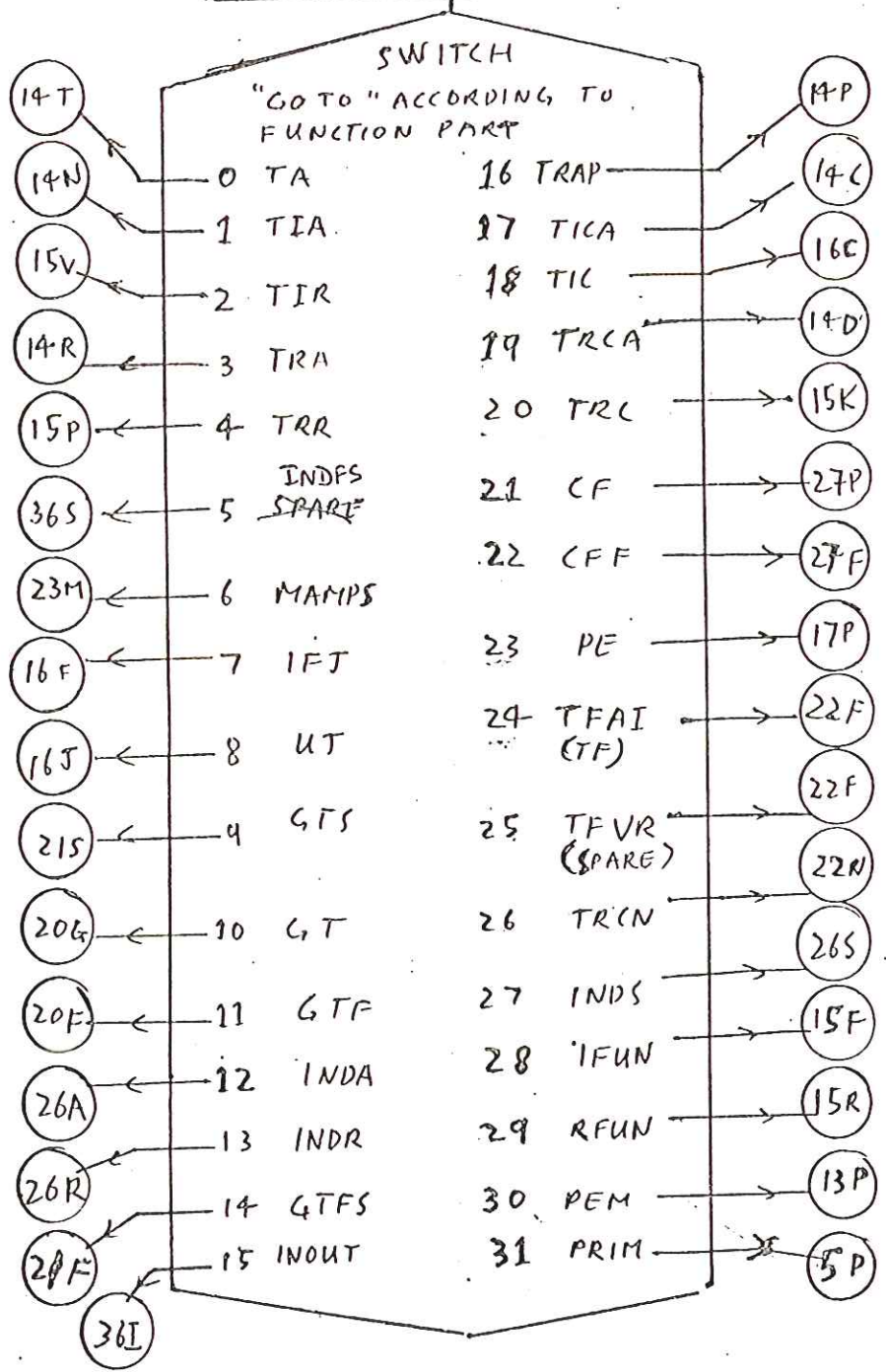
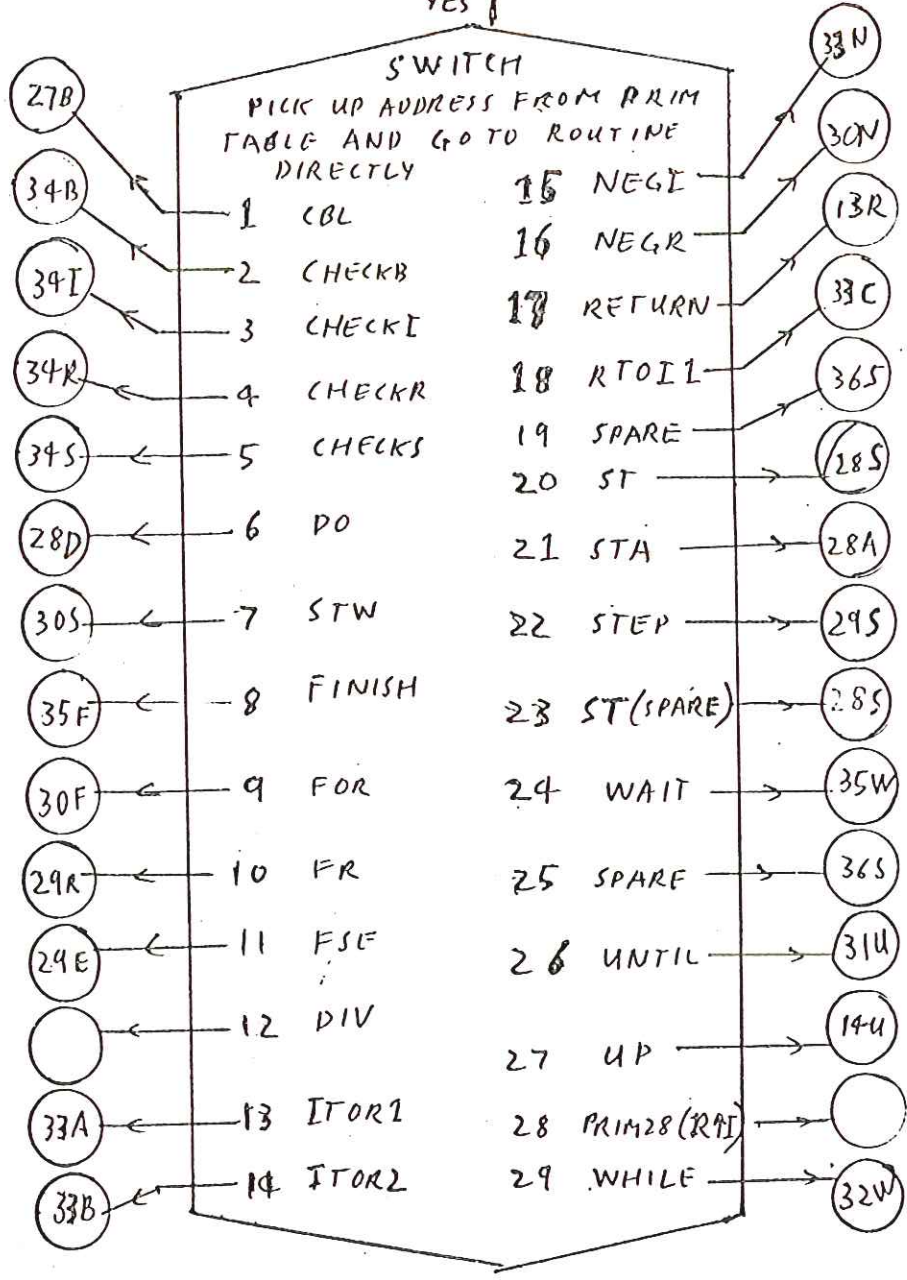
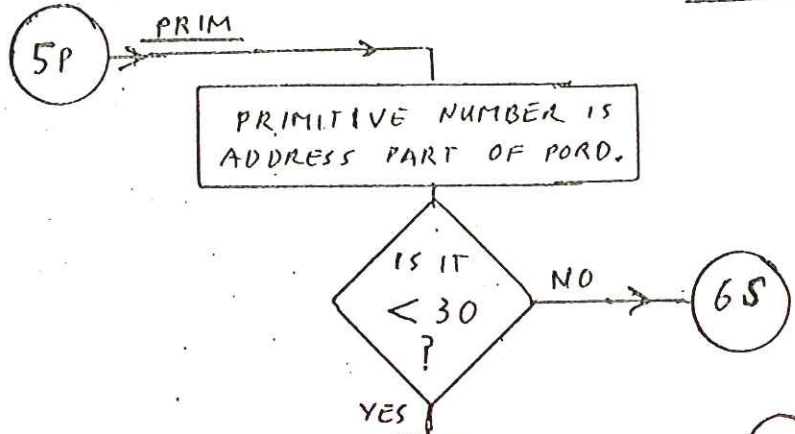


