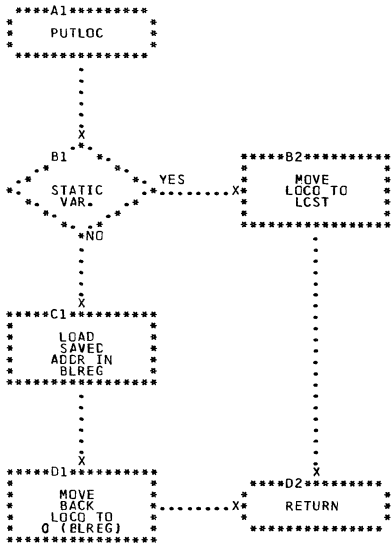
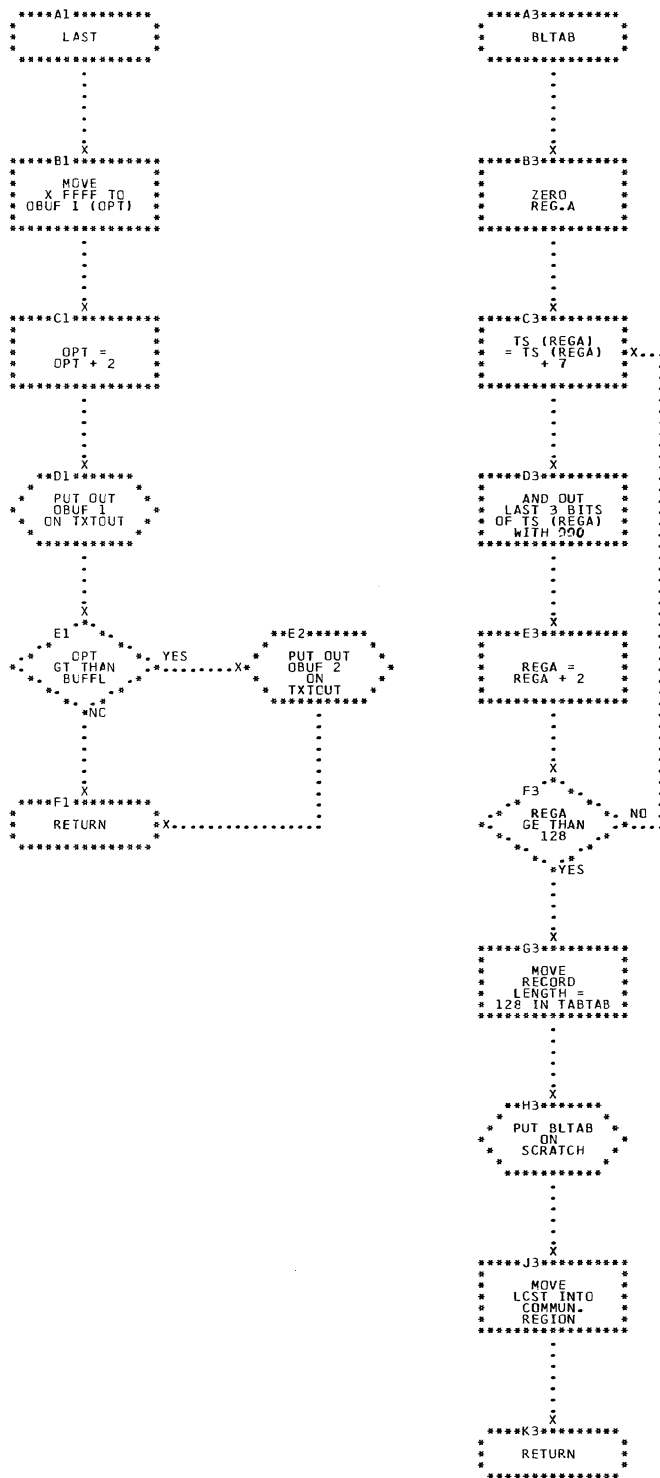
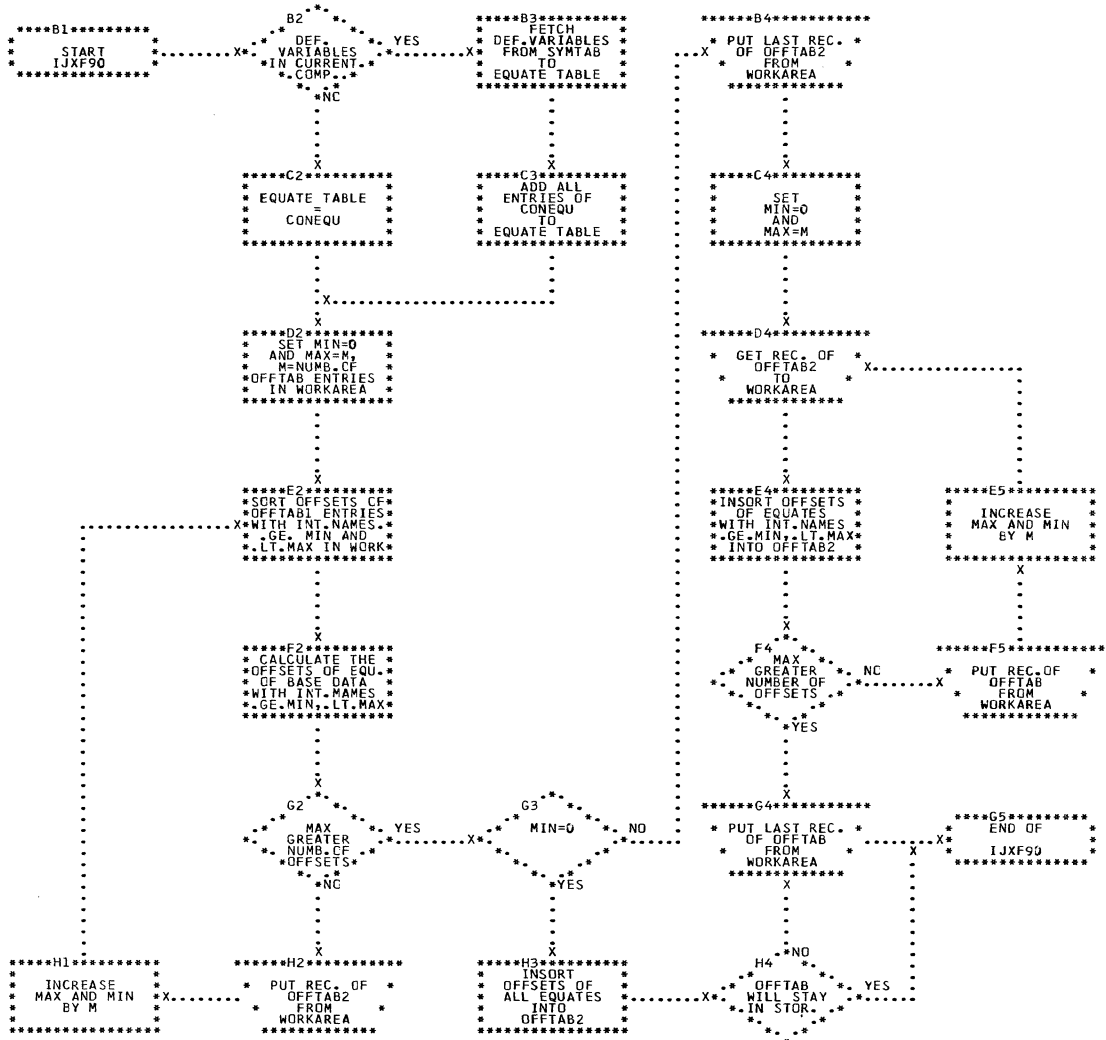


CHART 2J. IJXF75

OFFSET, GETLOC







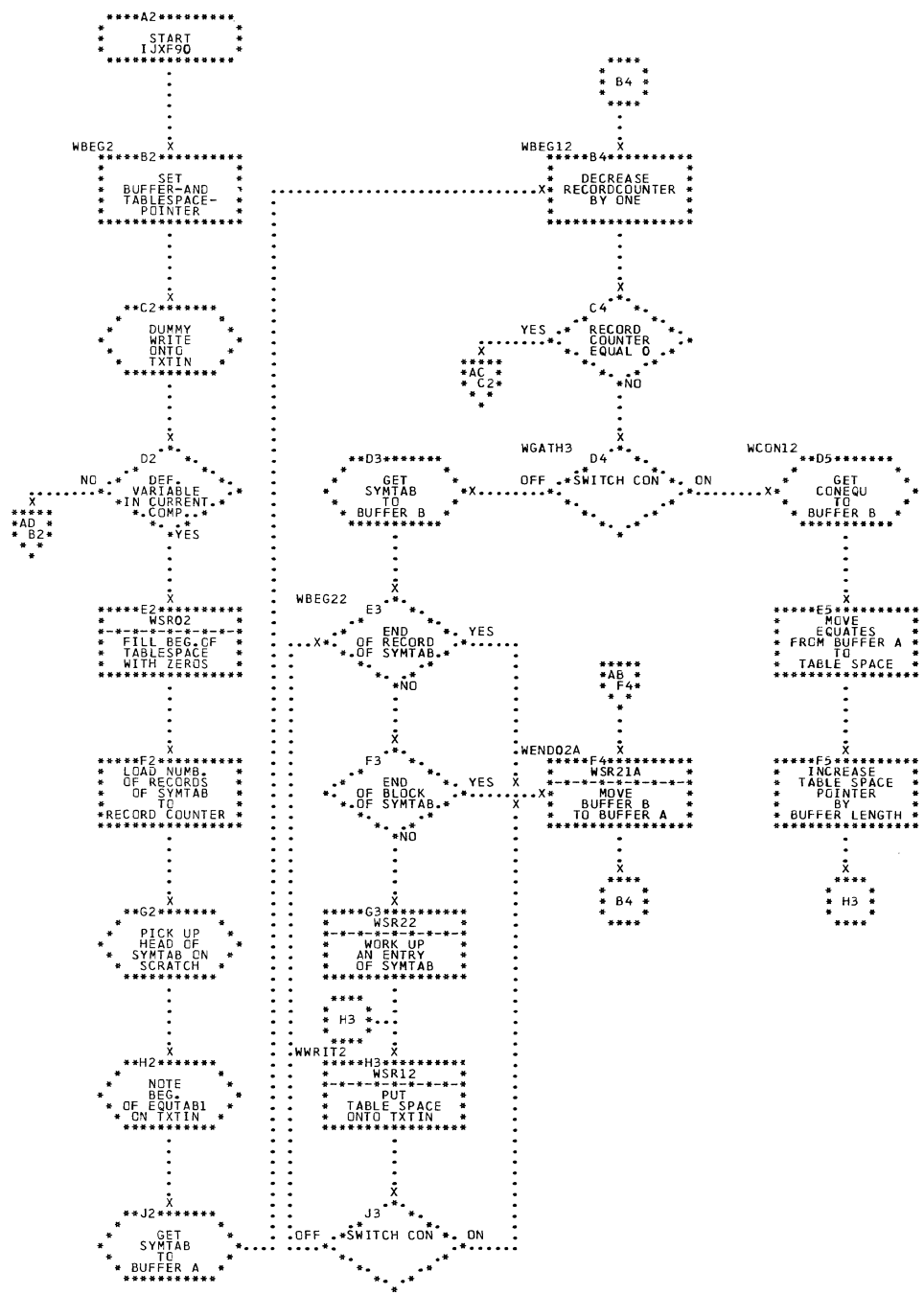
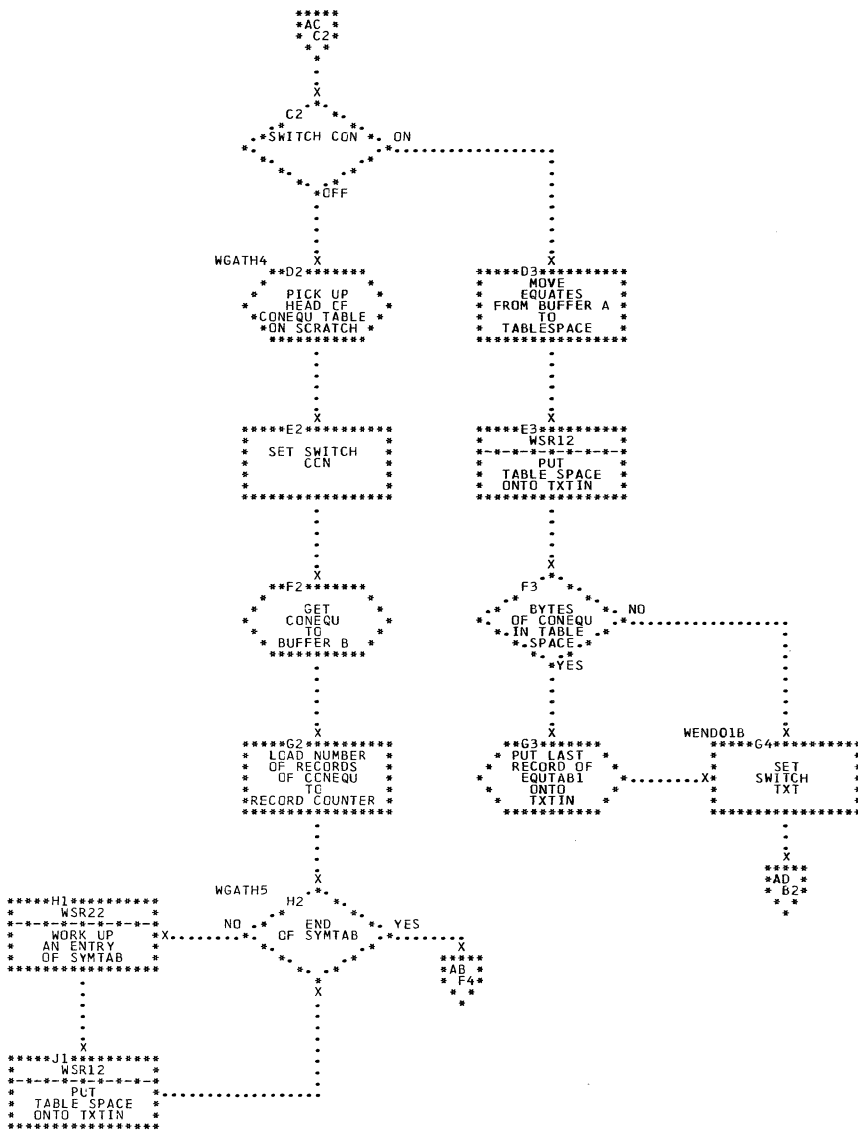


CHART AB. IJXF90

INITIAL. & GATHERING OF EQU.



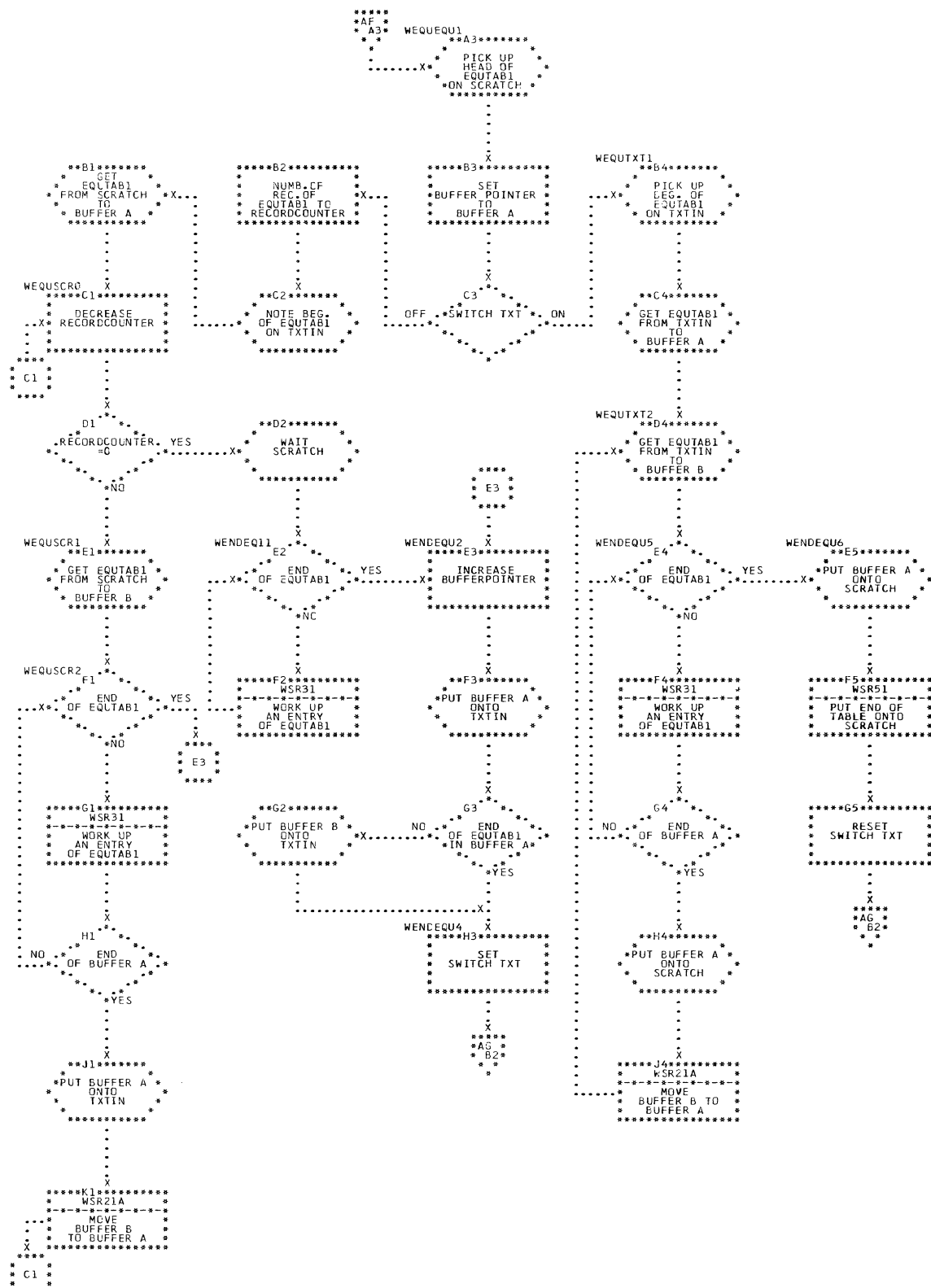
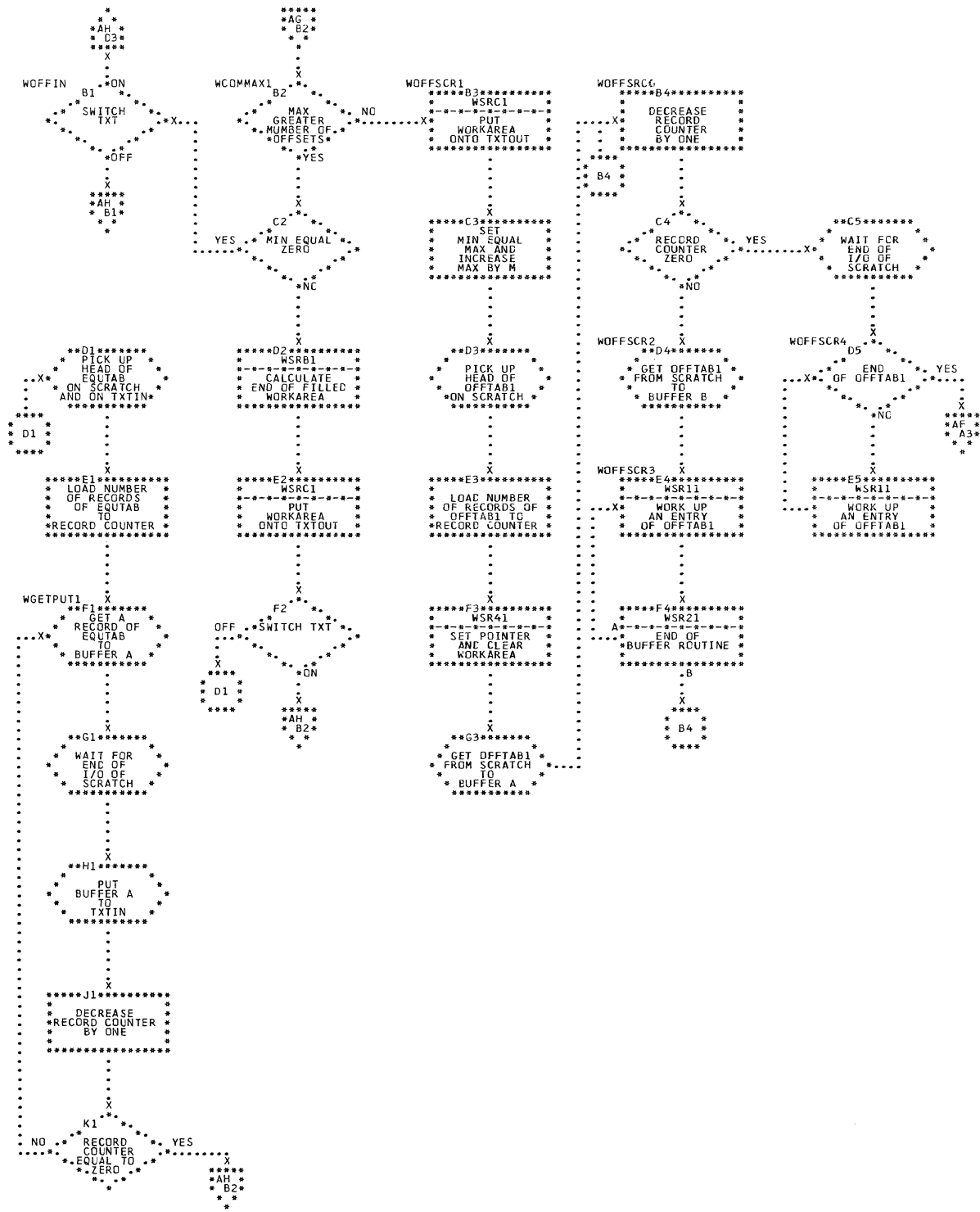


CHART AF. IJXF90

CALCULATING OFFSETS OF EQUATES



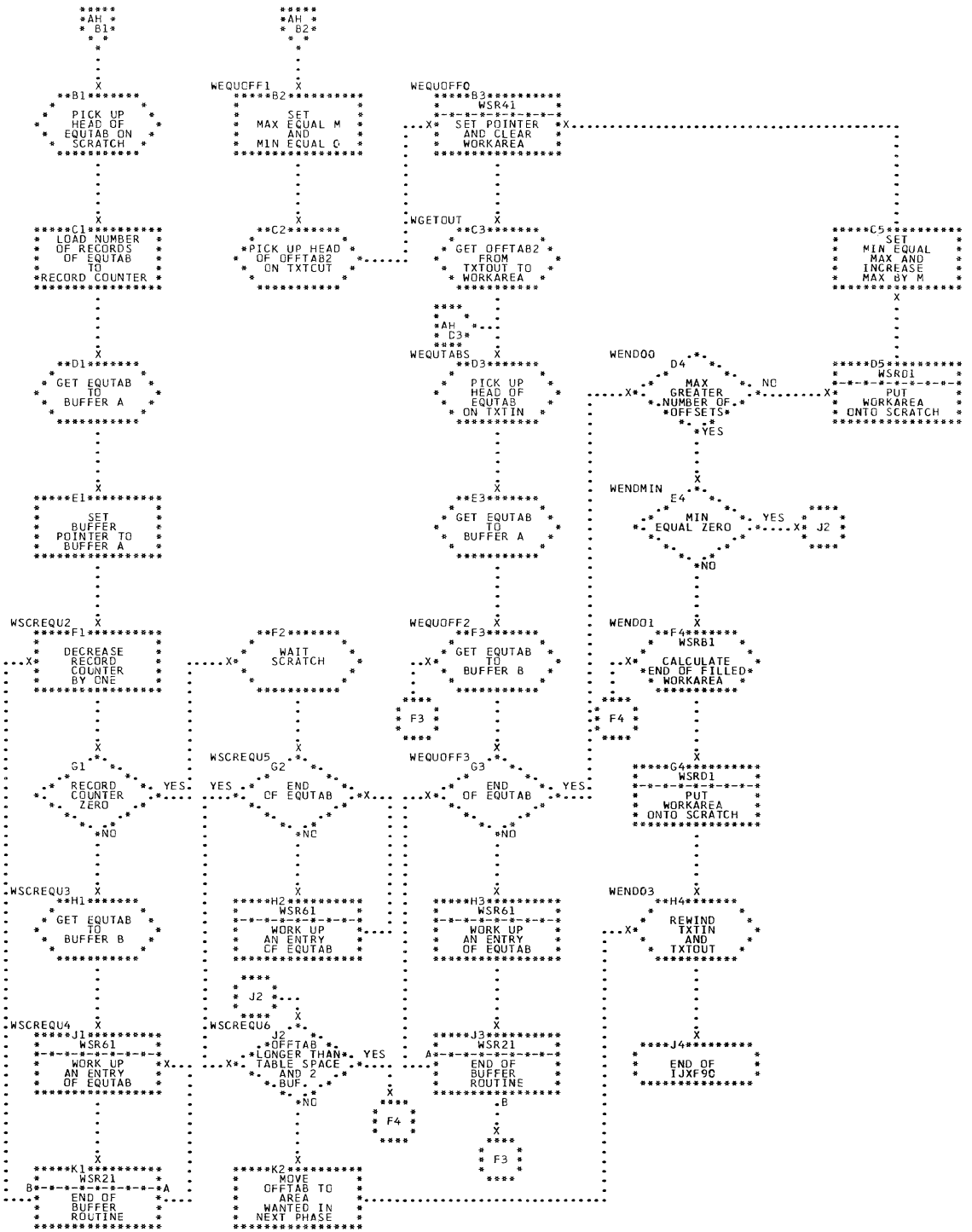
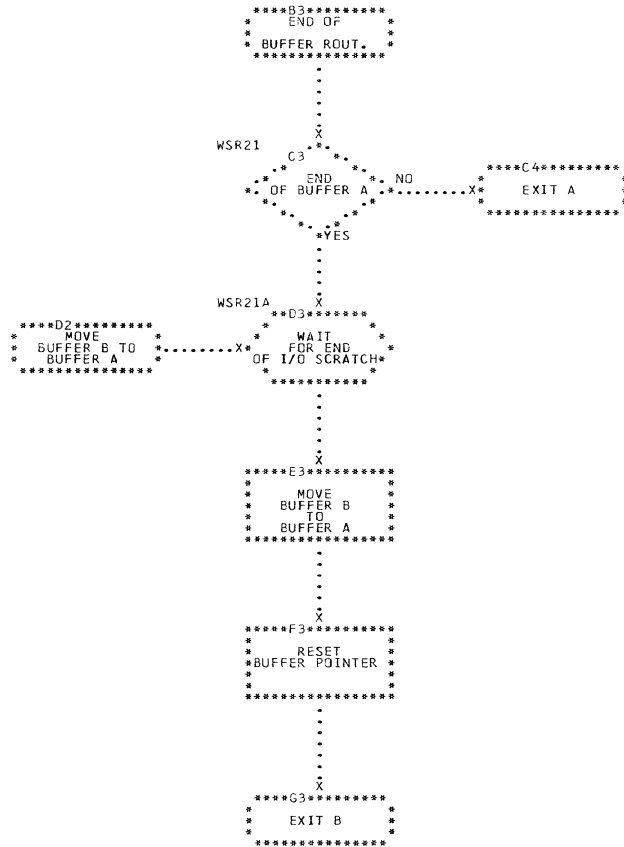
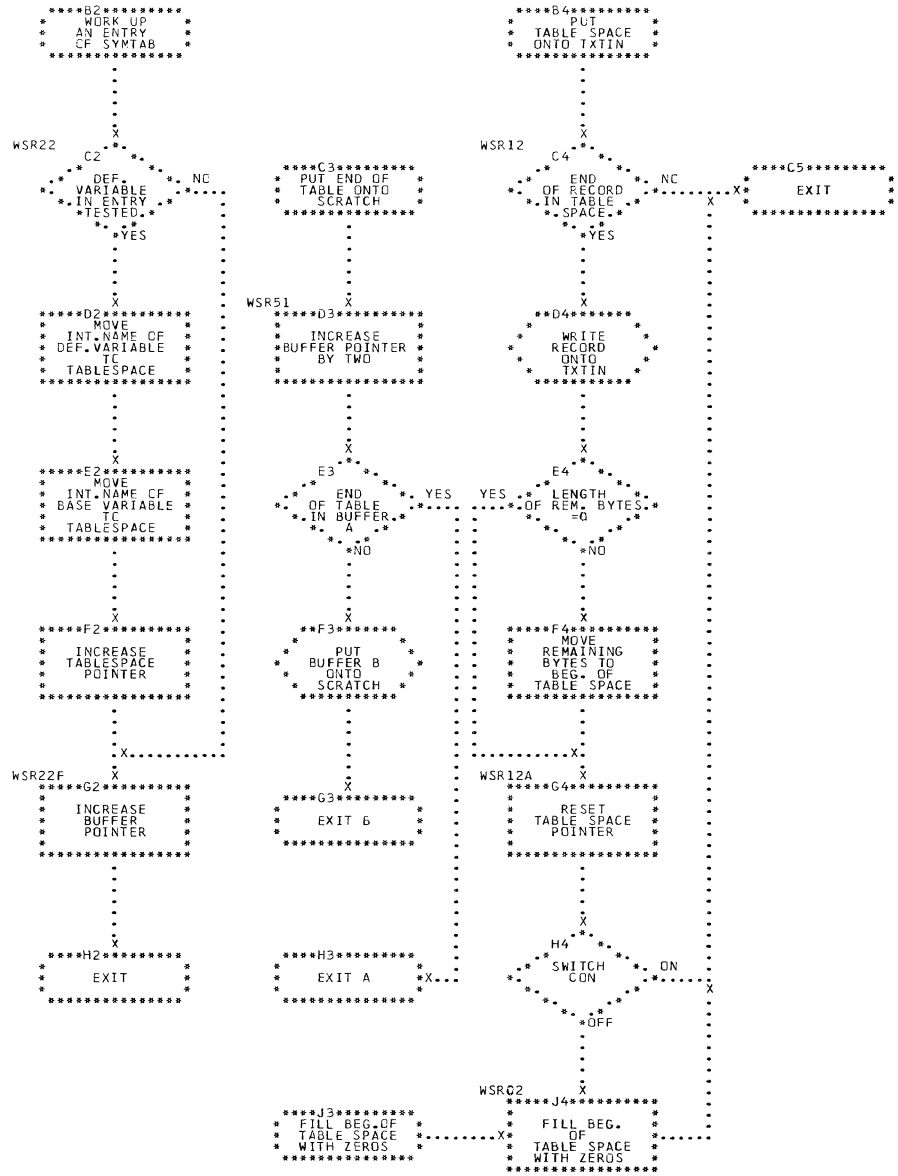
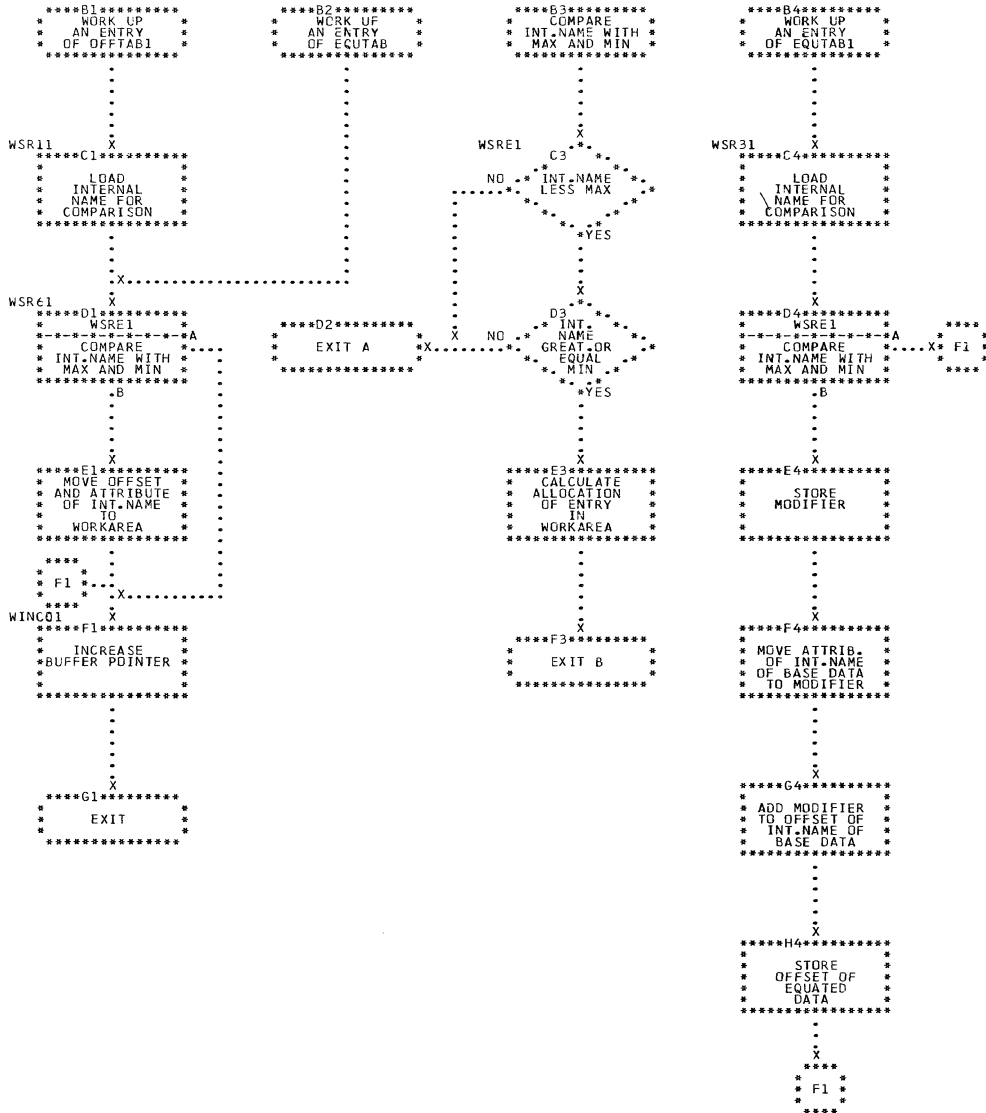


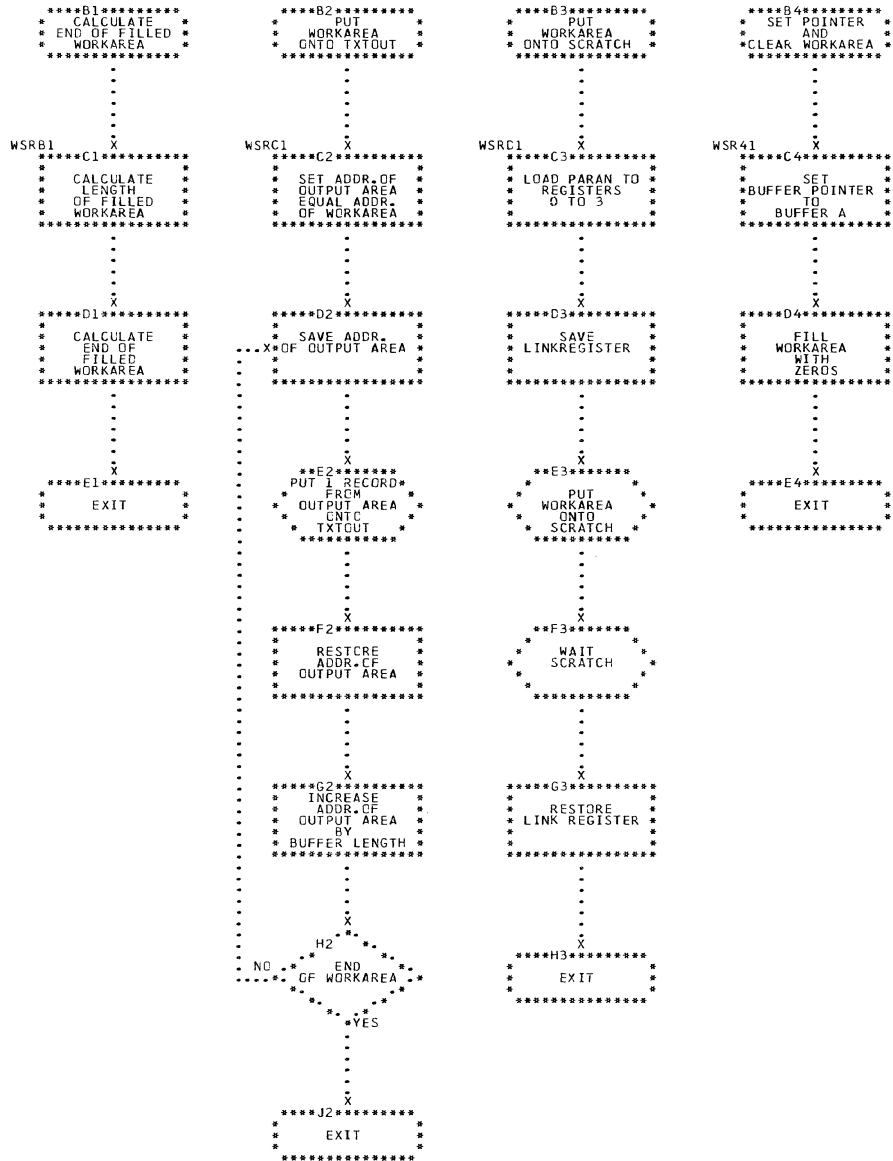
CHART AH. IJXF90

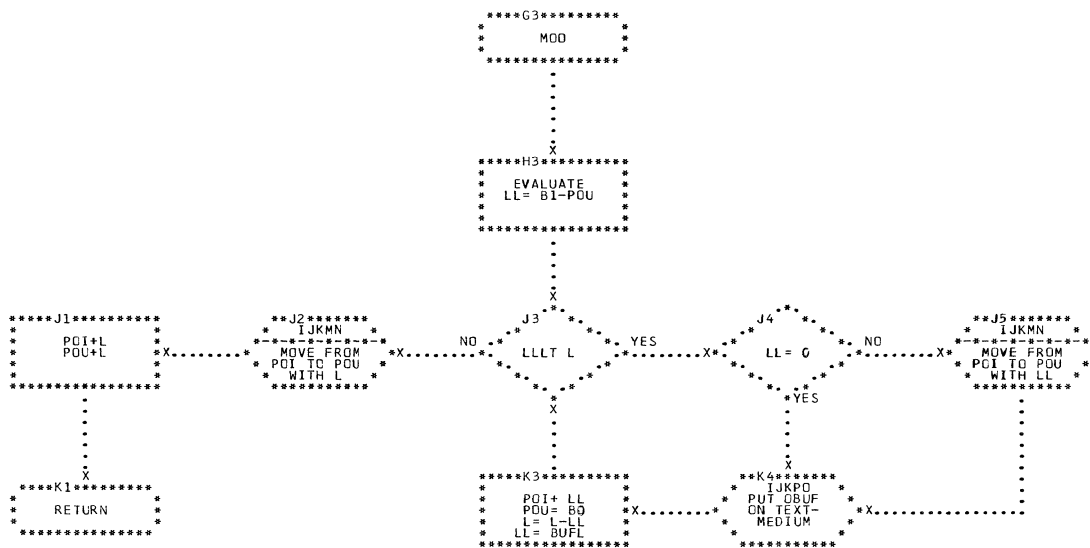
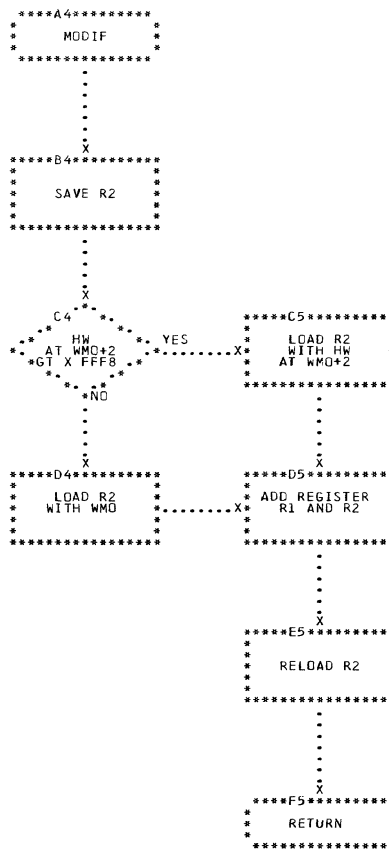
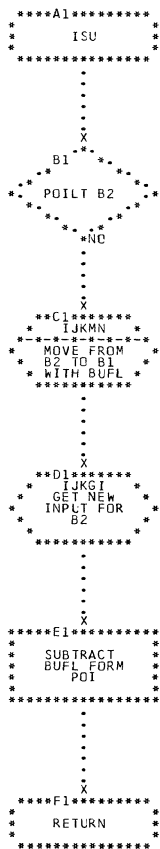
INSERTING OF OFFSETS OF EQUATES