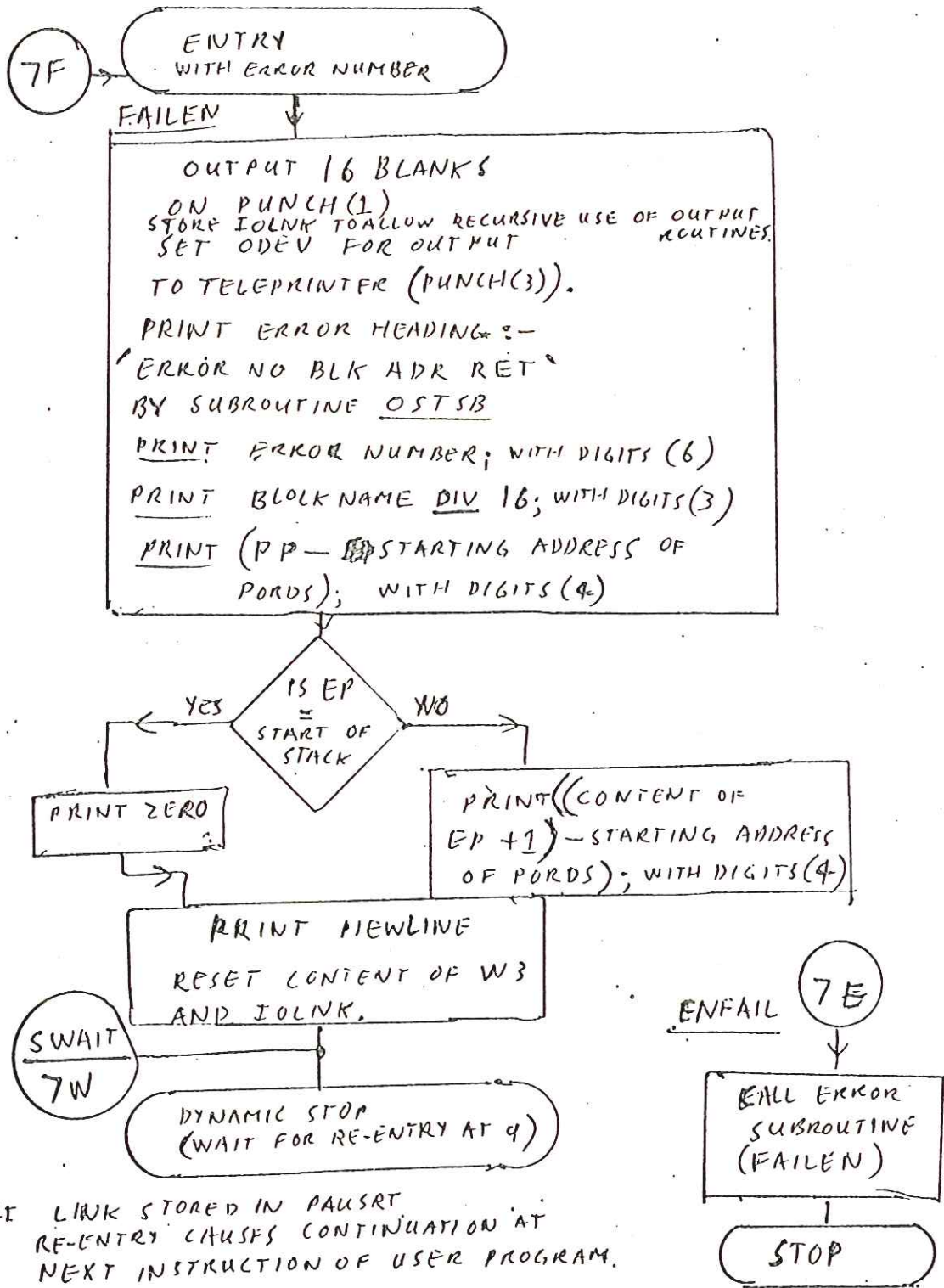
4N → WORD

PICK UP WORD INDICATED BY
WORD POINTER.
ADVANCE WORD POINTER BY ONE.
PICK OUT ADDRESS AND FUNCTION
PARTS FROM WORD. (STORE ADDRESS
IN ADDRPT)





FAIL Subroutine (FAILEN)



SUBRT LINK STORED IN PAUSRT
RE-ENTRY CAUSES CONTINUATION AT
NEXT INSTRUCTION OF USER PROGRAM.

Function PEM

13P

$n := ADPART$
STORE $EP - 3n$ at SP
 $FP := EP - 3n - 3$
store BN at $SP + 1$
 $SP := SP + 2$

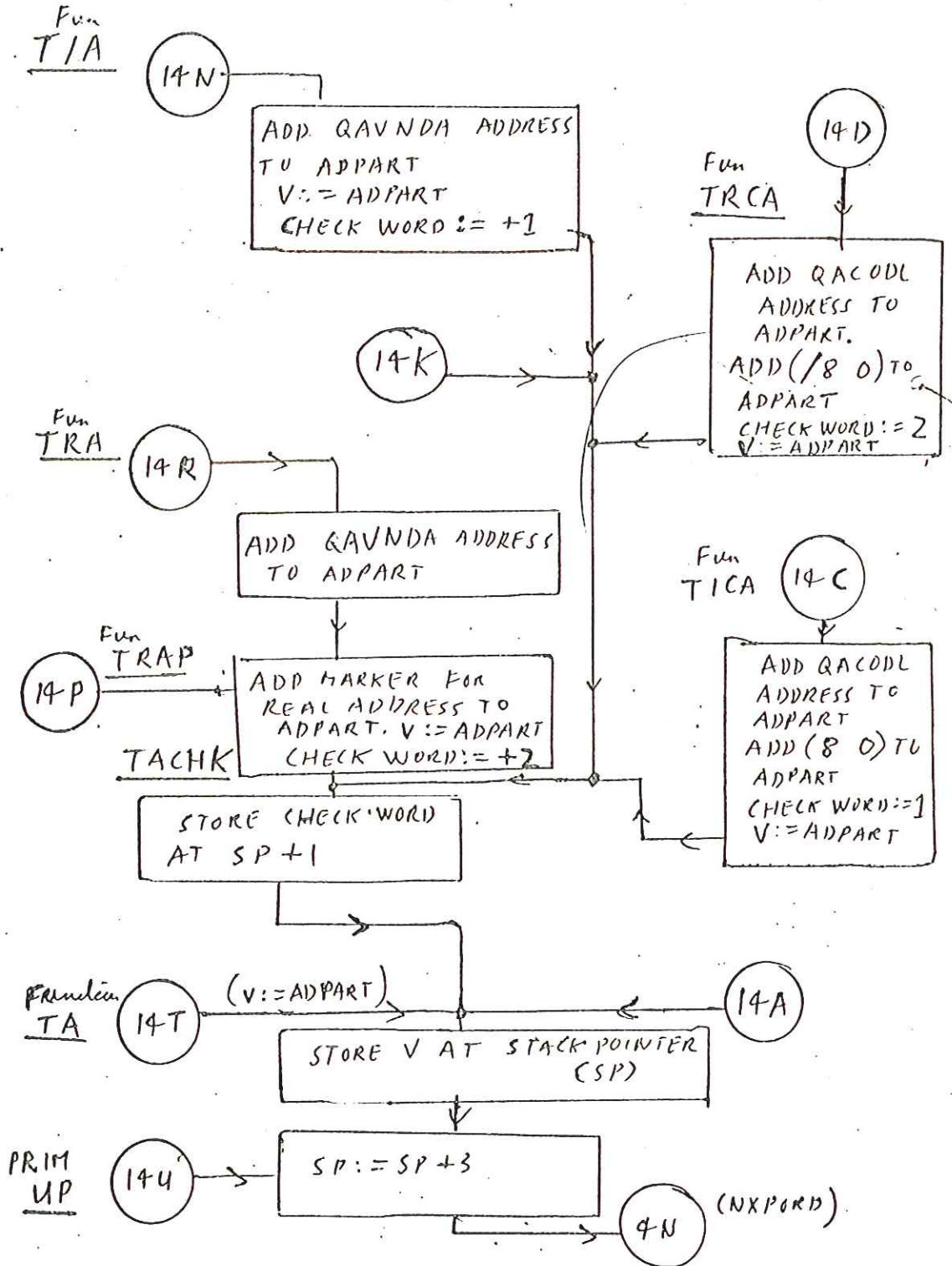
CALL MACHINE CODE SUBROUTINE
(LINK AT (PP) , ENTRY AT $((PP)+1)$)

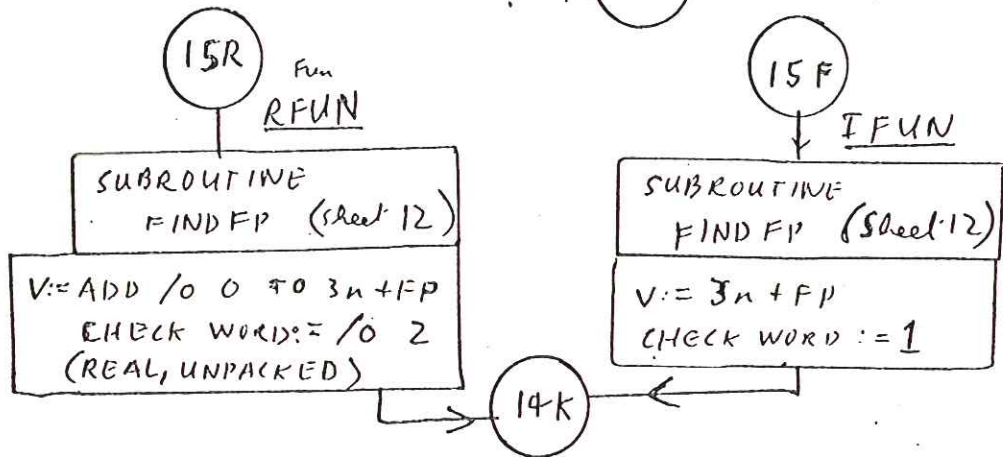
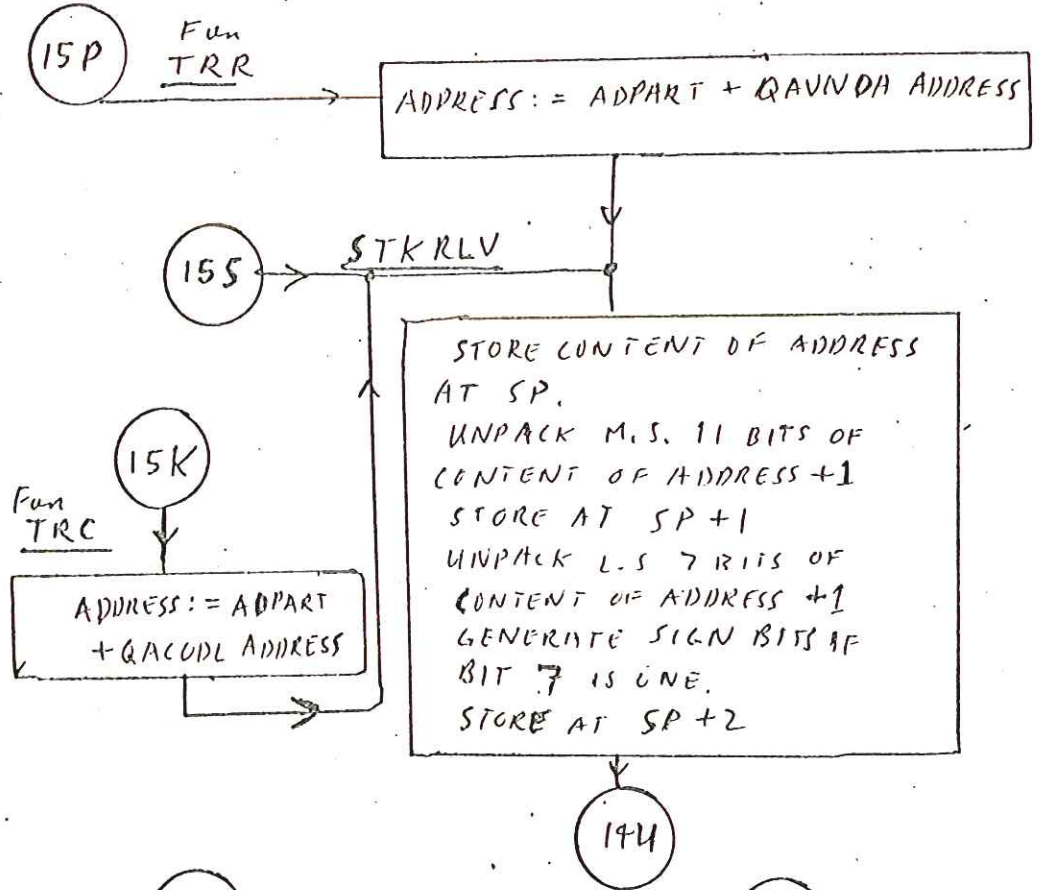
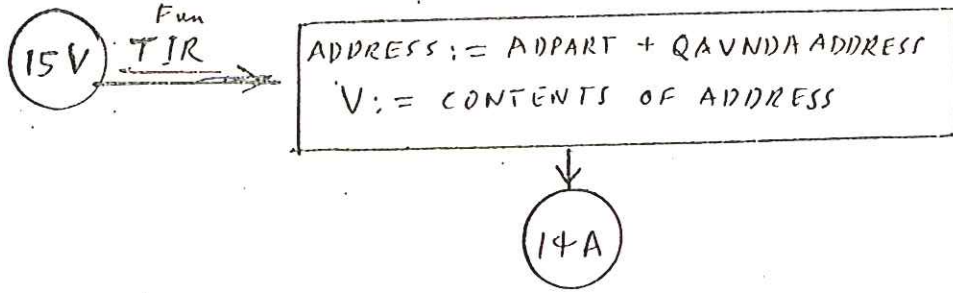
PRIM RETURN

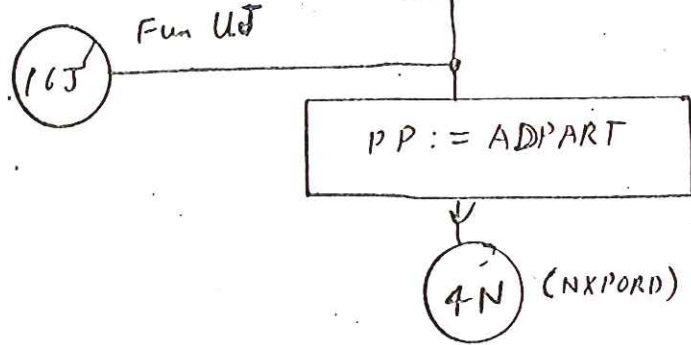
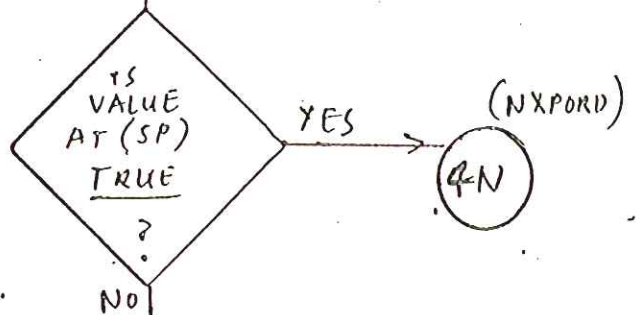
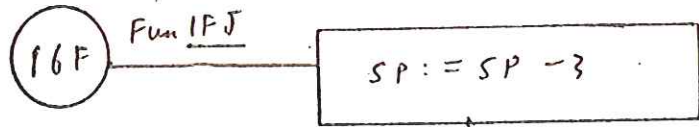
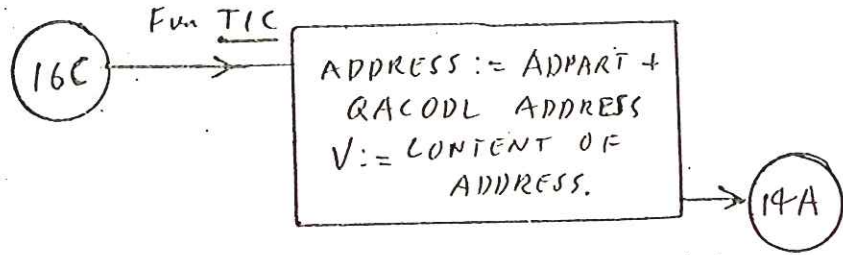
13R

$BN :=$ CONTENT OF $EP + 3$
 $SP :=$ CONTENT OF $EP + 2$
 $PP :=$ CONTENT OF $EP + 1$
 $EP :=$ CONTENT OF EP
 ~~$FP :=$ (CONTENT OF $EP + 2$) - 3~~

4N (NEXTORD)







17P Fun PE

```

PICK OUT BN' AND VA
FROM ADPART. NEXTAD :=
CONTENT OF (SP) := EP - 3m
EP := EP - 3m - 3;
CONTENT OF (SP+1) := BN (ORIGINAL);
BN := BN';
SP := SP + 2;

```

17R PCHK4

```

HAVE
ALL
PROCEDURE
PARAMETERS
BEEN
CHECKED?

```

YES → 18M

NO

PCHK1

```

PICK UP NEXT CHECK
WORD (CONTENT OF (PP));
PP := PP + 1;

```

```

IS
PARAM.
CALLED BY
VALUE?

```

YES → PCHK2

```

IS IT
AN ARRAY
(INTEGER
OR REAL)

```

YES → 19C

NO

```

PICK UP CODE CHECK
FROM (NEXTAD + 1)
IGNORING M.S. BIT

```

17N

PCHK5

```

NEXTAD :=
NEXTAD + 3

```

17M

7E

```

PICK UP ERROR
NO. 1 (PARAMETER
MISMATCH)

```

```

DOES
CHECK WORD
CODE
CORRESPOND?

```

YES → 18A

NO → 7E



PCHK3

18A SUBTRACT PROGRAM BASE ADDRESS FROM ADDRESS PART OF CHECK WORD. (RESULT = DIM)

WAS ADDRESS PART = 8191

YES () 17N

(IGNORE IF PARAMETER NOT USED IN PROCEDURE)

PCHK8

IS PARAMETER AN ARRAY

PICK OUT NUMBER OF DIMENSIONS

NO

IS IT A PROCEDURE

NO

PICK OUT NUMBER OF PARAMETERS FROM BLOCK HEAD

IS NUMBER = DIM ?

YES 17N

NO 17M

18M

MAMPS4

IS SP < WARN ADDRESS ?

YES 4N

(NXPORD)