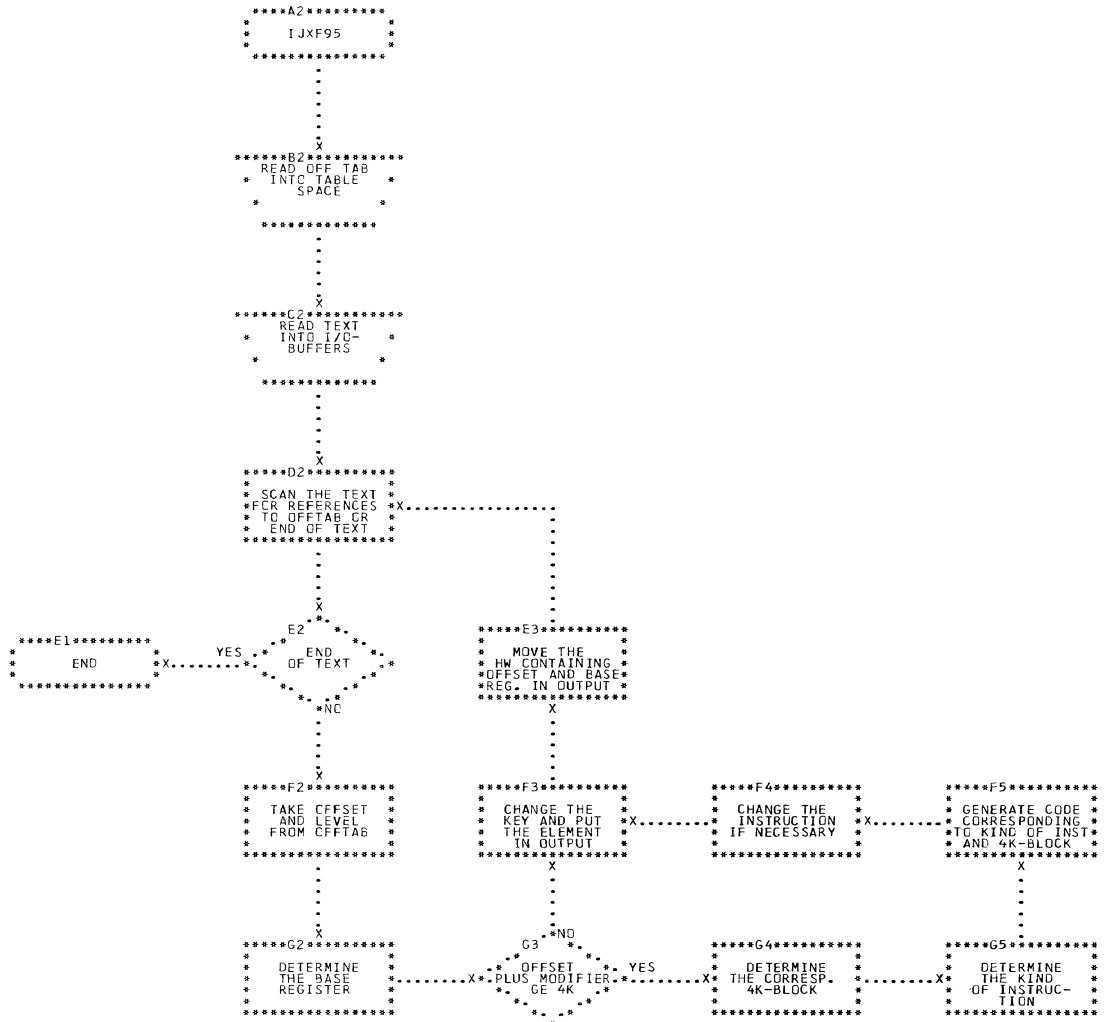


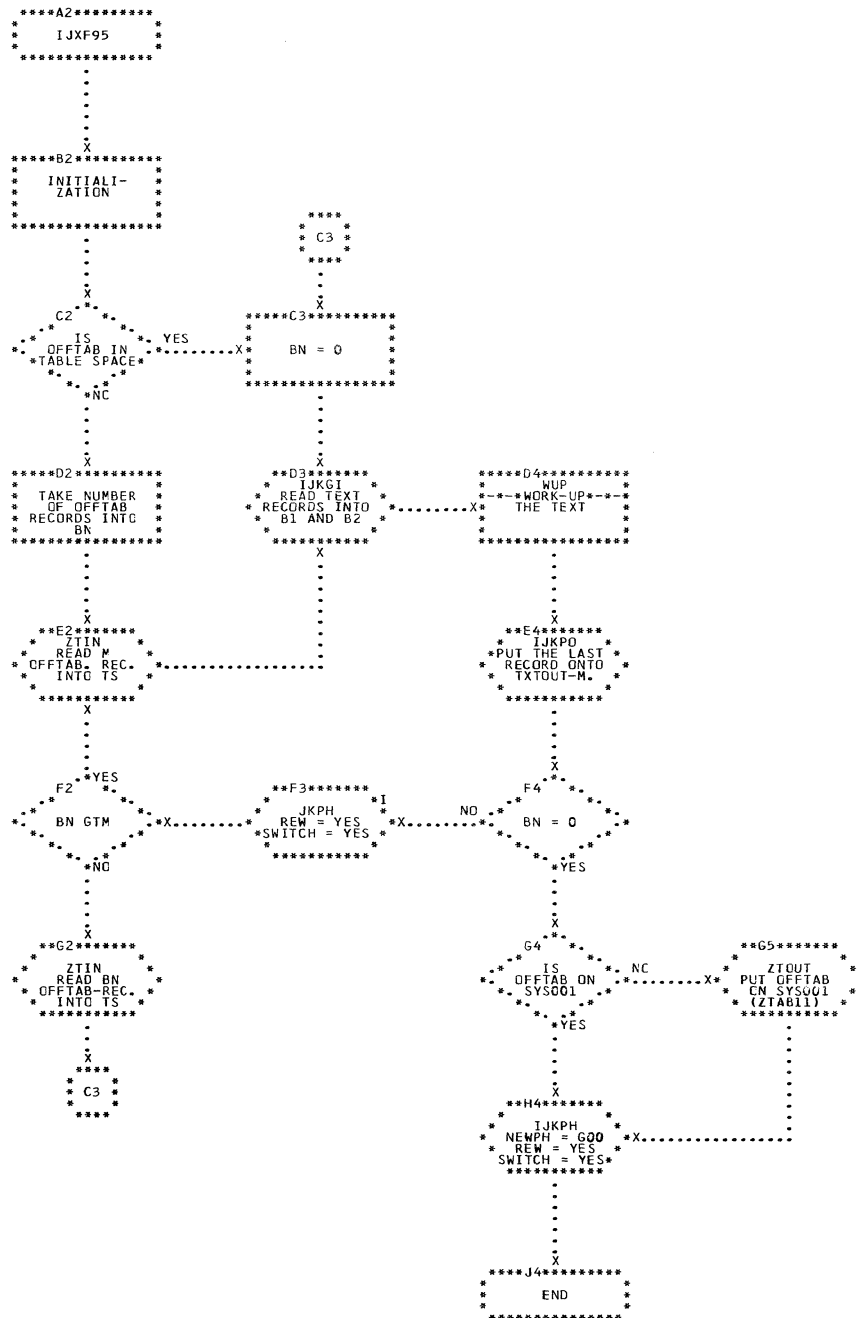


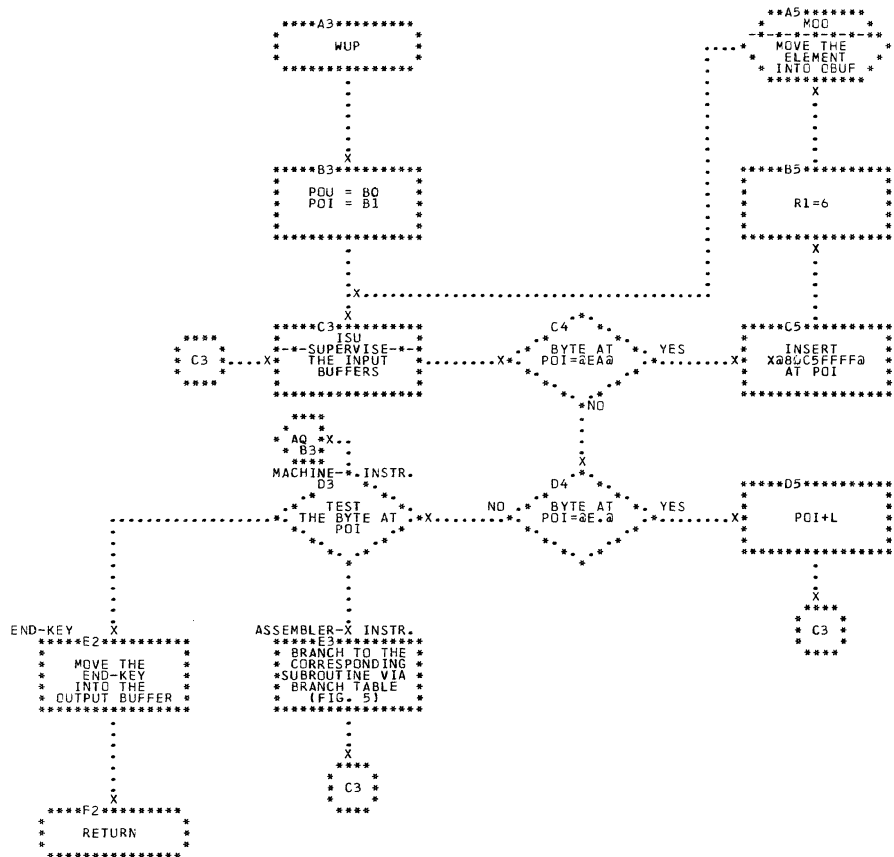


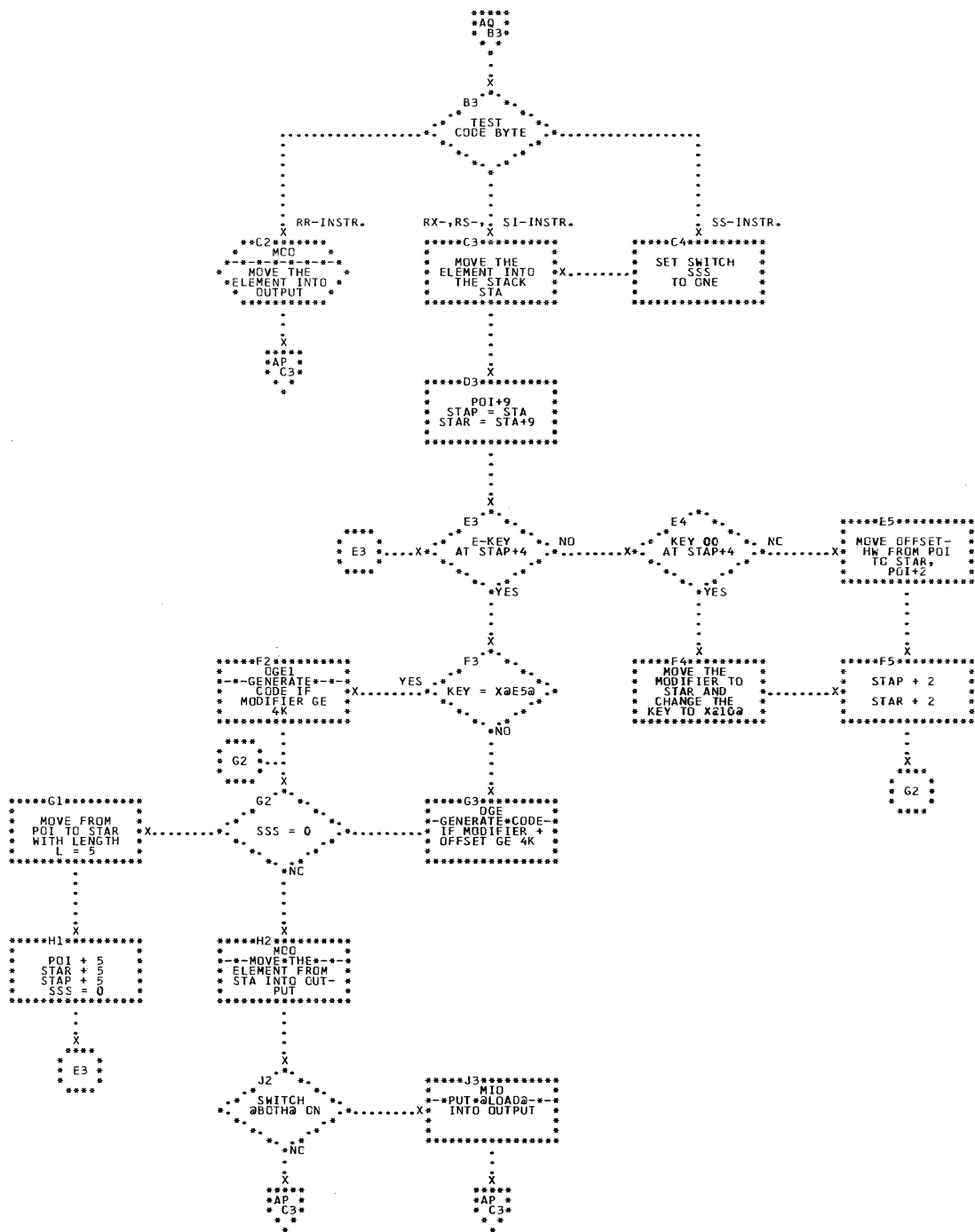


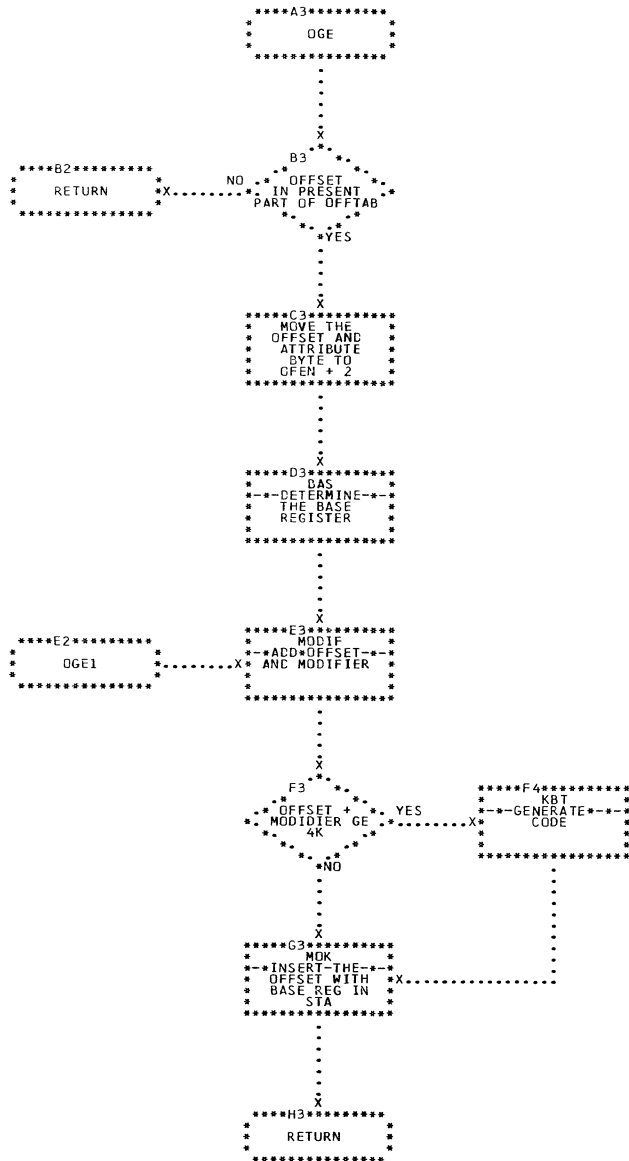


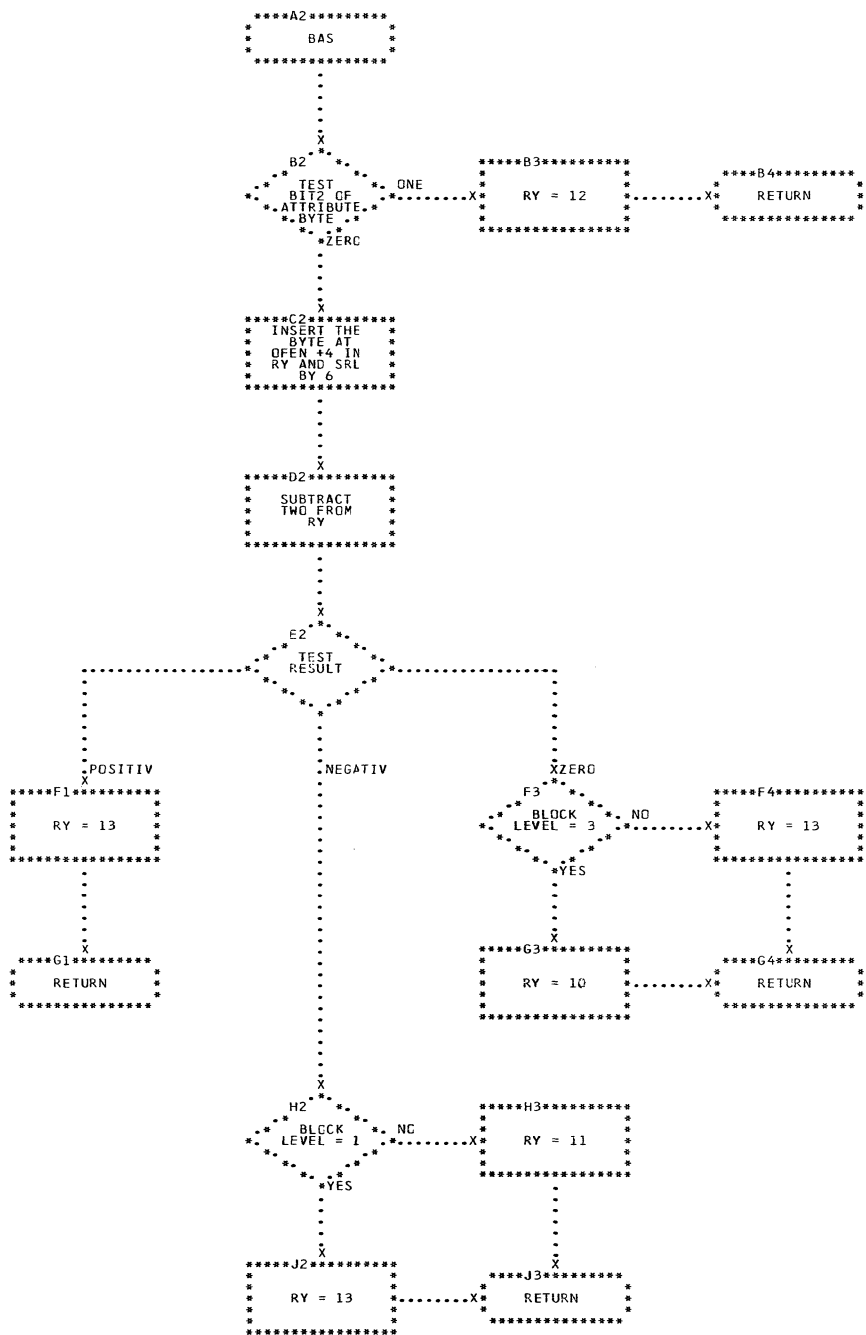












```

*****A1*****
MOK
  
```

```

*****B1*****
INSERT THE
BASE REGISTER
INTO THE LEFT
MOST 4 BITS
OF HW AT STAR
  
```

```

*****C1*****
MODIF
ADD MODIFIER
AND OFFSET
  
```

```

*****D1*****
INSERT THE
SUM INTO
THE RIGHT
MOST 12 BITS
OF HW AT STAR
  
```

```

*****E1*****
RETURN
  
```

```

*****A3*****
KBT
  
```

```

*****B3*****
DETERMINE
THE NUMBER
OF THE 4K-
BLOCK IN R1
  
```

```

*****C3*****
FRR
DETERMINE
THE FREE
REGISTER
(5 OR 6)
  
```

```

D3
R1 LT 3
  
```

```

E3
ANY
REGISTER (5,6)
FREE
  
```

```

F3
R1 LT 8
  
```

```

*****E2*****
SET SWITCH
BOTH, CN,
RO = STCRE
  
```

```

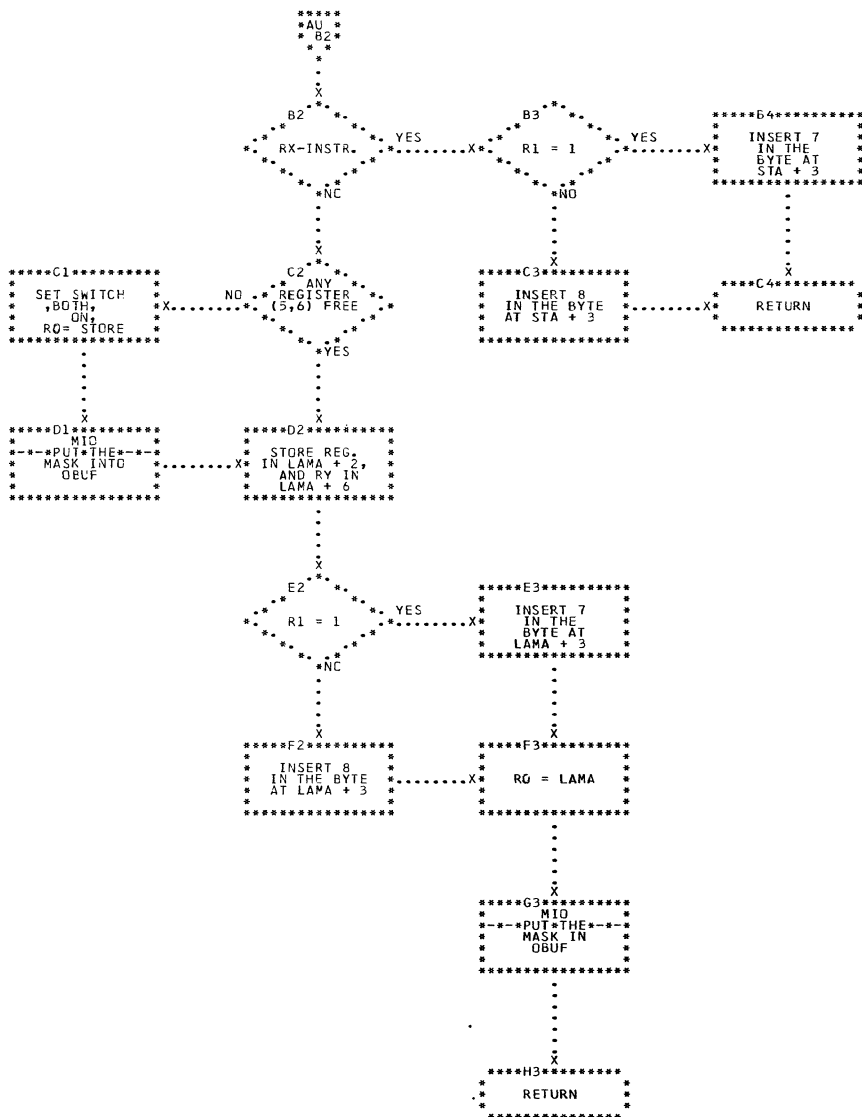
*****F2*****
MIO
PUT, STORE,
INSTRUCTION
INTC OUTPUT
  
```

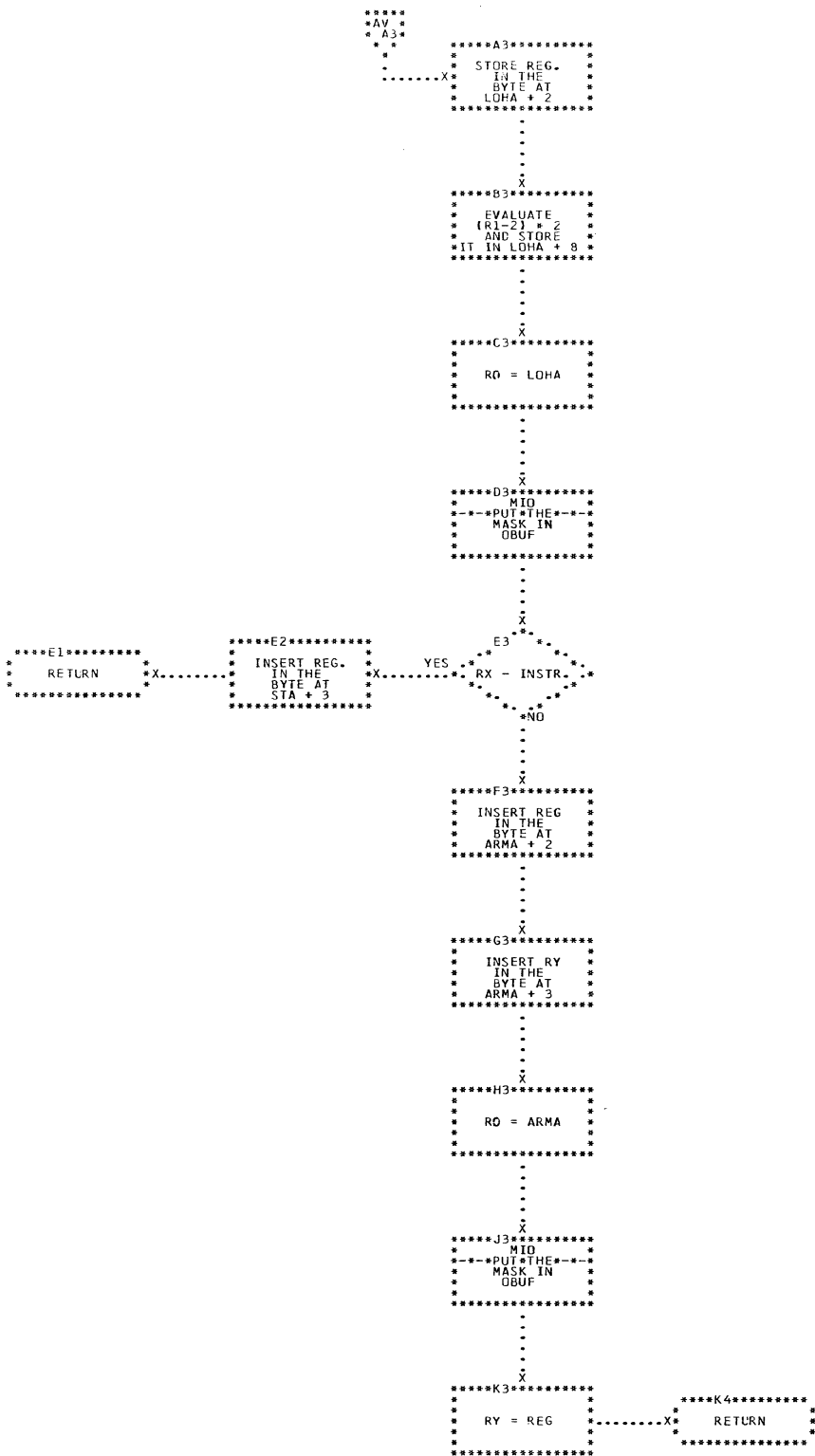
```

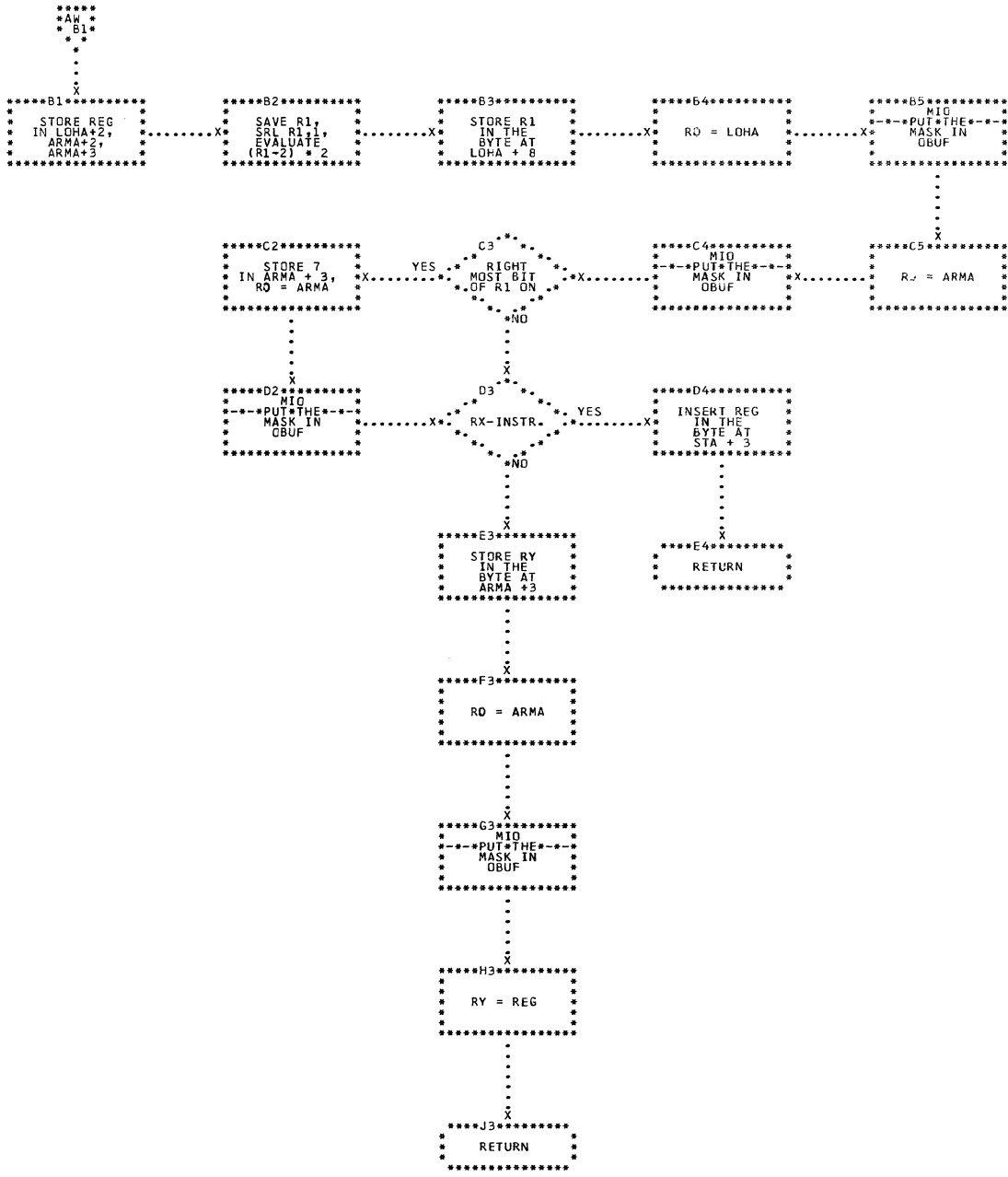
*****
AU
B2
  
```

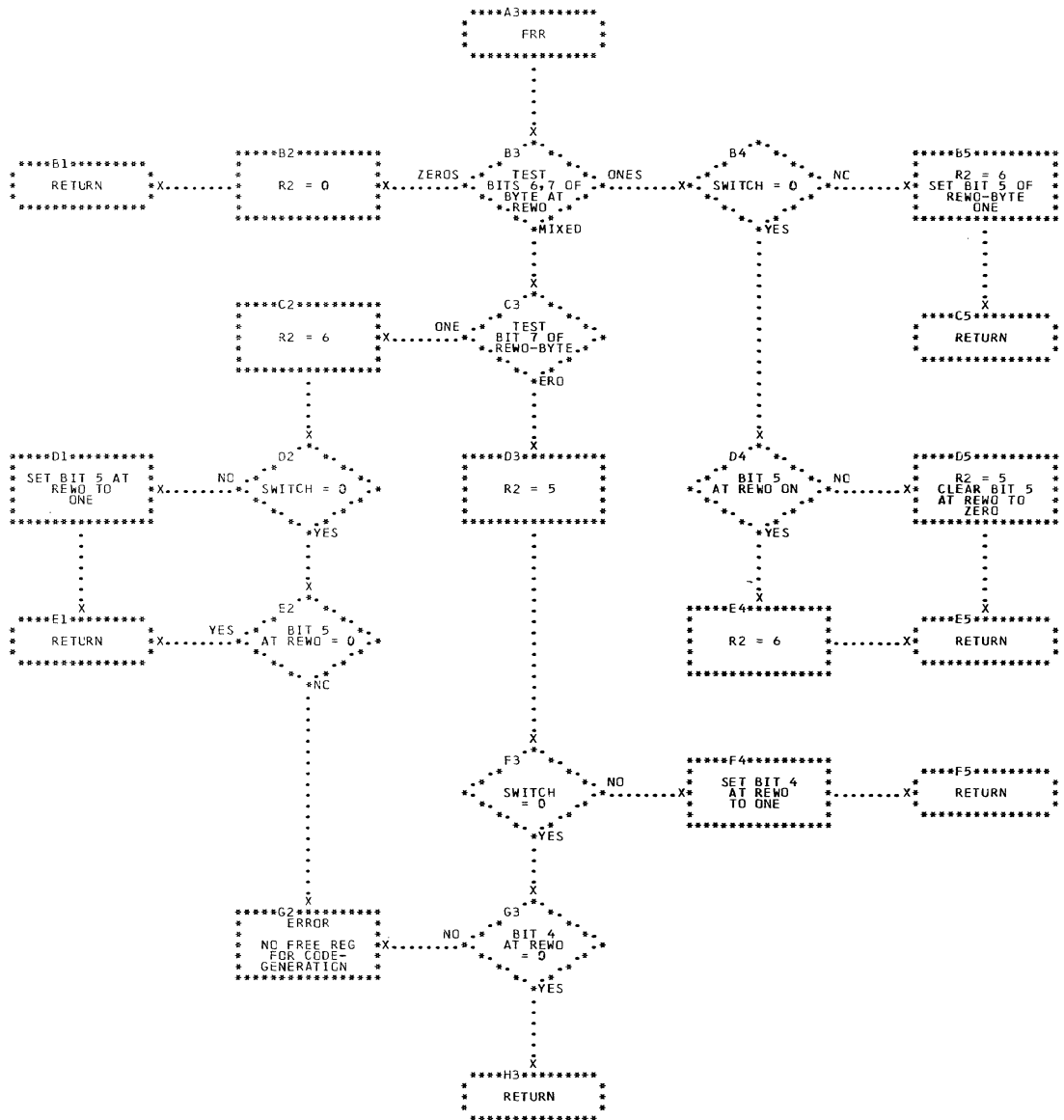
```

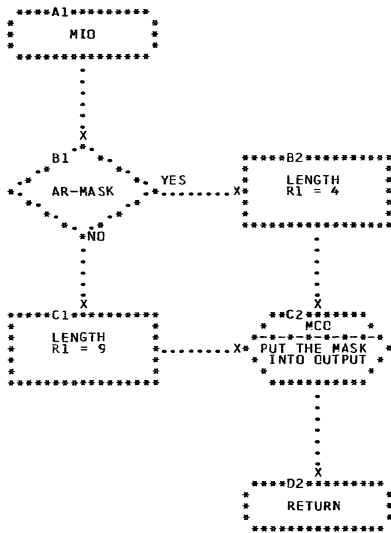
*****
AV
A3
  
```











*****A1*****
MULTI

.....
X
.....

*****B1*****
R1= 4

.....
X
.....

*****C1*****
MOD
TO MOVE THE
ELEMENT TO
DBUF

.....
X
.....

*****D1*****
RETURN

*****B2*****
DCAL3

.....
X
.....

*****C2*****
R1= 7

.....
X
.....

*****A4*****
DCX

.....
X
.....

*****B4*****
LOAD HW
AT P01+2
IN R1

.....
X
.....

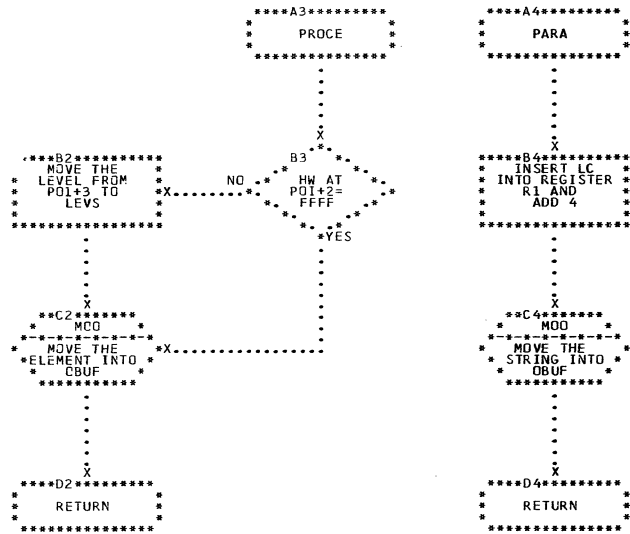
*****C4*****
ADD 4 ON
R1

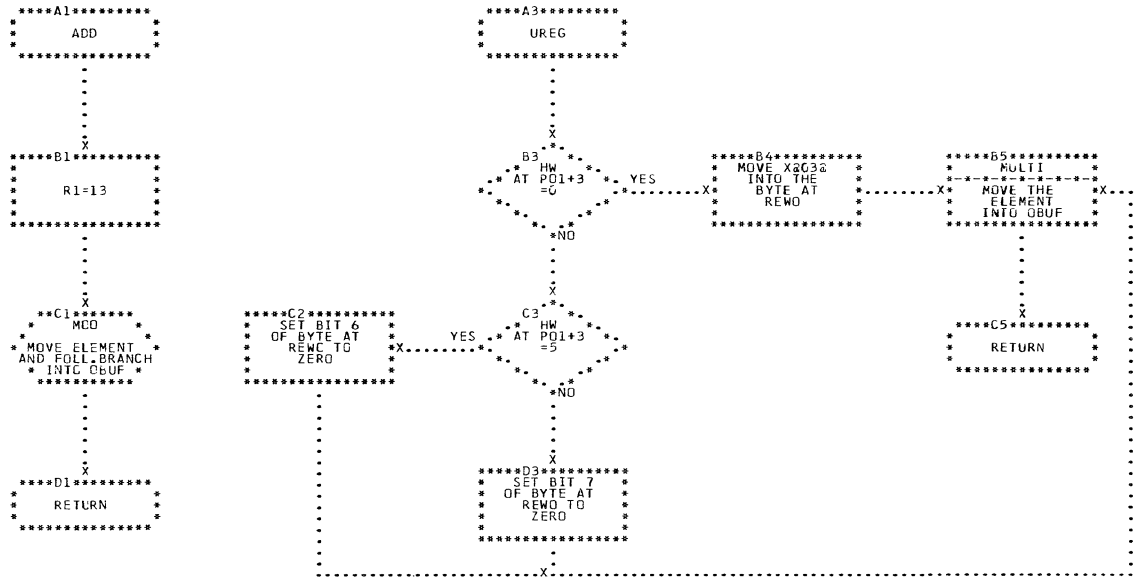
.....
X
.....

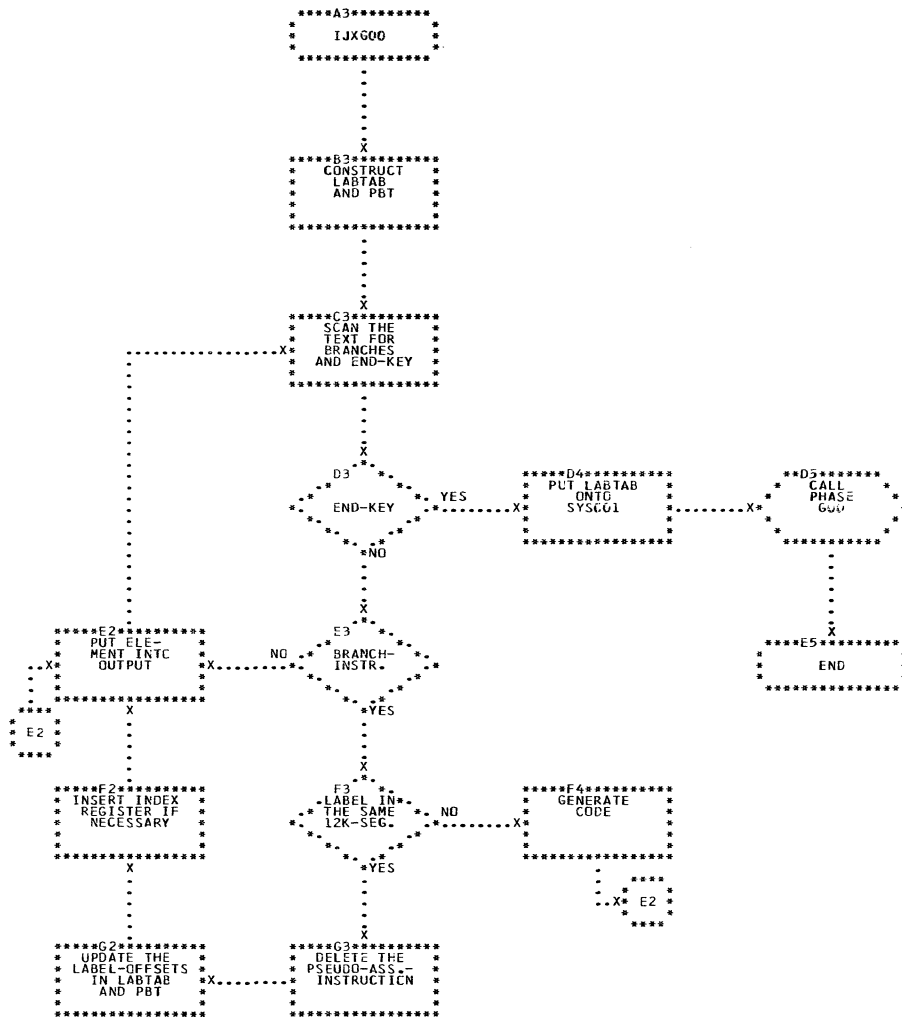
*****D4*****
MOD
TO MOVE THE
ELEMENT TO
DBUF

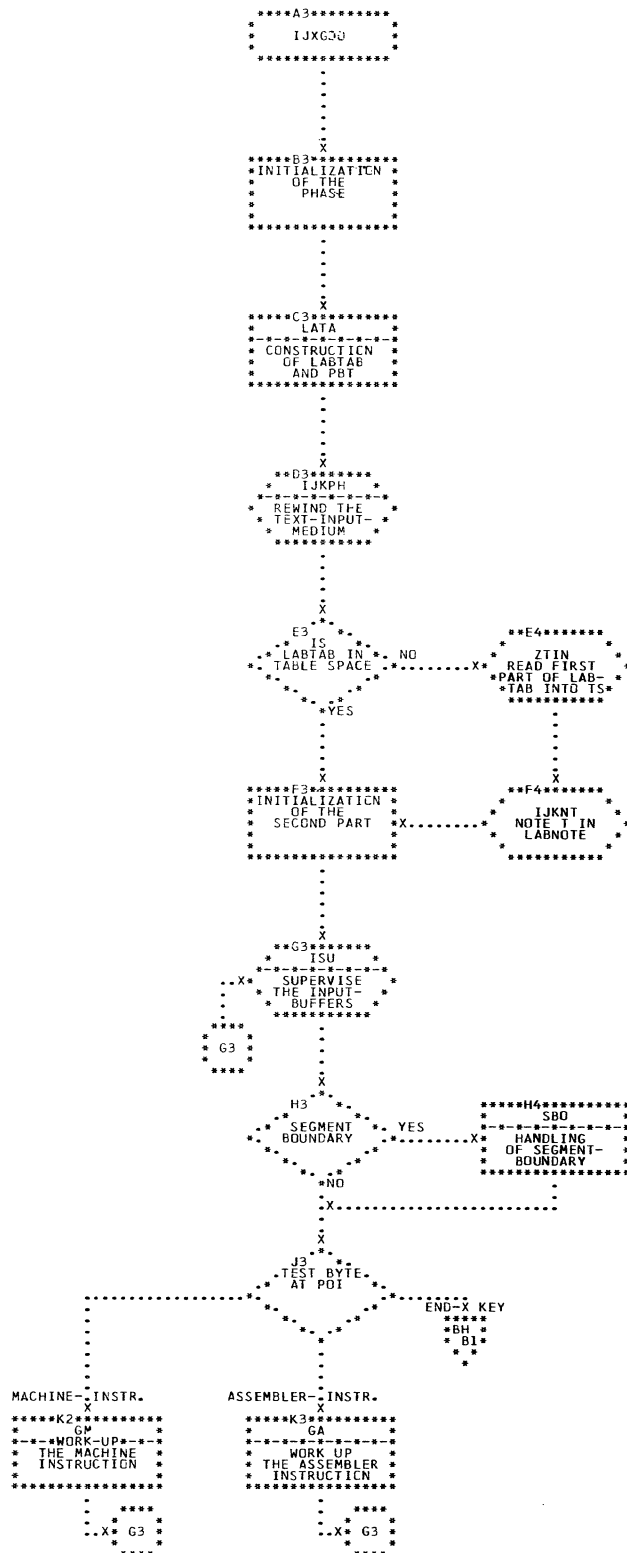
.....
X
.....