NO

SET WARN TRUE

18E

MAMPS5

ERROR NUMBER 2 STACK OVERFLOW

7E

IS SP < STACK END

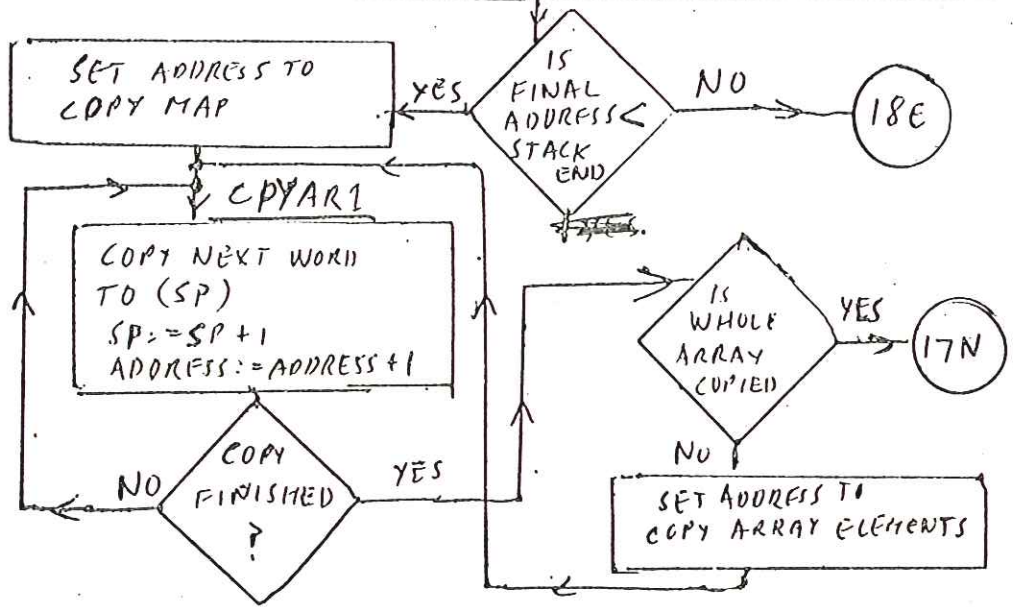
YES 4N

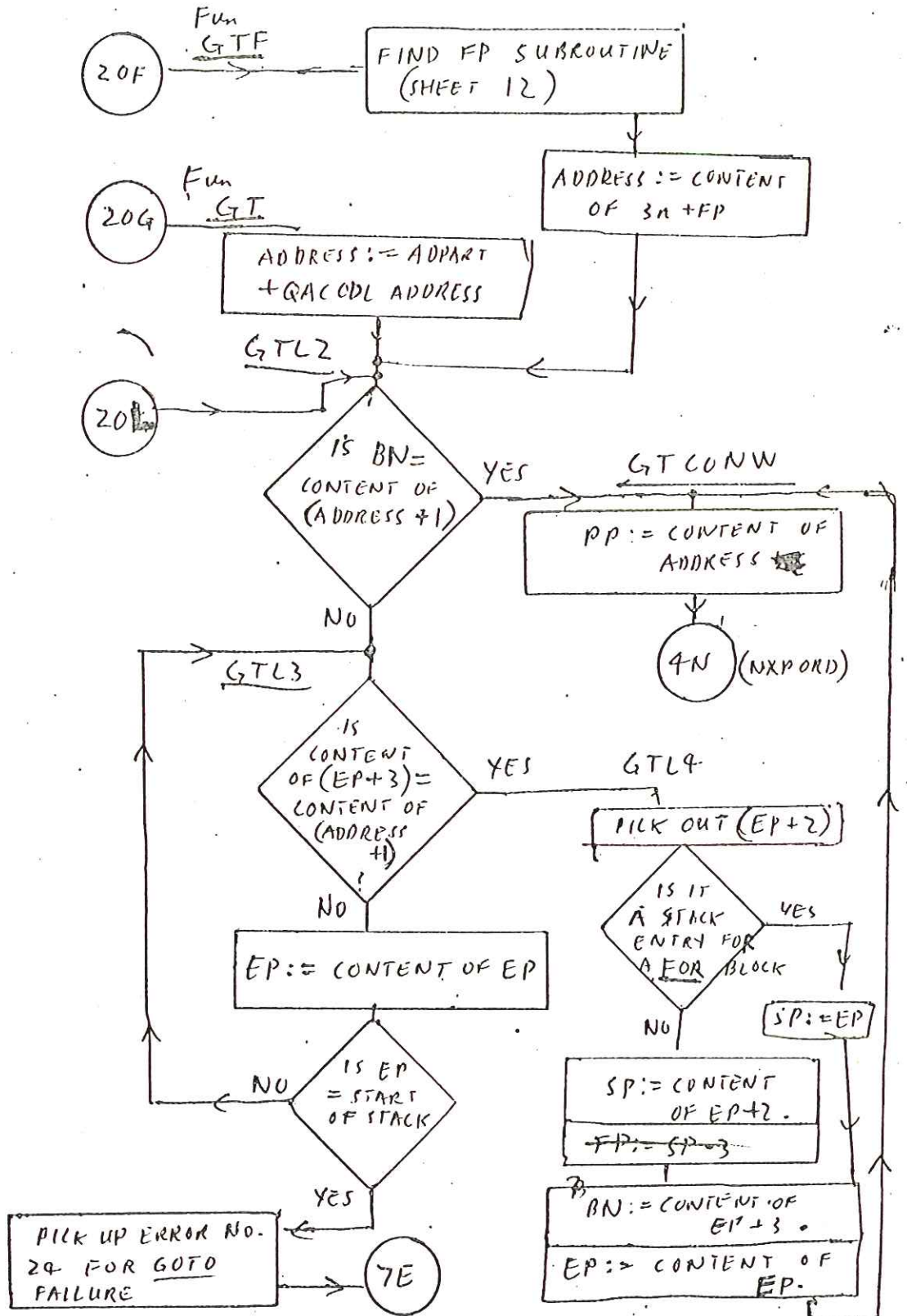
NO

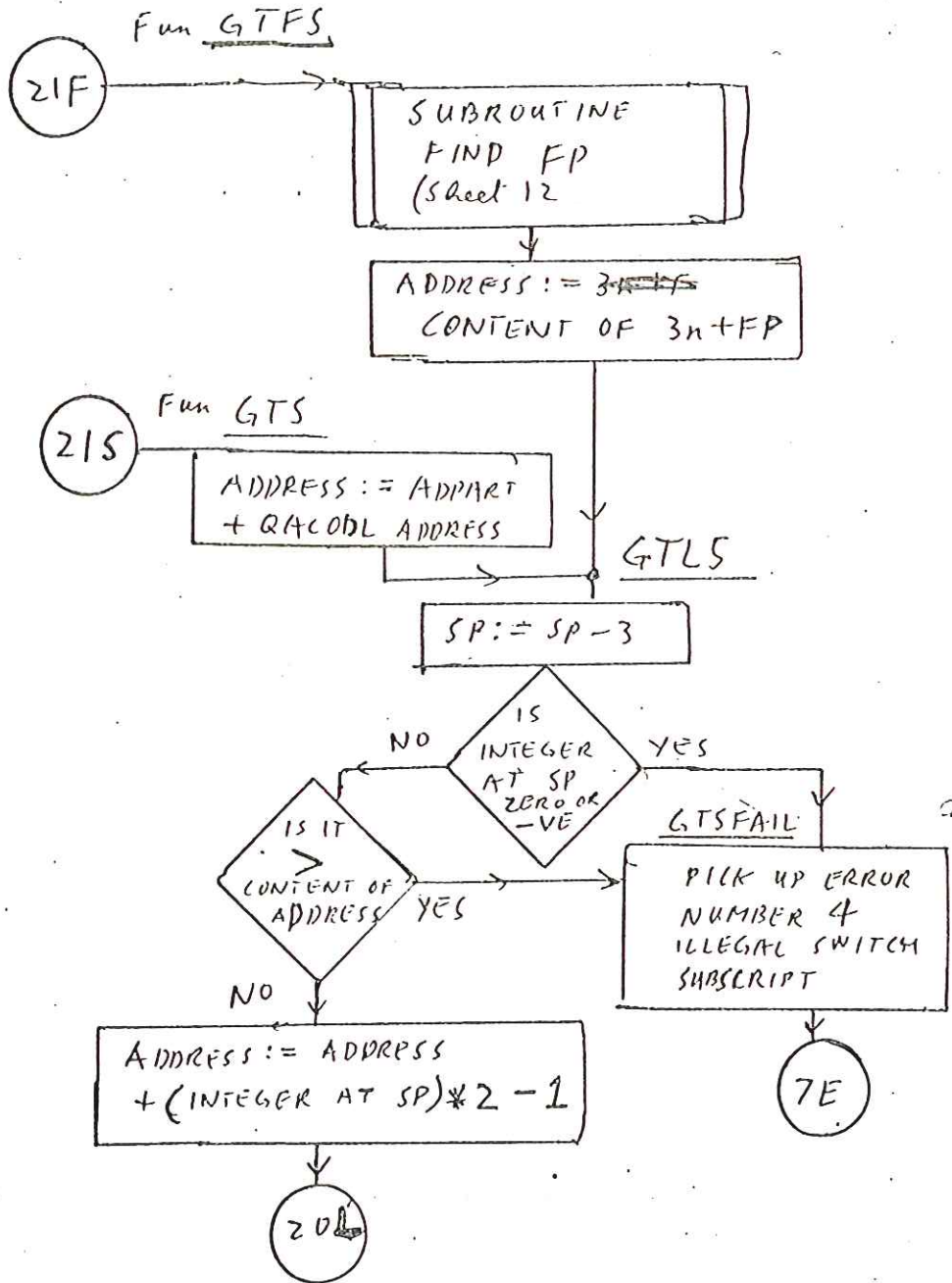
COPYAR

19C

PICK OUT ADDRESS OF
ARRAY PAIR FROM PARAMETER
SPACE.
REPLACE BY ADDRESS OF ARRAY PAIR
~~ARRAY~~ COPY (= SP)
PICK OUT ADDRESS OF ARRAY
MAP (SUBROUTINE GARAD)
PICK OUT NUMBER OF DIMENSIONS
ADDRESS OF ARRAY COPY
= NUMBER * 2 + SP + 4 ;
~~ADD~~
ADD INDICATOR BIT FOR REAL/
INTEGER AND STORE IN ARRAY
PAIR COPY.
PICK OUT NUMBER OF DIMENSIONS
AND FORM INTO SECOND WORD OF
ARRAY PAIR COPY. RELATIVE
ADDRESS POINTS TO NEXT WORD
STORE ADDRESS OF ARRAY MAP
COPY (= SP+3) IN (SP)+2
FINAL ADDRESS := ADDRESS OF COPY
~~SIZE~~ SIZE OF ARRAY. ~~+~~







(TFM)
TFAR
TFVI
TFVR)

22F

Fun TF

SUBROUTINE FIND FP
(Sheet 12)
ADDRESS := 3n + FP

CONTENT OF (SP+2) := CONTENT OF ADDRESS+2.
CONTENT OF (SP+1) := CONTENT OF ADDRESS+1.
CONTENT OF (SP) := CONTENT OF ADDRESS.

14M

Fun. TRCN

22N

SUBROUTINE FIND FP
(Sheet 12)

CHECK CONTENT OF 3n + FP + 1
FOR SIGN BIT (UNPACKED
NUMBER INDICATOR)

IS NUMBER
UNPACKED ?

ADDRESS := CONTENT OF
3n + FP.