*****E4*****
RETURN









```

*****
* BH *
* B1 *
*   *
*   *
* X *
*****B1*****
* SET LOC1 ON *
* DW-BOUNDARY *
* BY GENERATING *
* A CORR. DS *
*****
*   *
*   *
* X *
*****C1*****
* INSERT THE *
* LENGTH OF *
* OBJECT PROGRAM *
* INTO ZTABC5 *
*****
*   *
*   *
* X *
*****D1*****
* PUT THE *
* END-KEY *
* INTO DBUF *
*****
*   *
*   *
* X *
*****E1*****
* IJKPO *
* PUT LAST *
* TXT-RECORD *
* ONTO TXTOUT *
*****
*   *
*   *
* X *
*****F1*****
* IJKPH *
* CALL PHASE G01 *
* REWIND ALL *
* SWITCH *
*****
*   *
*   *
* X *
*****G1*****
* END *
*****

```

```

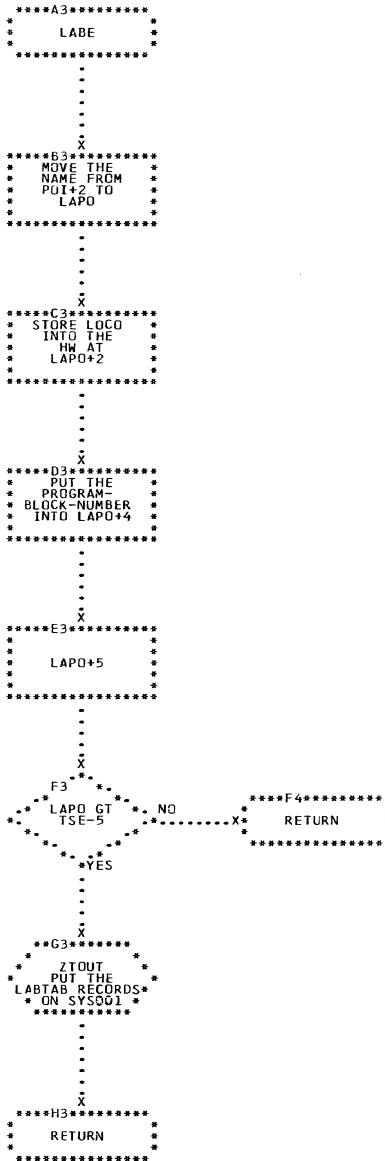
*****A3*****
* GM *
*   *
*   *
* X *
*****B3*****
* DETERMINE *
* THE FORMAT *
* AND THE *
* LENGTH *
*****
*   *
*   *
* X *
*****C3*****
* ADJUST THE *
* LOCATION COUNT *
* CORRESPONDINGLY *
* (LOC0,LOC1) *
*****
*   *
*   *
* X *
*****D3*****
* MOO *
* MOVE THE *
* ELEMENT *
* INTO DBUF *
*****
*   *
*   *
* X *
*****E3*****
* RETURN *
*****

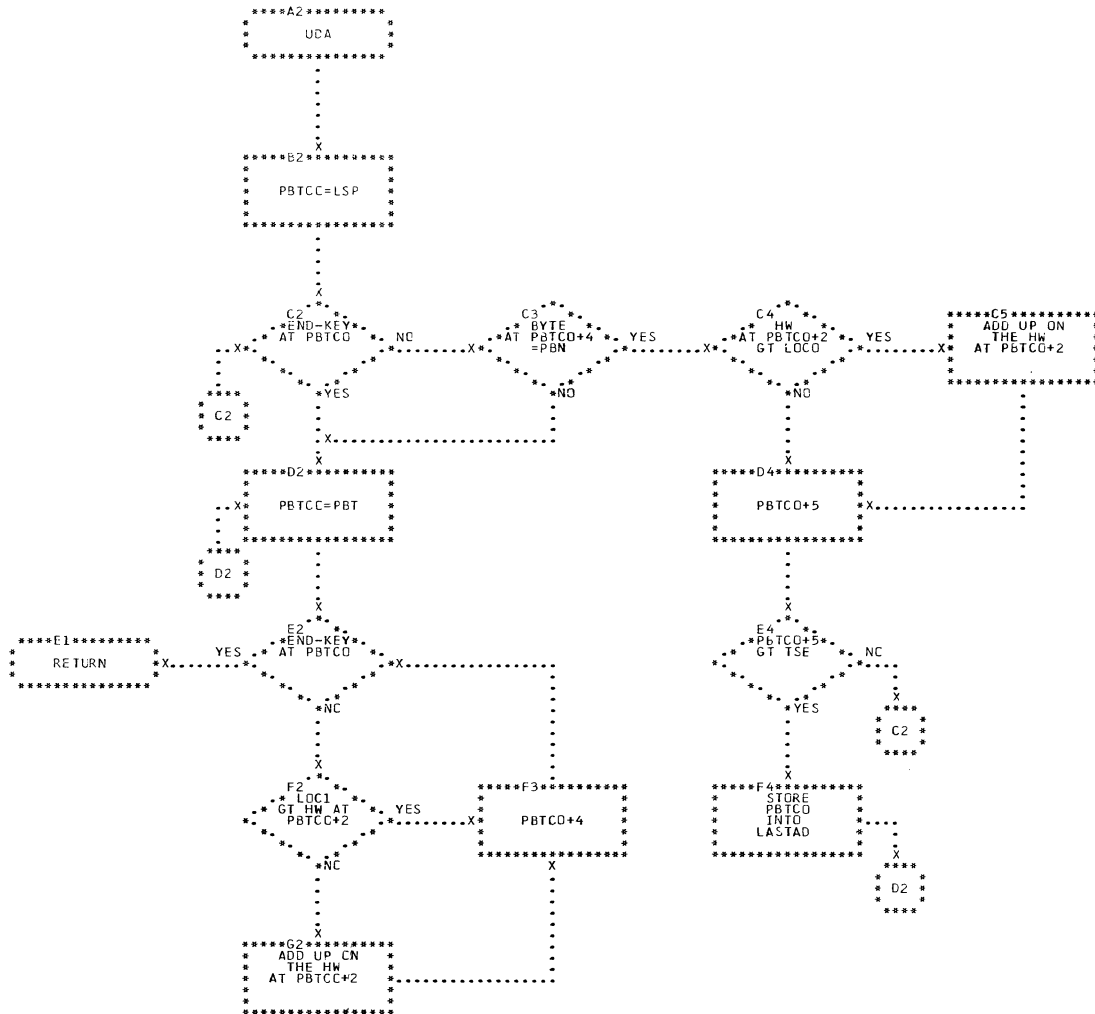
```

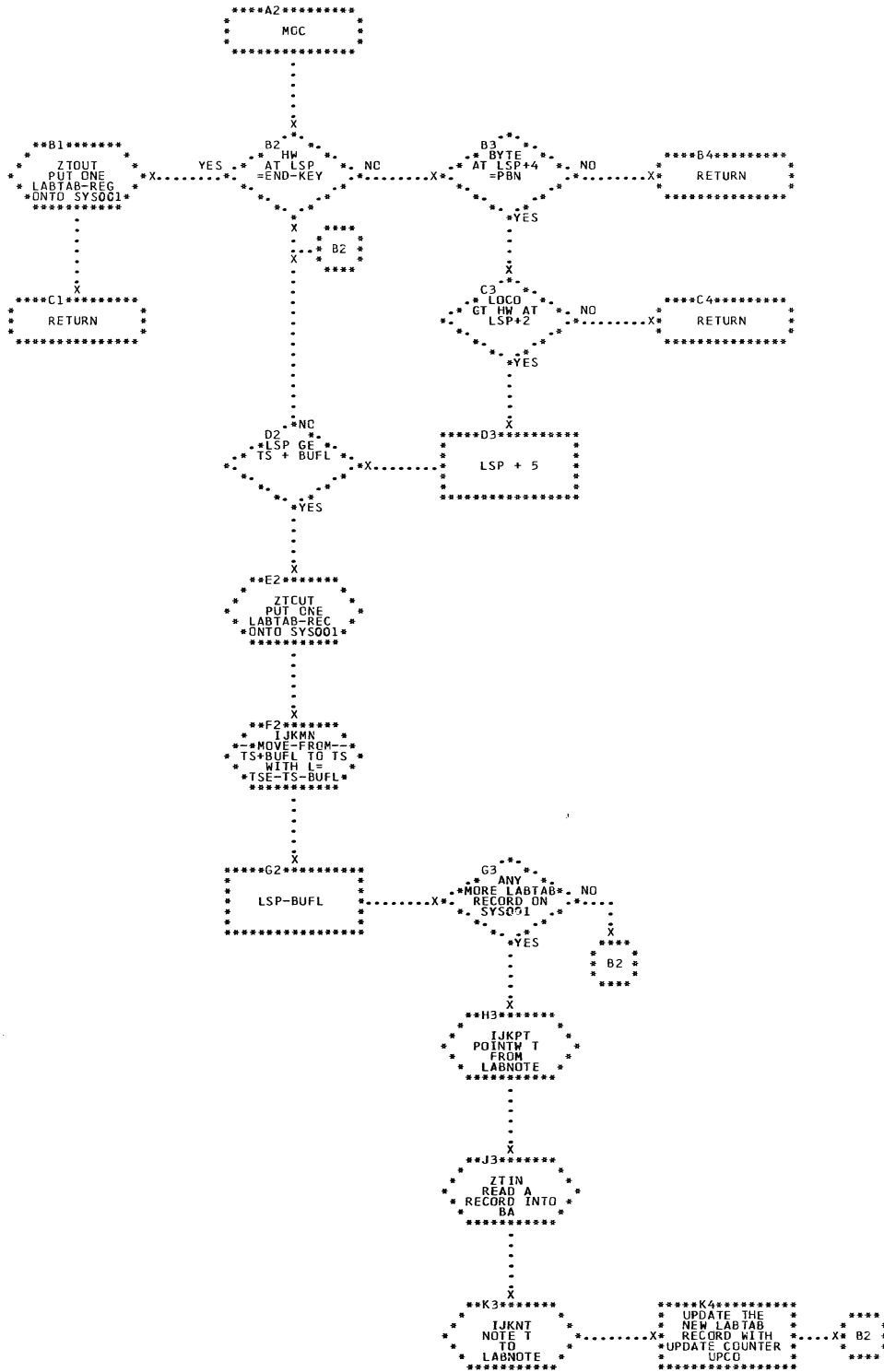
```

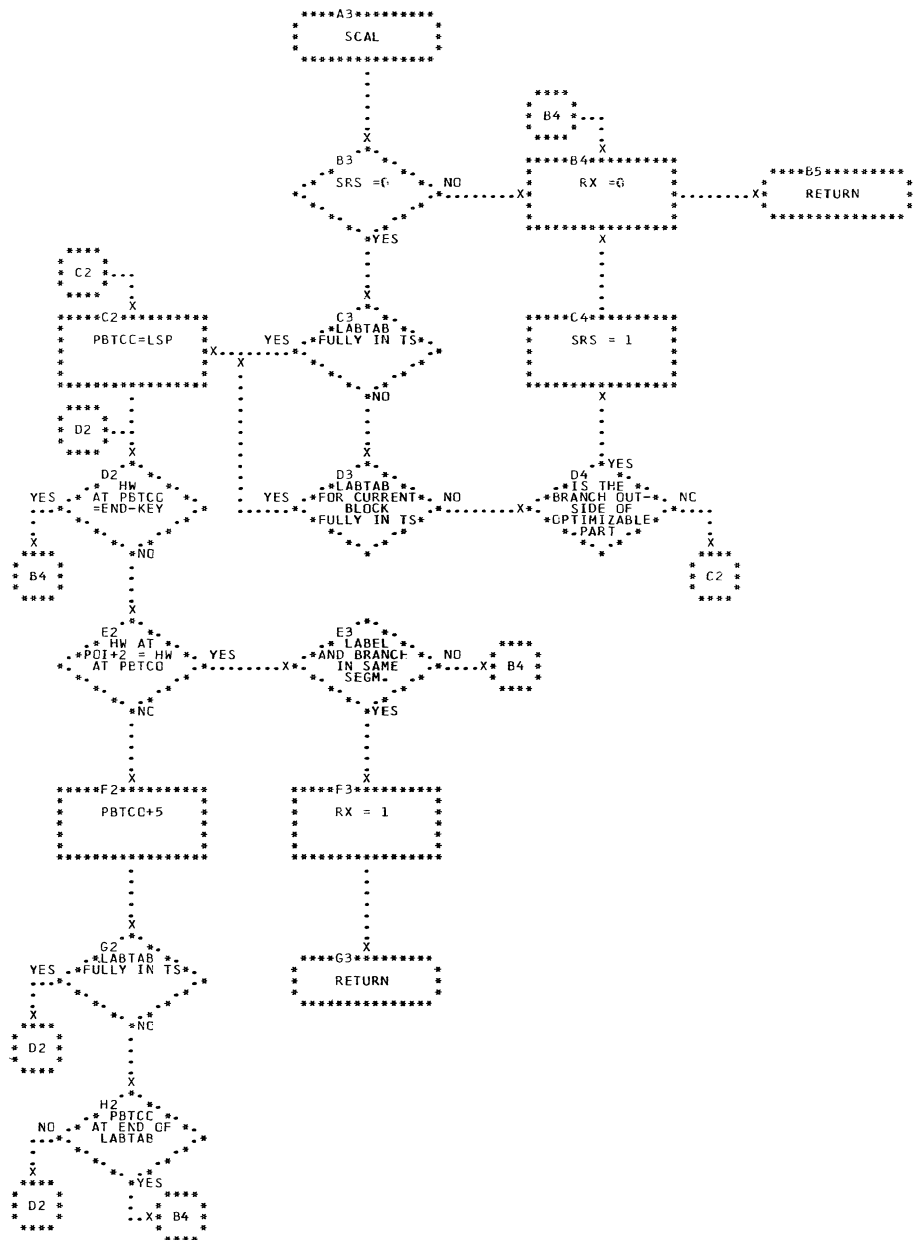
*****A5*****
* GA *
*   *
*   *
* X *
*****B5*****
* BRANCH TO THE *
* CORRESPONDING *
* SUBROUTINE BY *
* MEANS OF THE *
* BRANCH TABLE *
*****
*   *
*   *
* X *
*****C5*****
* RETURN *
*****

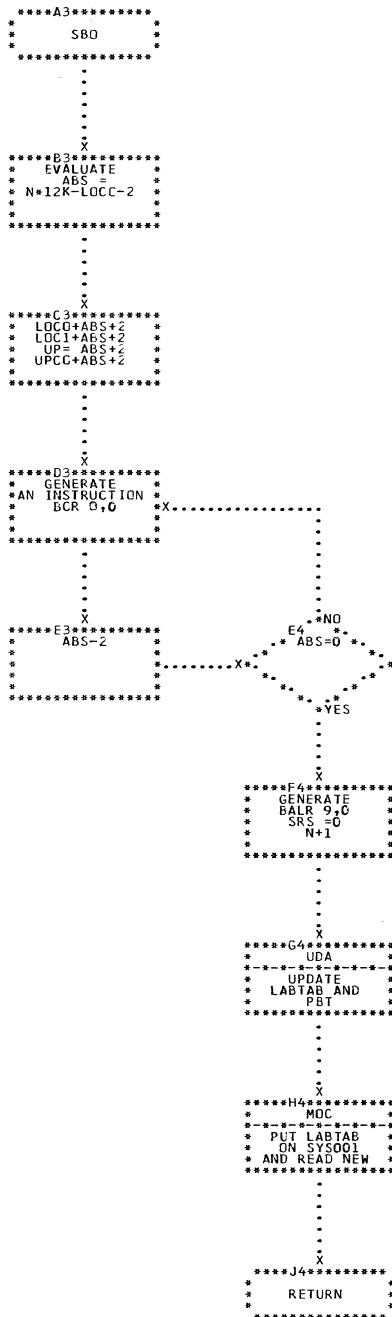
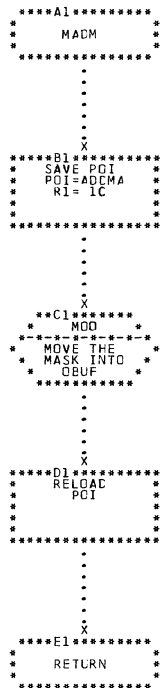
```