PICK UP CONTENT OF 3n + FP
(ADDRESS)

22S

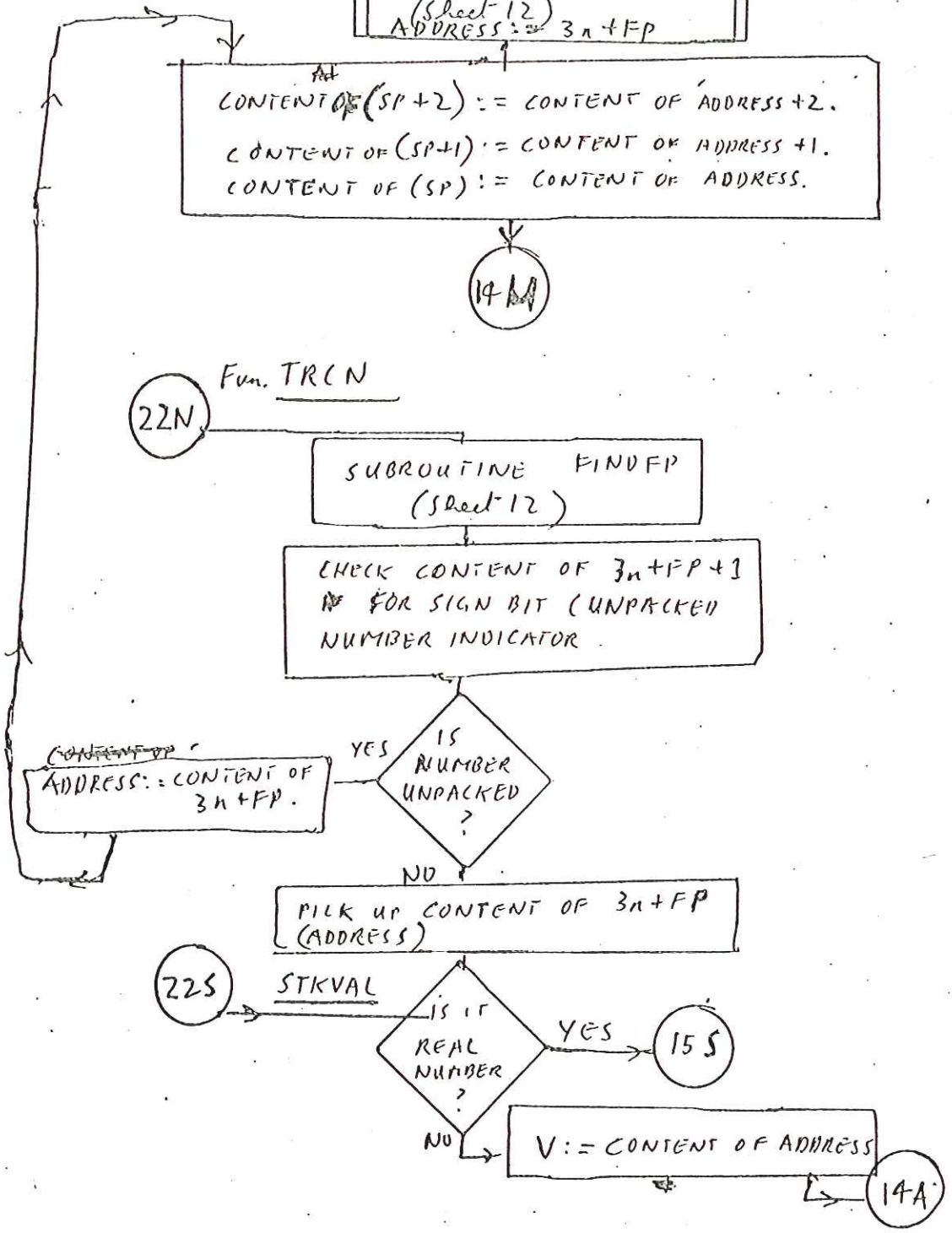
STKVAL

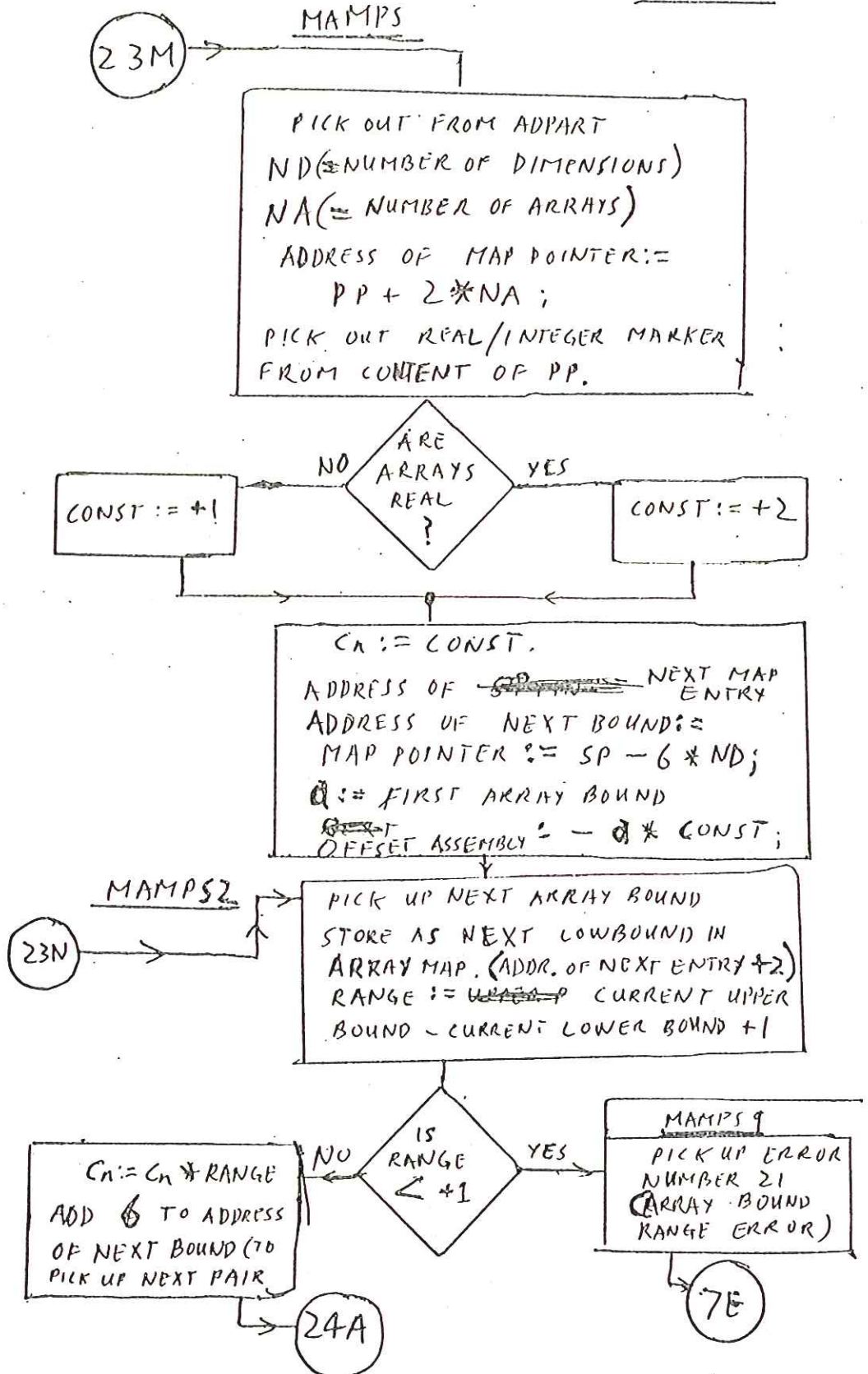
IS IT
REAL
NUMBER ?

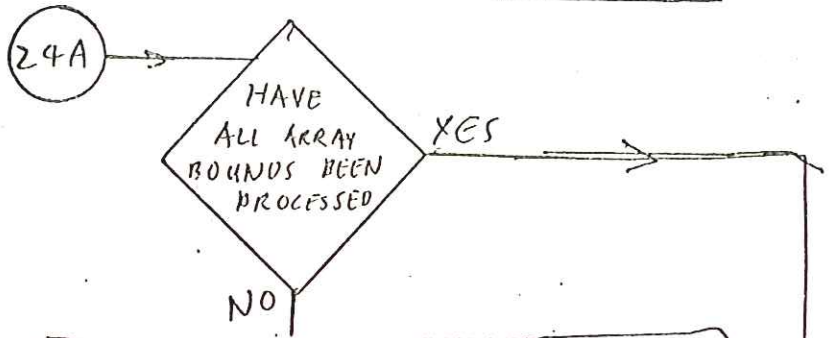
15S

V := CONTENT OF ADDRESS

14A







STORE NEW VALUE OF CONSTANT C_n
 IN ARRAY MAP (ADDRESS OF NEXT ~~MAP~~
 ENTRY + 3).
 ADD +2 TO ADDRESS OF NEXT ENTRY.
~~MAP~~
 OFFSET ASSEMBLY := OFFSET ASSEMBLY
 - (NEXT LOWBOUND * C_n);



MAMPS1
 STORE TOTAL SIZE (= CURRENT VALUE of C_n)
 IN FIRST LOCATION OF MAP;
 SECOND LOCATION OF MAP :=
 OFFSET ASSEMBLY;
 ADDRESS OF NEXT ARRAY :=
 ADDRESS OF NEXT MAP ENTRY + 3
 + REAL/INTEGER MARKER;