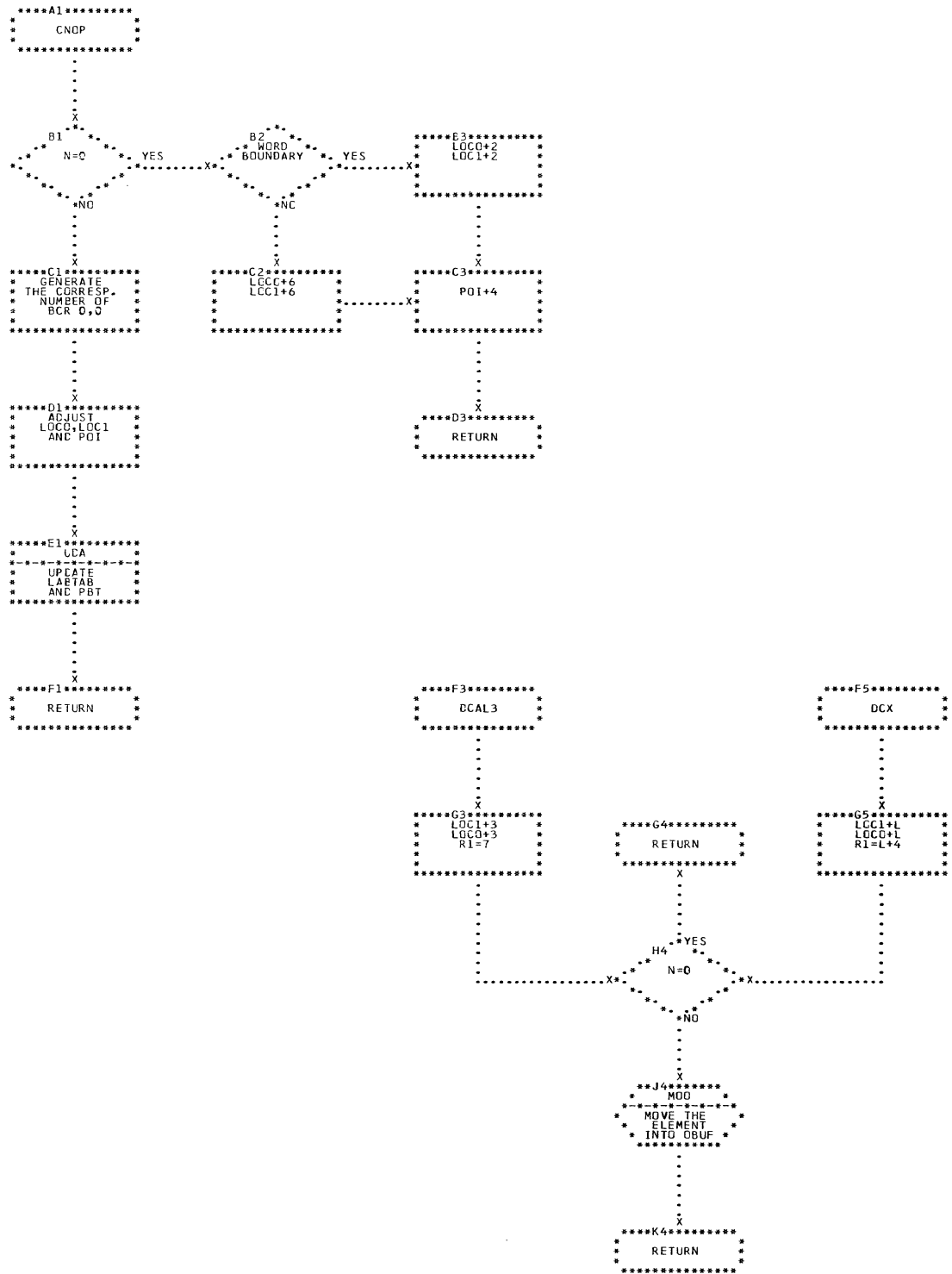



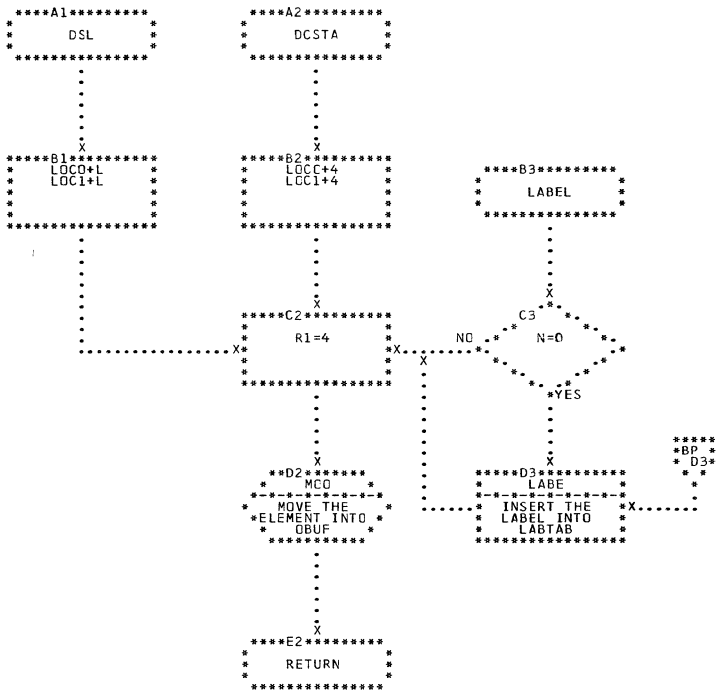


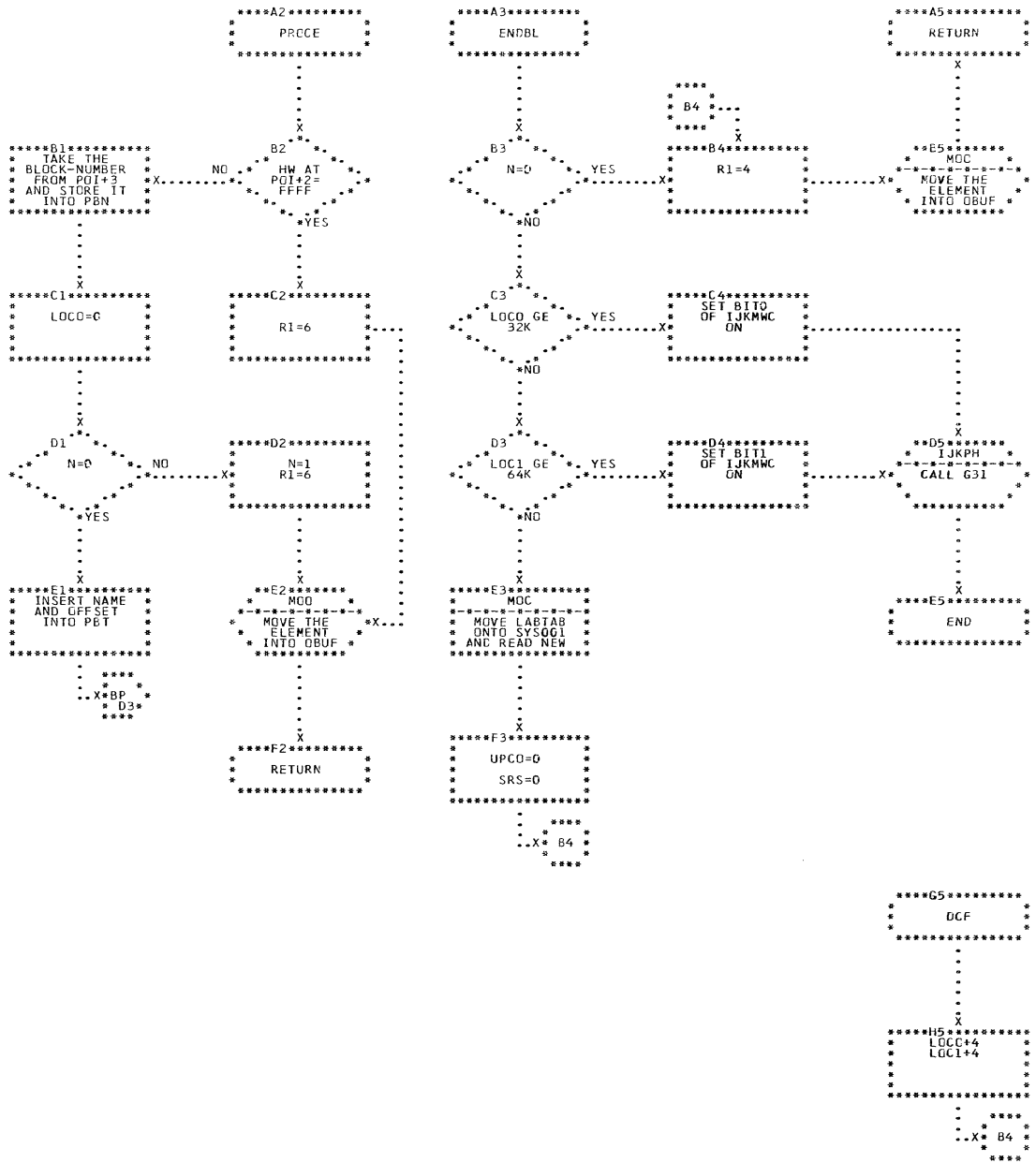


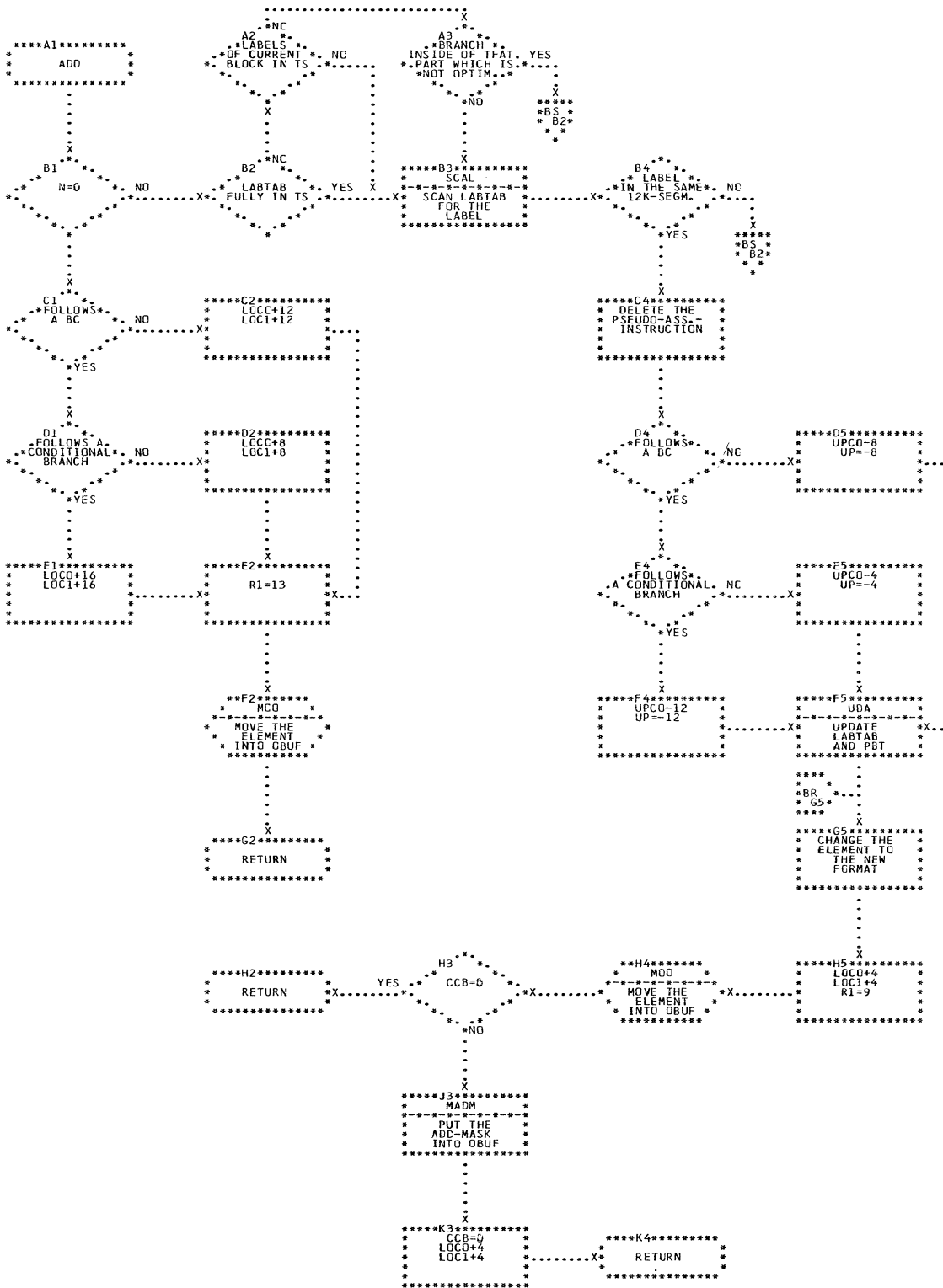


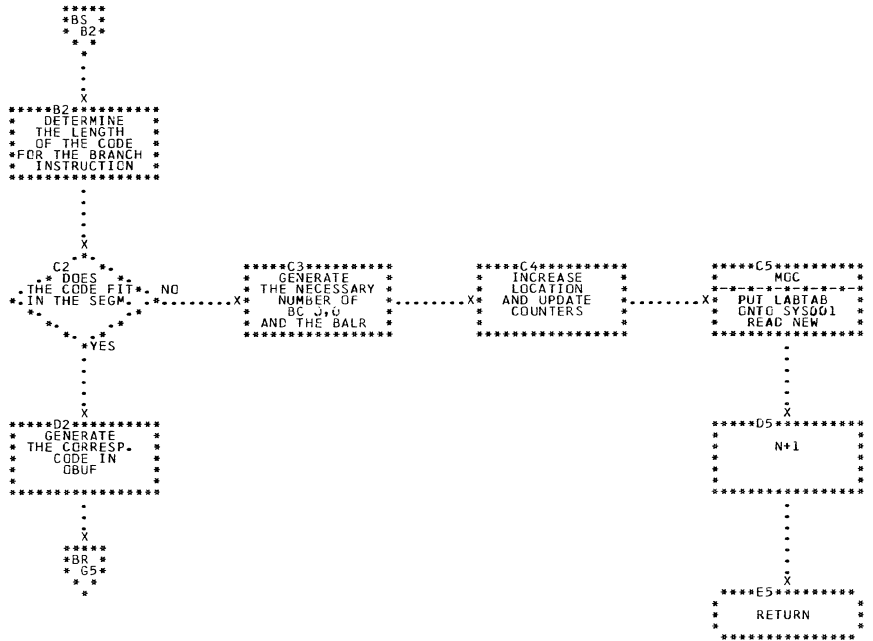


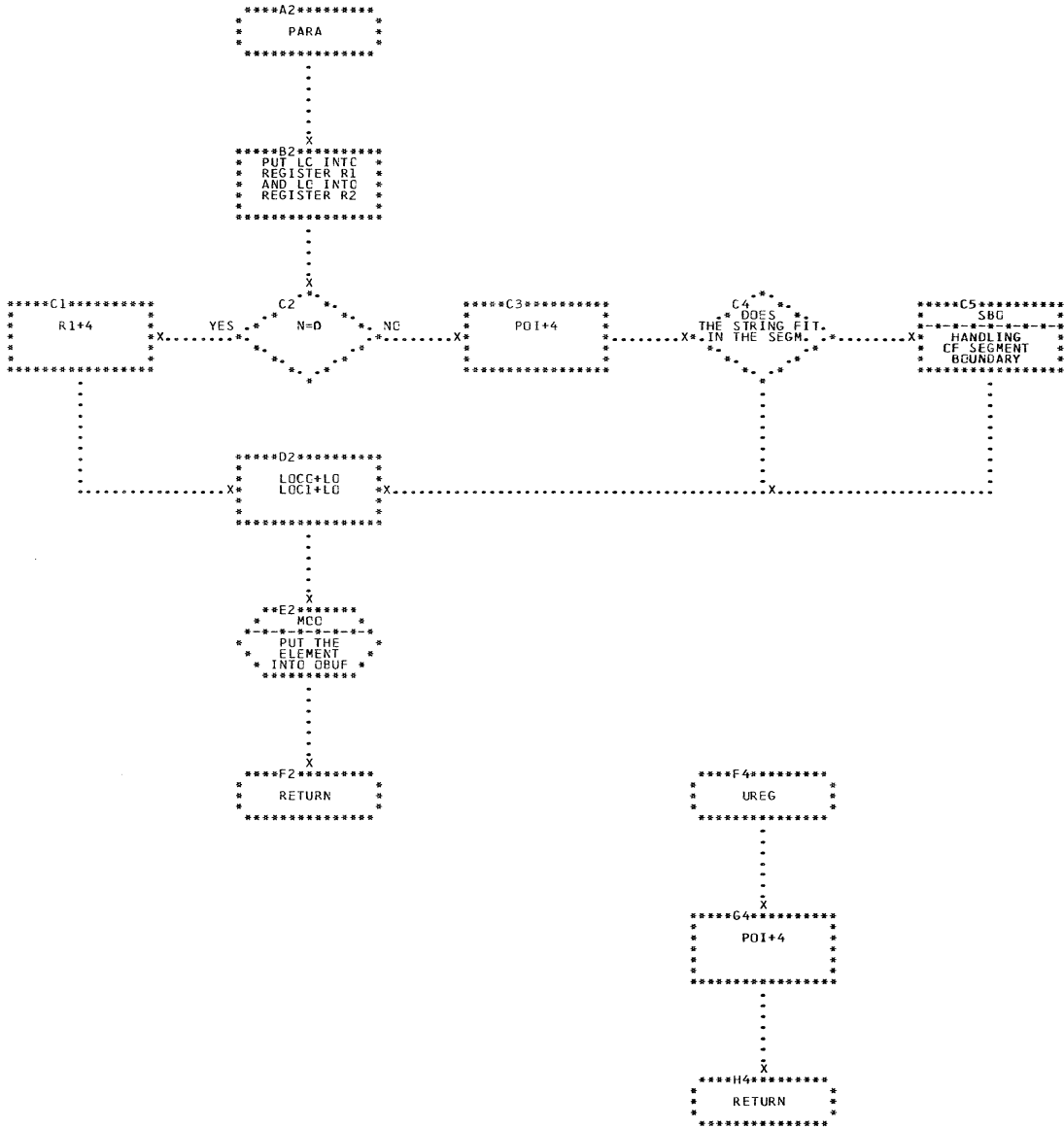


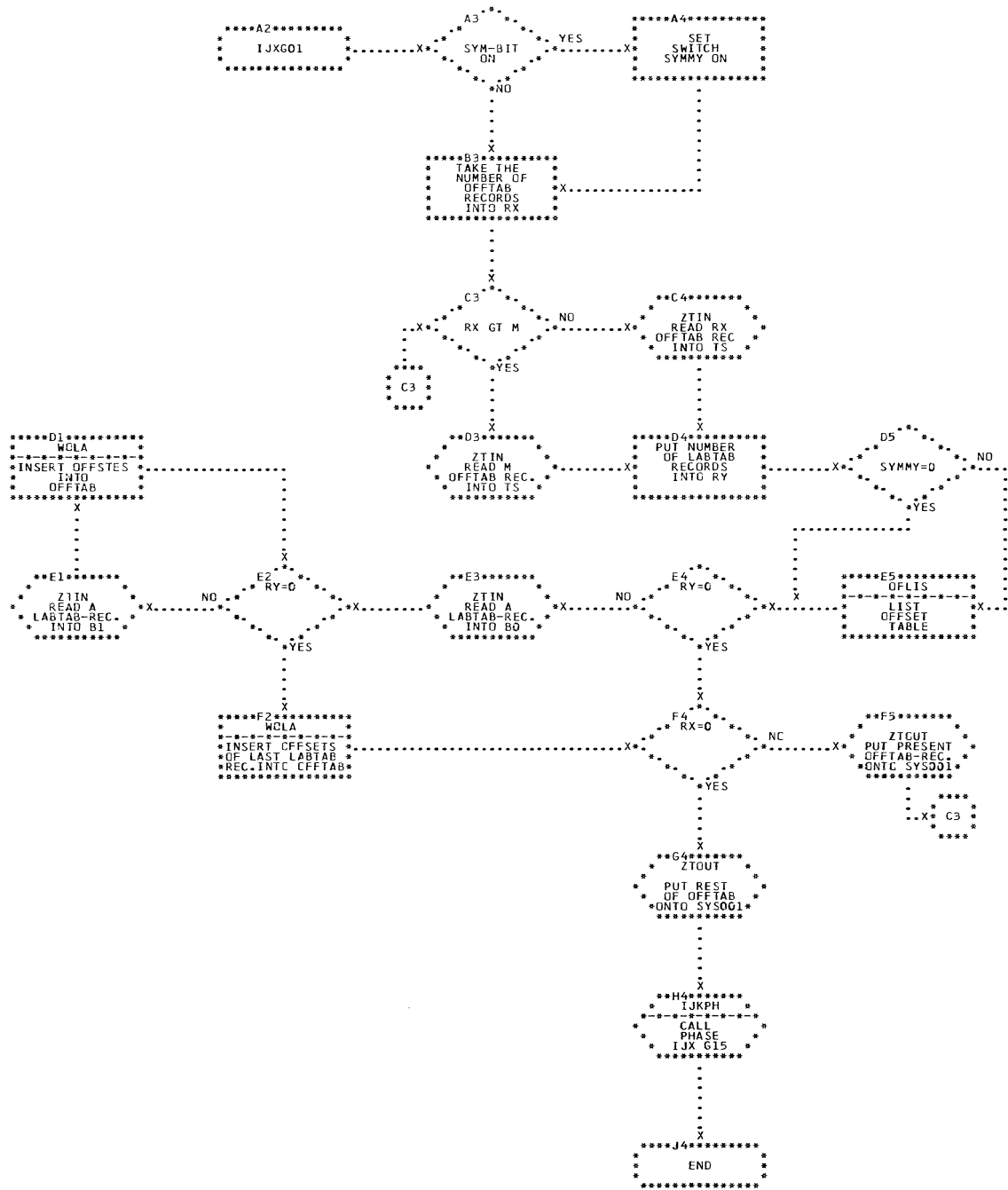


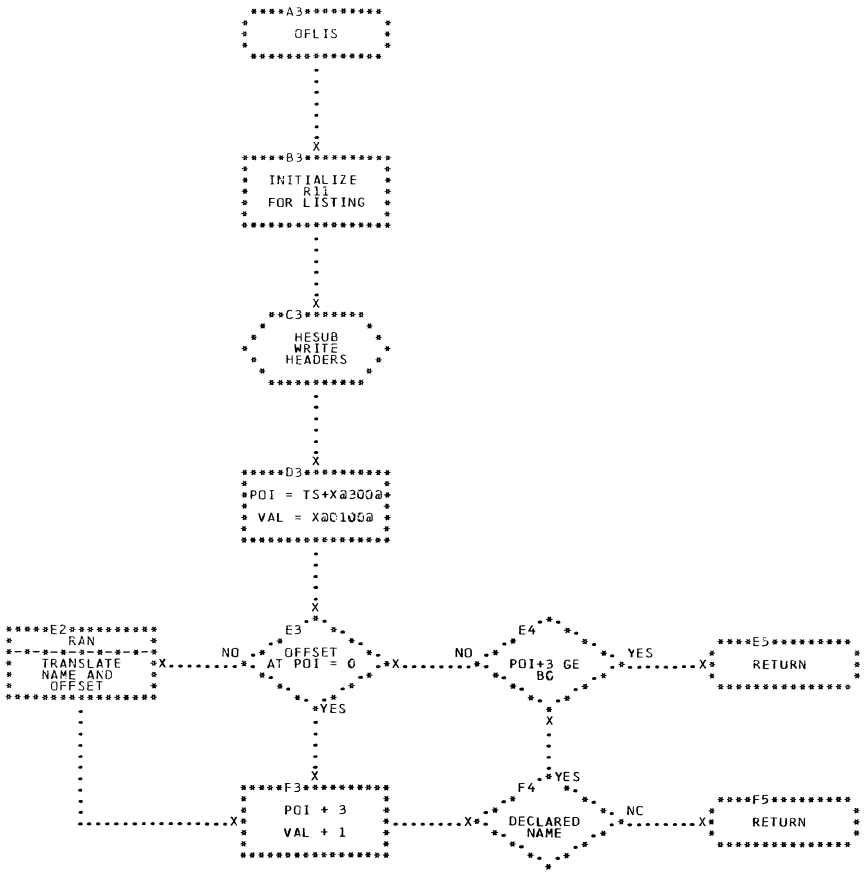


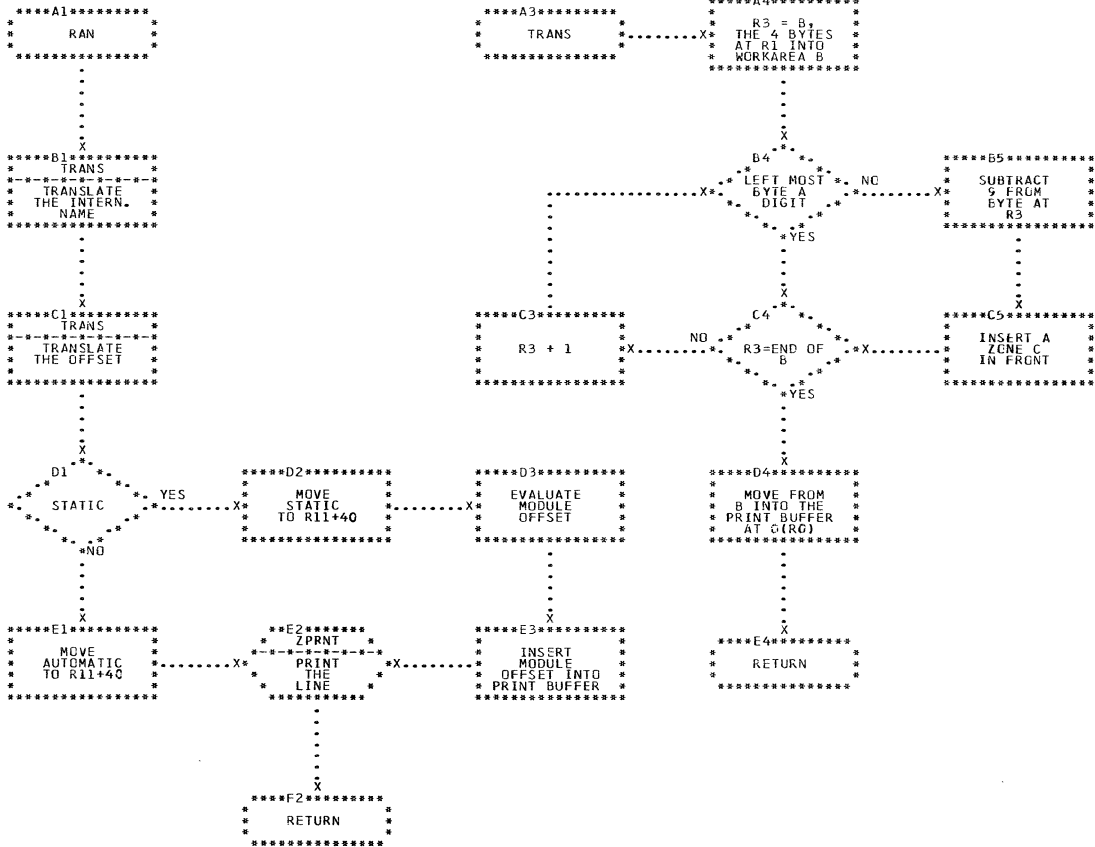


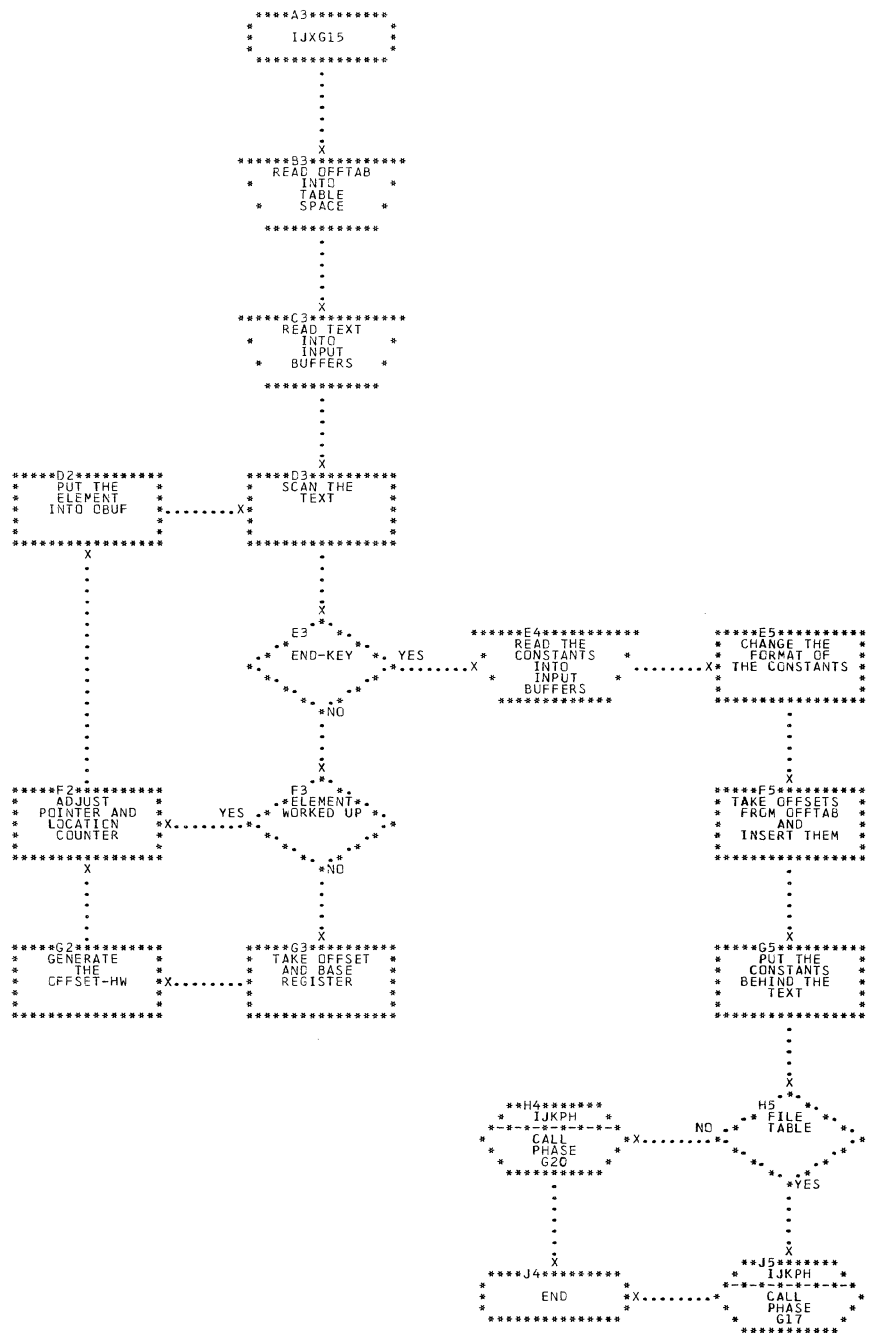


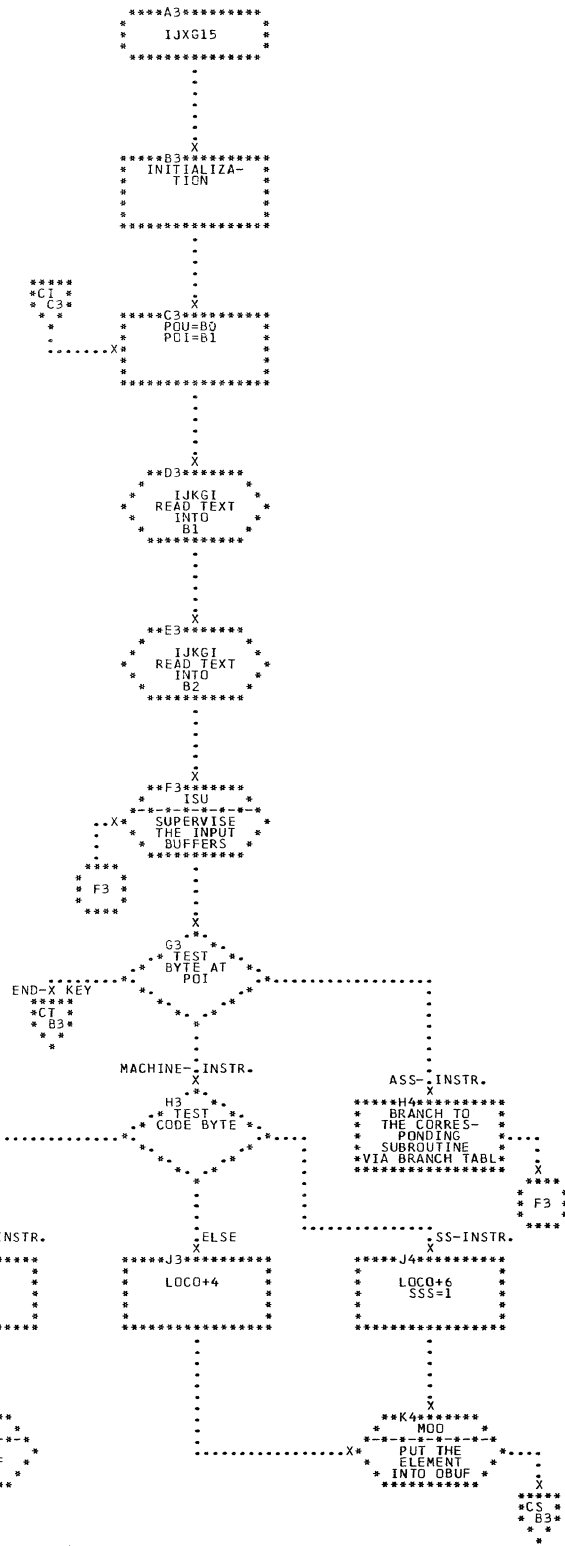


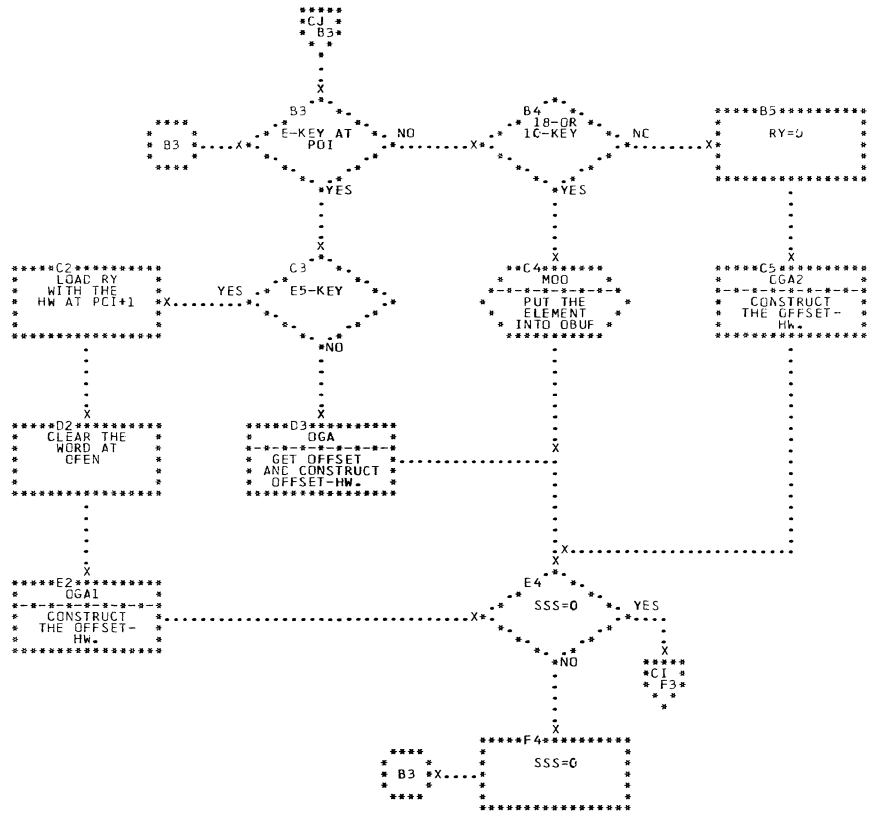


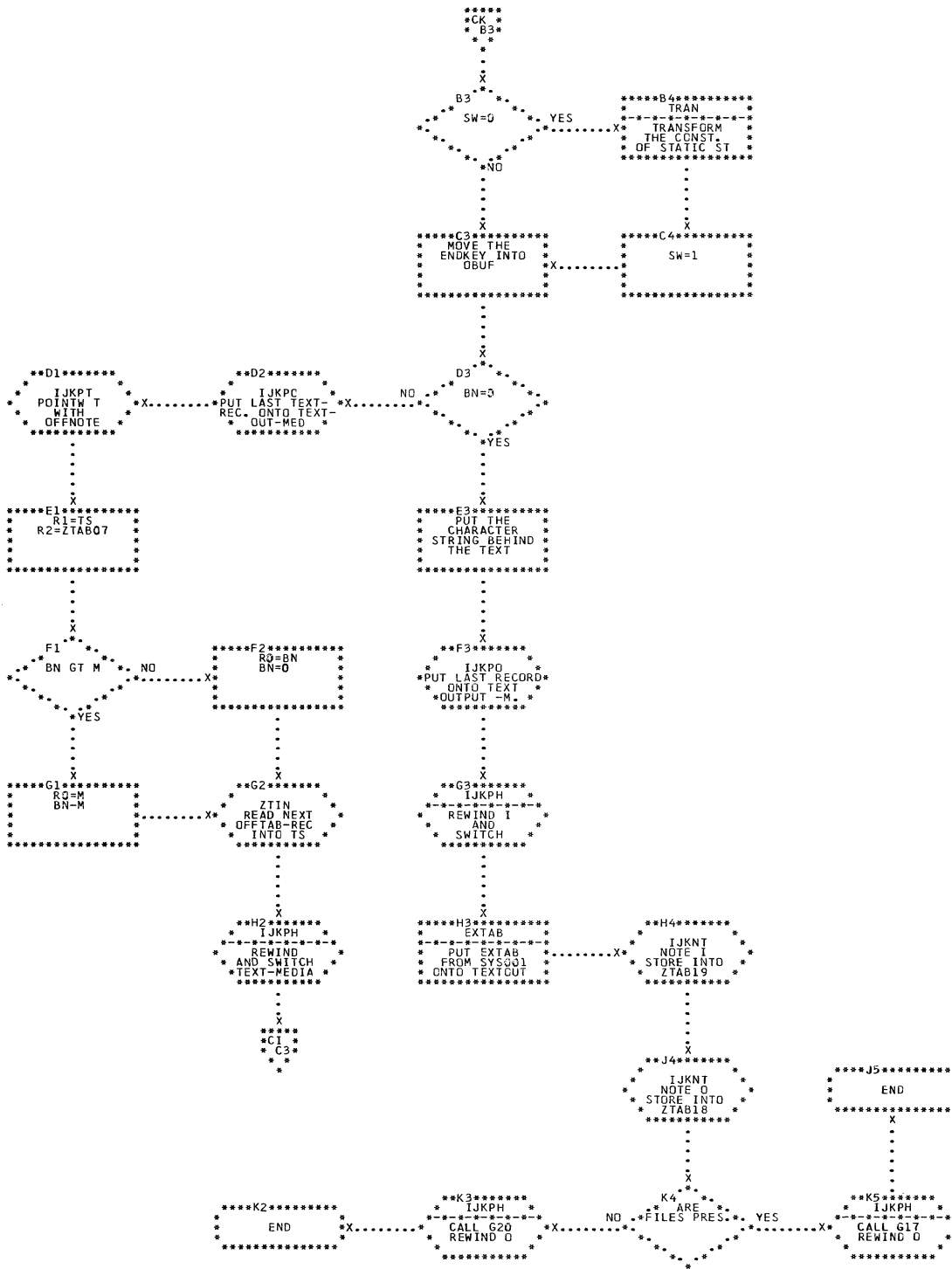


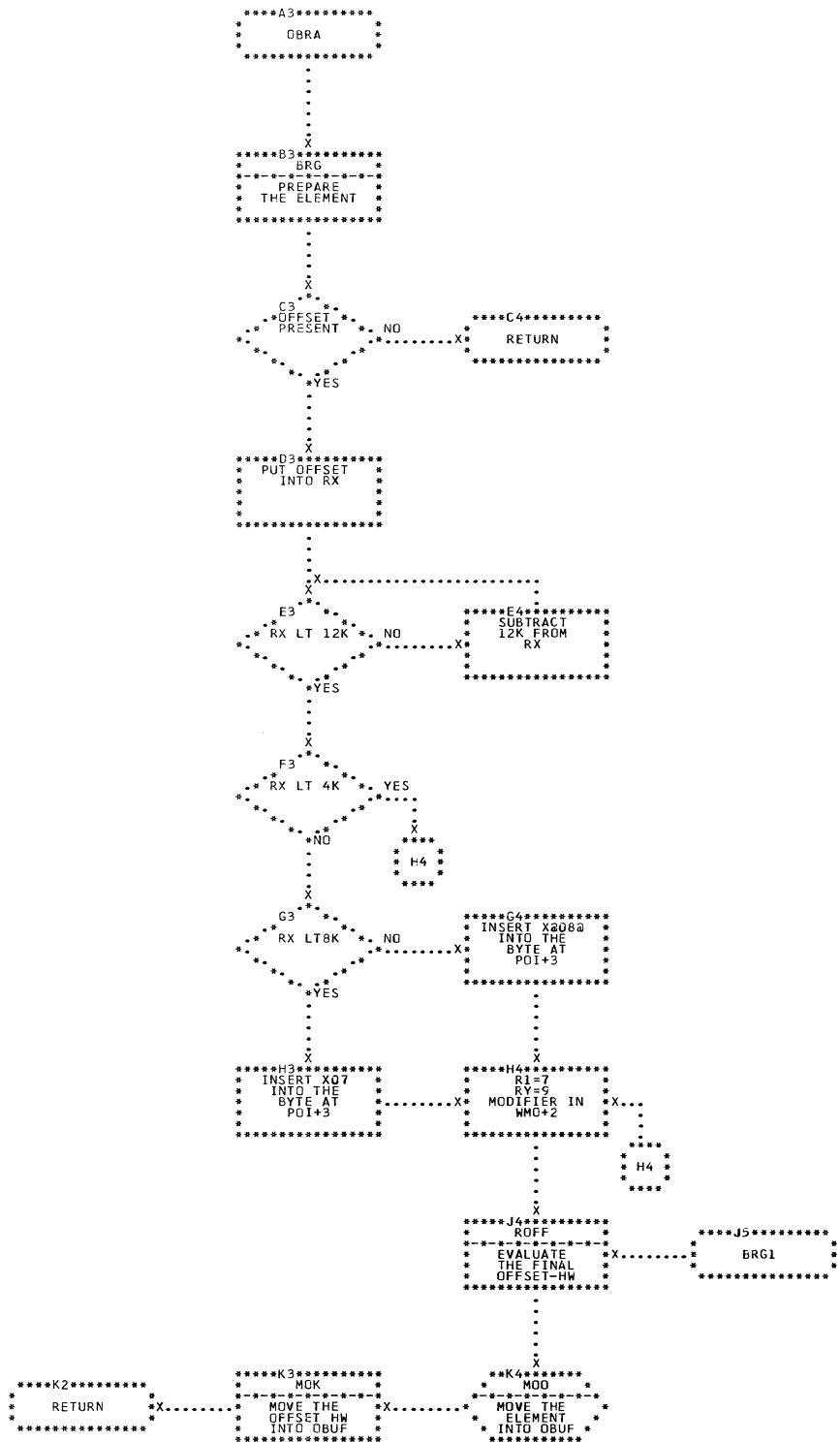


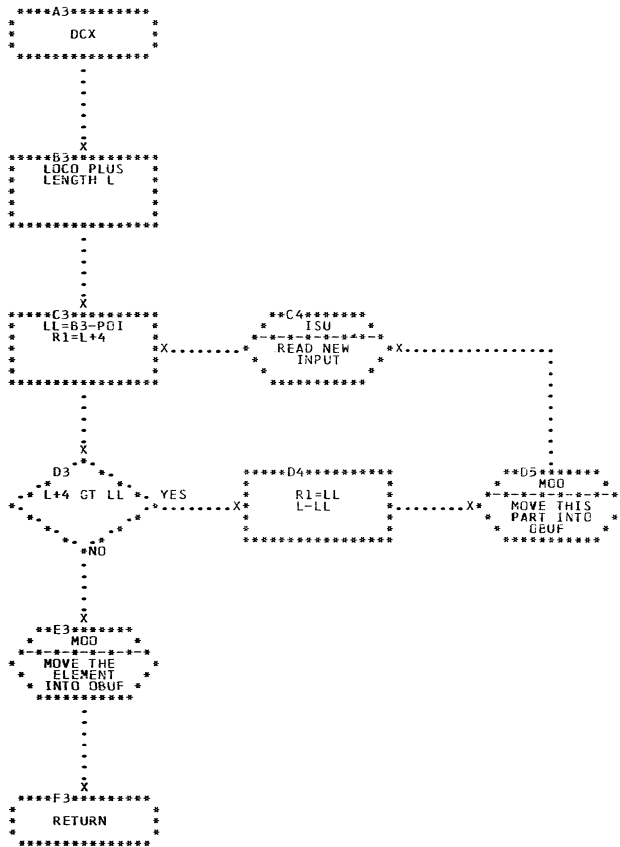
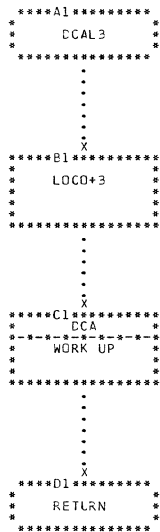


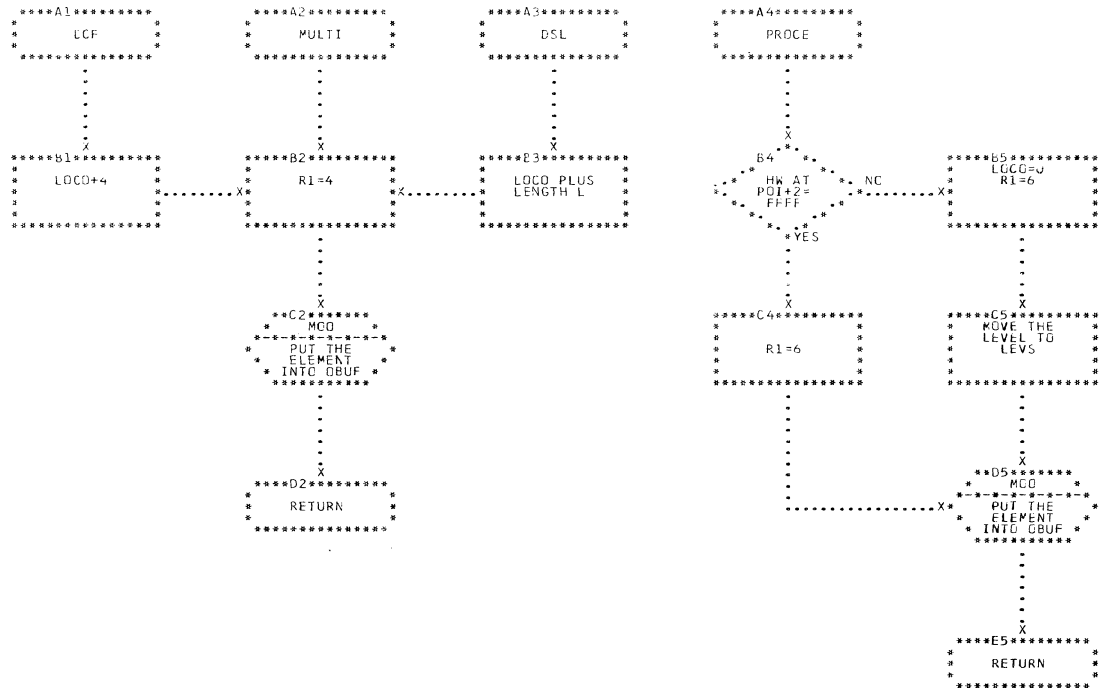


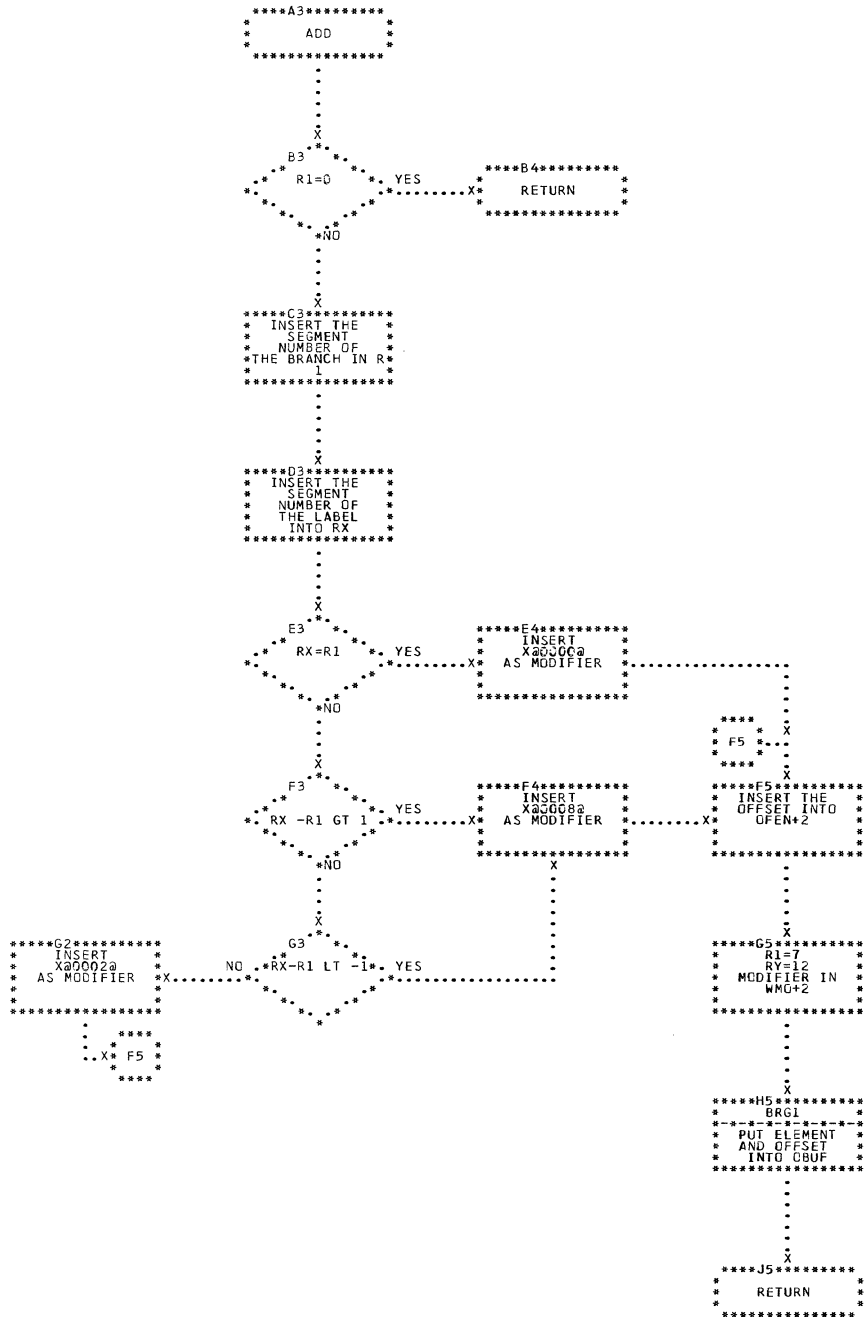
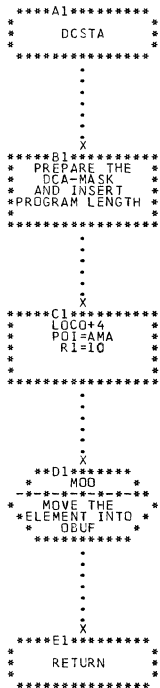