MAMPS3

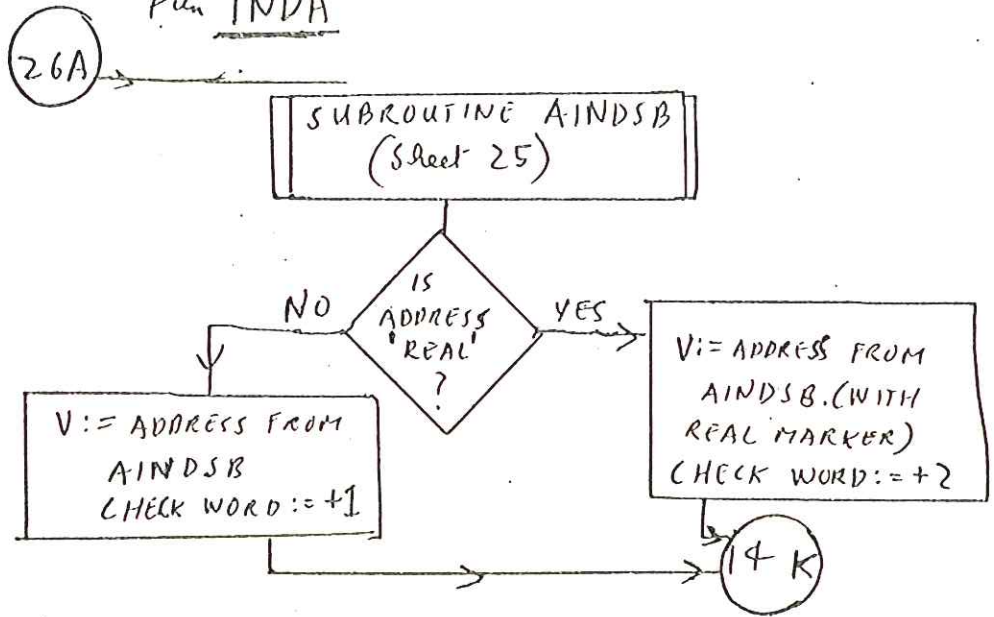
STORE ADDRESS OF NEXT ARRAY AT
 PP (ADDRESS OF NEXT ARRAY PAIR)
 ADDRESS OF NEXT ARRAY :=
 ADDRESS OF NEXT ARRAY + C_n
 PP := PP + 2



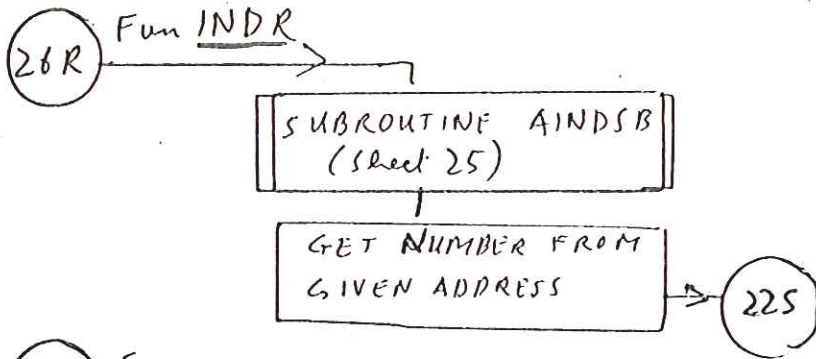
PP := PP + 1
 SP := ADDRESS OF
 NEXT ARRAY WITHOUT
 MARKER BIT