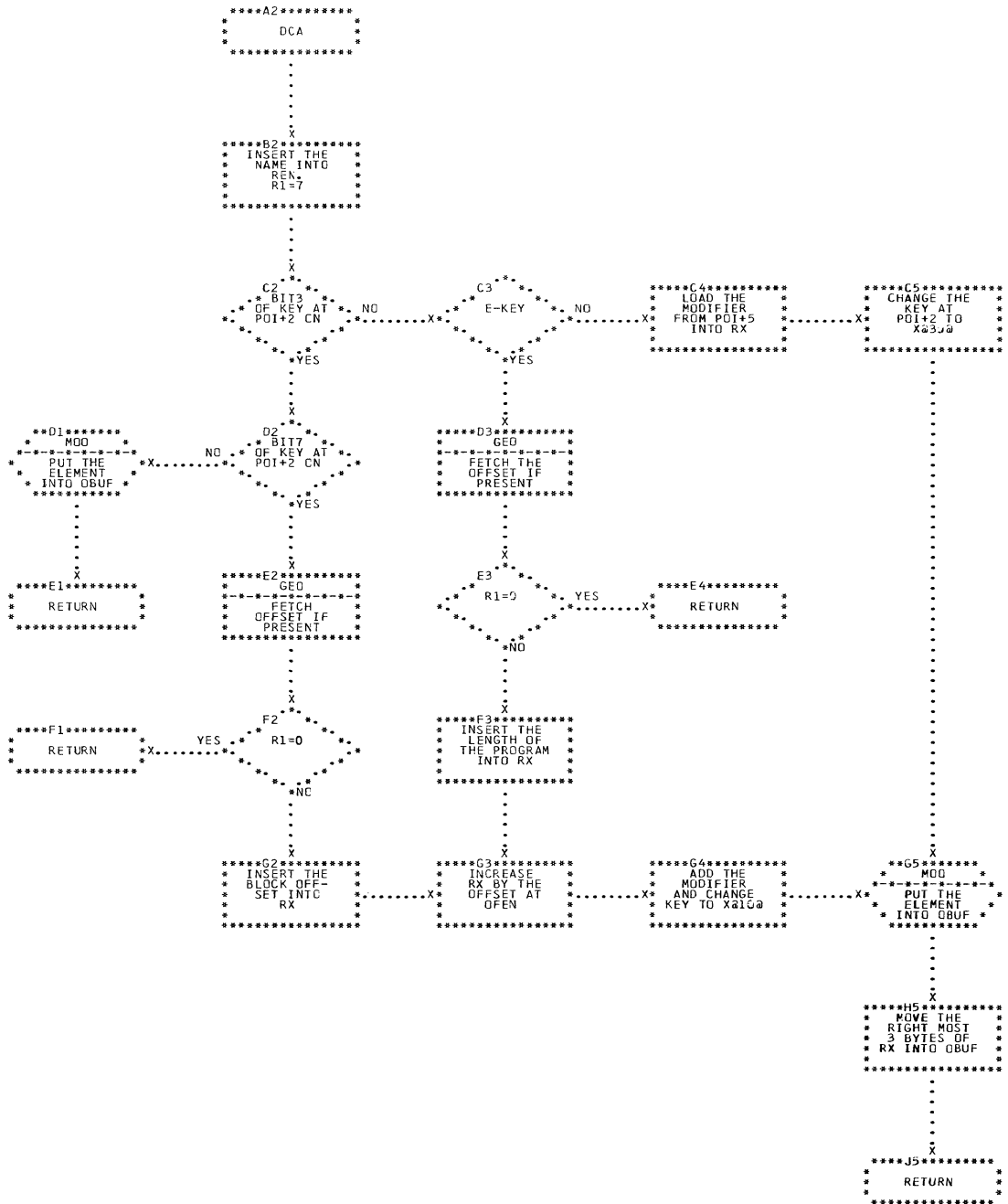


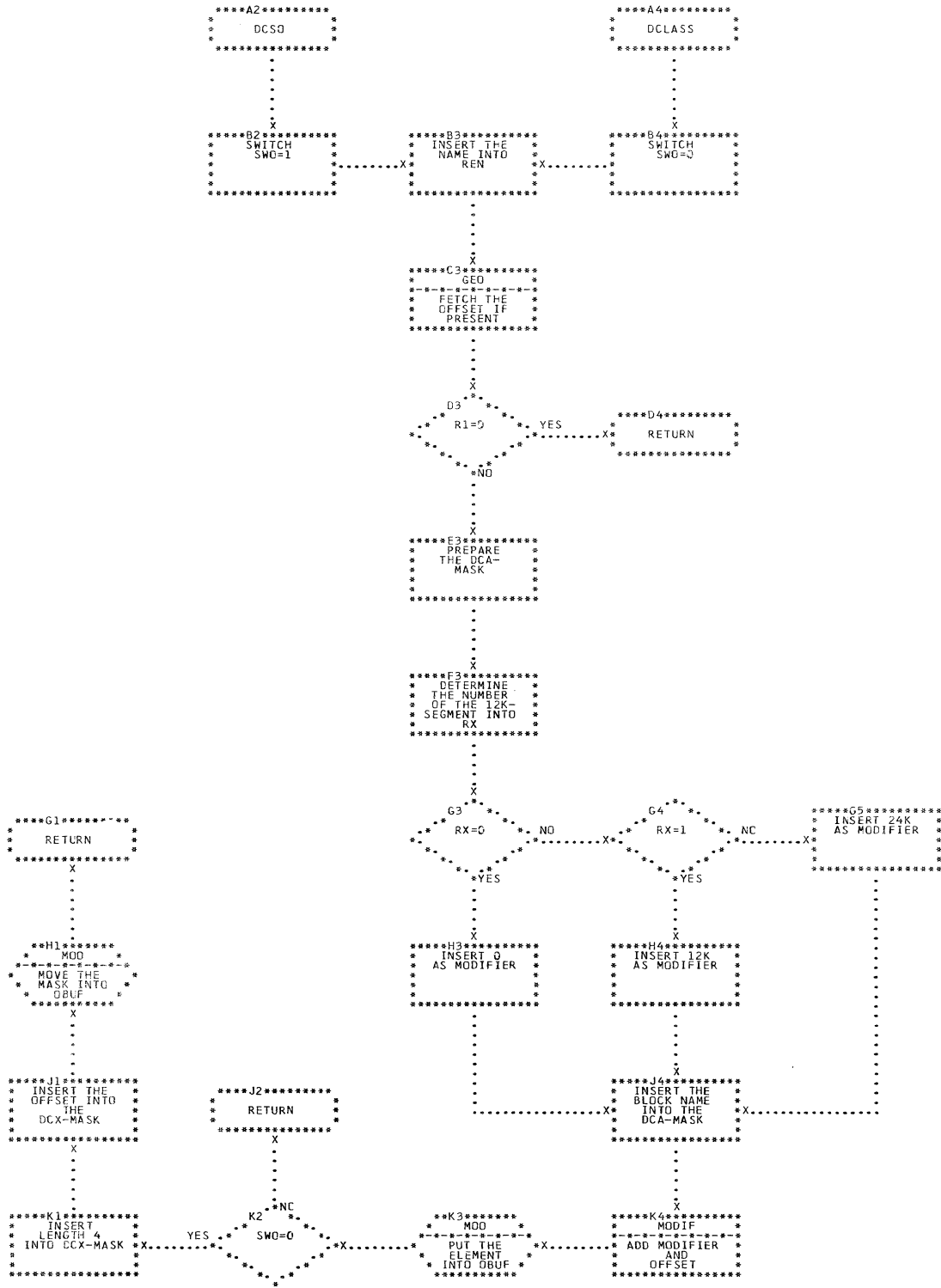


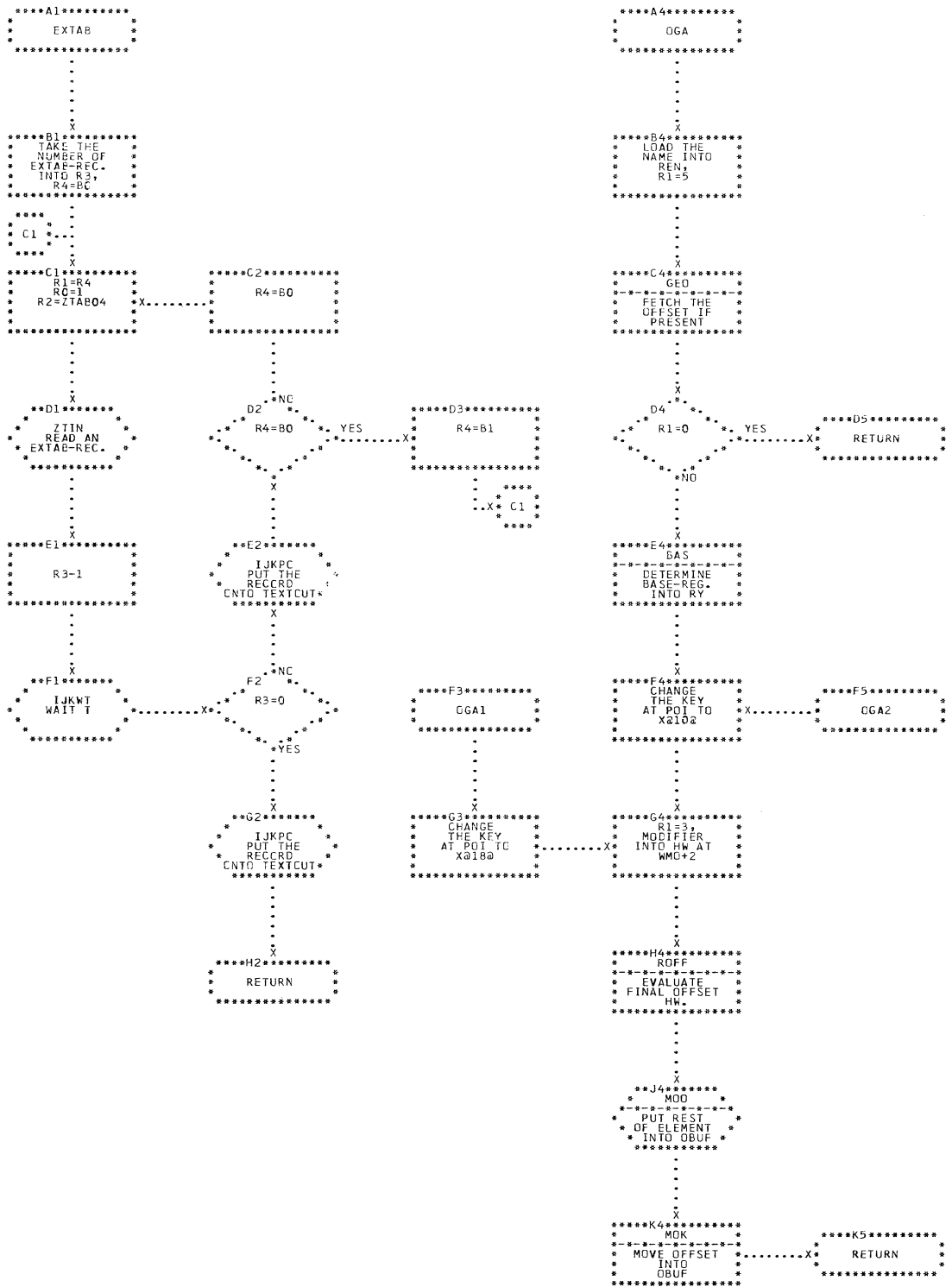


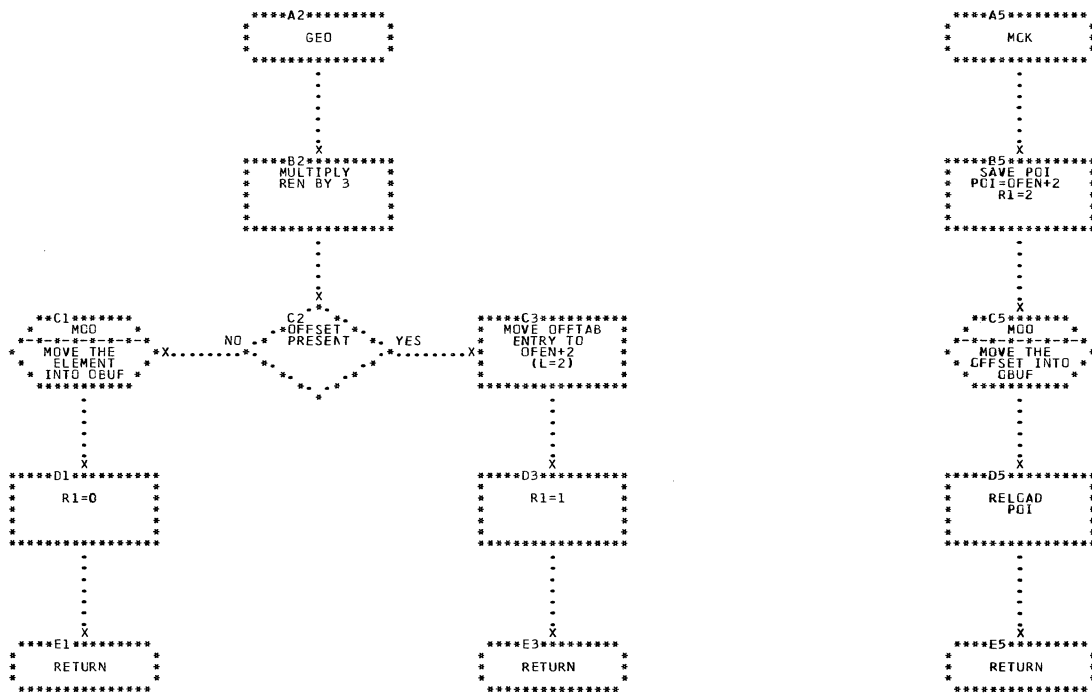


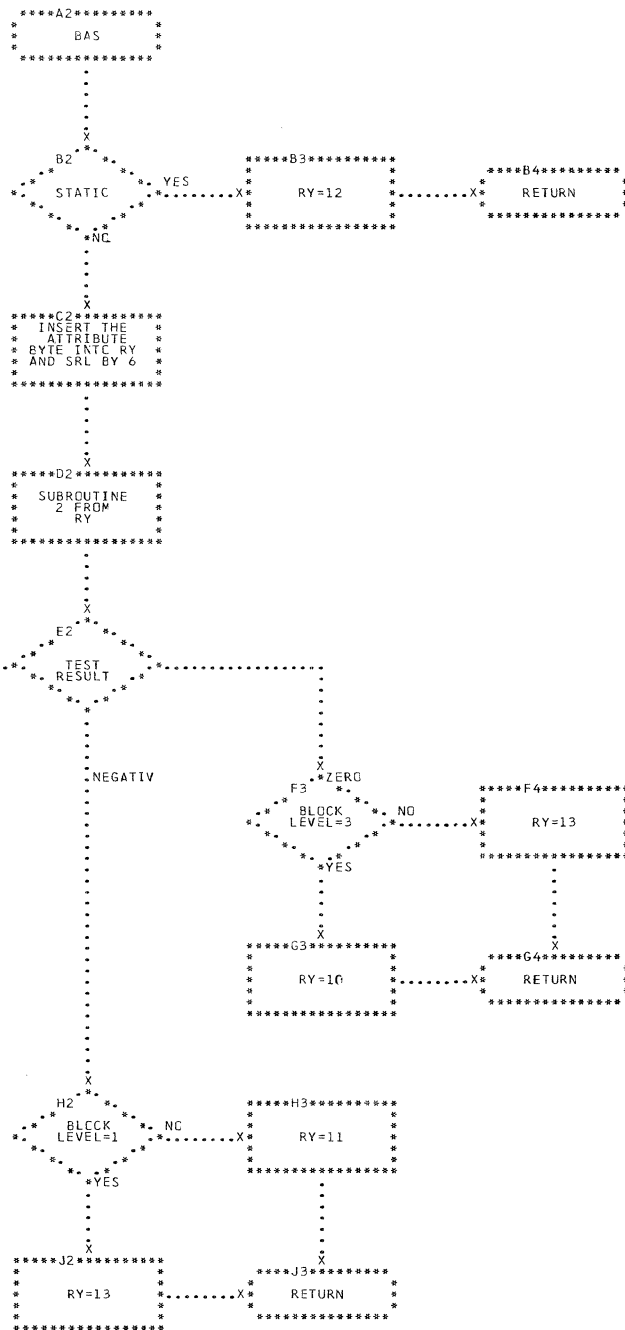


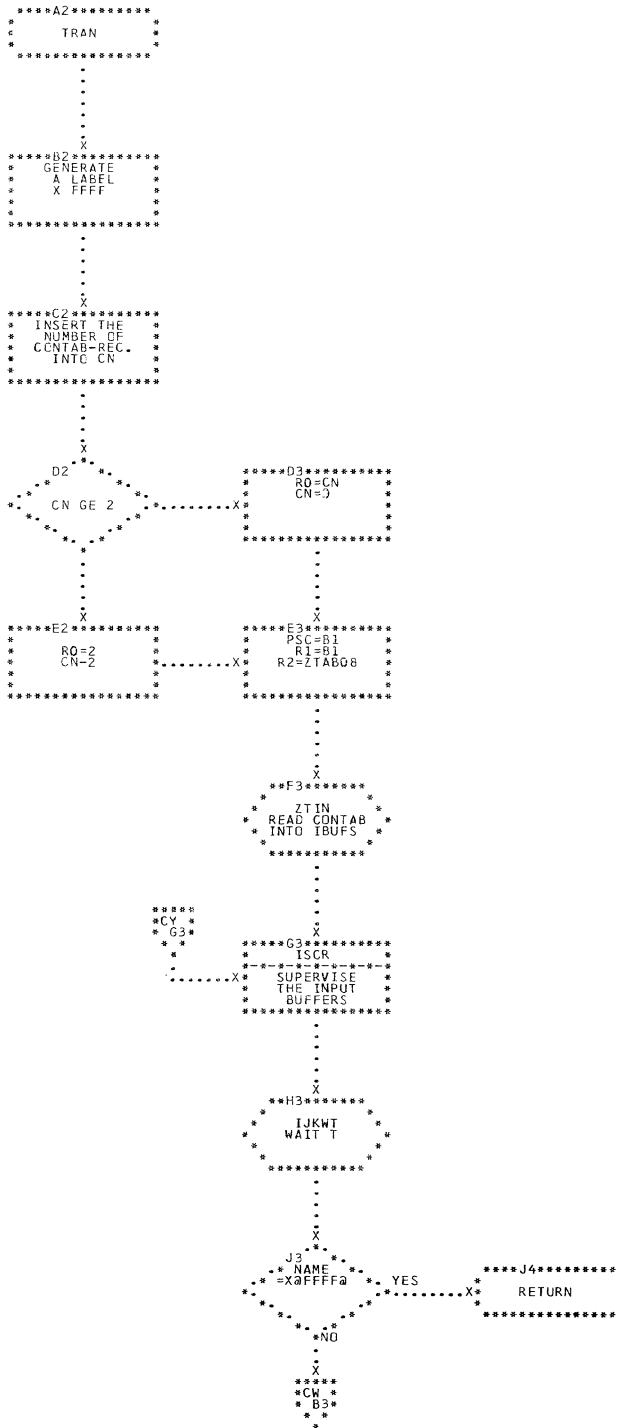


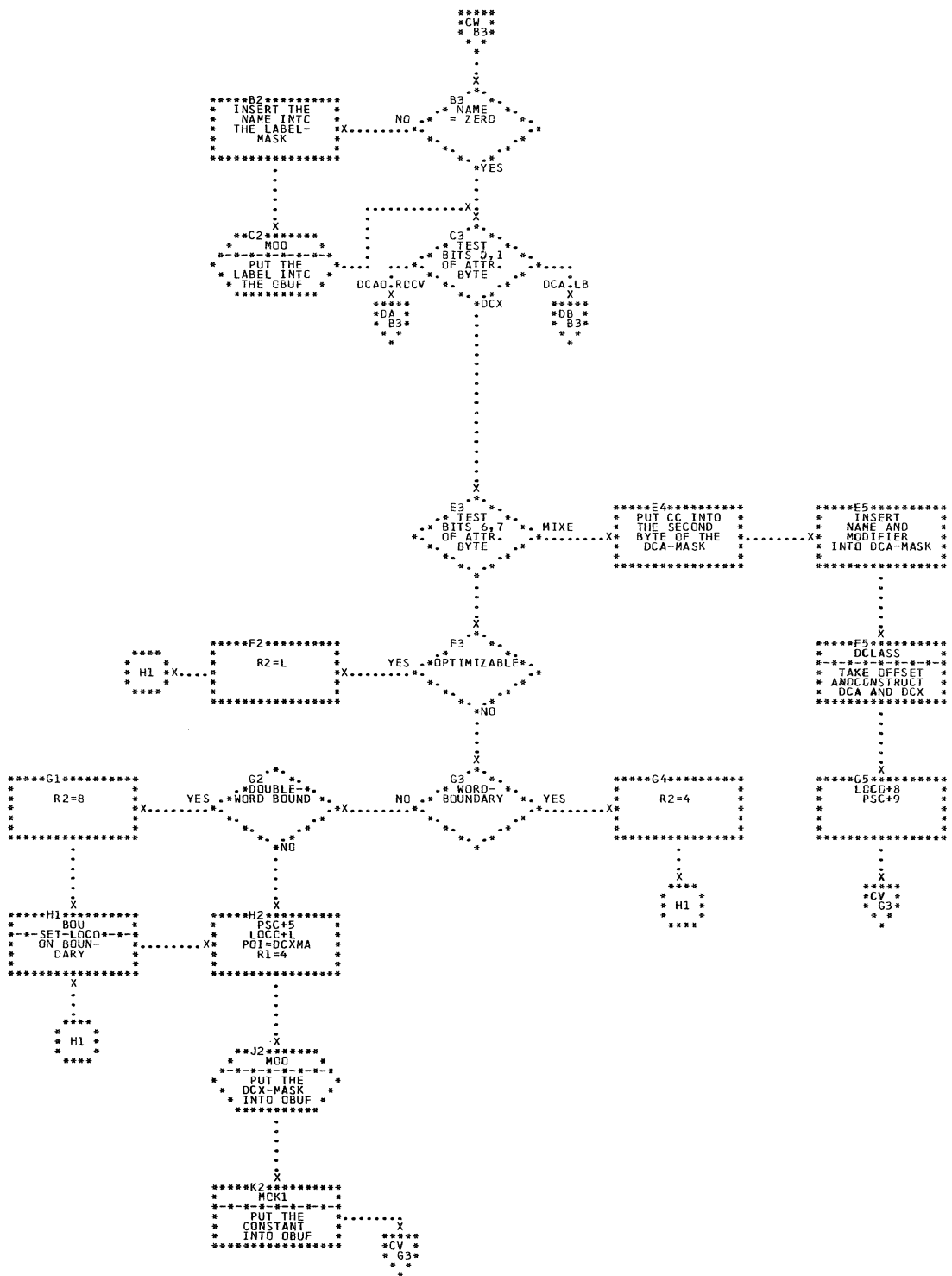


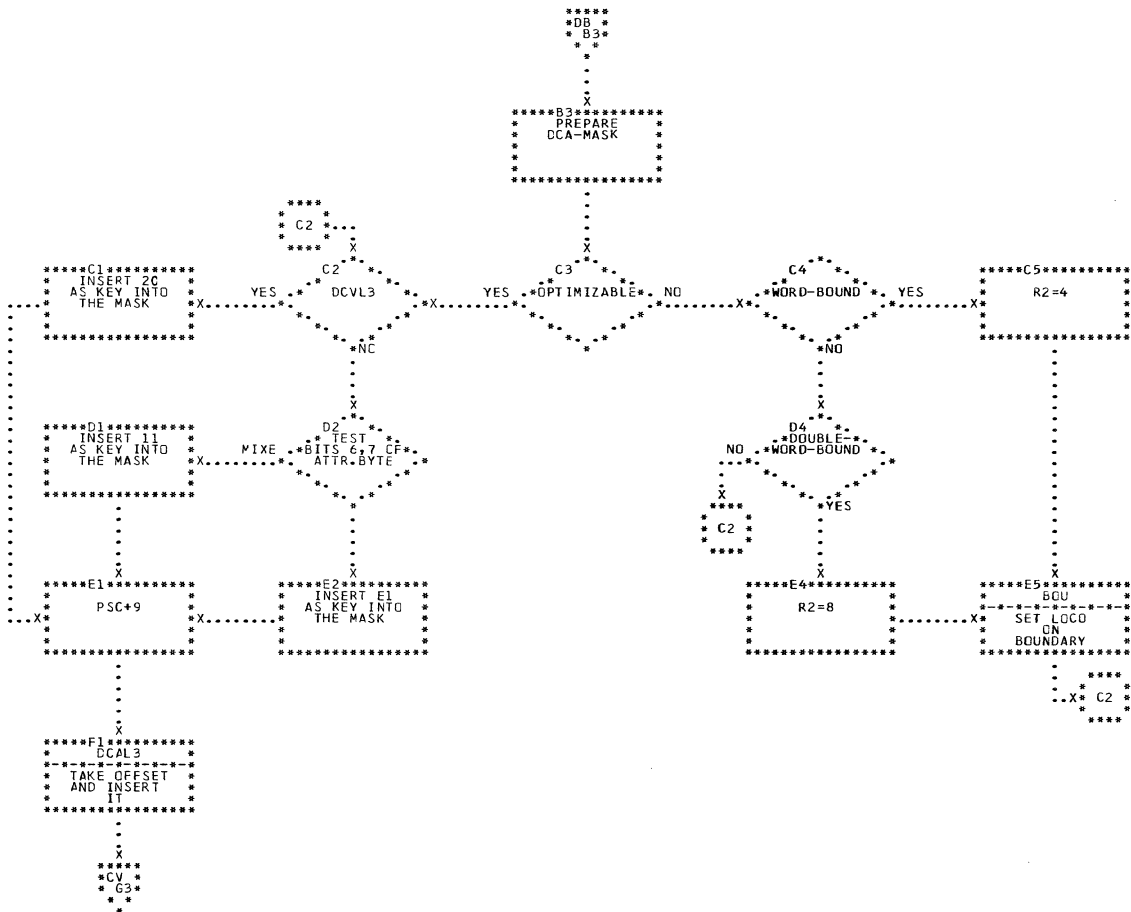


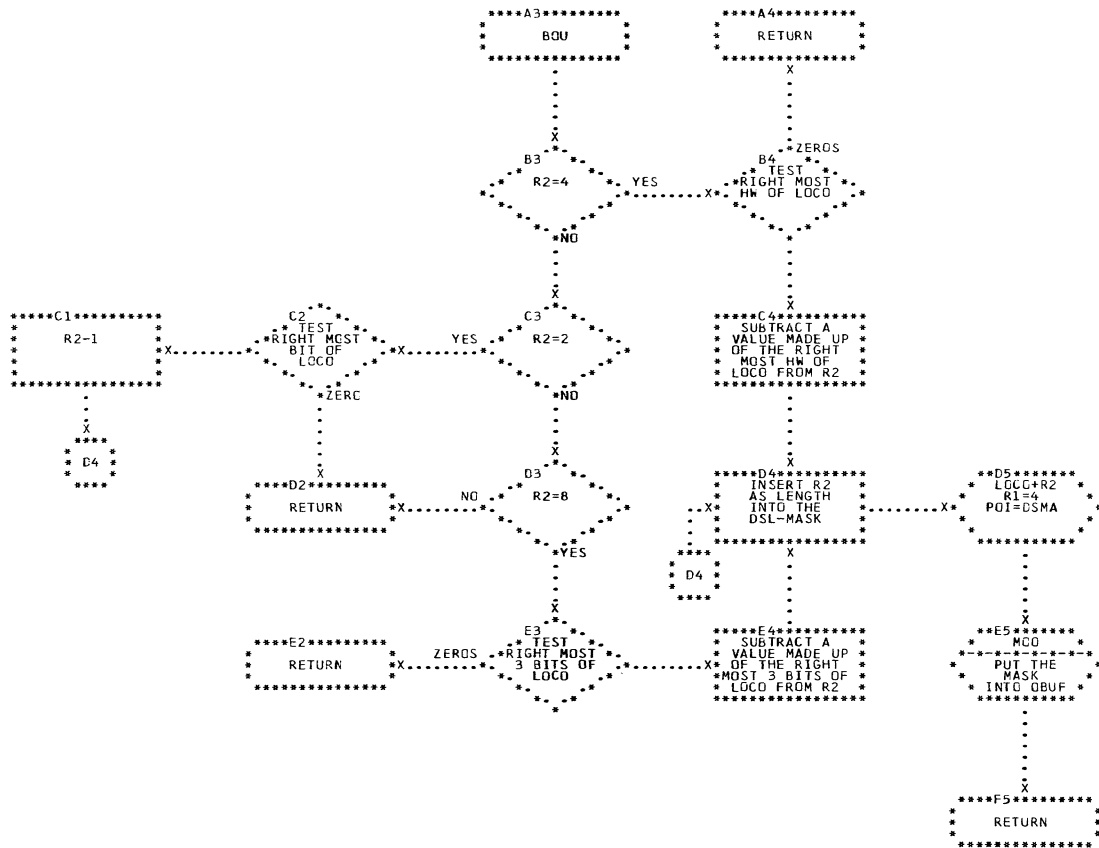


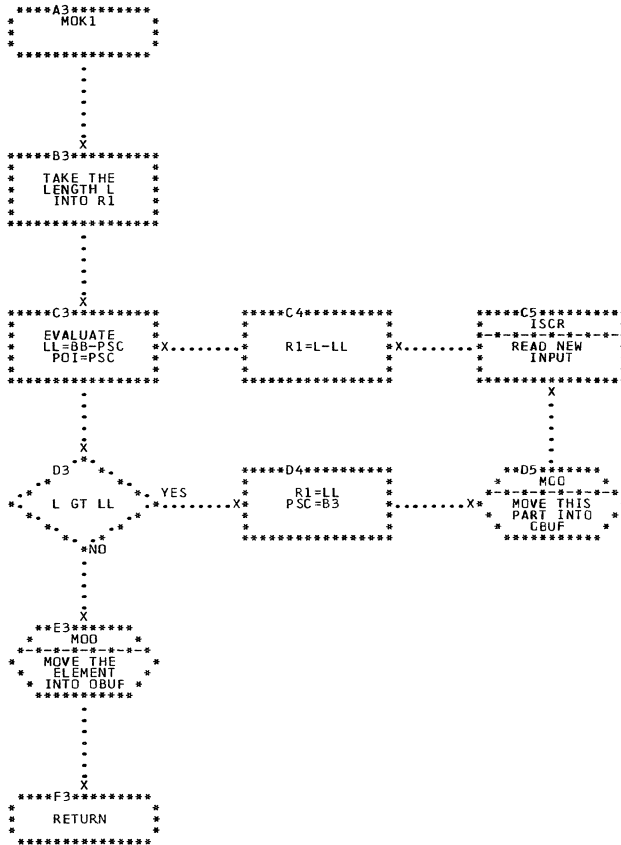


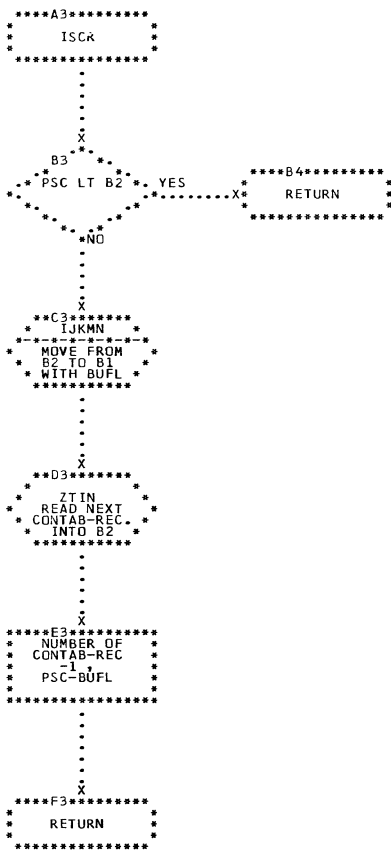


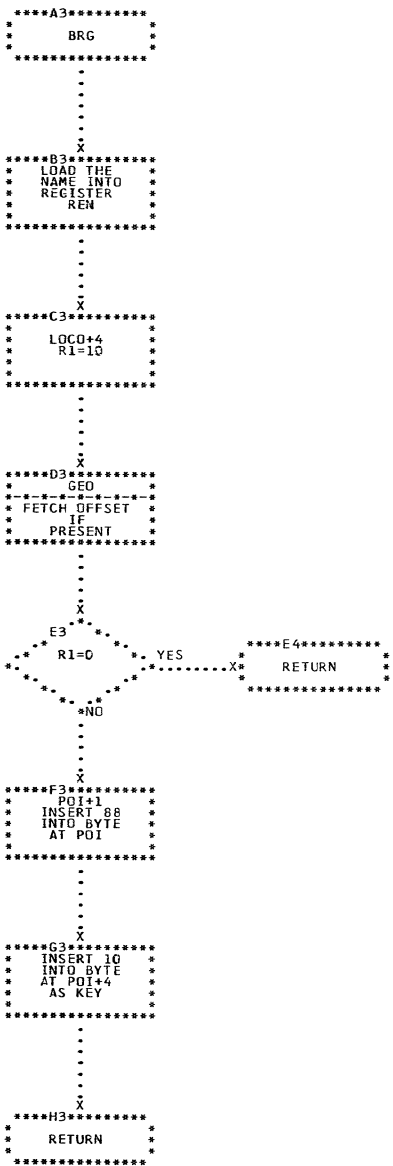












BEGIN

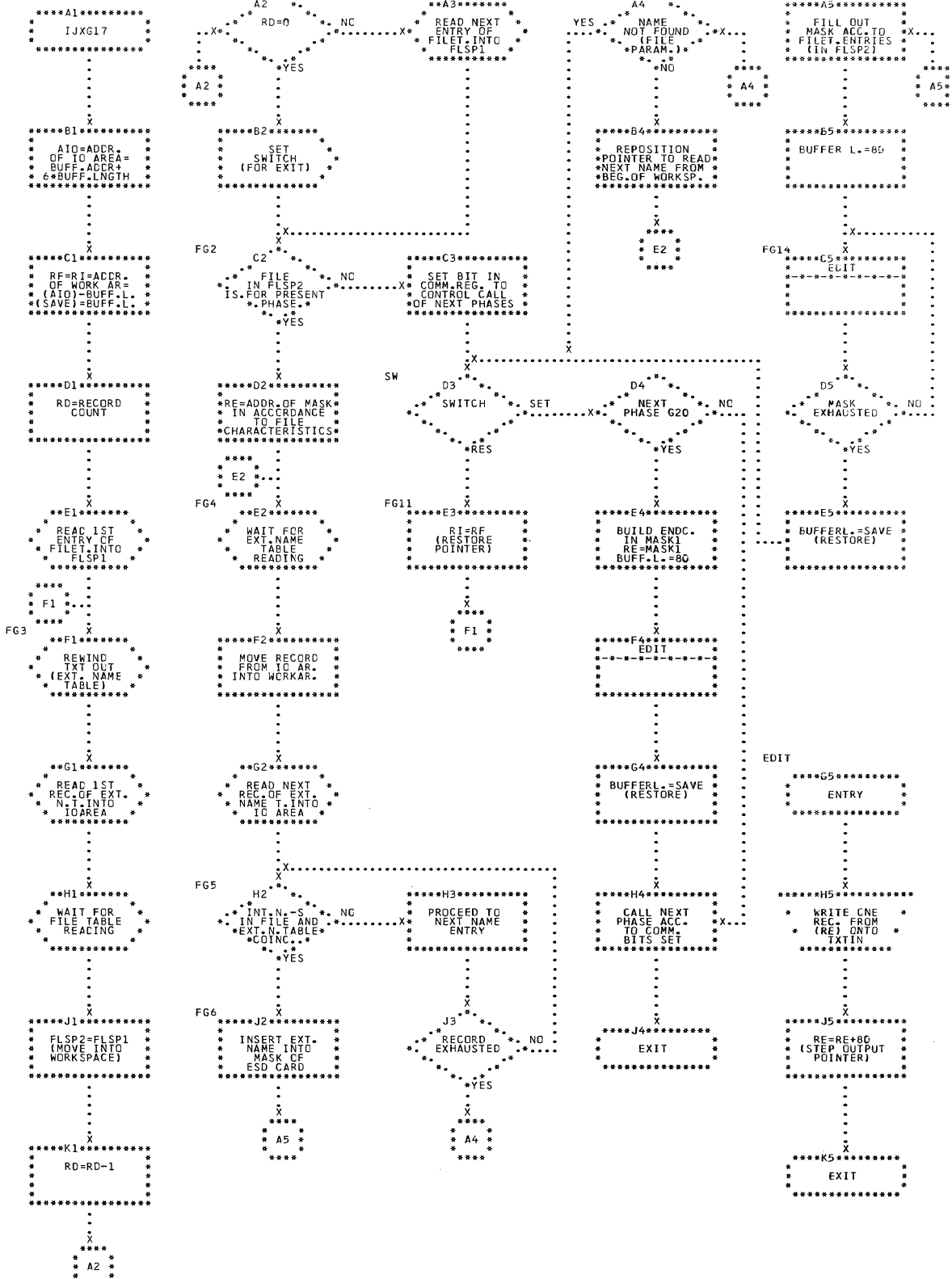
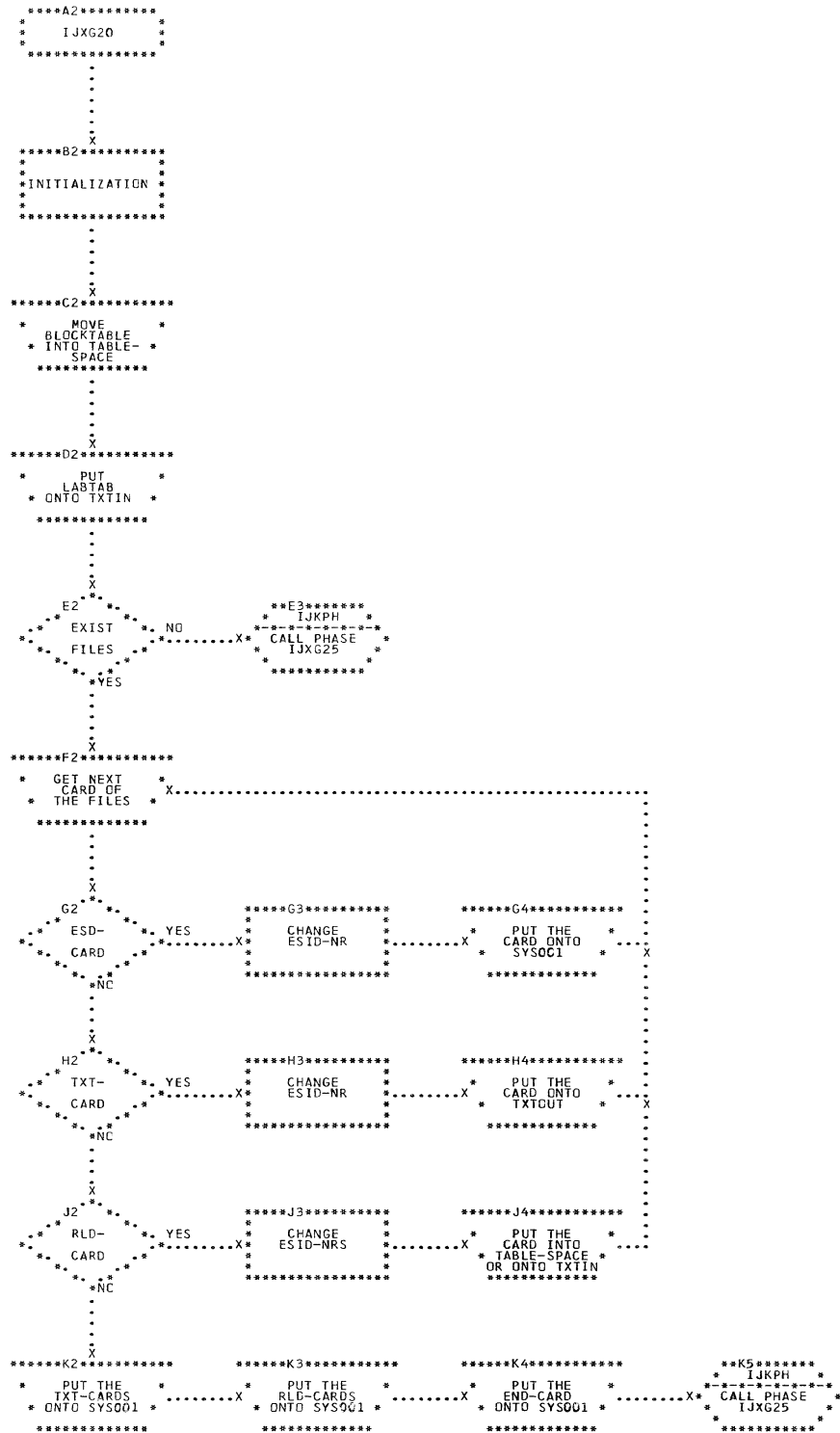


CHART DK. IJXG17

GENERATION OF DTF APP. AND TABLES



```
*****A2*****
* IJXG20 *
*****
```

```
*****B2*****
* INI 1 *
*-----*
*INITIALIZATION*
*****
```

```
*****C2*****
* LOAD *
* PARAMETERS *
*****
```

```
*****D2*****
* ZTIN *
* READ BLCK- *
* TABLE IN *
* TABLE SPACE *
*****
```

```
*****E2*****
* LABTAB *
*-----*
* MOVE LABTAB *
* FROM SYS001 *
* INTO TXIN *
*****
```

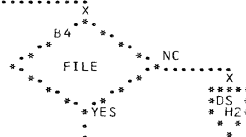
```
*****F2*****
* SAVE BASE- *
* REGISTER *
*****
```

```
*****G2*****
* POINTS *
* REWIND *
* SYS001 *
*****
```

```
*****H2*****
* RELOAD *
* BASE-REG *
*****
```

```
*****J2*****
* LOAD NOTE- *
* INFORMATION *
* OUT OF KSAVE 8 *
*****
```

```
*****K2*****
* IJKPT T *
* SET SYS001 *
* TO END OF *
* DTFTABLE *
*****
```



```
*****C4*****
* INI 2 *
*-----*
*INITIALIZATION*
*****
```

```
*****D4*****
* LOAD NOTE- *
* INFORMATION *
* OUT OF ZTAB18 *
*****
```

```
*****E4*****
* STORE THE *
* NOTEINFORM *
* INTO NOTEX *
*****
```

```
*****F4*****
* IJKPT 0 *
* SET TXOUT *
* TO END OF EX- *
* TERNAL NAME TAB *
*****
```

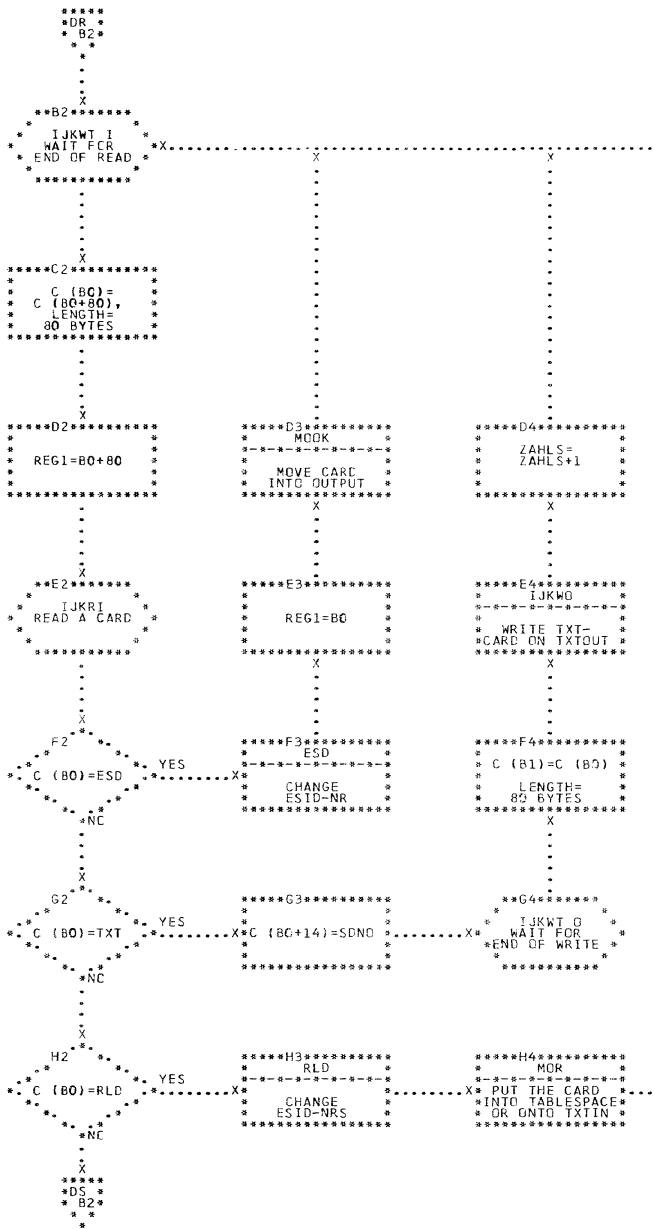
```
*****G4*****
* LOAD NOTE- *
* INFORMATION *
* OUT OF ZTAB19 *
*****
```

```
*****H4*****
* IJKPT 1 *
* SET TXIN TO *
* BEGINNING OF *
* THE FILES *
*****
```

```
*****J4*****
* BUFS=IJKBHL *
* IJKMBL= BC *
* RA= C *
* REG1= B0+80 *
*****
```

```
*****K4*****
* IJKGI *
* READ A CARD *
*****
```

```
*****
* DR *
* B2 *
*****
```



```

*****
*DS*
*E2*
*
*
*
*
*****B2*****
*TXSYS*
*-----*
*PUT TXTCARDS*
*INTC OUTPUT*
*-----*
*
*
*
*
*****C2*****
*RLDSYS*
*-----*
*PUT RLD-CARDS*
*INTC OUTPUT*
*-----*
*
*
*
*
*****D2*****
*
*REG1= BC*
*
*-----*
*
*
*
*
*****E2*****
*MCCK*
*-----*
*MOVE END-CARD*
*INTC OUTPUT*
*-----*
*
*
*
*
*****F2*****
*
*IKMBL=
*BUFSA*
*
*-----*
*
*
*
*
*****G2*****
*
*C (B5)=PGUS
*LENGTH=
*1 BYTE*
*-----*
*
*
*DS*
*H2*
*
*
*****H2*****
*
*REG1= NCTE*
*
*-----*
*
*
*
*
*****J2*****
*
*IKPT I
*SET TXTIN
*TC BEGIN CF
*LABTAB*
*-----*
*
*
*
*
*****K2*****
*IKPH*
*-----*
*CALL G25
*REWIND TXTCUT*
*-----*

```

```

*****A2*****
*   INI 1   *
*****
*   .   *
*   .   *
*   .   *
*   X   *
*****B2*****
* SET TRANSFER *
* BIT OF ZTAB13 *
* TC ZERC *
*****
*   .   *
*   .   *
*   .   *
*   X   *
*****C2*****
* SET TRANSFER *
* BIT OF ZTAB20 *
* TC ZERC *
*****
*   .   *
*   .   *
*   .   *
*   X   *
*****D2*****
* BC= IJKMBS *
* +2=IJKBML *
*   *
*   *
*   *
*   X   *
*****E2*****
* B1=B0+IJKMEL *
*   *
*   *
*   .   *
*   .   *
*   .   *
*   X   *
*****F2*****
*   RETURN   *
*****

```

****A2*****
INI 2

X
****B2*****
B1= B0+160

X
****C2*****
B3=B0+IJKMBL

X
****D2*****
BA=B3+8

X
****E2*****
BB=BA+320

X
****F2*****
SET BUFFER-
LENGTH IN
ZTABLE TC 320

X
****G2*****
SET TRANSFER
BIT OF ZTABLE
TO ZERO

X
****H2*****
SDNC=0

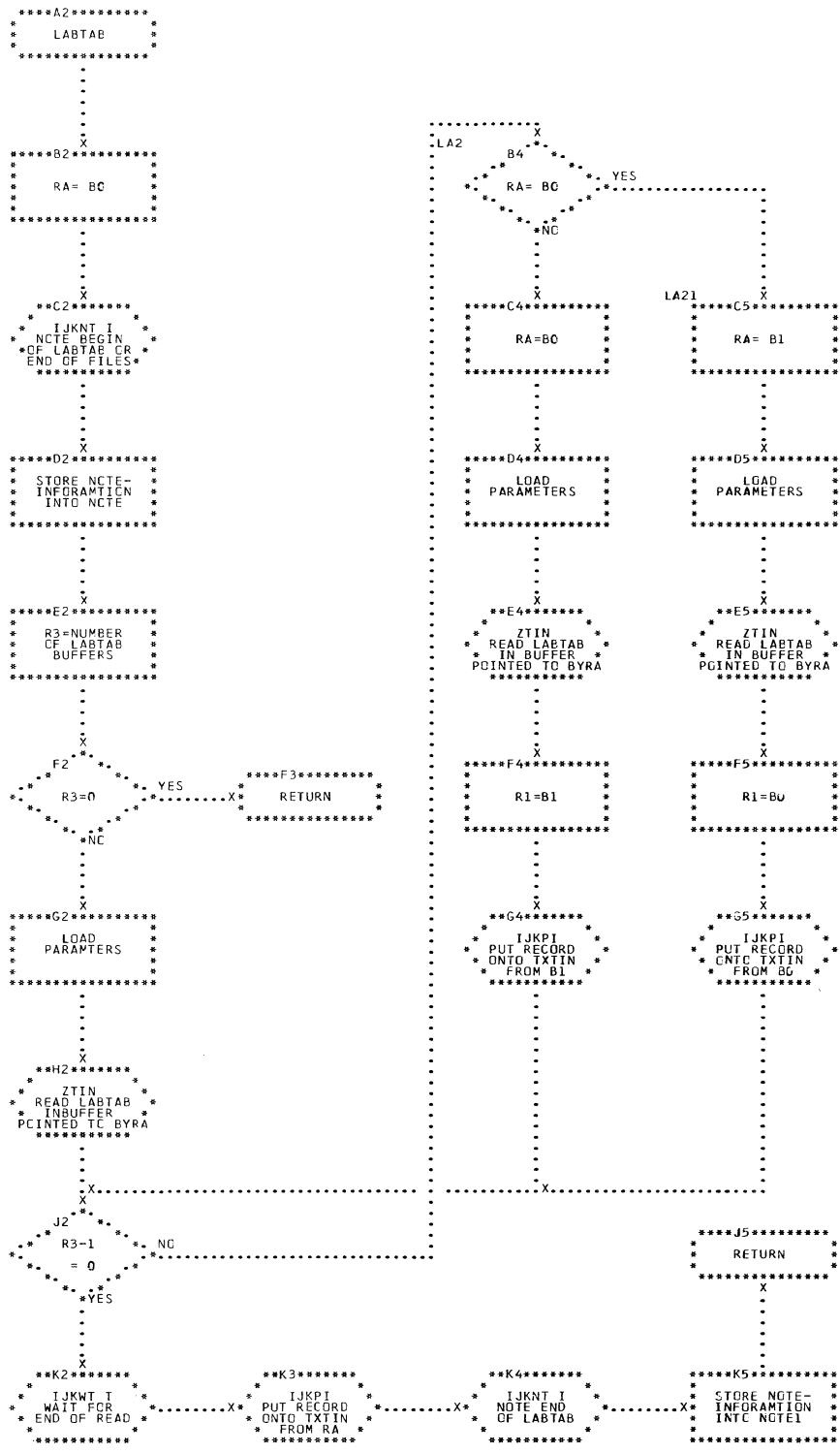
X
****J2*****
ESID= 1

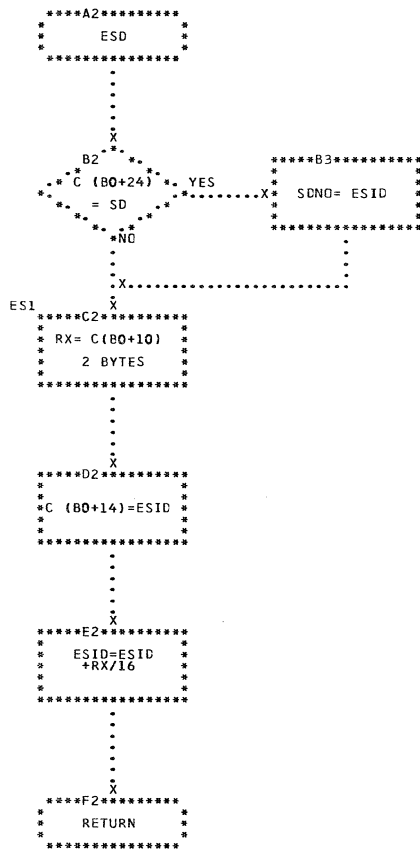
X
****B4*****
CSP=BA

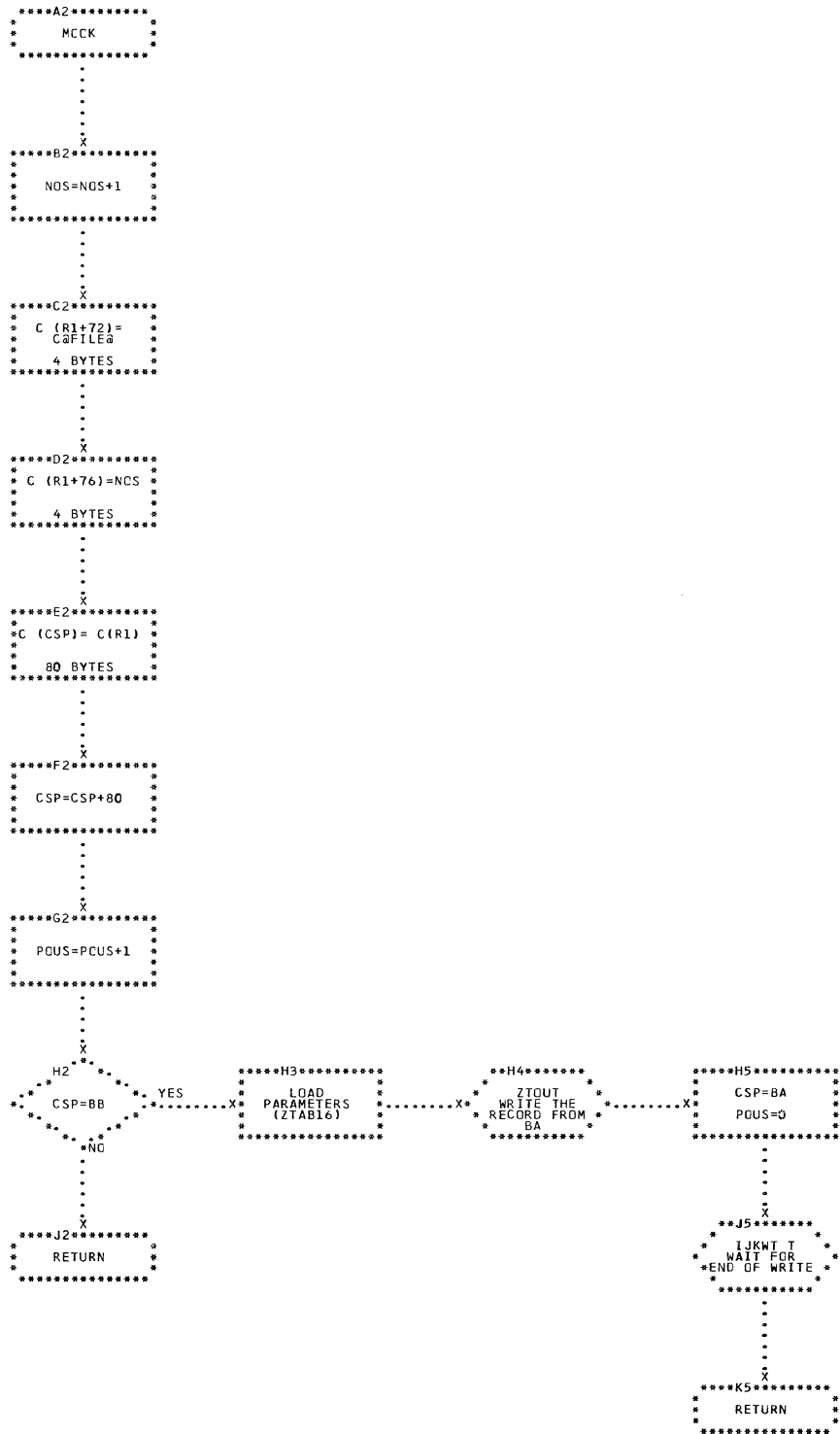
X
****C4*****
TSB=IJKMTS
+128

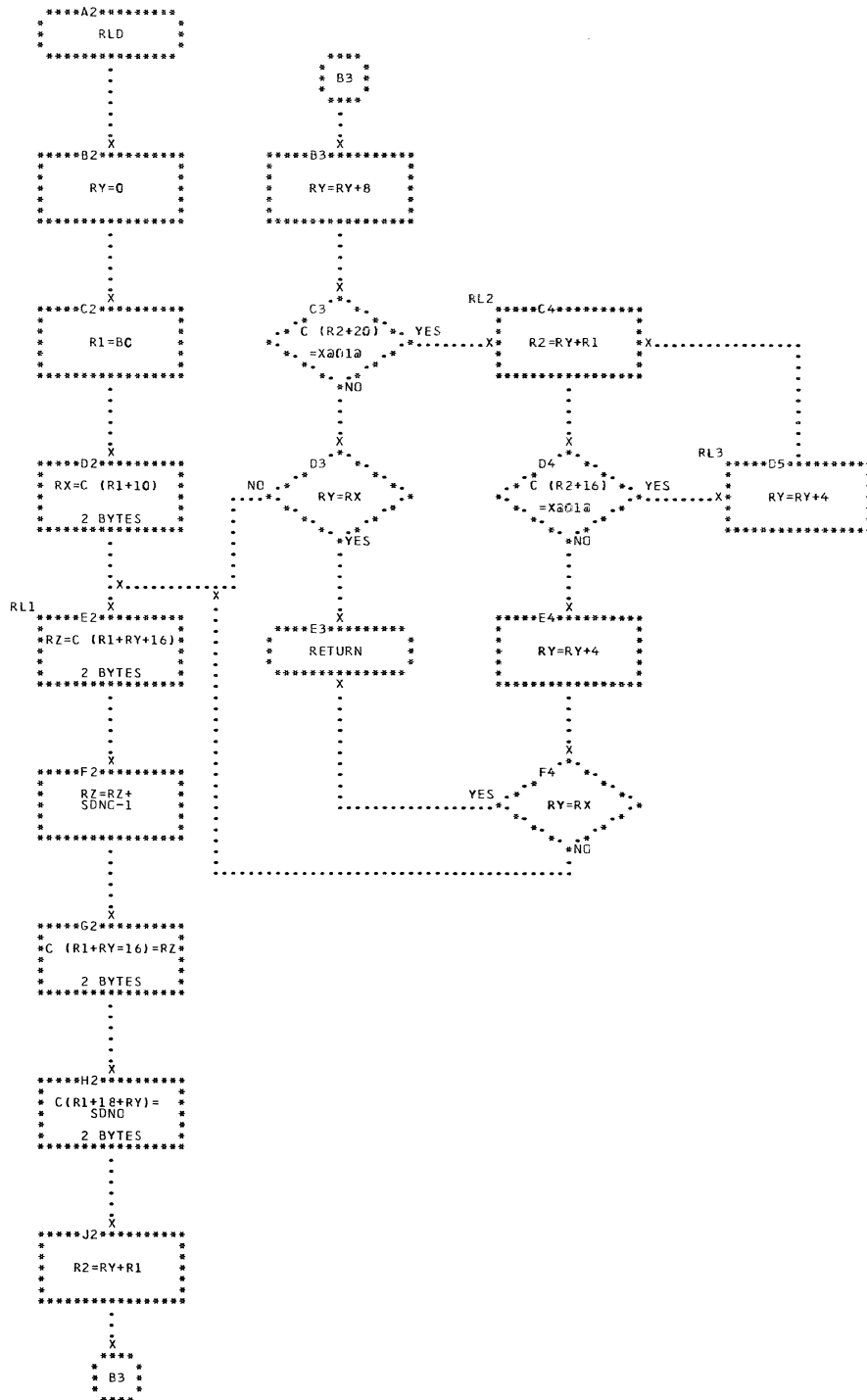
X
****D4*****
SPEI=IJKMTS
+128

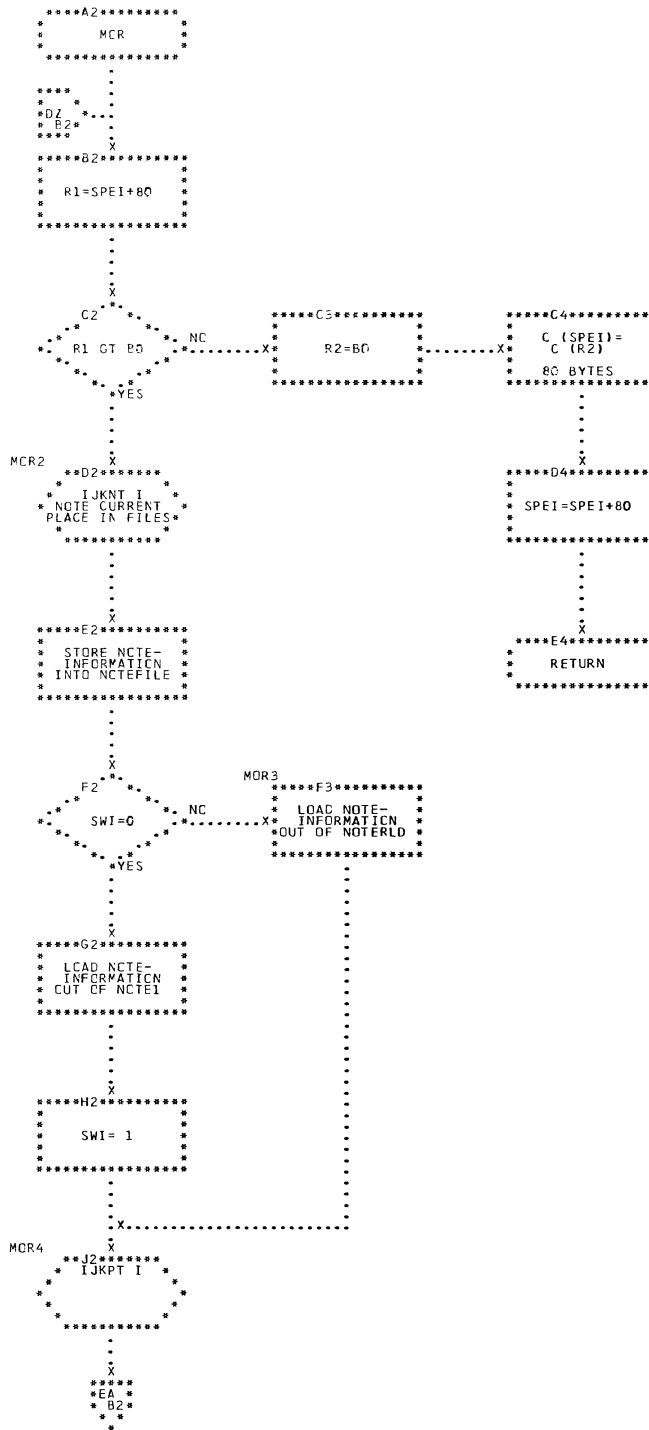
X
****E4*****
RETURN

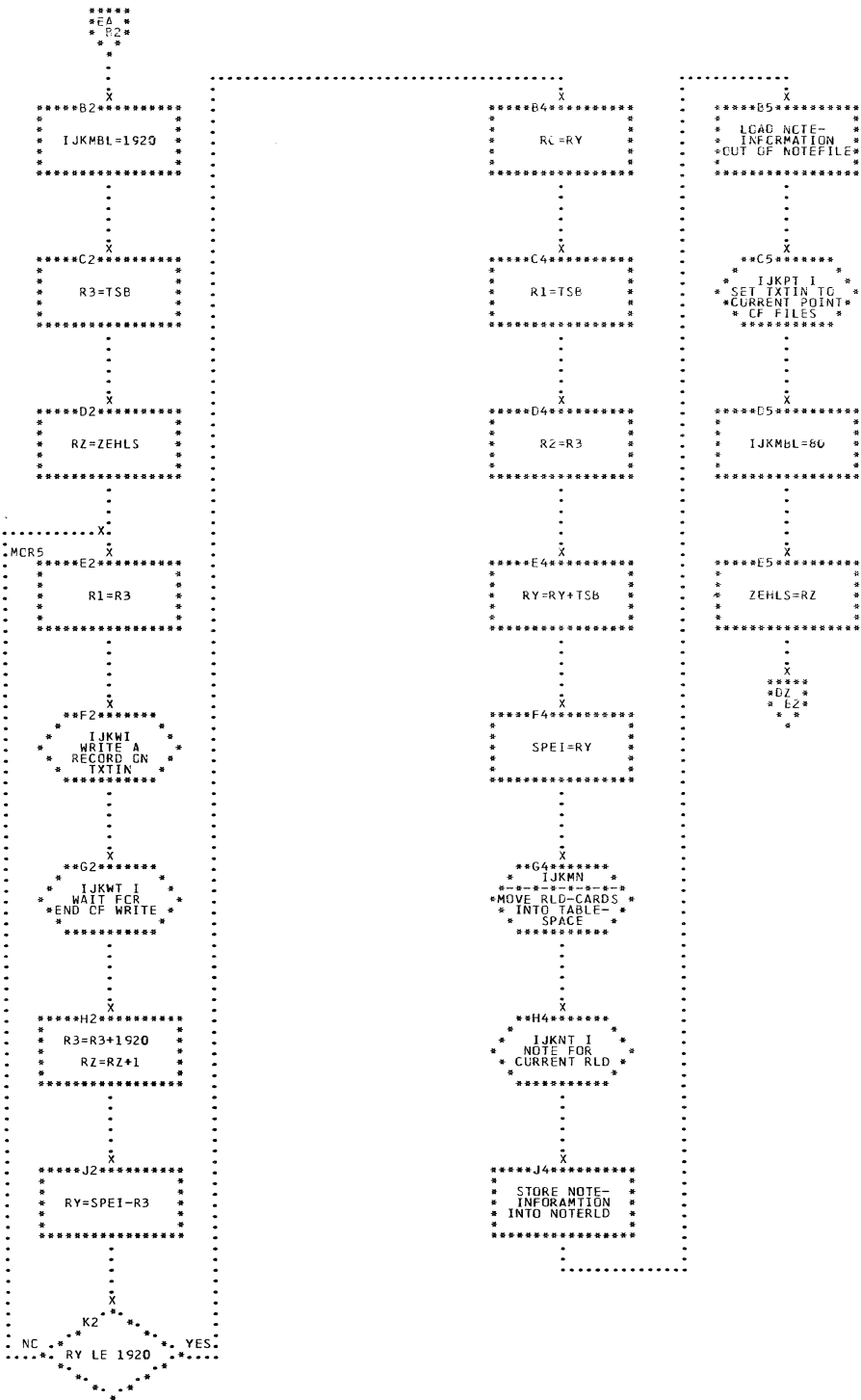












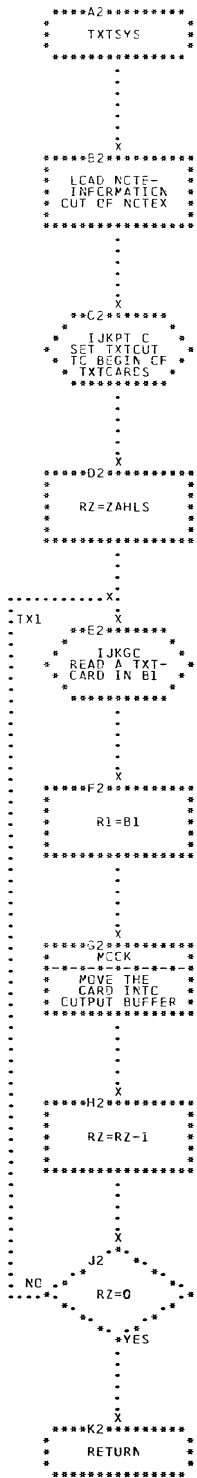
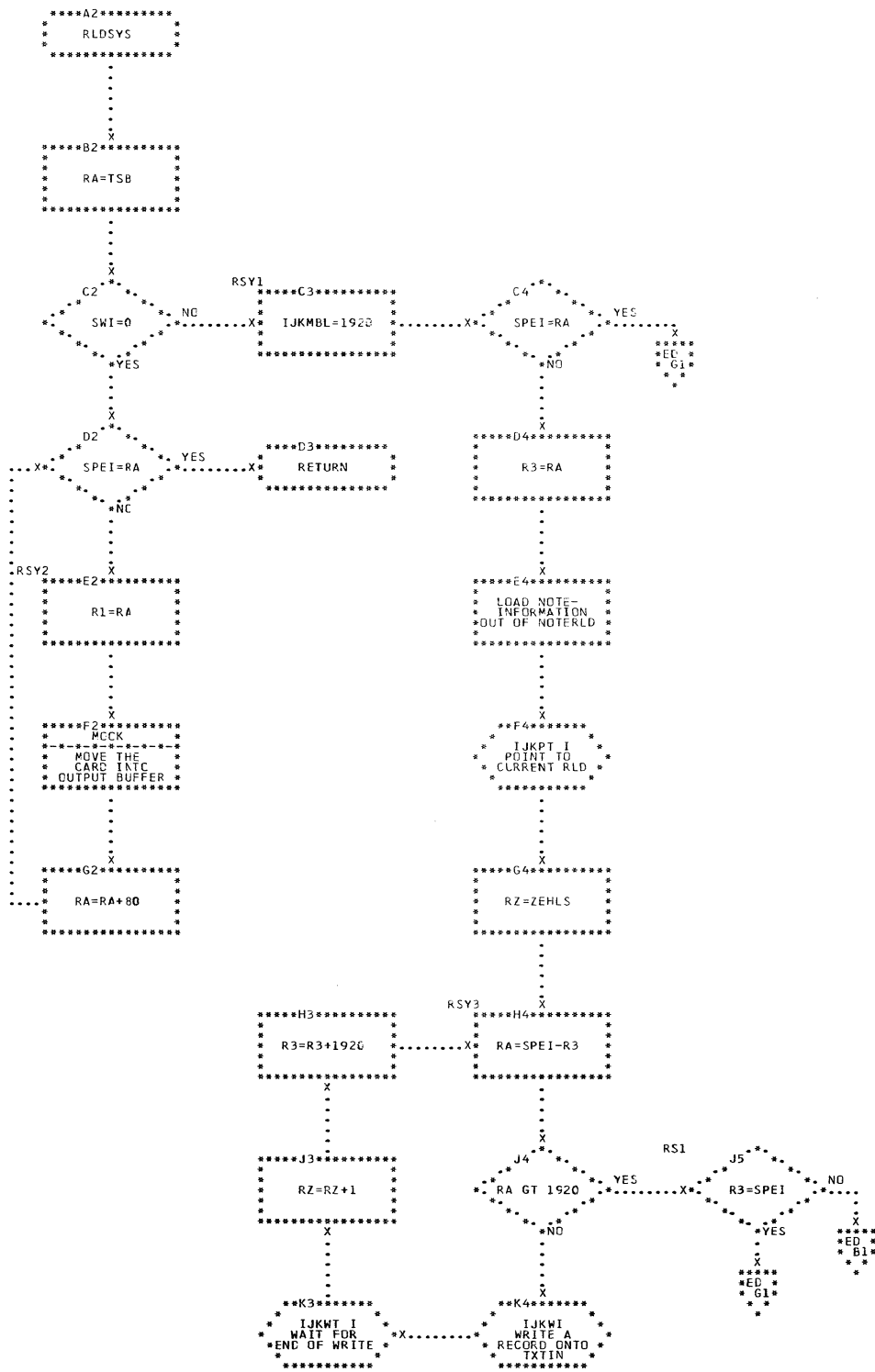
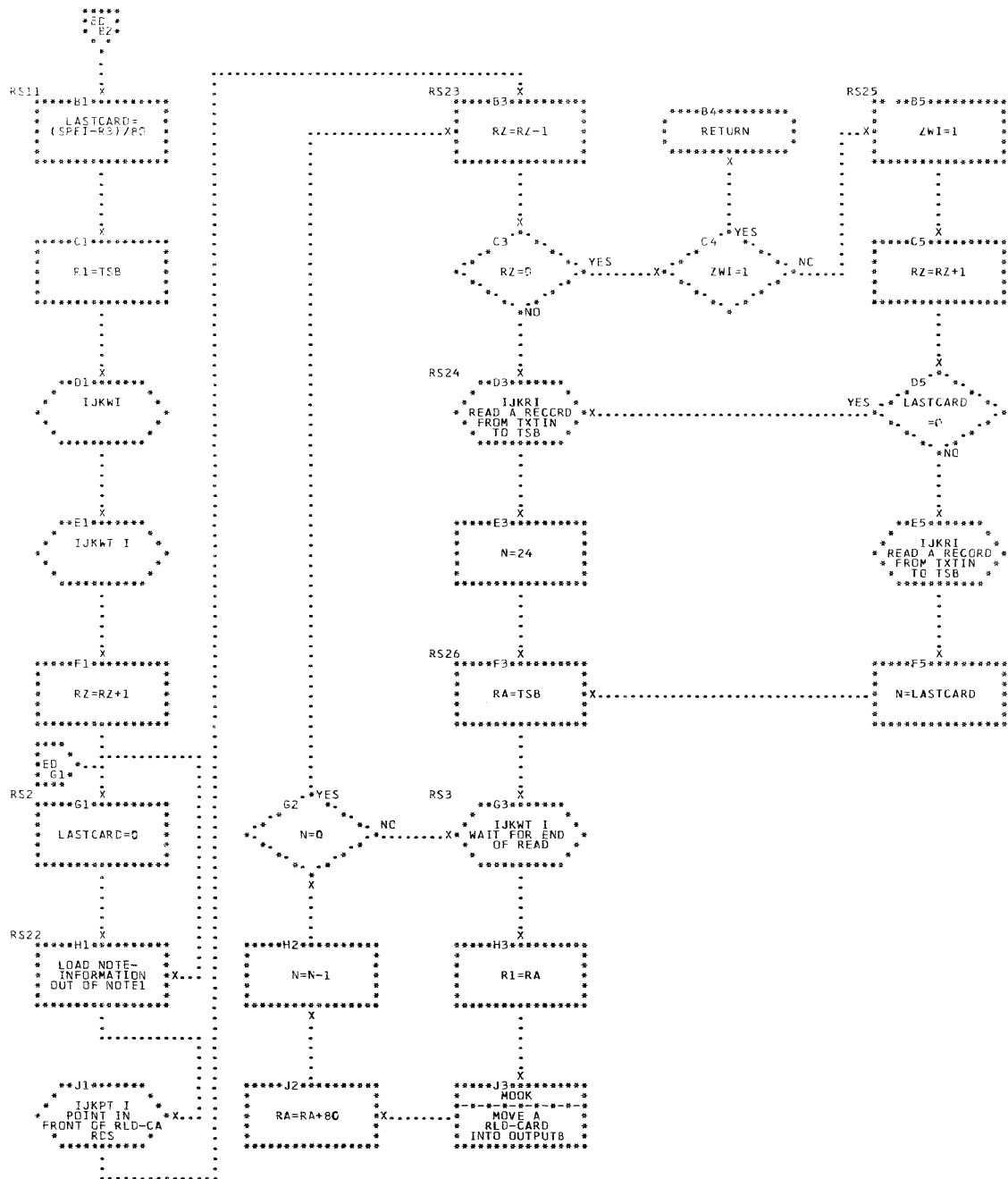
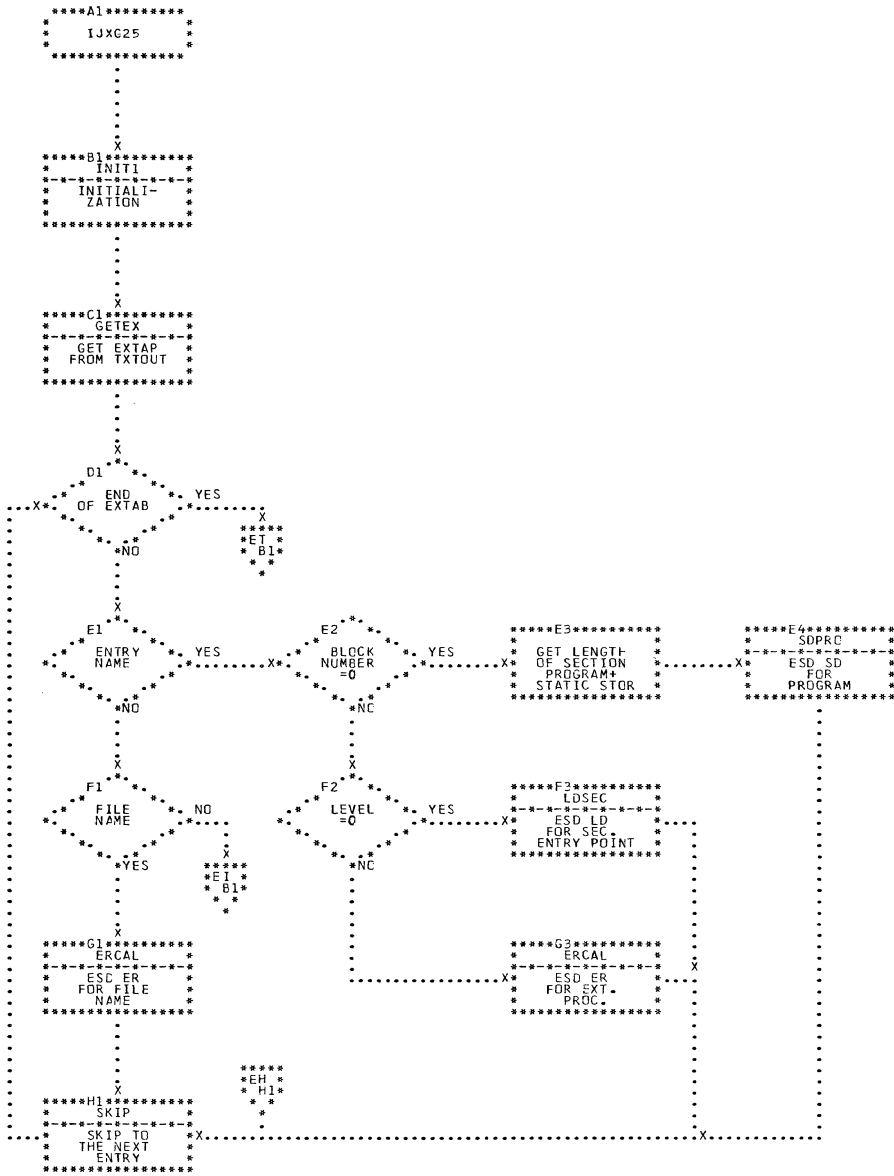


CHART EB. IJXG20 TXT SYS







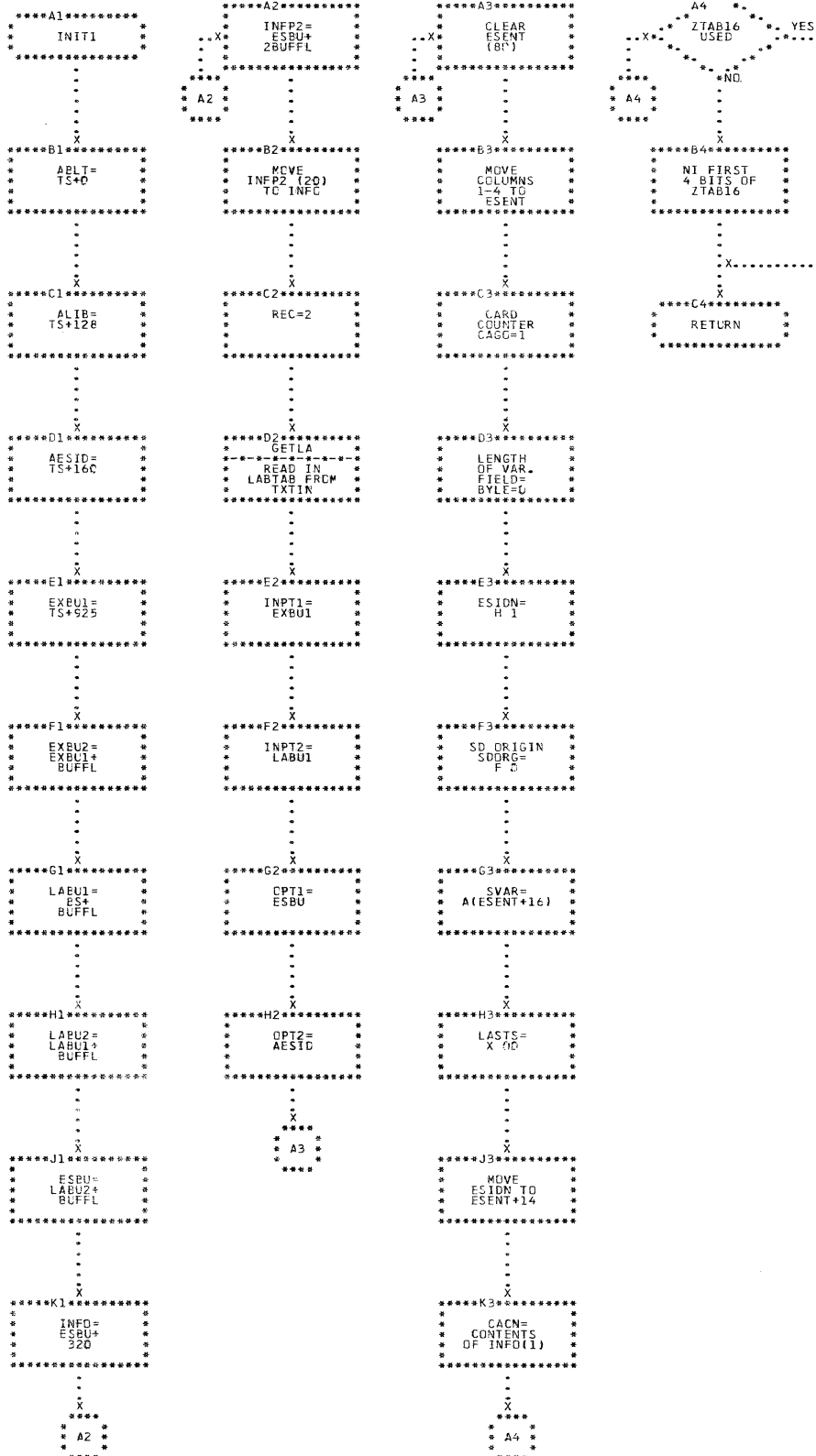
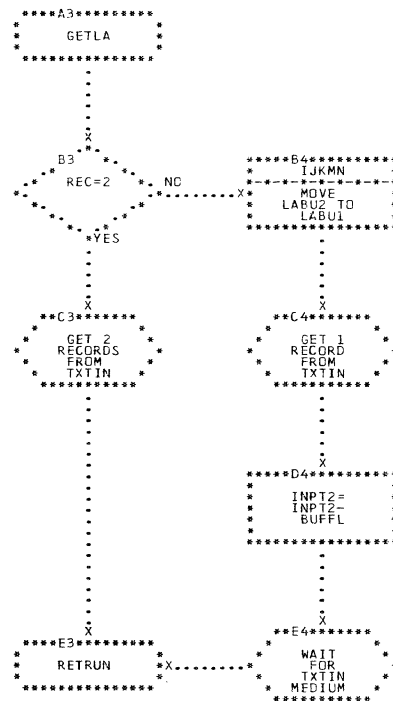
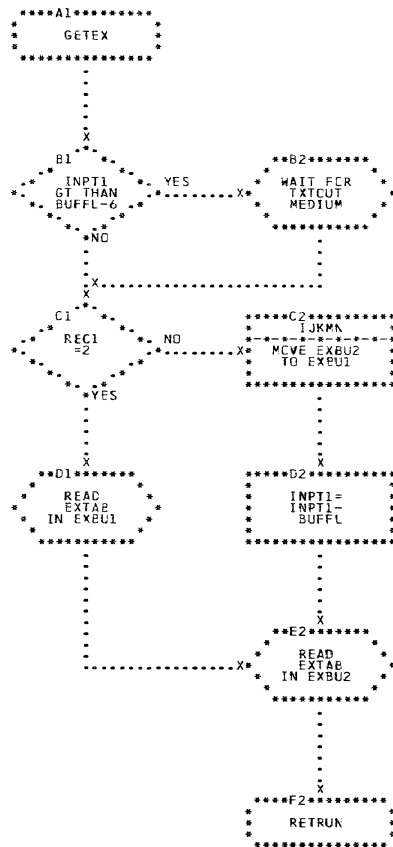


CHART EJ. IJXG25 INIT1



```
*****A1*****
SCPRC
*****
```

```
*****A2*****
MESID
*****
```

```
*****B1*****
MOVE
EXT. NAME
TO SVAR
*****
```

```
*****B2*****
MOVE
INT. NAME
TO OPT2
*****
```

```
*****C1*****
MOVE
ORIGIN=0
TO SVAR+9
*****
```

```
*****C2*****
MOVE
ESIDN
TO OPT2+2
*****
```

```
*****D1*****
MOVE
TYPE = X 00
TO SVAR+8
*****
```

```
*****D2*****
OPT2=
OPT2+3
*****
```

```
*****E1*****
MOVE
LENGTH
TO SVAR+13
*****
```

```
*****E2*****
RETURN
*****
```

```
*****F1*****
MESID
MAKE
ESID
ENTRY
*****
```

```
*****G1*****
ESFIN
BUILD
ESD
CARD
*****
```

```
*****H1*****
ESIDN=
ESIDN+
1
*****
```

```
*****J1*****
RETURN
*****
```

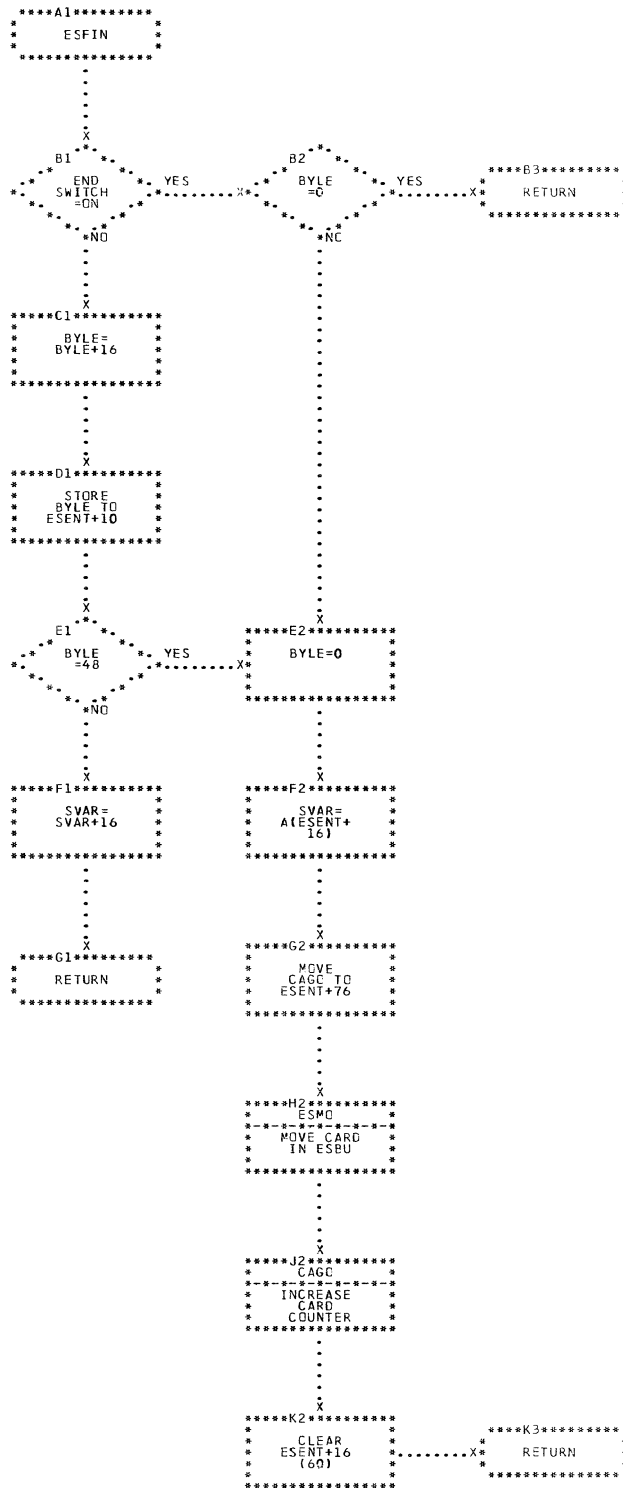
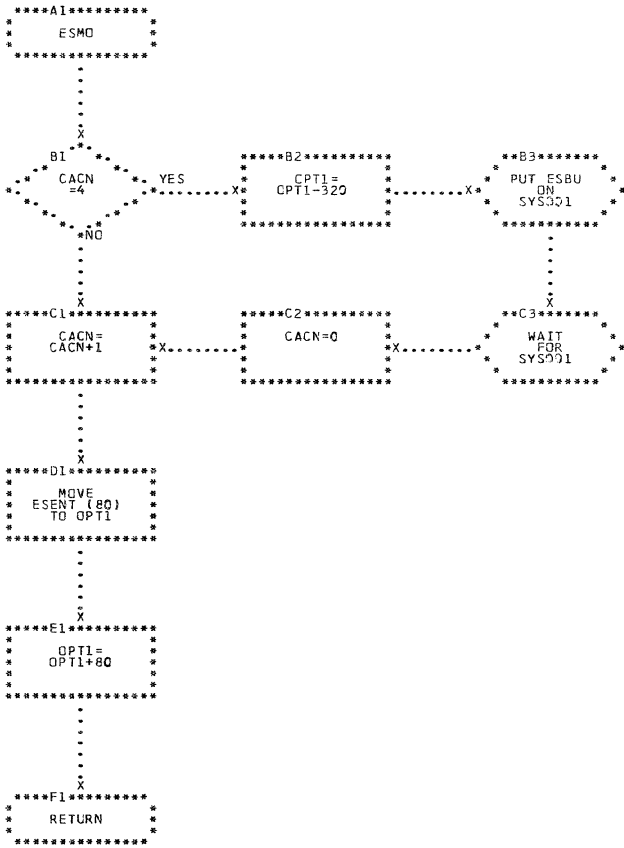
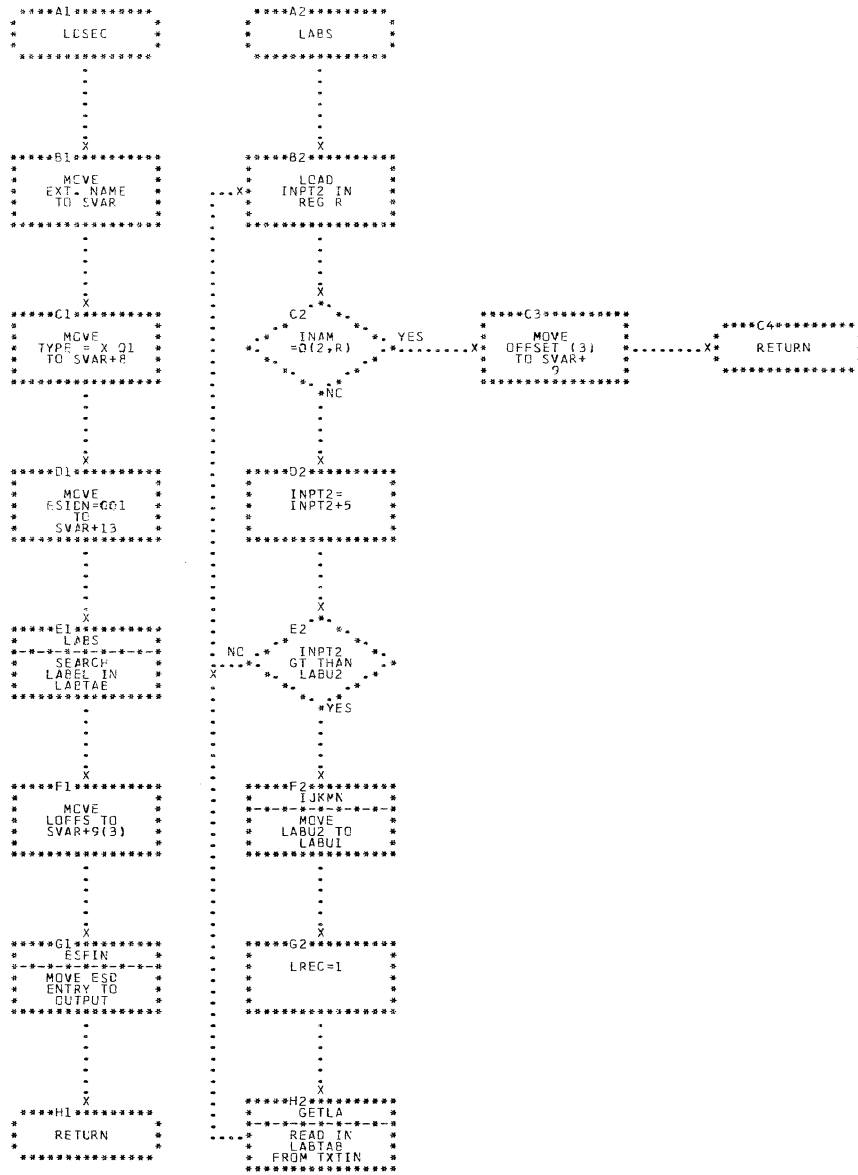


CHART EM. IJXG25 ESFIN

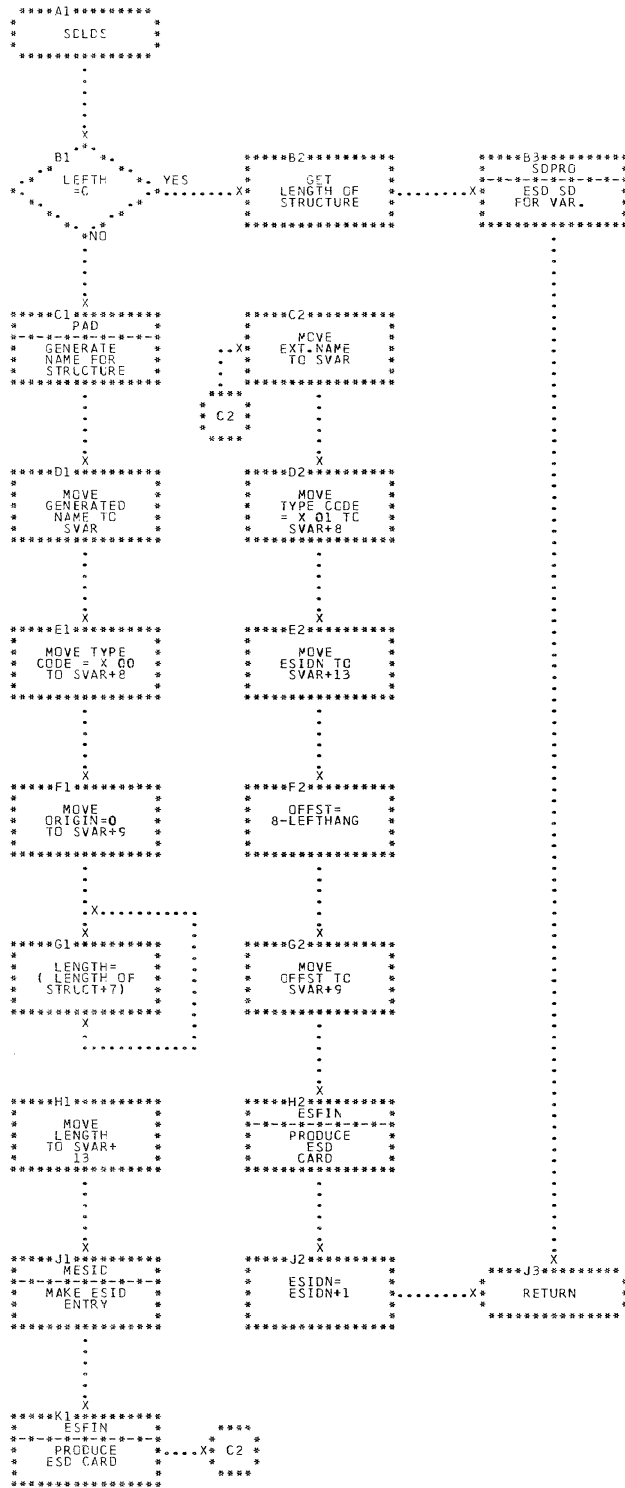


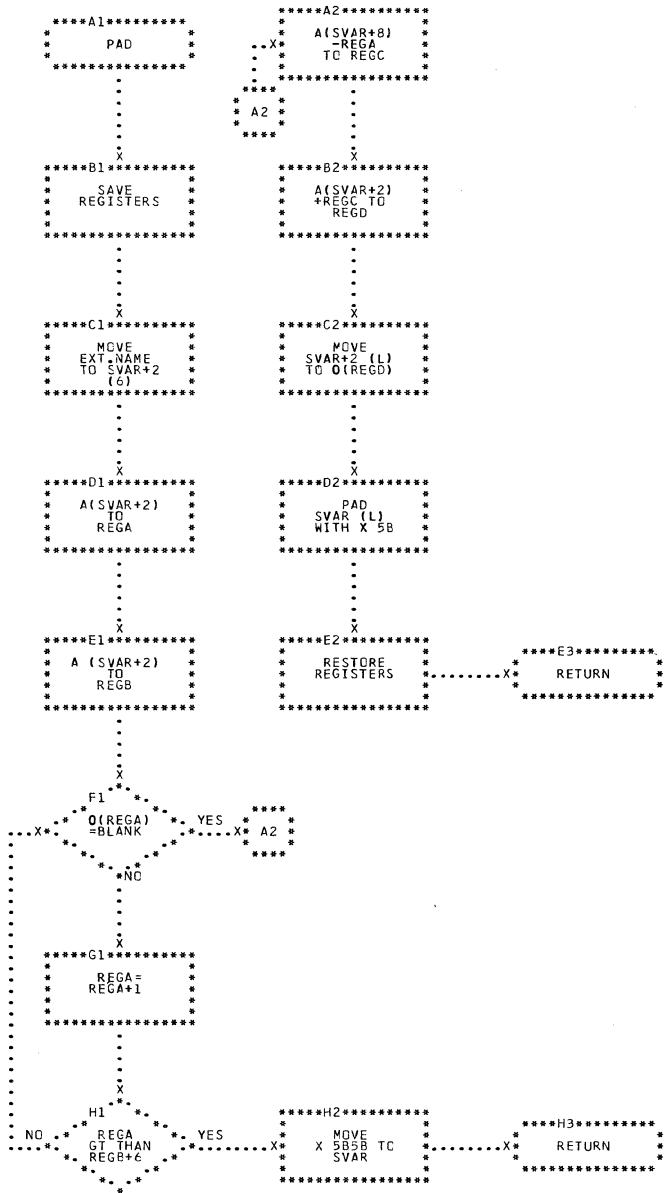


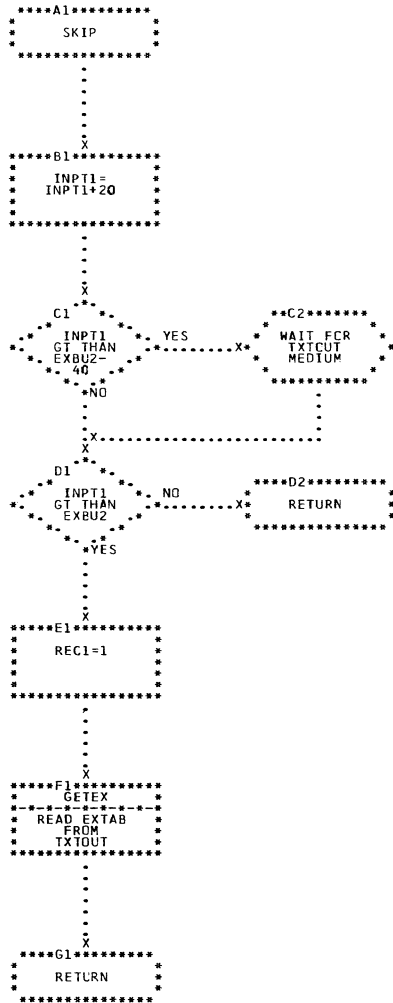
```

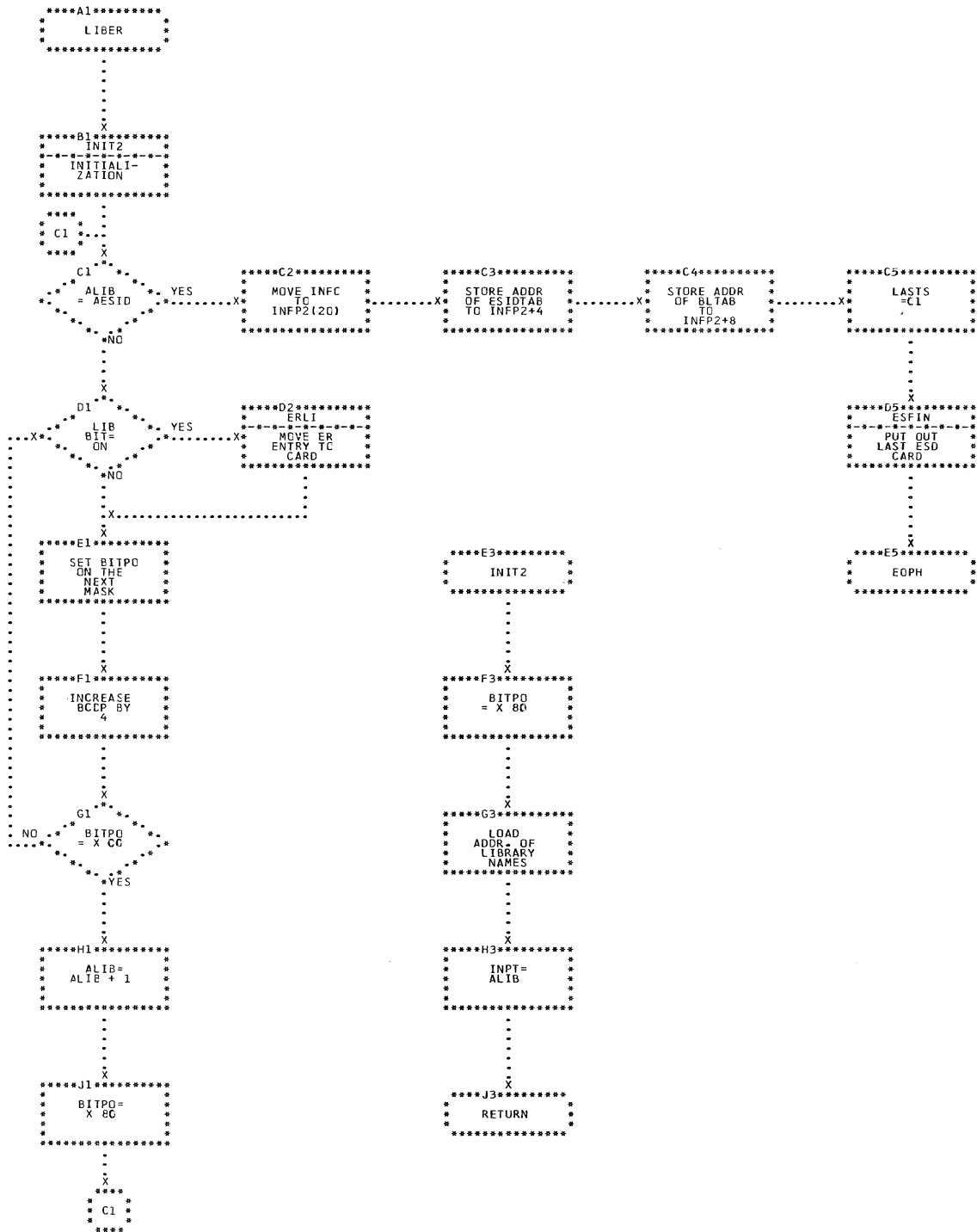
*****A1*****
*   ERCAL   *
*****
*   .       *
*   .       *
*   .       *
*   X       *
*****B1*****
*   MOVE    *
*   EXTERNAL*
*   NAME (8) *
*   TO SVAR *
*****
*   .       *
*   .       *
*   X       *
*****C1*****
*   MOVE    *
*   TYPE=2  *
*   TO      *
*   SVAR+8  *
*****
*   .       *
*   .       *
*   X       *
*****D1*****
*   MOVE    *
*   ORIGIN=0*
*   TO      *
*   SVAR+9  *
*****
*   .       *
*   .       *
*   X       *
*****E1*****
*   MESTIC  *
*   MAKE    *
*   ESTD    *
*   ENTRY   *
*****
*   .       *
*   .       *
*   X       *
*****F1*****
*   ESFIN   *
*   ESD     *
*   CARD    *
*****
*   .       *
*   .       *
*   X       *
*****G1*****
*   ESTDN=  *
*   ESTDN+1 *
*****
*   .       *
*   .       *
*   X       *
*****H1*****
*   RETURN  *
*****

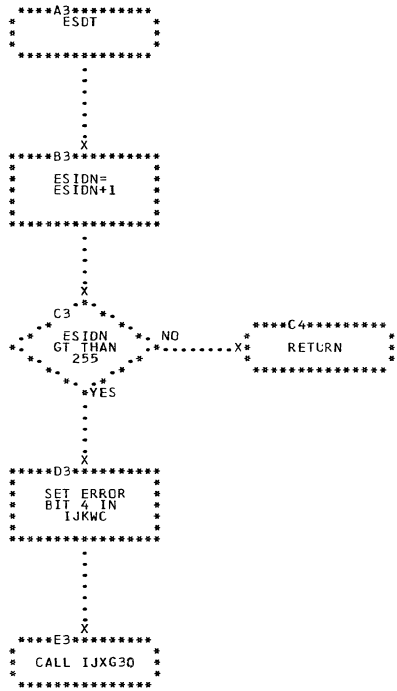
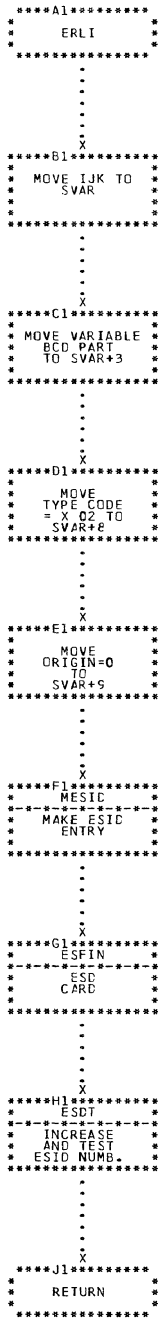
```