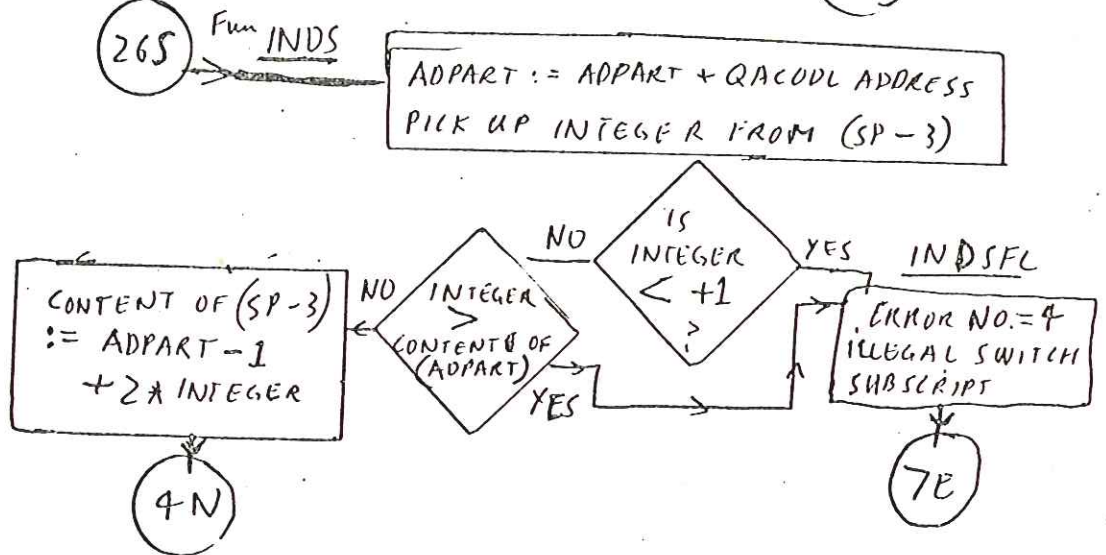
Fun INDA

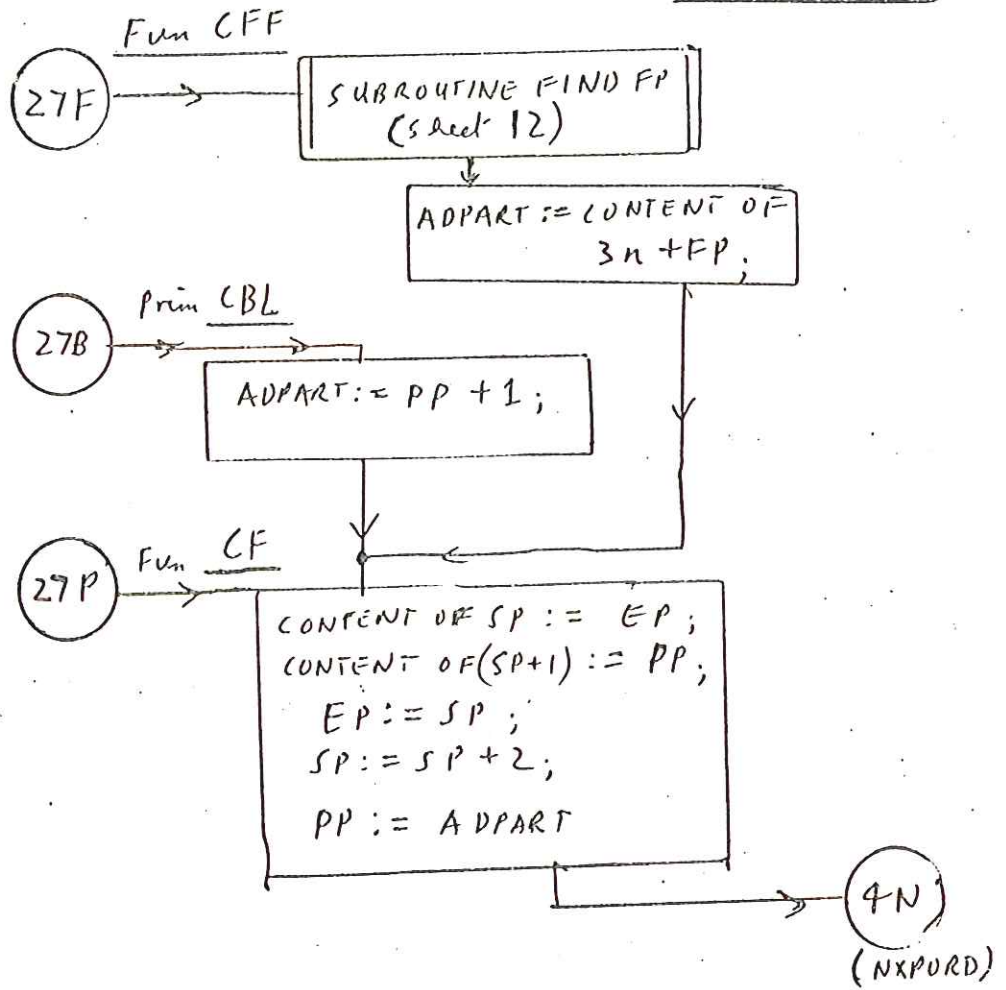


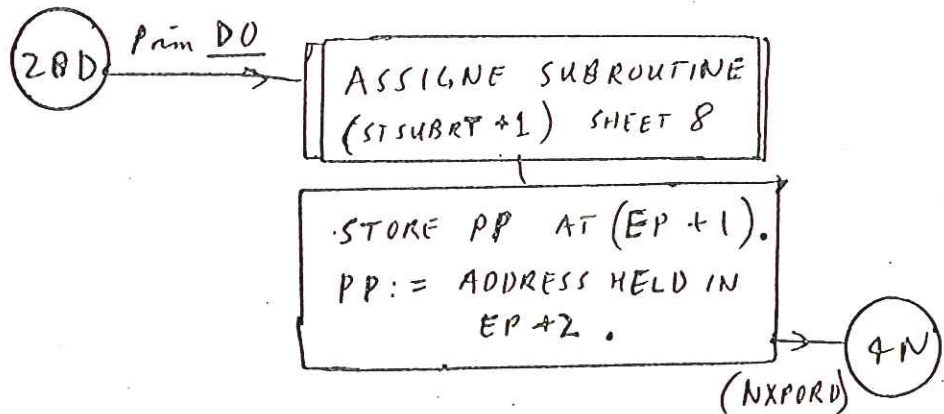
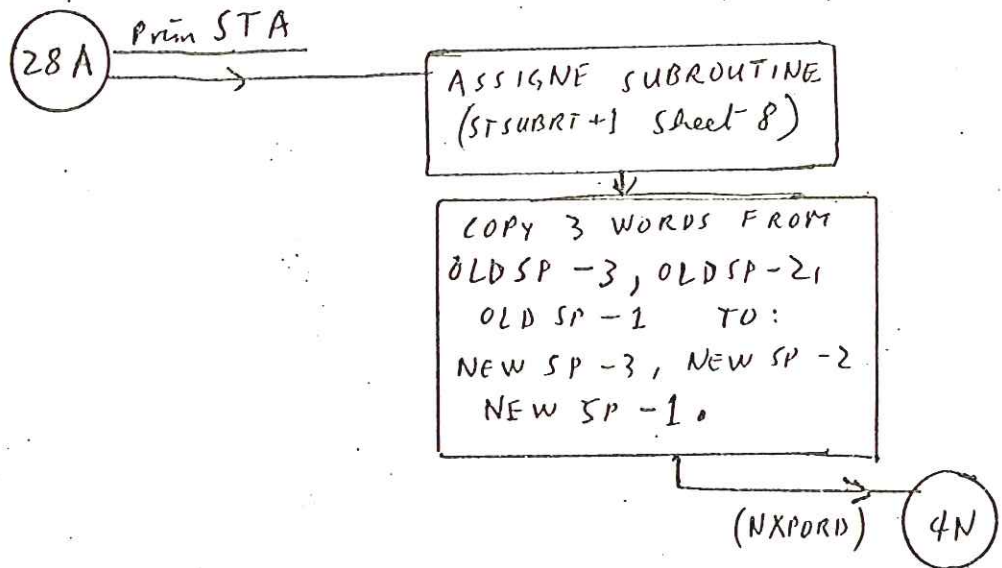
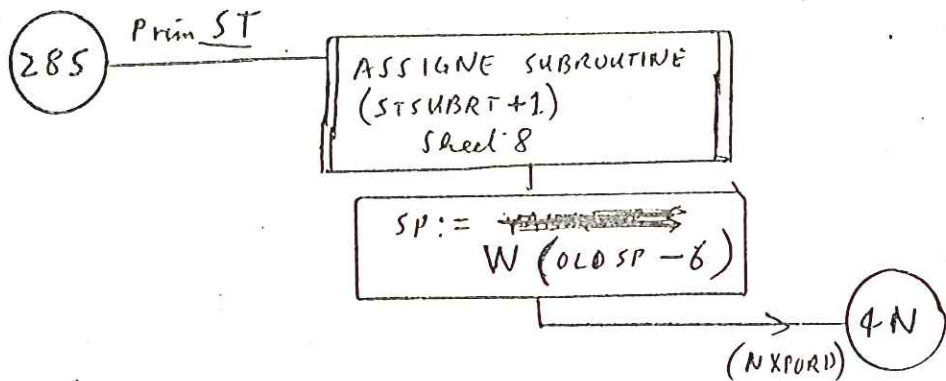
Fun INDR

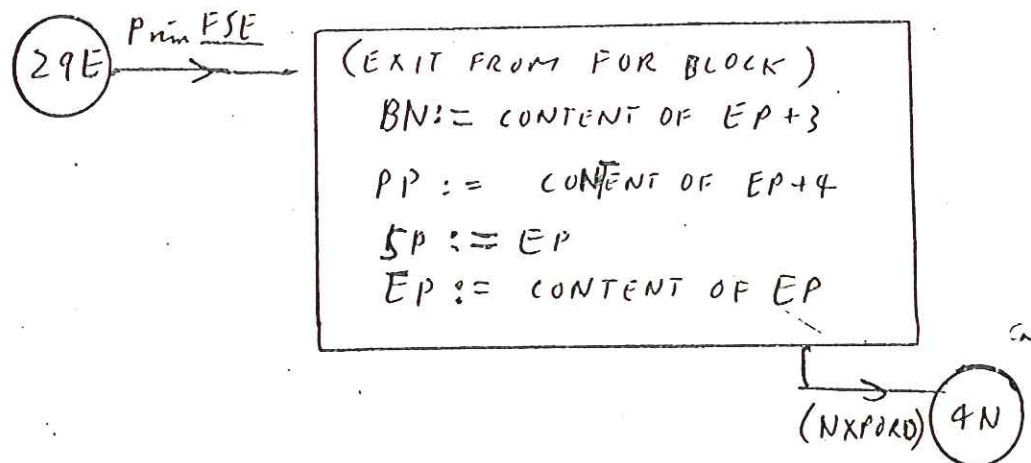
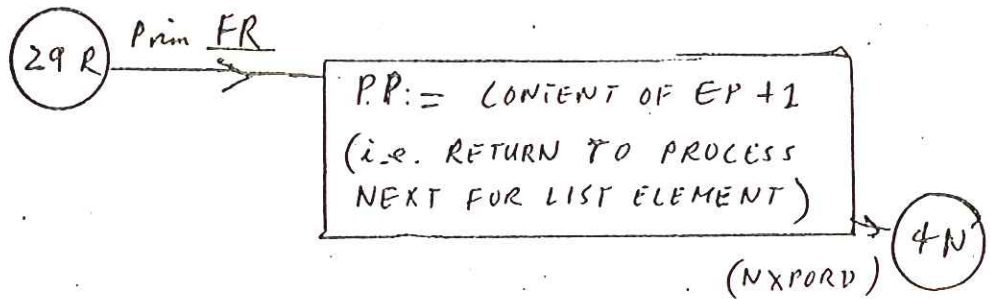
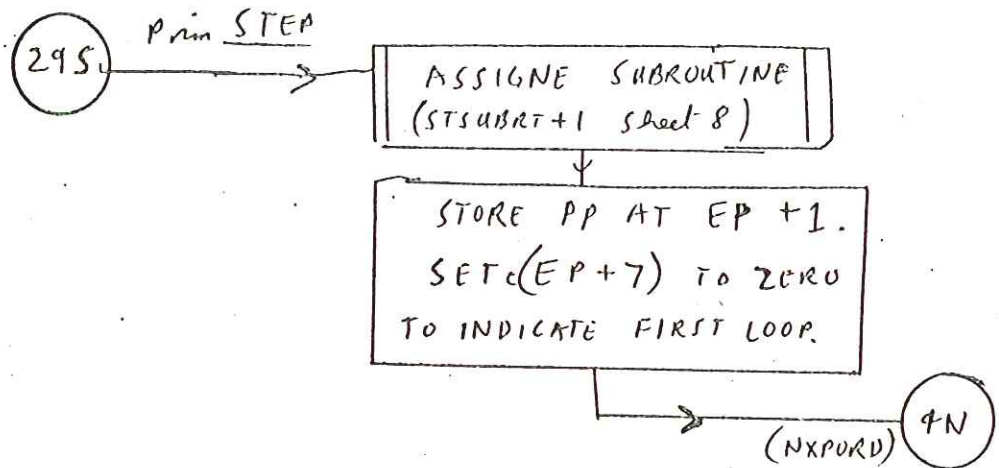


Fun INDS









30F Prim FOR

SET UP FOR BLOCK TO CONTROL FOR LOOP.
ADDRESS OF BLOCK = SP
C(BLOCK) := EP.
L(BLOCK+2) := ADDRESS 1 (= CONTENT OF PP).
L(BLOCK+3) := BN;
BN := CONTENT OF PP+1
CONTENT OF (BLOCK+4) := ADDRESS 2 (= CONTENT OF PP+2)
EP := SP
SP := SP+5; PP := PP+3

(NXPORD) 4N

30S Prim STW

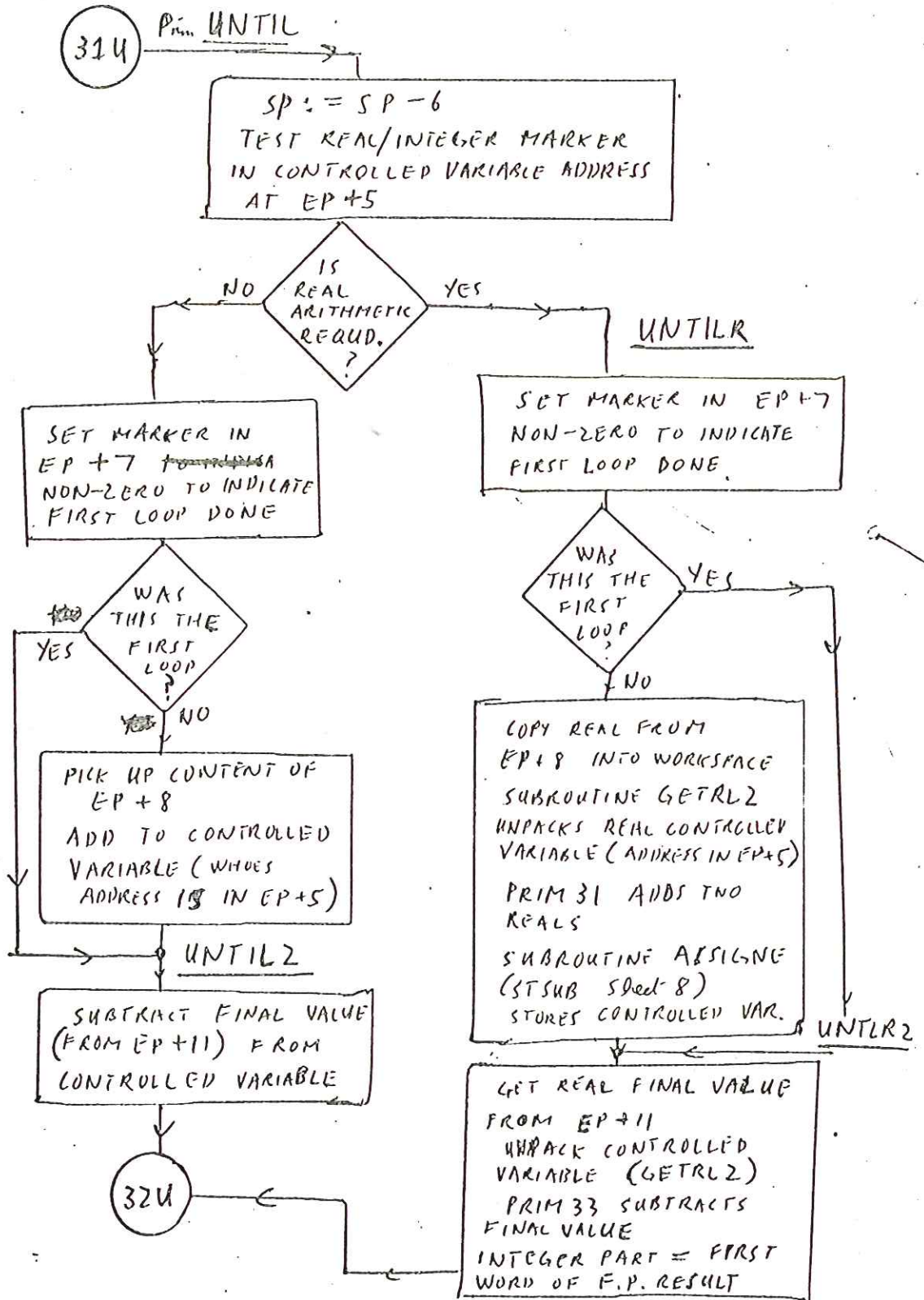
ASSIGNE SUBROUTINE
(STSUBRT +] sheet 8)