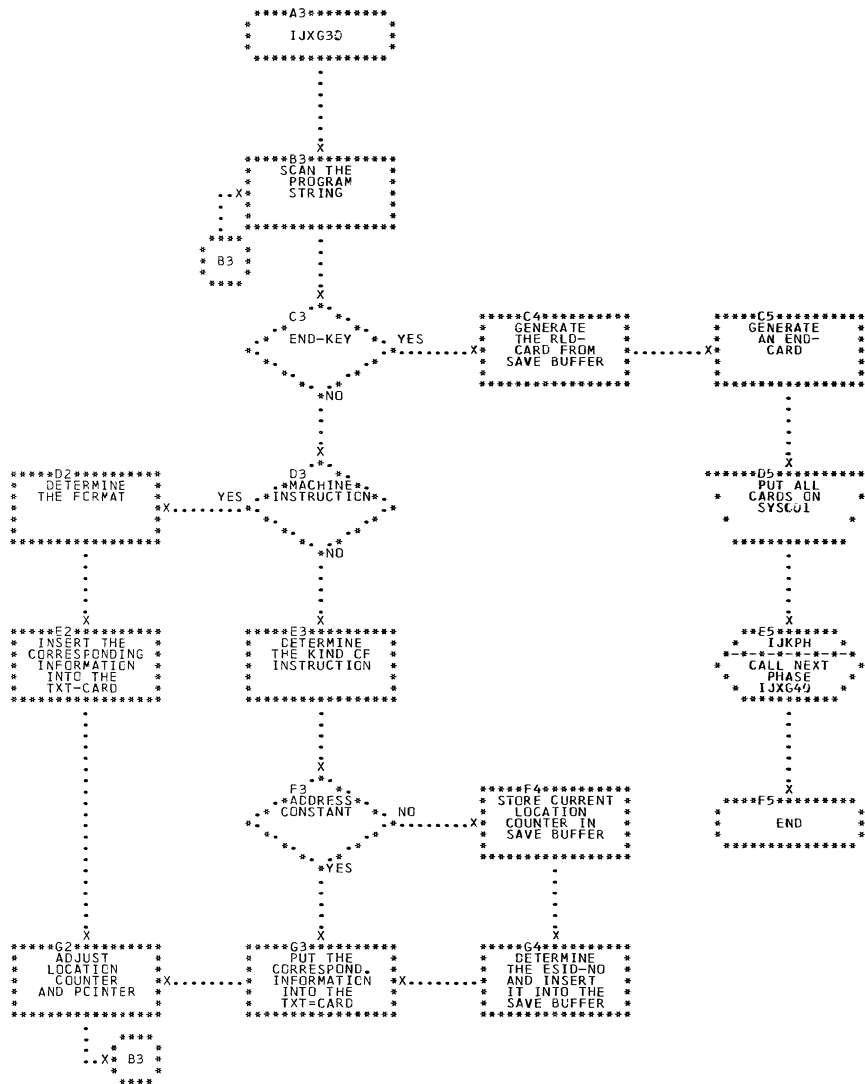



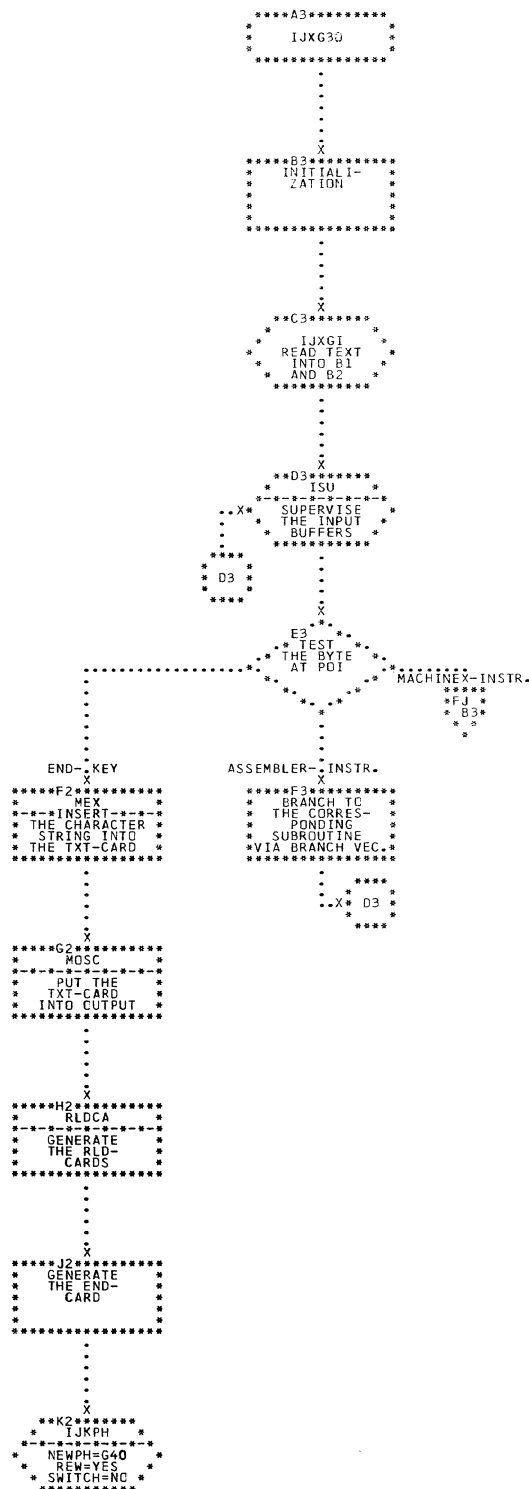













```

*****A1*****
* INRE *
*-----*
* . *
* . *
* . *
* X *
*-----*
*****B1*****
* INSERT THE *
* REGISTERS AT *
* POI+2 AND POI+3 *
* IN ONE BYTE *
* AT POI+2 *
*-----*
* . *
* . *
* . *
* X *
*-----*
*****C1*****
* RETURN *
*-----*

```

```

*****A3*****
* MEX *
*-----*
* . *
* . *
* . *
* X *
*-----*
* B3 *
* DOES *
* THE ELEMENT * NO *
* FIT IN THE *-----*
* CARD * *
*-----*
* . *
* . *
* . *
* YES *
*-----*
* B3 *
*-----*
* . *
* . *
* . *
* X *
*-----*
*****C3*****
* INSERT THE *
* ELEMENT *
* INTO THE *
* CARD *
*-----*
* . *
* . *
* . *
* X *
*-----*
*****D3*****
* RETURN *
*-----*

```

```

*****B4*****
* INSERT AS *
* MUCH OF *
* THE ELEMENT *-----*
* AS POSSIBLE *
*-----*

```

```

*****B5*****
* PGSC *
*-----*
* PUT THE *
* CARD INTO *
* OUTPUT *
*-----*
* . *
* . *
* . *
* X *
*-----*
*****C5*****
* INITIALIZE *
* THE NEXT *
* CARD *
*-----*
* . *
* . *
* . *
* X * B3 *
*-----*

```

```

*****A1*****
MOSC
  
```

```

*****B1*****
INSERT THE
NUMBER OF
INFORMATION
BYTES INTO
THE CARD
  
```

```

*****C1*****
INSERT THE
CARD-
NUMBER
  
```

```

*****D1*****
MOVE THE
CARD INTO
THE CLPUT
BUFFER
  
```

```

*****E1*****
END-CARD
  
```

```

*****F1*****
ZTCUT
PUT LAST
CARD-RECORD
ON SYSC01
  
```

```

*****G1*****
RETURN
  
```

```

*****D2*****
RETURN
  
```

```

*****E2*****
RECORD FULL
  
```

```

*****F2*****
ZTCUT
PUT CARD-
RECORD ON
SYSC01
  
```

```

*****G2*****
RETURN
  
```

```

*****A4*****
MOKK
  
```

```

*****B4*****
REST
OF SAVE-
BUFFER GT 4
  
```

```

*****C4*****
FILL THE
SAVE BUFFER
  
```

```

*****D4*****
IJKPO
PUT T85
RECORD ON
XTIOUT
  
```

```

*****E4*****
MOVE THE
REST OF THE
4 BYTES TO
THE BEGIN OF
THE SAVEBUFFER
  
```

```

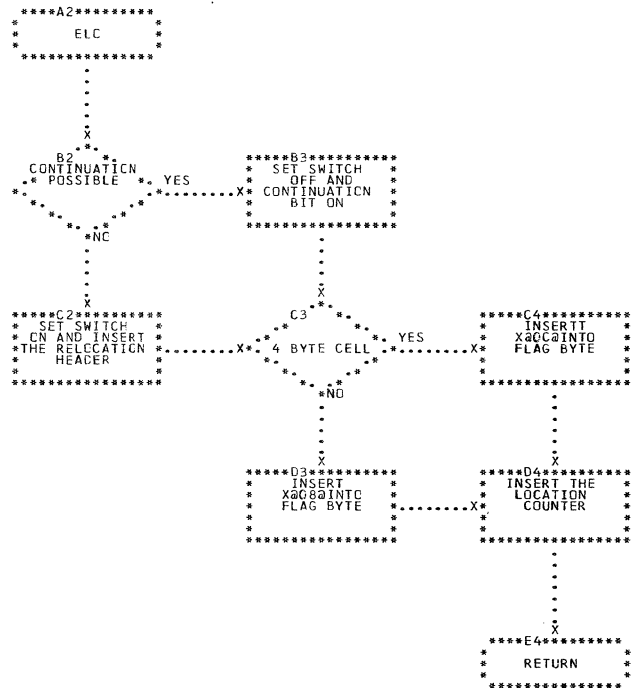
*****F4*****
RETURN
  
```

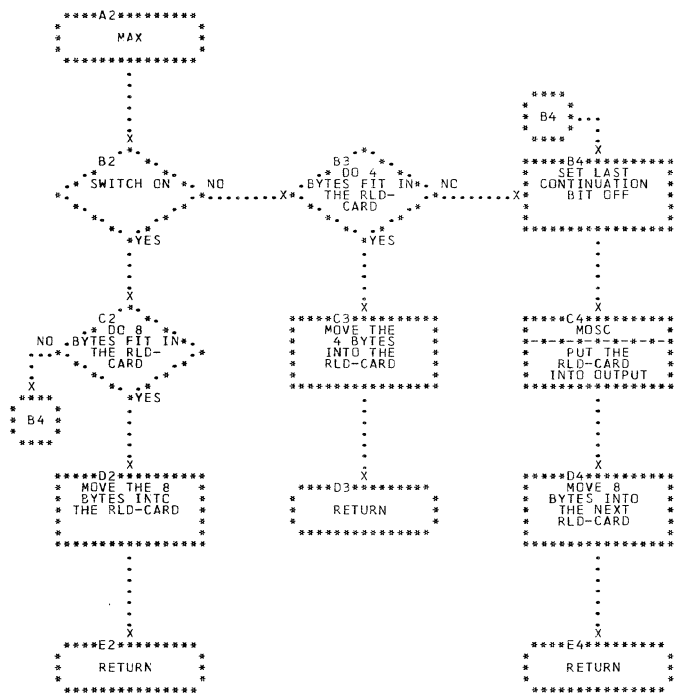
```

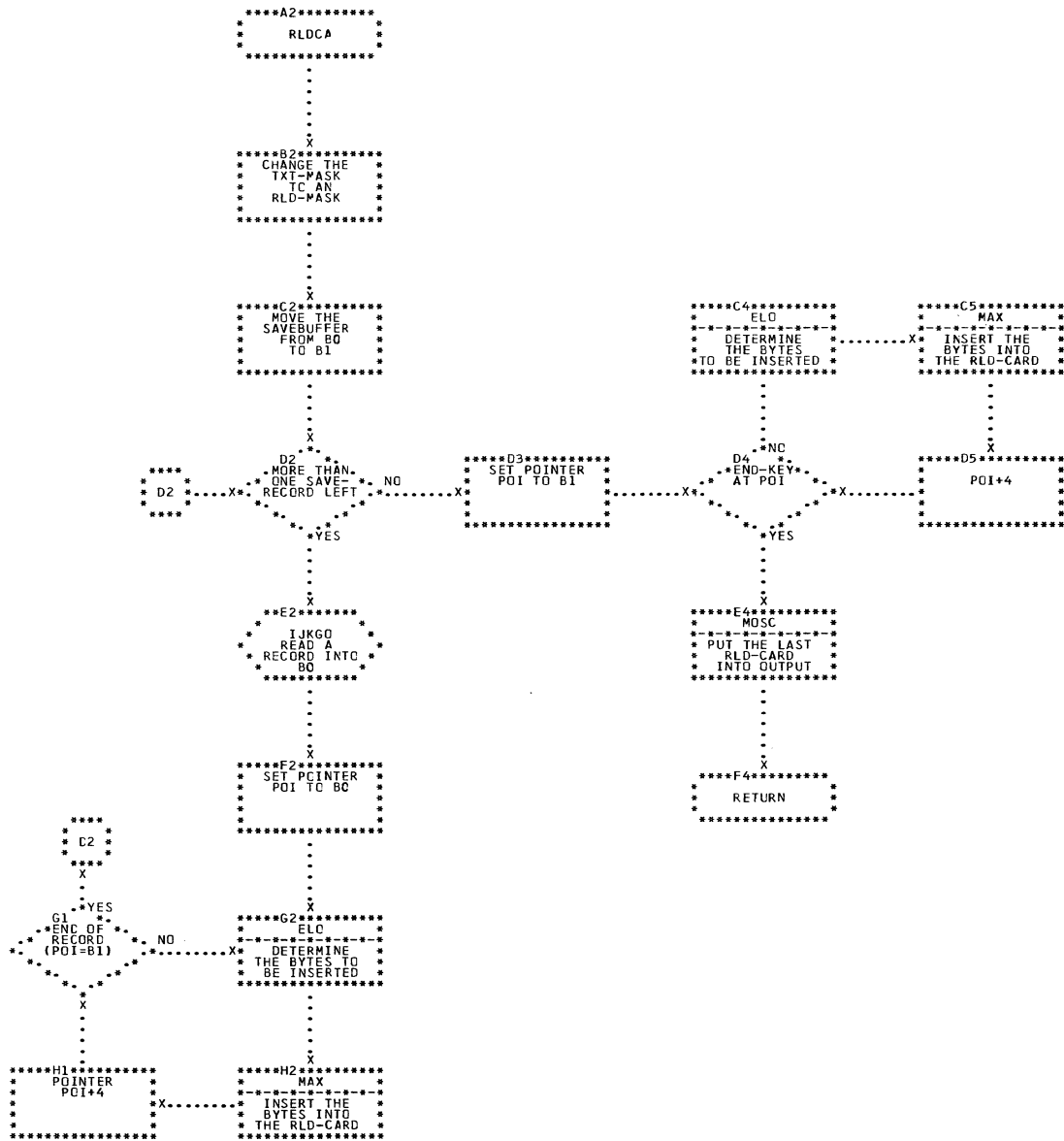
*****B5*****
MOVE THE 4
BYTES(LDC-
NO) INTO THE
SAVE BUFFER
  
```

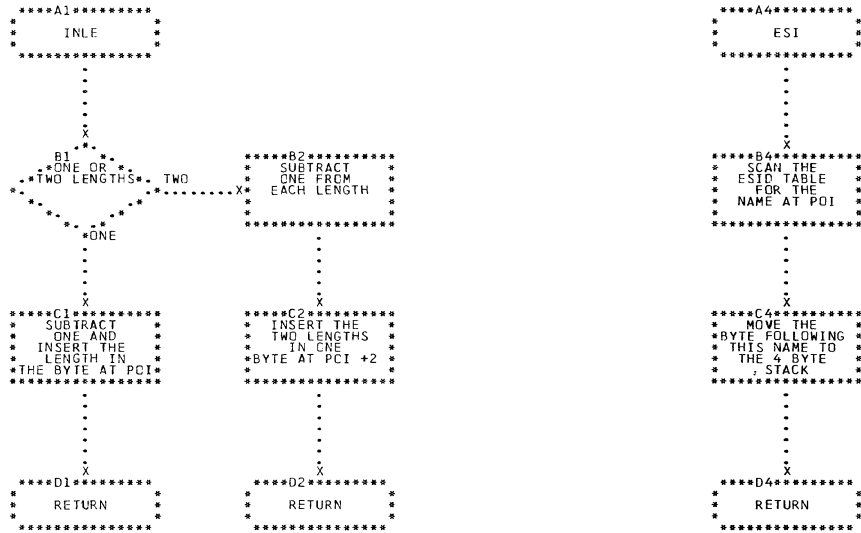
```

*****C5*****
RETURN
  
```









```
*****A1*****
* DCAL3 *
*****
```

```
*****B1*****
* STORE THE LO- *
* CATION COUNTER *
* INTO THE FIRST *
* 3 BYTES OF *
* 4-B. STACK *
*****
```

```
*****C1*****
* DCVL3 *
* NO *
* YES *
```

```
*****D1*****
* EST *
* --FETCH-THE *
* ESID-NO FROM *
* ESID-TABLE *
*****
```

```
*****C2*****
* MCVE 1 INTO *
* THE RIGHTMOST *
* BYTE OF THE *
* 4 BYTE-STACK *
* (AS ESID-NC) *
*****
```

```
*****D2*****
* MCKK *
* MCVE THE 4 *
* BYTES INTO *
* SAVE BUFFER *
*****
```

```
*****E2*****
* MEX *
* INSERT THE *
* OFFSET INTO *
* THE TXT-CARD *
*****
```

```
*****F2*****
* LCI+3 *
* POI+4 *
*****
```

```
*****G2*****
* RETURN *
*****
```

```
*****A4*****
* DCX *
*****
```

```
*****B4*****
* MEX *
* INSERT THE *
* CONSTANT INTO *
* THE TXT-CARD *
*****
```

```
*****C4*****
* POI+4+L *
* LCI+L *
*****
```

```
*****D4*****
* RETURN *
*****
```


*****A1*****
DSL

*****B1*****
LOC1+L

*****C1*****
L GT 8

*****D1*****
MOVC
PLT CURRENT
TXT-CARD
INTO OUTPUT

*****E1*****
INSERT LOC1
AS NEW START
ADDRESS INTO
THE TXT-MASK

*****F1*****
POI+4

*****G1*****
RETURN

*****D2*****
REST
OF THE CARD
GT LENGTH
L

*****E2*****
INSERT L
ZERCS INTO
THE TXT-CARD

*****F2*****
POI+4

*****G2*****
RETURN

*****A3*****
LABEL

*****B3*****
POI+4

*****C3*****
RETURN

*****A5*****
PRCCE

*****B5*****
POI+6

*****C5*****
RETURN

*****E4*****
DCF

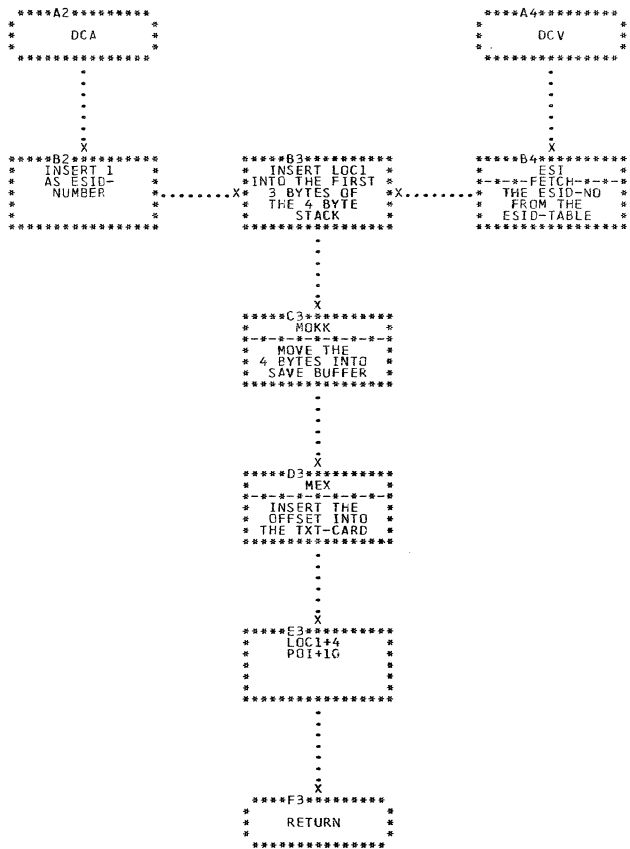
*****F4*****
TAKE THE
LENGTH OF
DSA FROM
THE BLOCK-TABLE

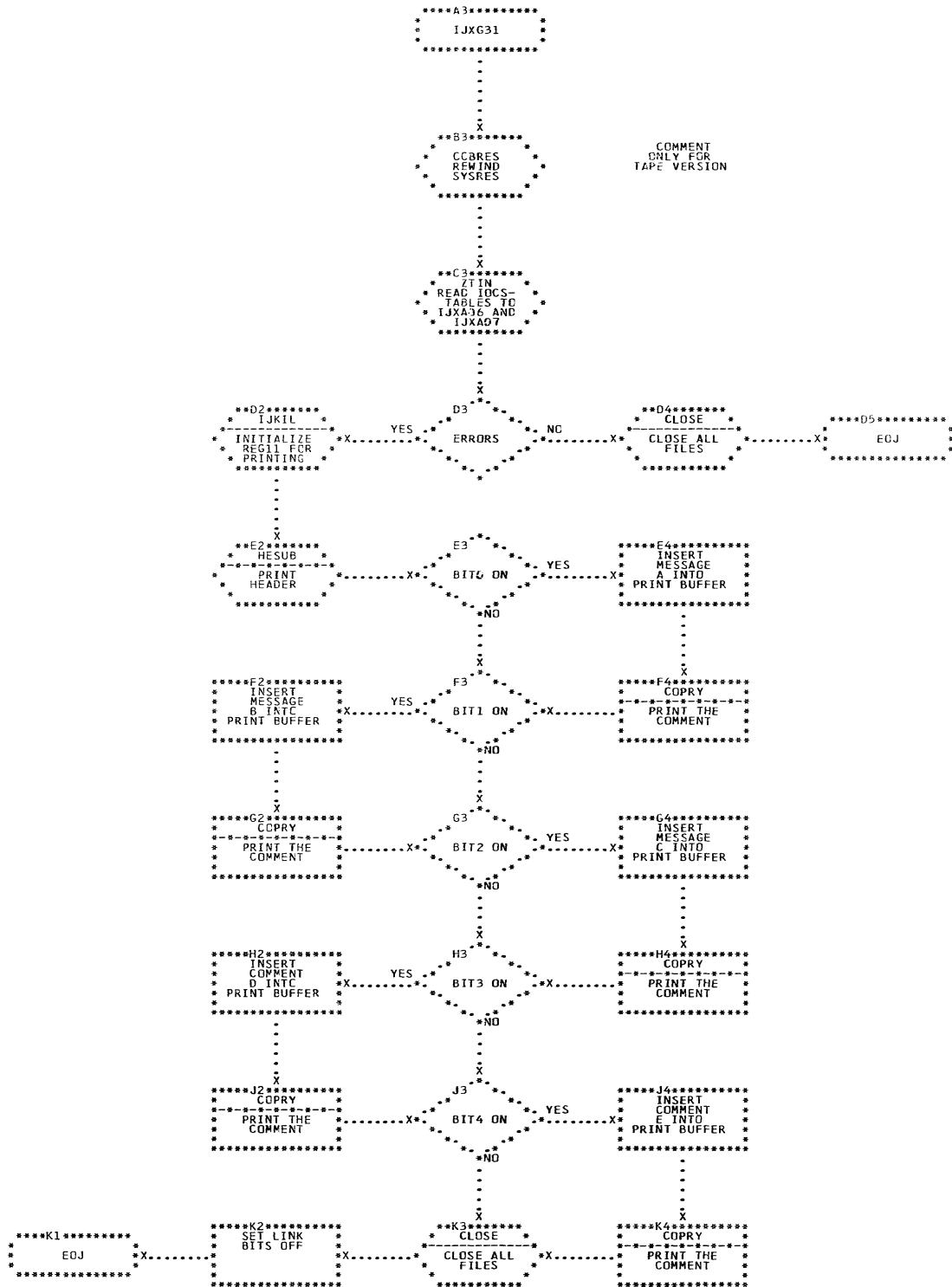
*****G4*****
EXPAND THE
HW TO A
FULLWORD

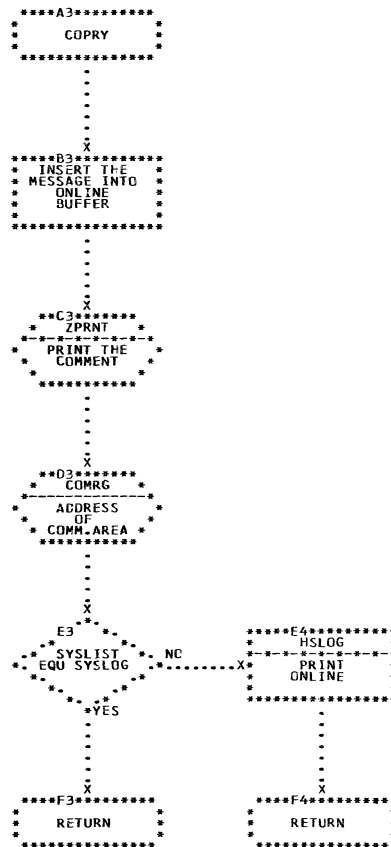
*****H4*****
MEX
INSERT THIS
CONSTANT INTO
THE TXT-CARD

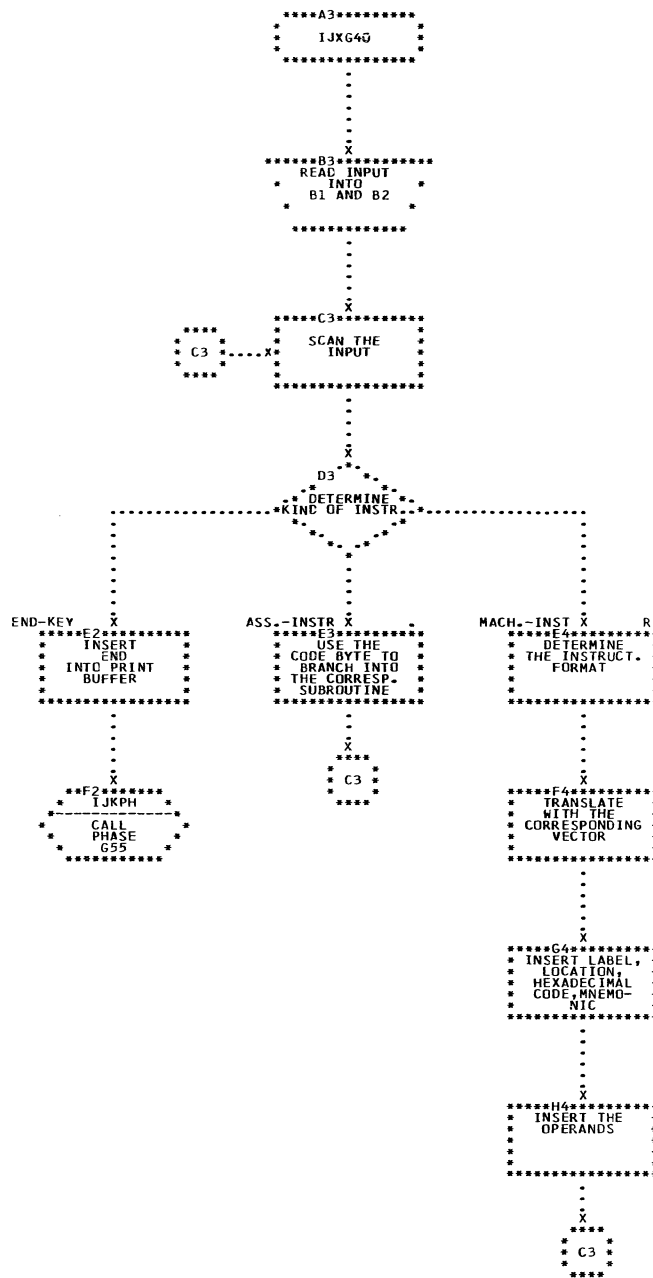
*****J4*****
POI+4
LOC1+4

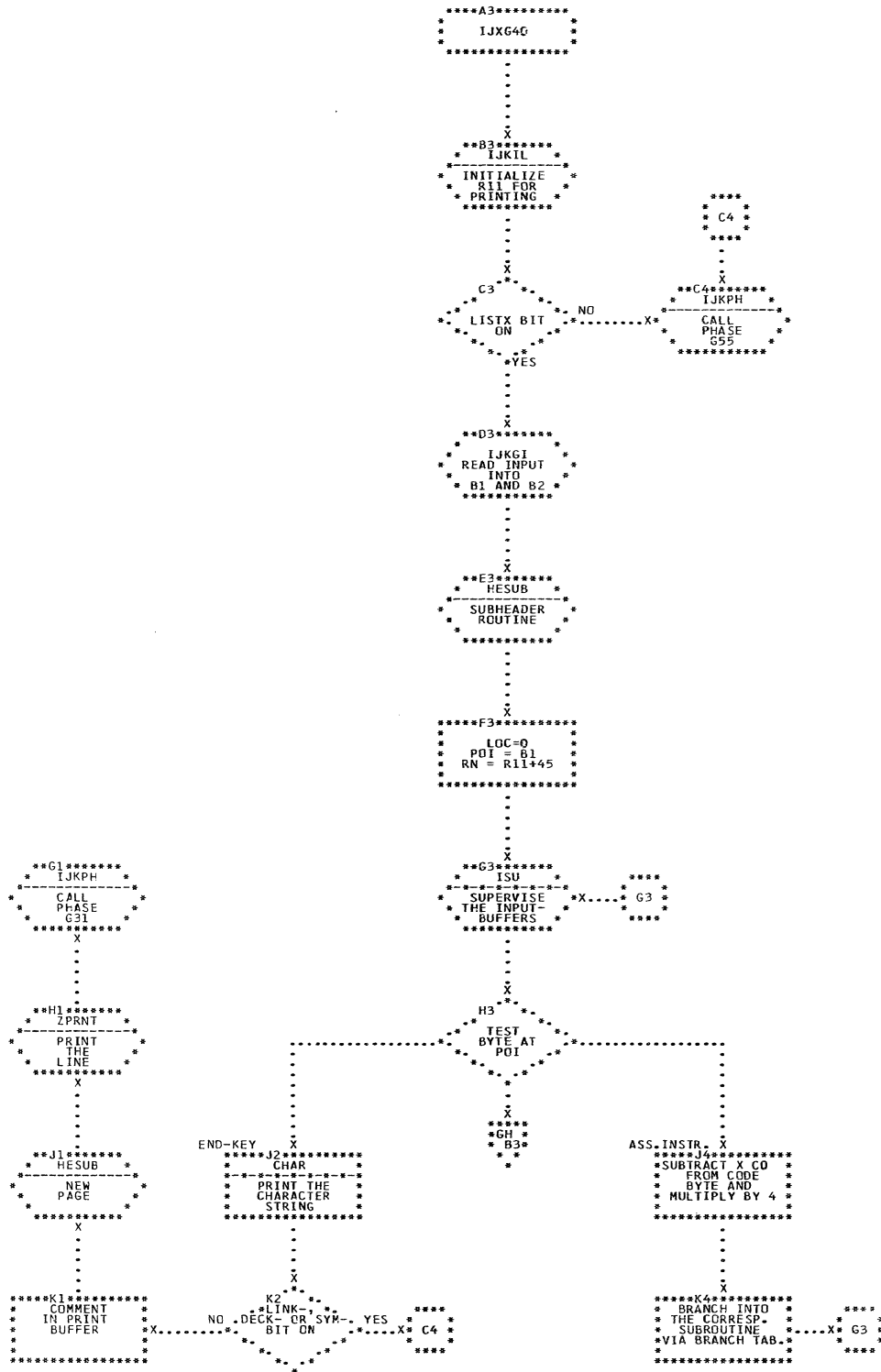
*****K4*****
RETURN

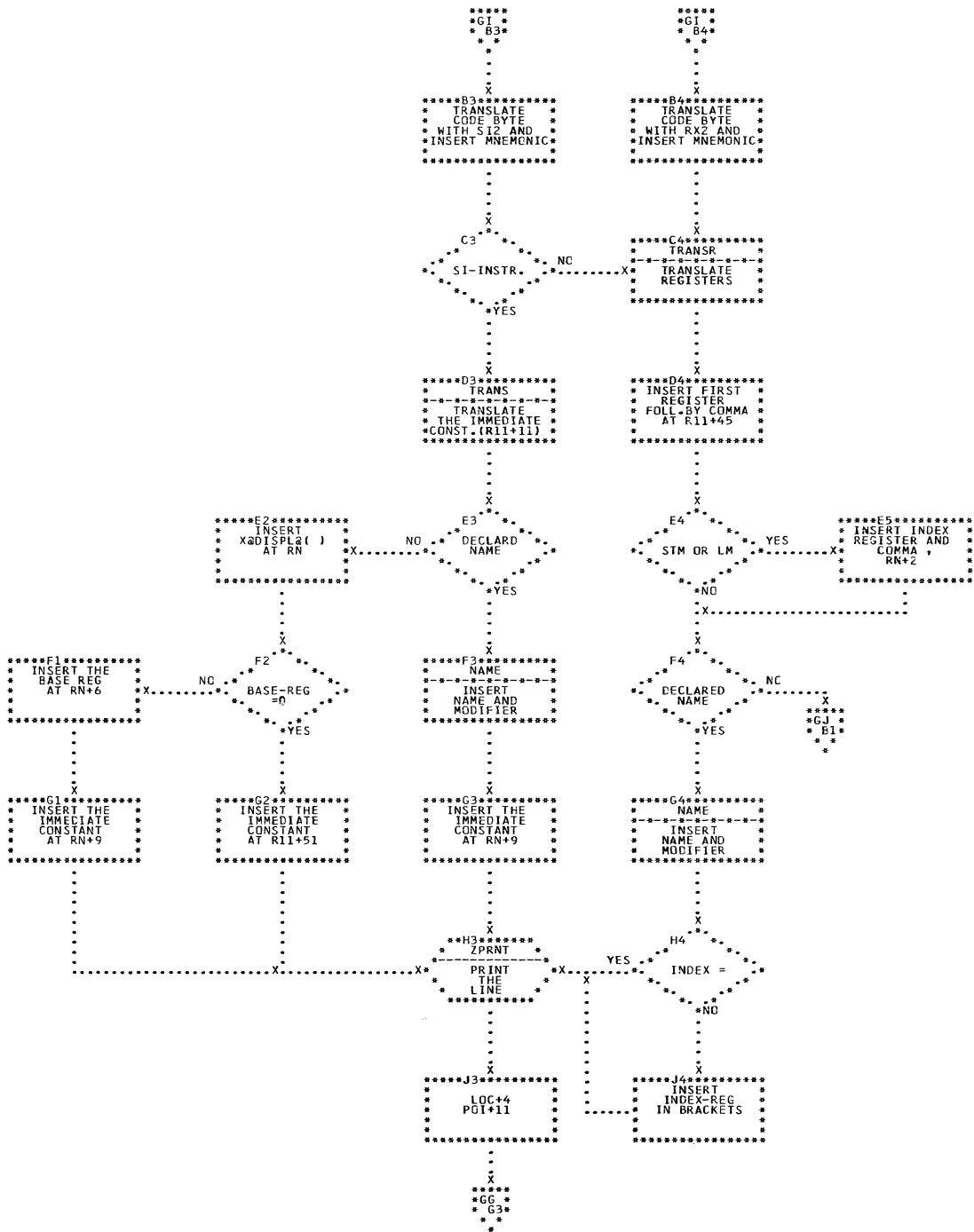


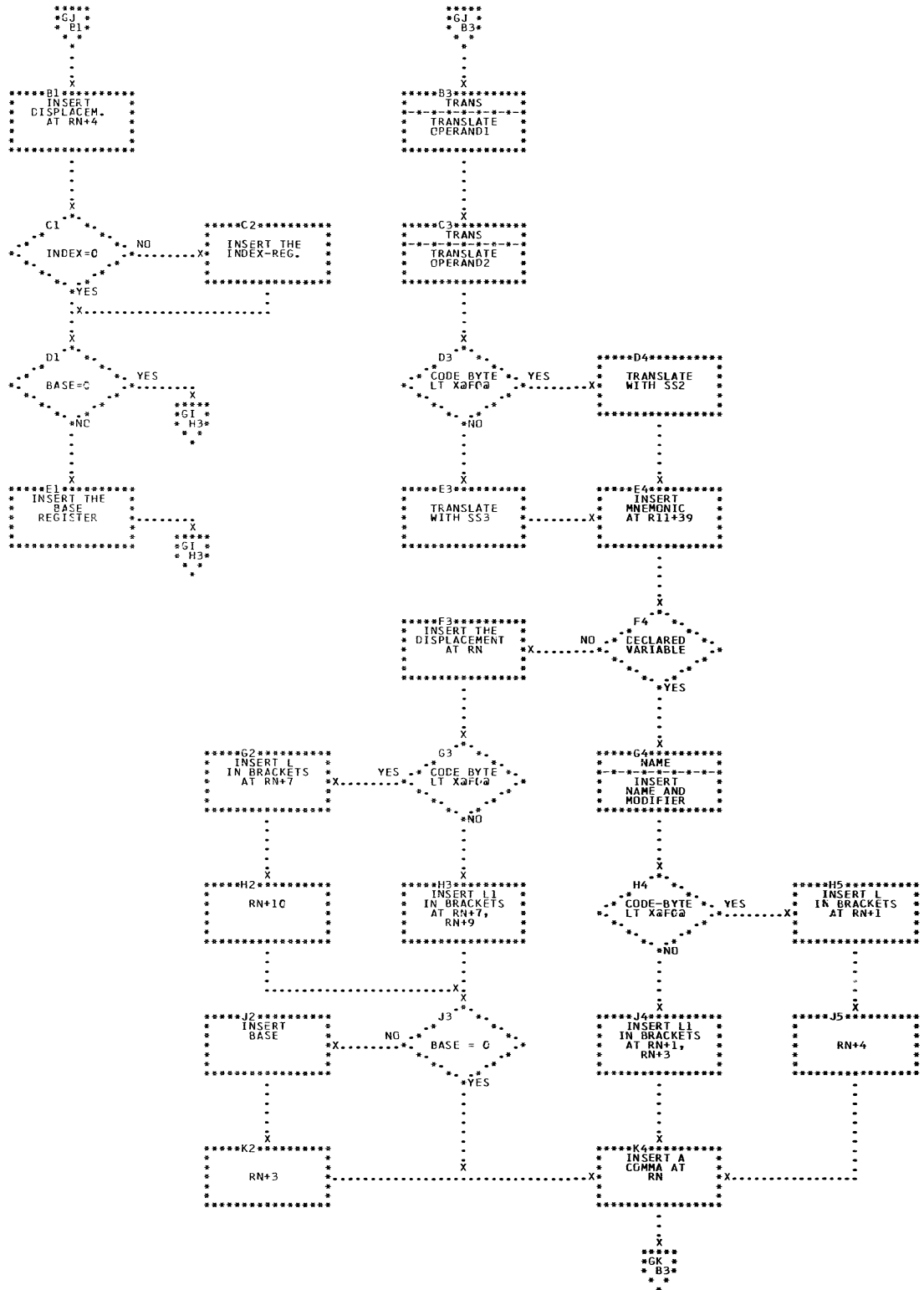


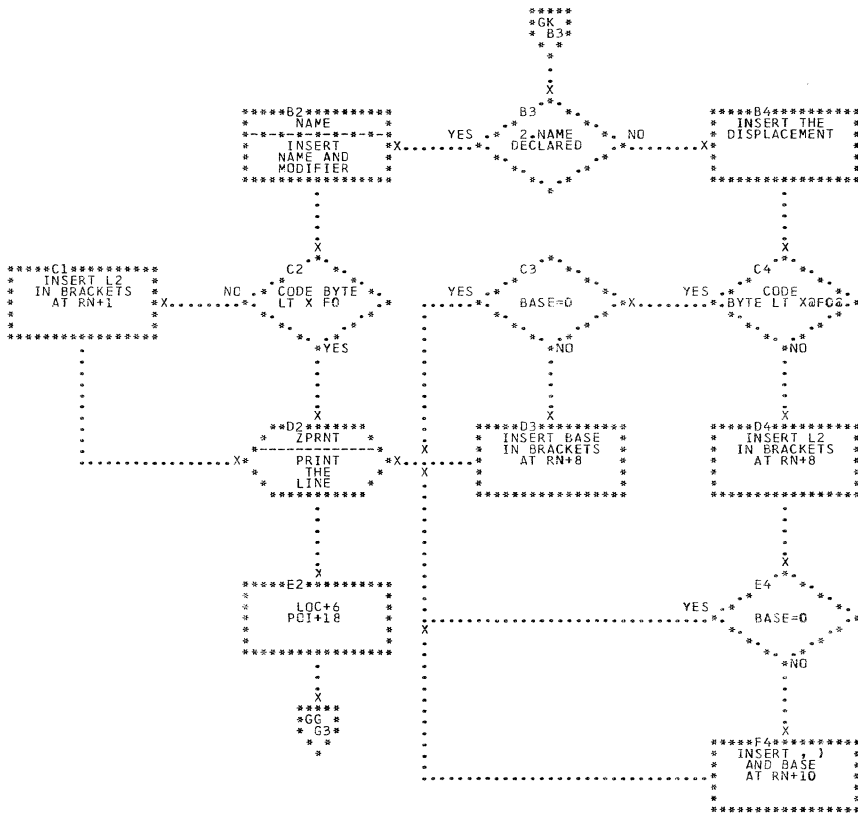


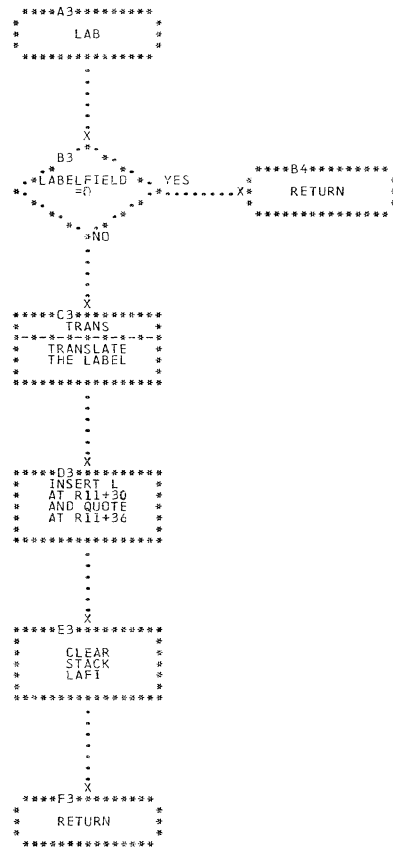
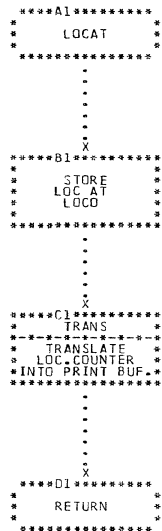


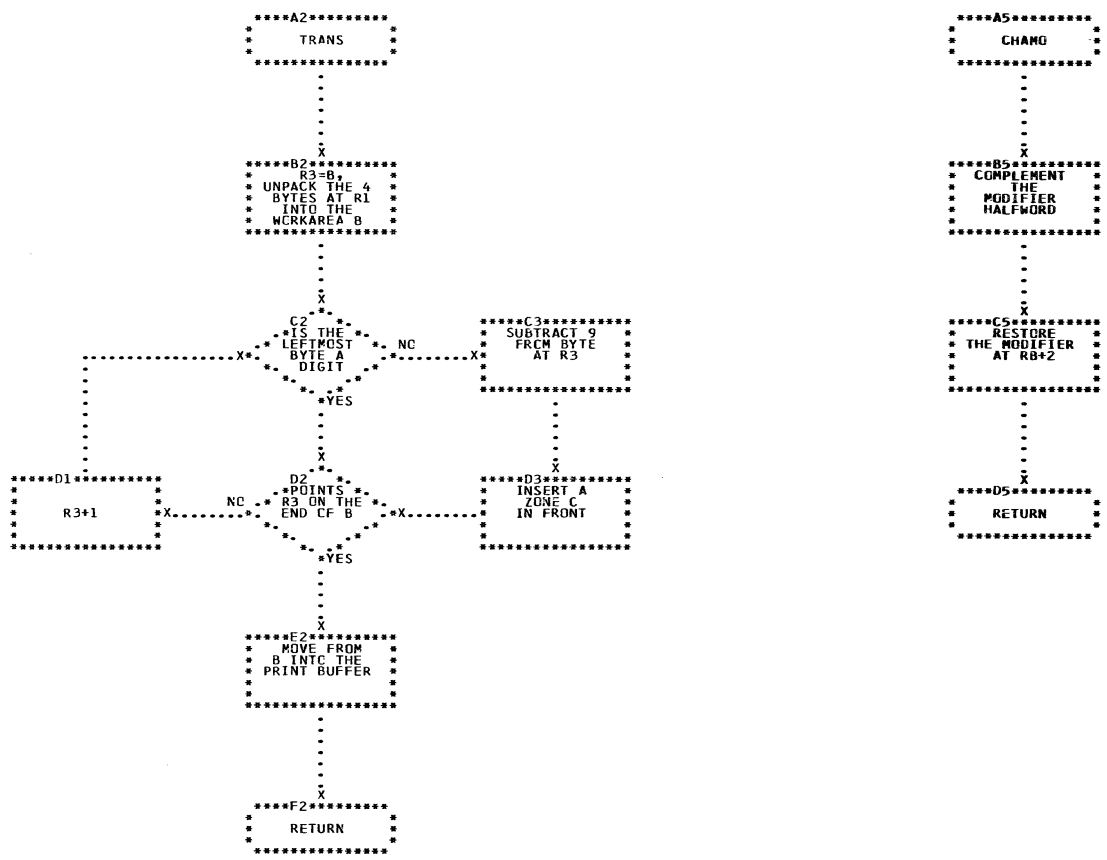


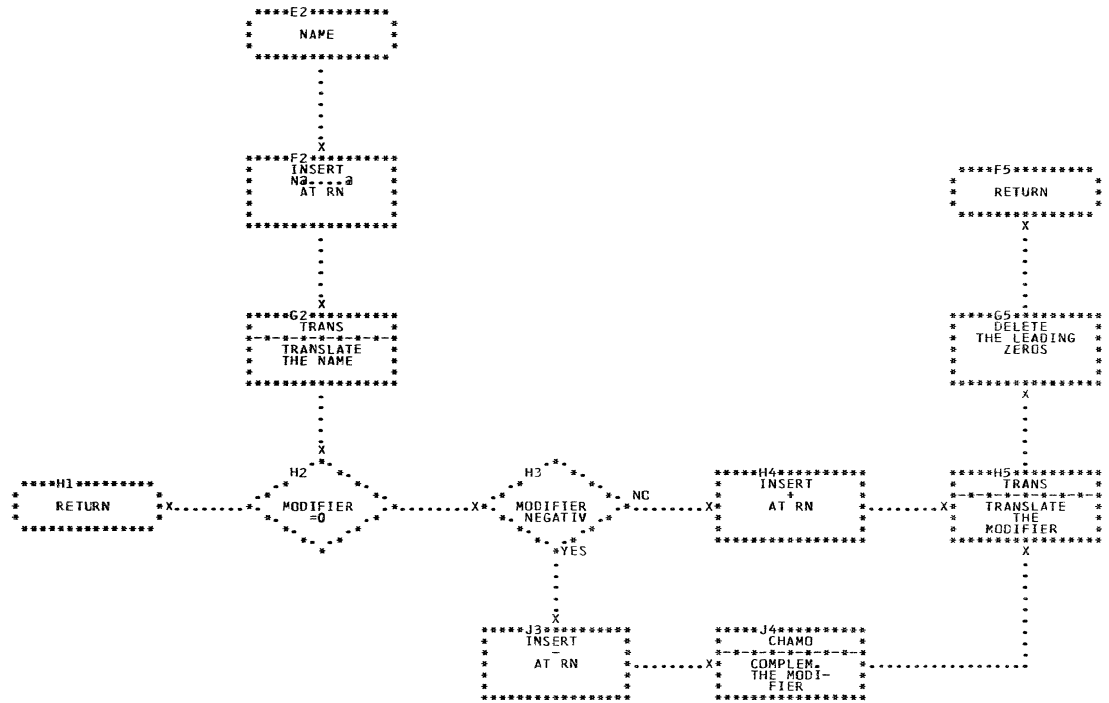
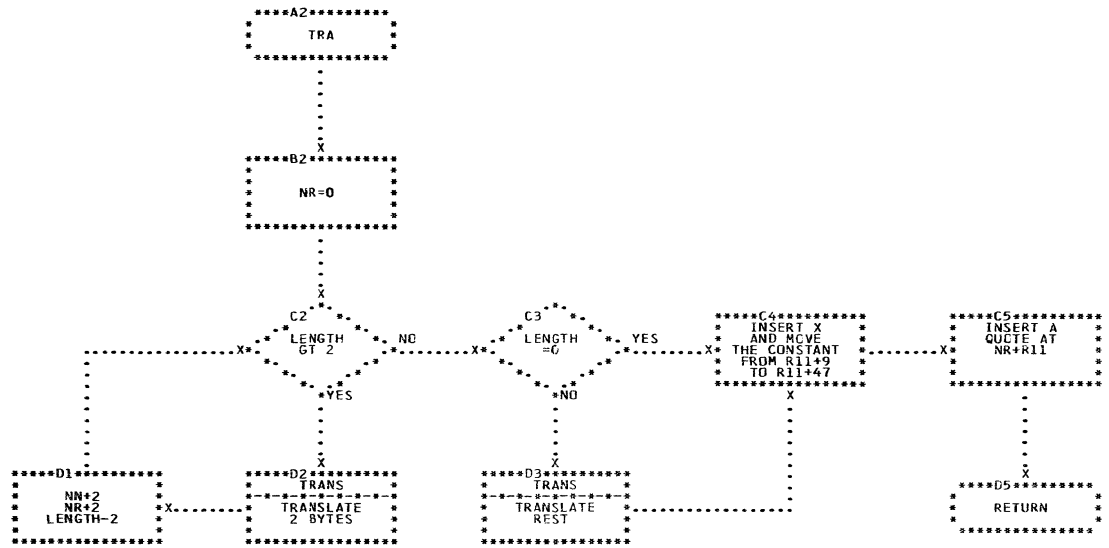


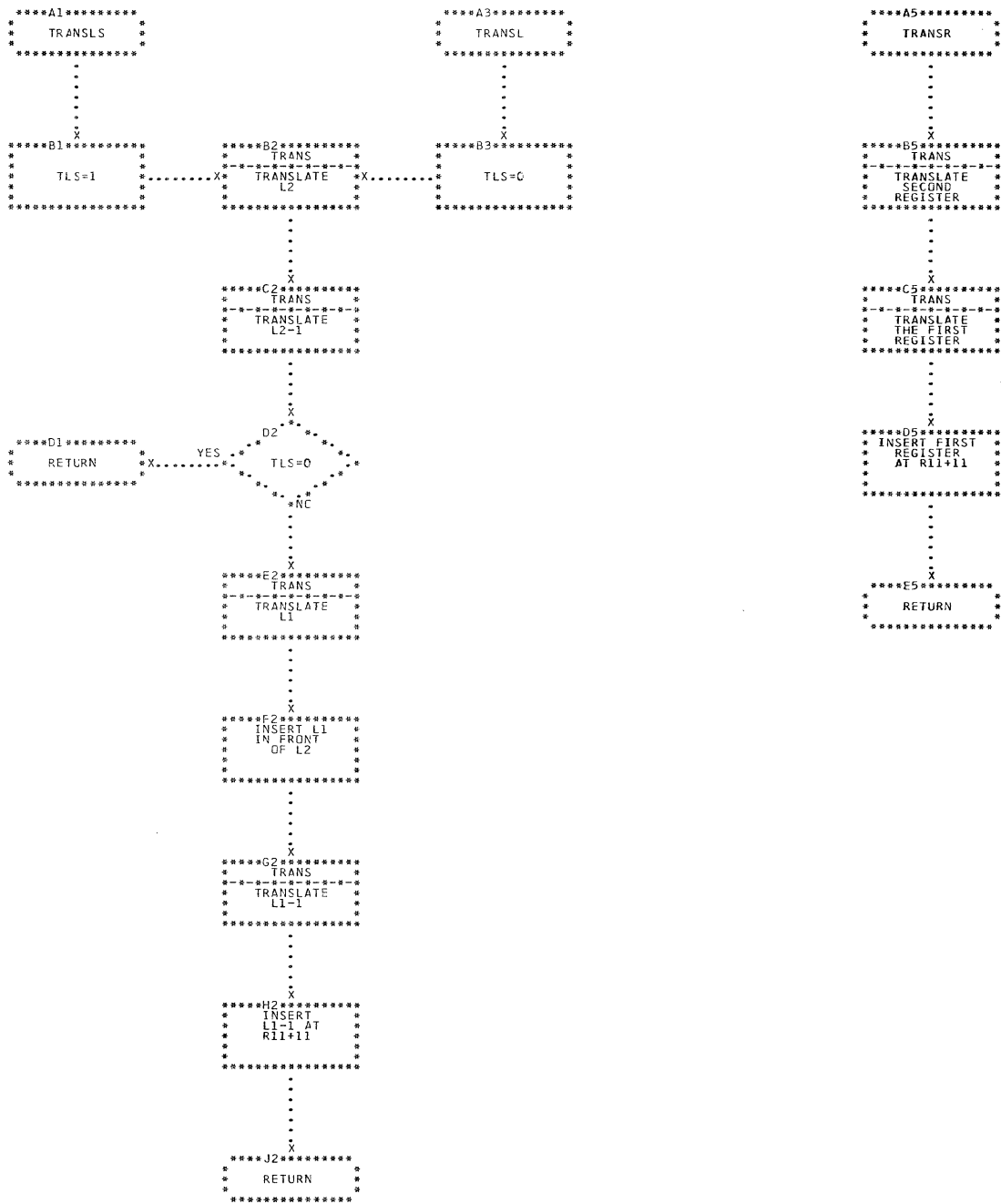


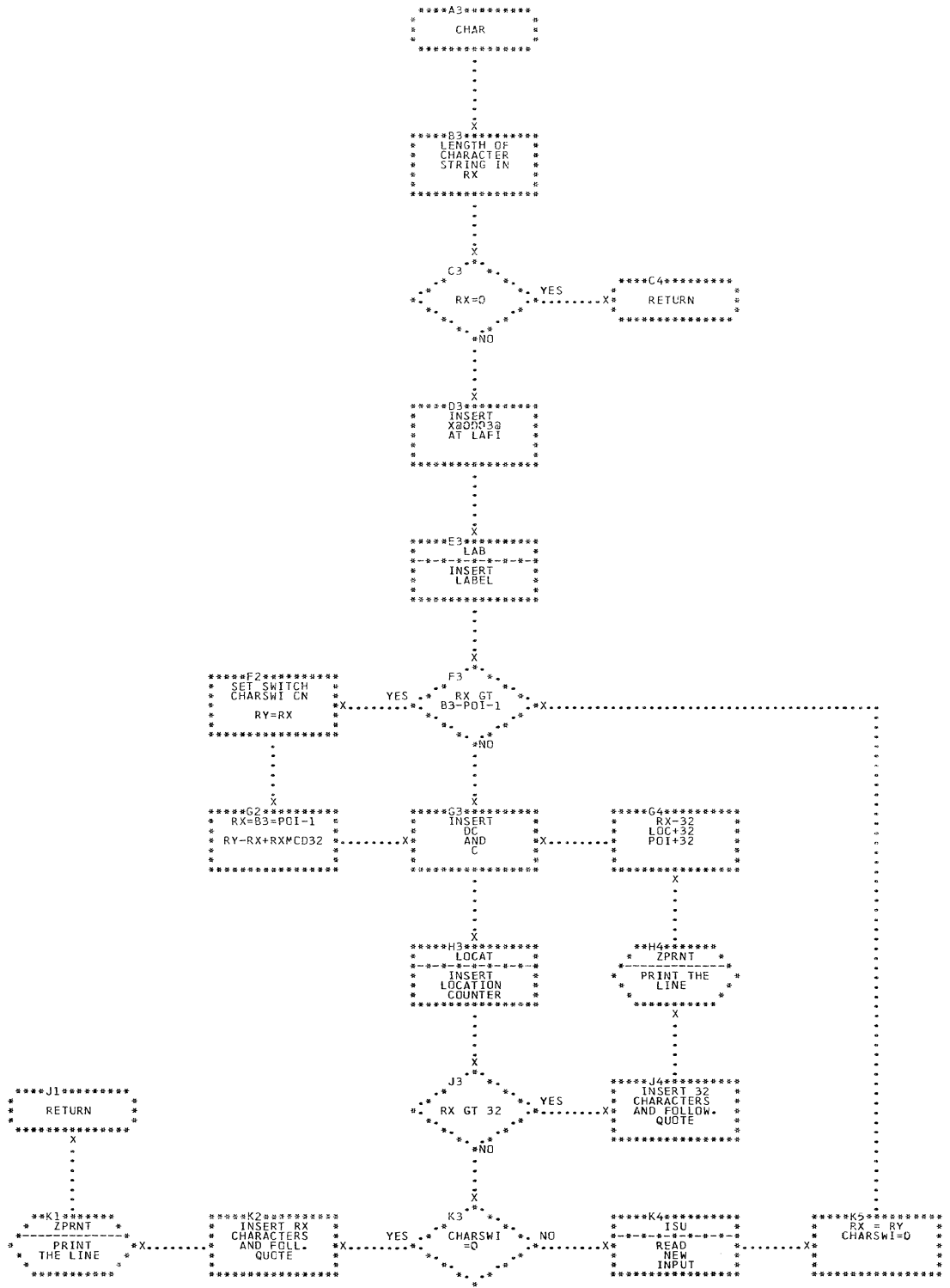


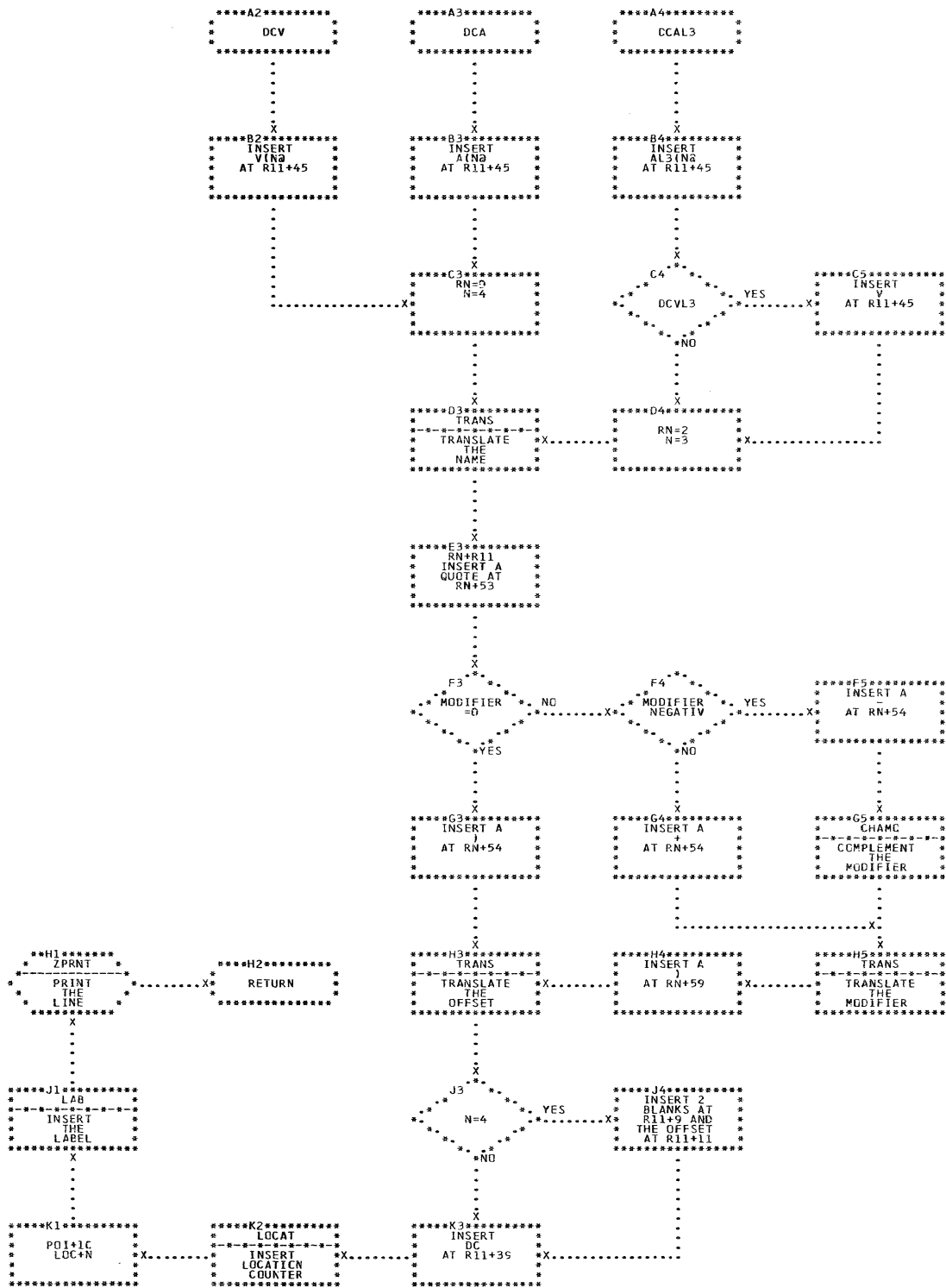












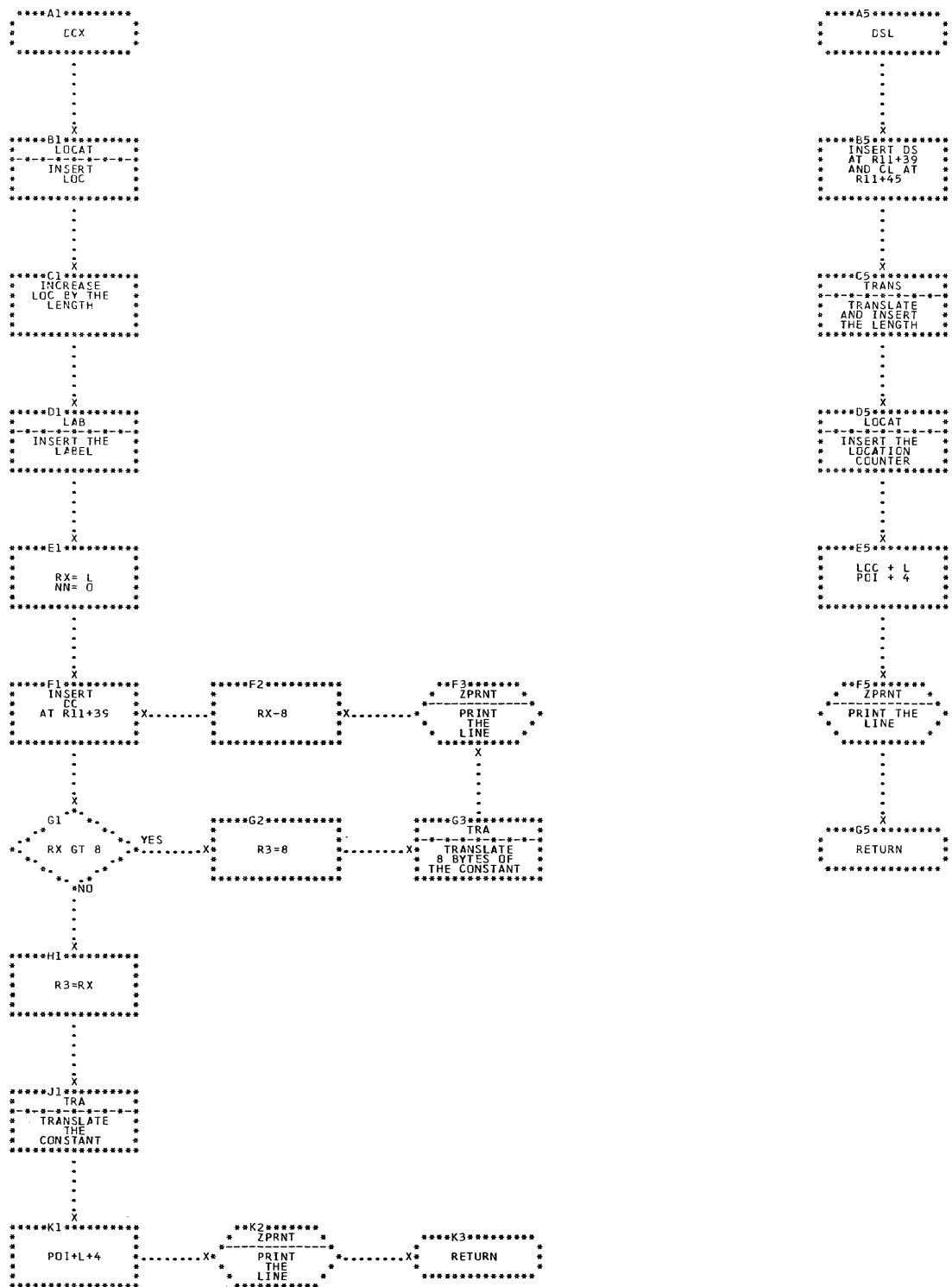
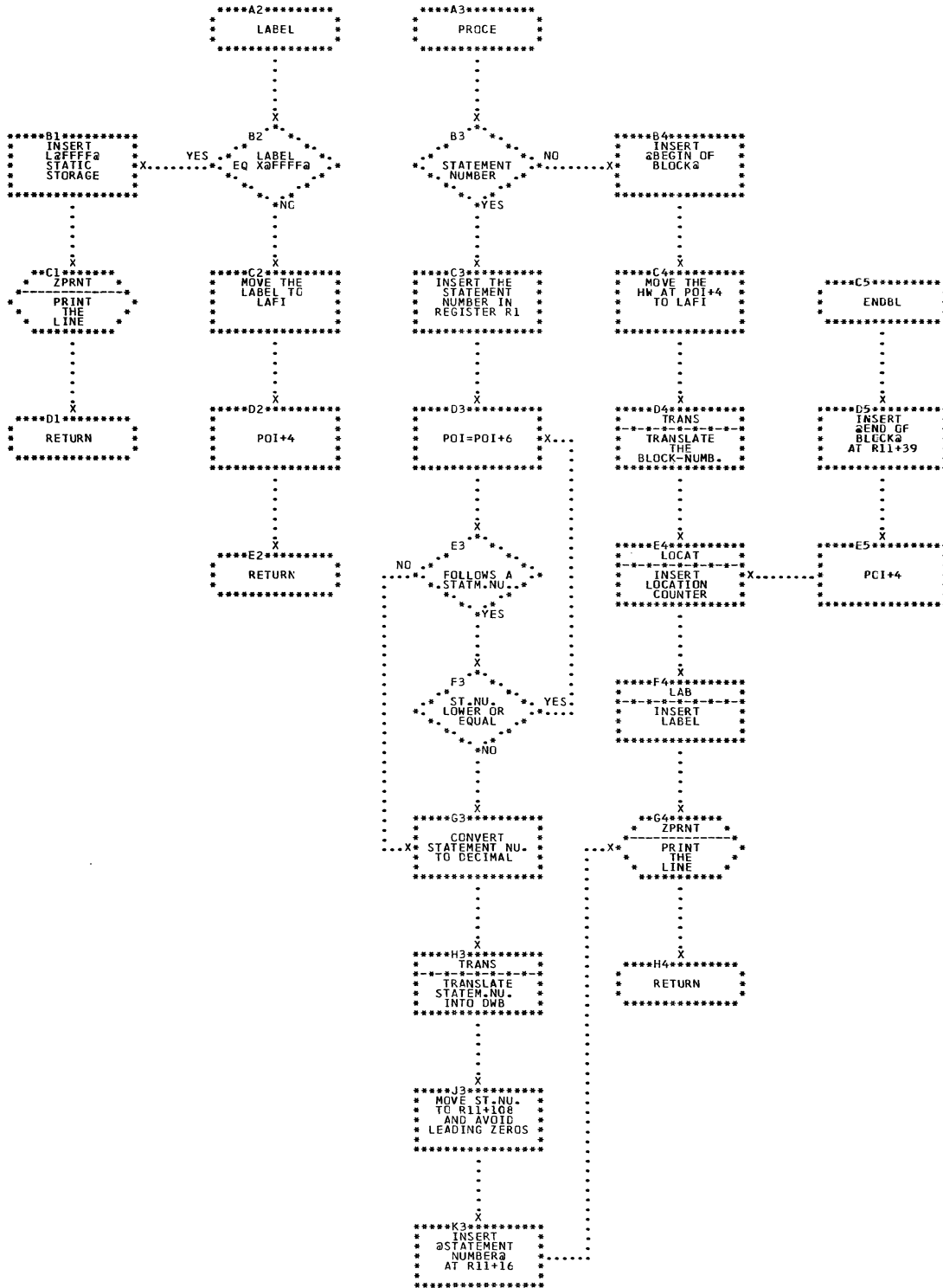


CHART GR. IJXG40

DCX,DSL



.....A3.....
DCF
.....

.....B3.....
EVALUATE THE
ADDRESS OF
THE ENTRY
IN THE BLOCK-
TABLE
.....

.....C3.....
TRANS
TRANSLATE
THE LENGTH
OF DSA
.....

.....D3.....
INSERT
LEADING
ZEROS
AT R11+9
.....

.....E3.....
INSERT DC
AT R11+39
.....

.....F3.....
INSERT
LENGTH OF
BLOCK
AT R11+45
.....

.....G3.....
TRANS
TRANSLATE
BLOCK-NO
.....

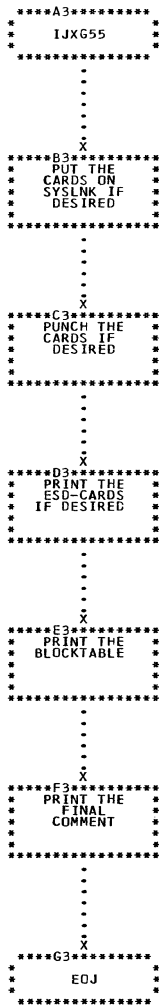
.....H3.....
LOCAT
INSERT
LOCATION
COUNTER
.....

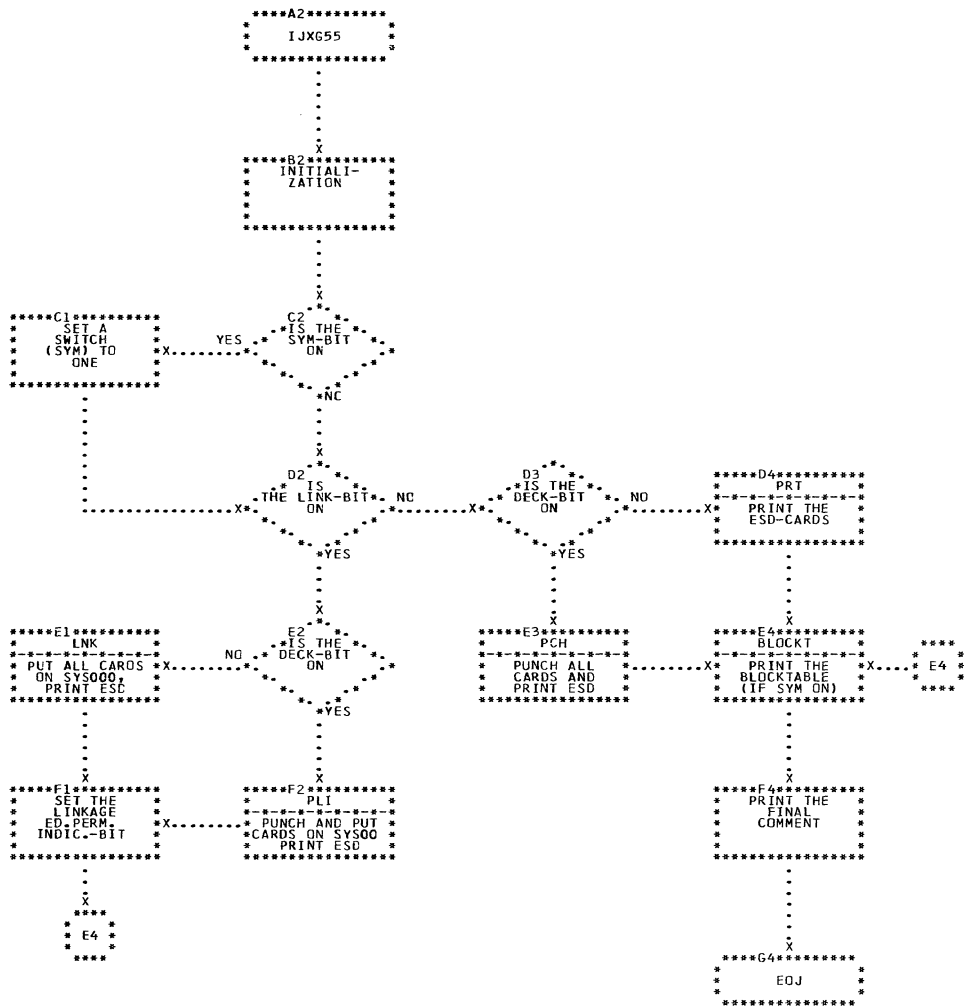
.....J3.....
POI+4
LOC+4
.....

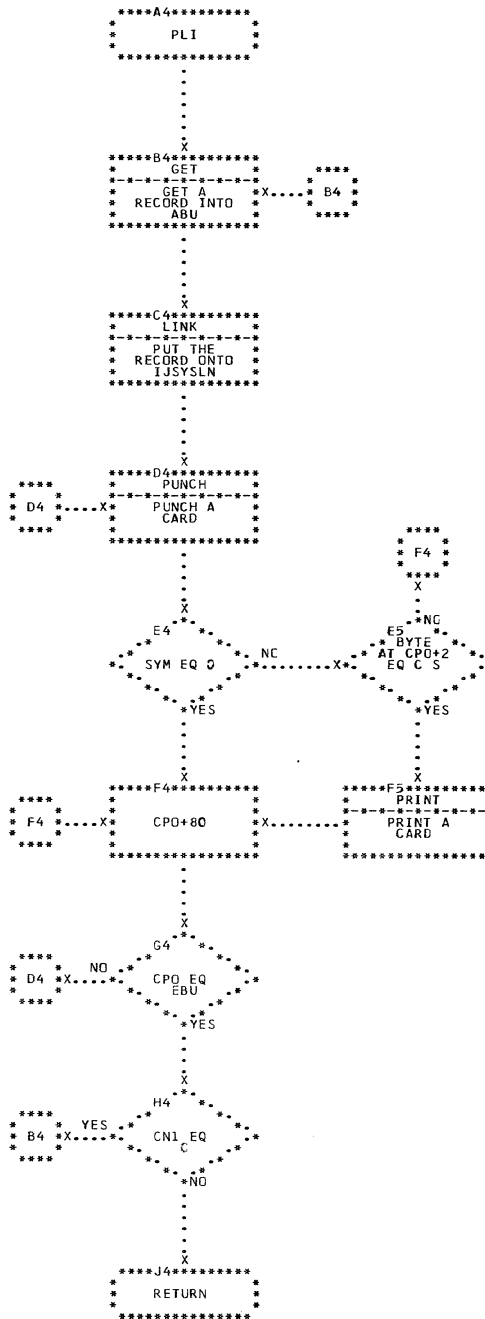
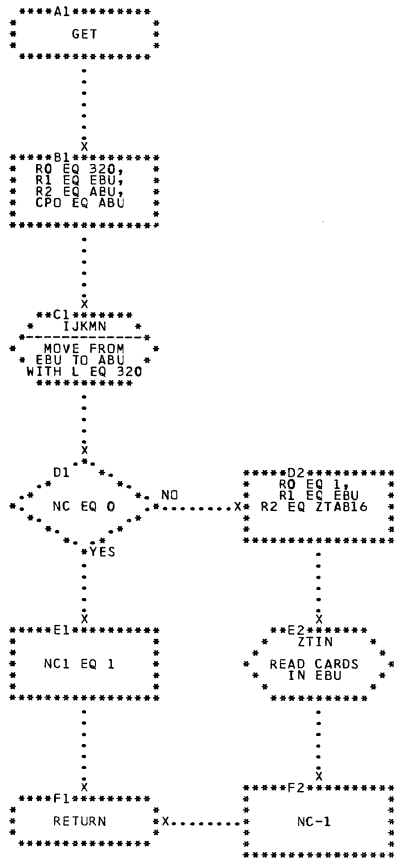
.....K3.....
LAB
INSERT
LABEL
.....

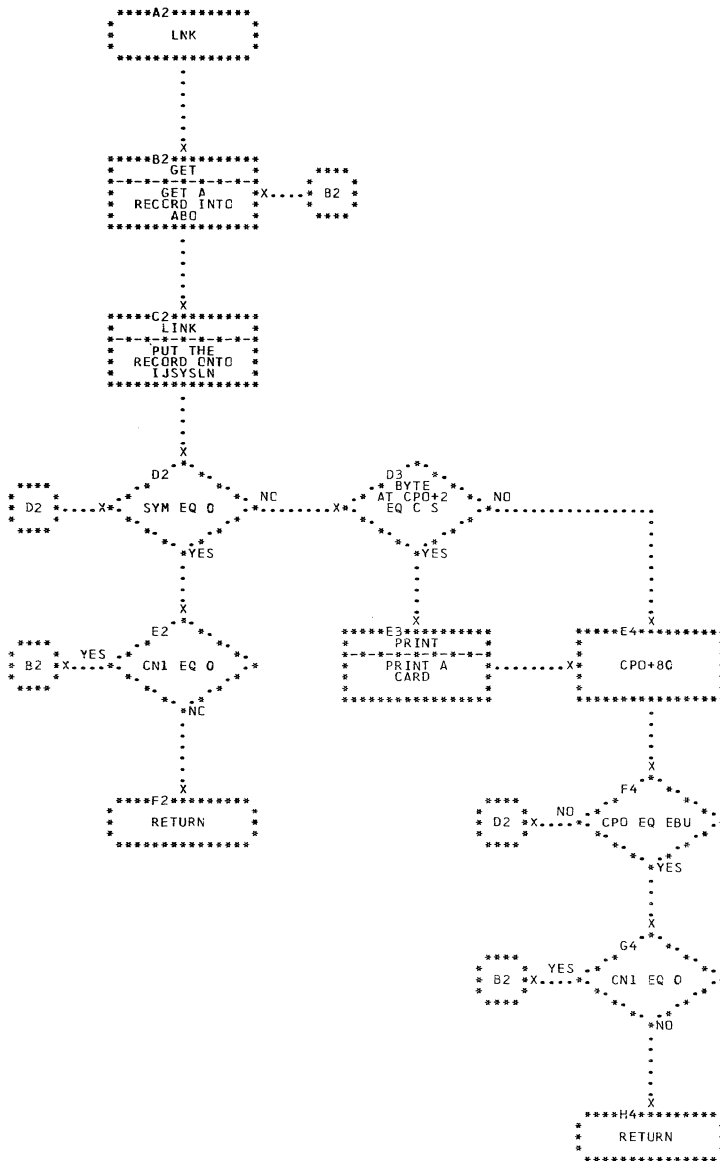
.....K4.....
ZPRNT
PRINT
THE LINE
.....

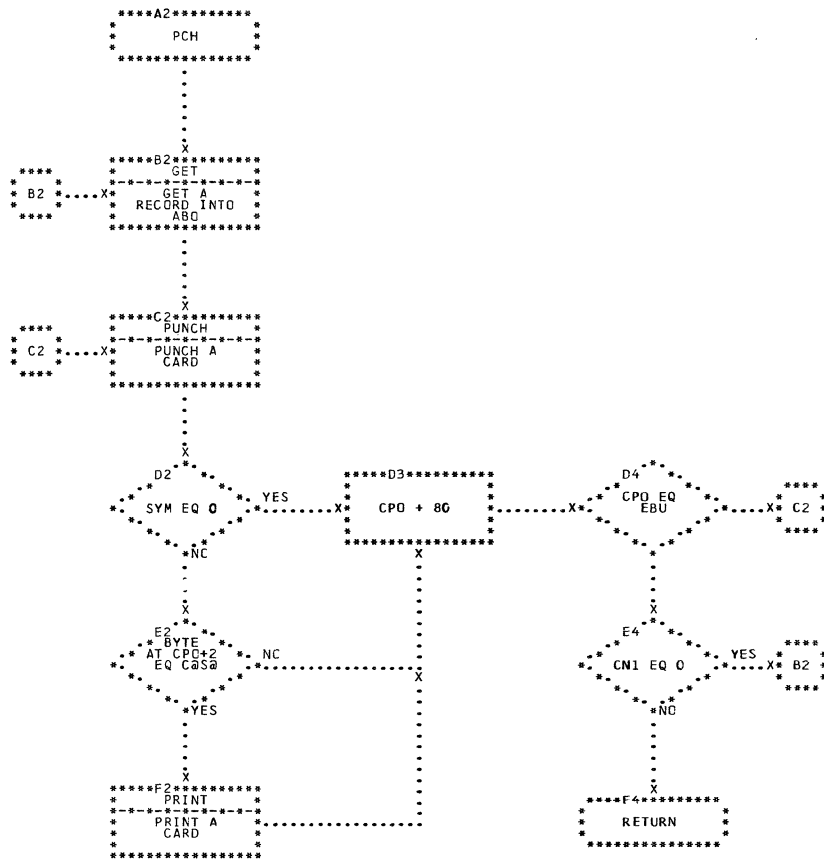
.....K5.....
RETURN
.....

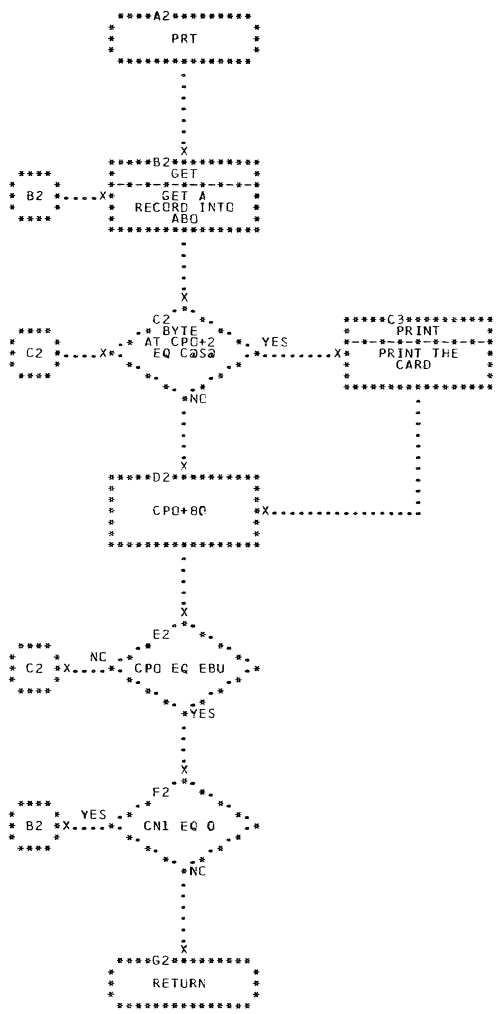


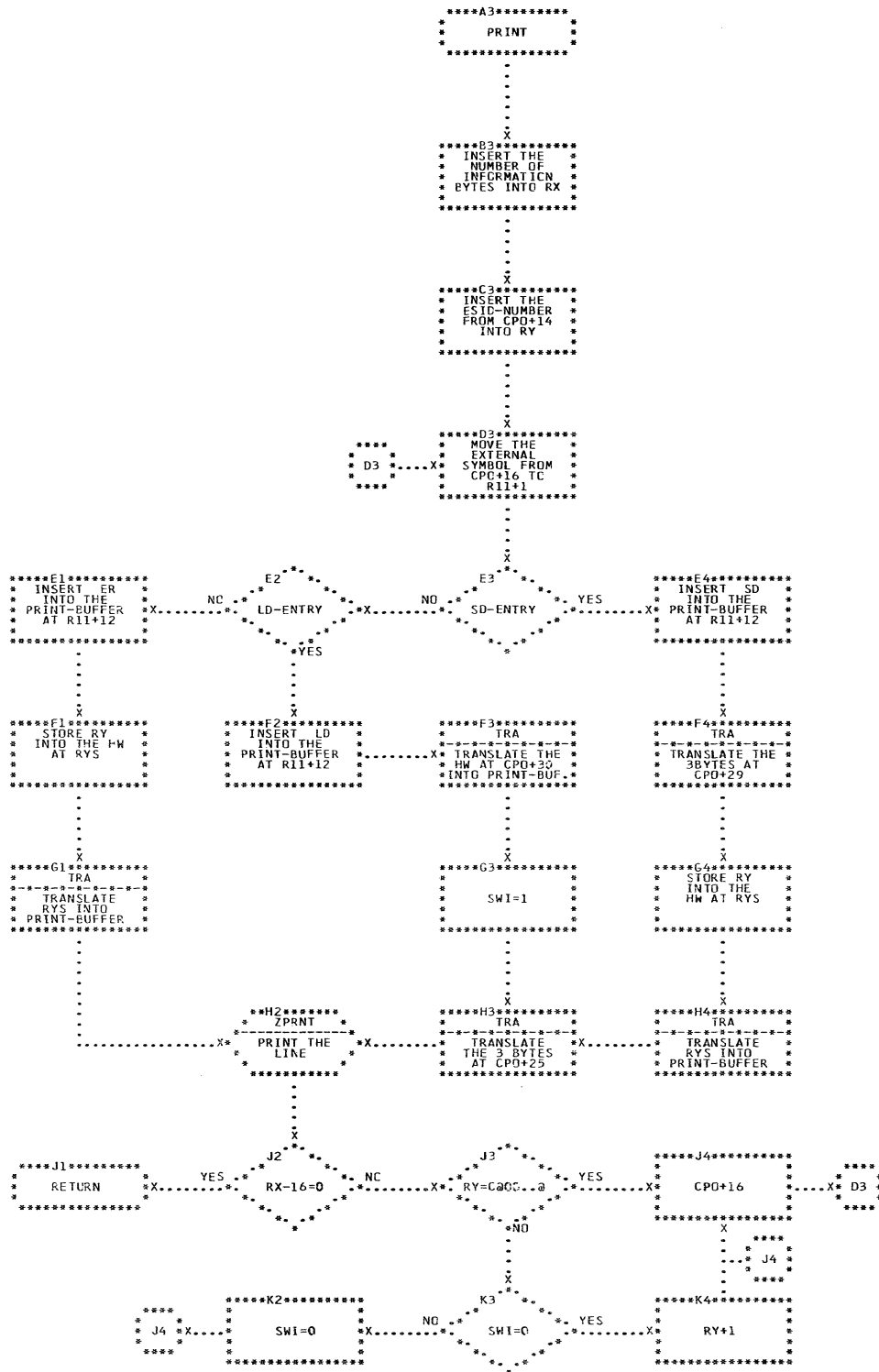













```
*****A1*****
BLOCKT
*****
```

```
.....
.....
.....
X
.....
```

```
*****B1*****
HESUB
*****
INSERT AND
PRINT THE
SUBHEADER
*****
```

```
.....
.....
X
.....
```

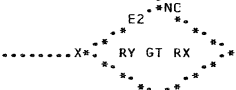
```
*****C1*****
CPD=1JKMBC
RX=1JKMBC
RY=1
*****
```

```
.....
.....
X
.....
```

```
*****D1*****
PRIBLO
PRINT THE
CURRENT
DSA-LENGTH
*****
```

```
.....
.....
X
.....
```

```
*****E1*****
RY+1
CPD+2
*****
```



```
.....
.....
X
*****F2*****
RETURN
*****
```

```
*****A4*****
PRIBLO
*****
```

```
.....
.....
.....
X
.....
```

```
*****B4*****
TRA
*****
TRANSLATE
BLOCKNUMBER
INTO R11+3
*****
```

```
.....
.....
X
.....
```

```
*****C4*****
TRA
*****
TRANSLATE
LENGTH OF
DSA INFO R11+13
*****
```

```
.....
.....
X
.....
```

```
*****D4*****
ZPRNT
PRINT THE
LINE
*****
```

```
.....
.....
X
.....
```

```
*****E4*****
RETURN
*****
```





International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

READER'S COMMENT FORM

IBM System/360
DOS/TOS PL/I PLM

Form Y33-9012-0

- How did you use this publication?

As a reference source
As a classroom text
As a self-study text

- Based on your own experience, rate this publication . . .

As a reference source:
Very Good Fair Poor Very
Good

As a text:
Very Good Fair Poor Very
Good

- What is your occupation?
- We would appreciate your other comments; please give specific page and line references where appropriate. If you wish a reply, be sure to include your name and address.

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

YOUR COMMENTS PLEASE . . .

This SRL bulletin is one of a series which serves as reference sources for systems analysts, programmers and operators of IBM systems. Your answers to the questions on the back of this form, together with your comments, will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

Fold

Fold

FIRST CLASS
PERMIT NO. 1359
WHITE PLAINS, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES



POSTAGE WILL BE PAID BY . . .

IBM Corporation
112 East Post Road
White Plains, N. Y. 10601

Attention: Department 813

Fold

Fold



International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]