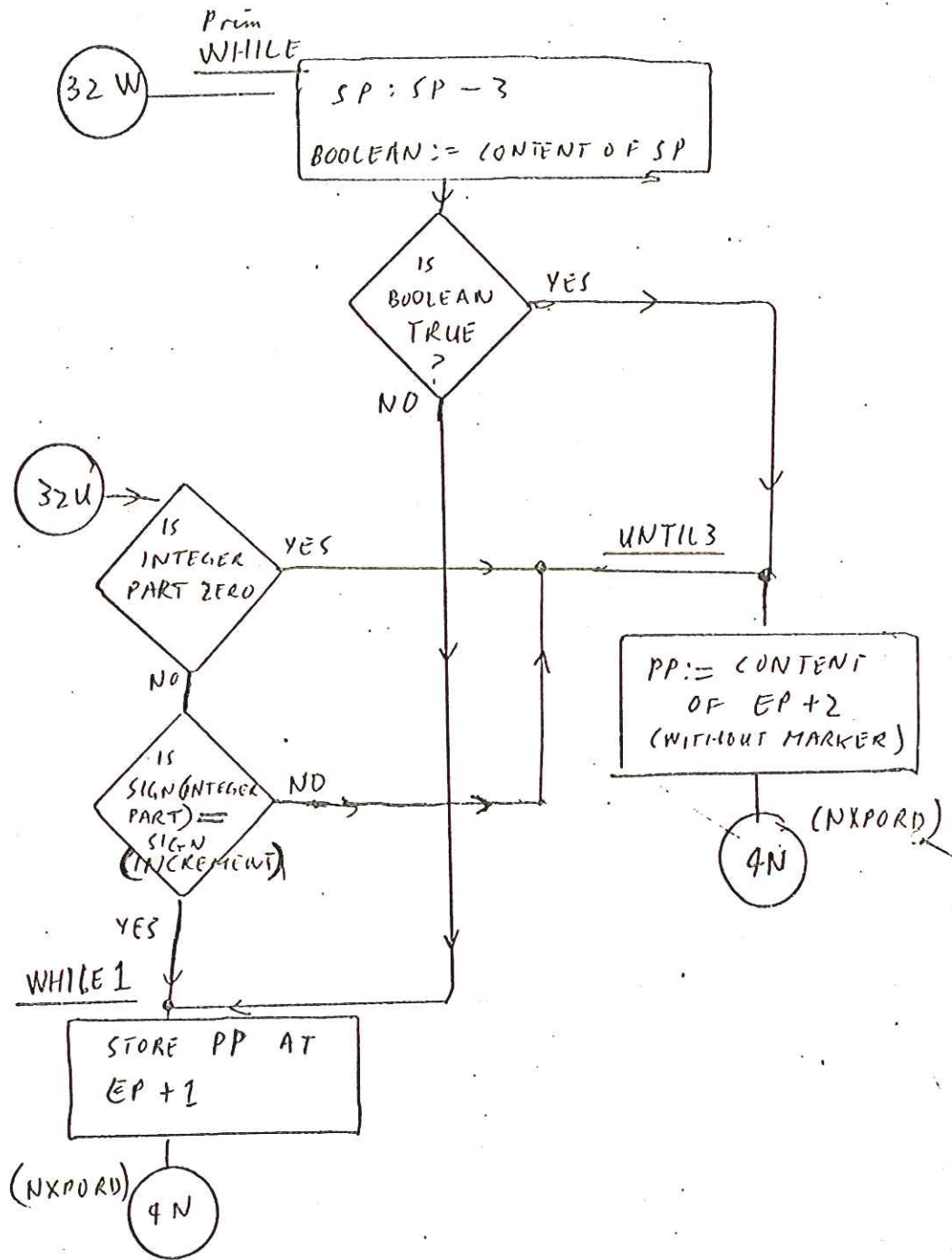
(NXPORD) 4N

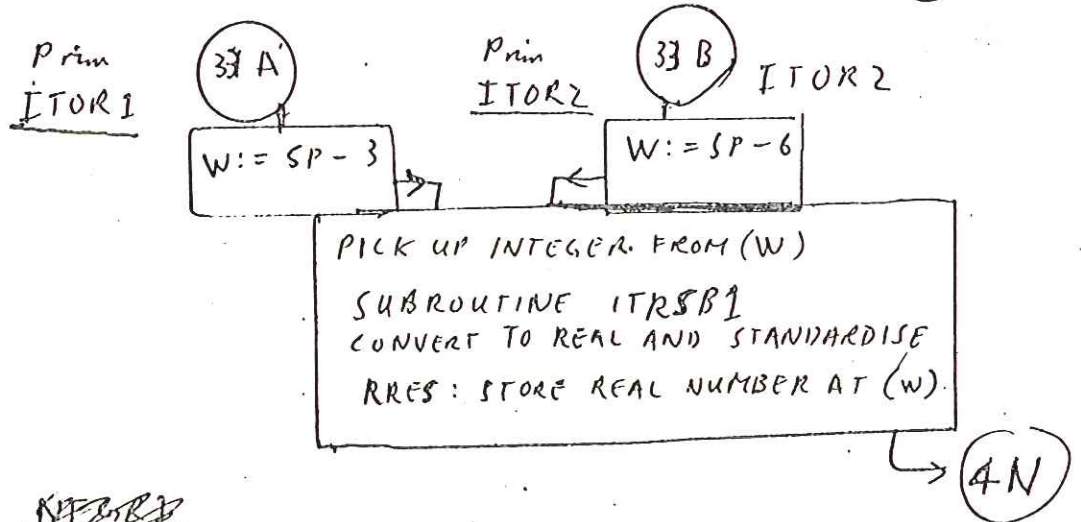
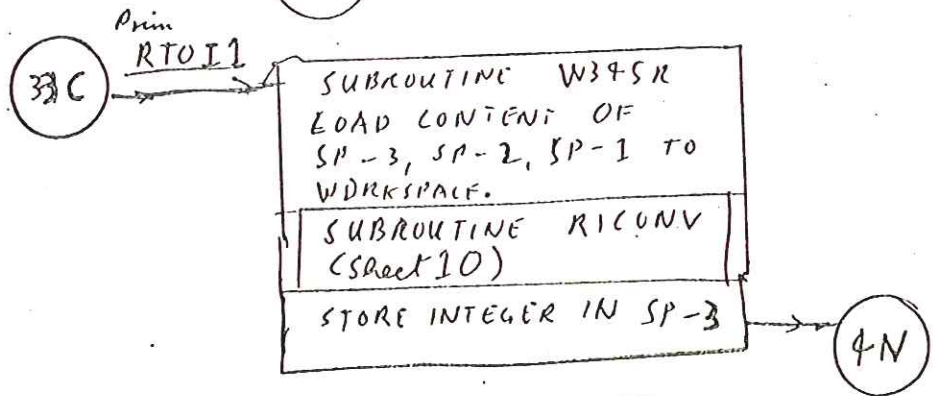
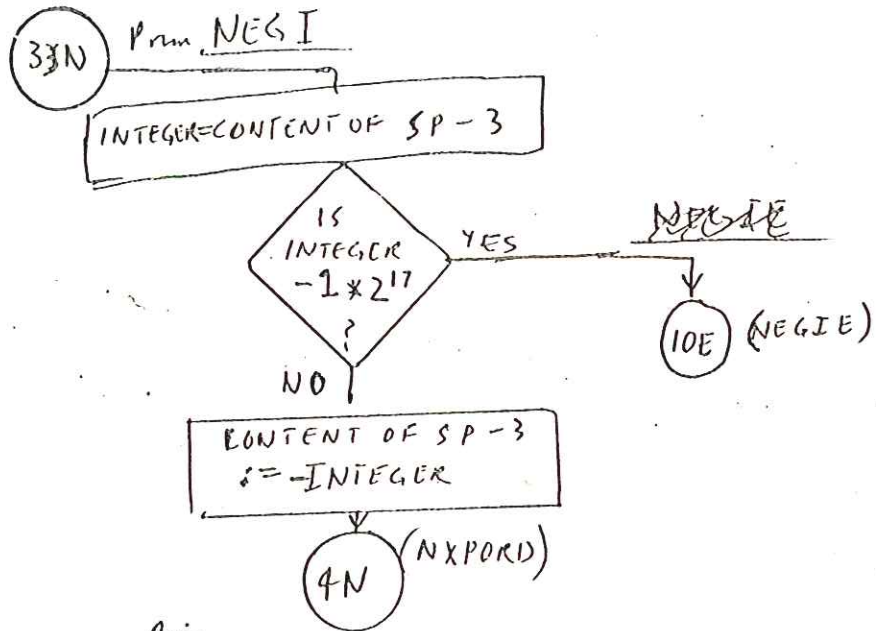
30N Prim NEGR

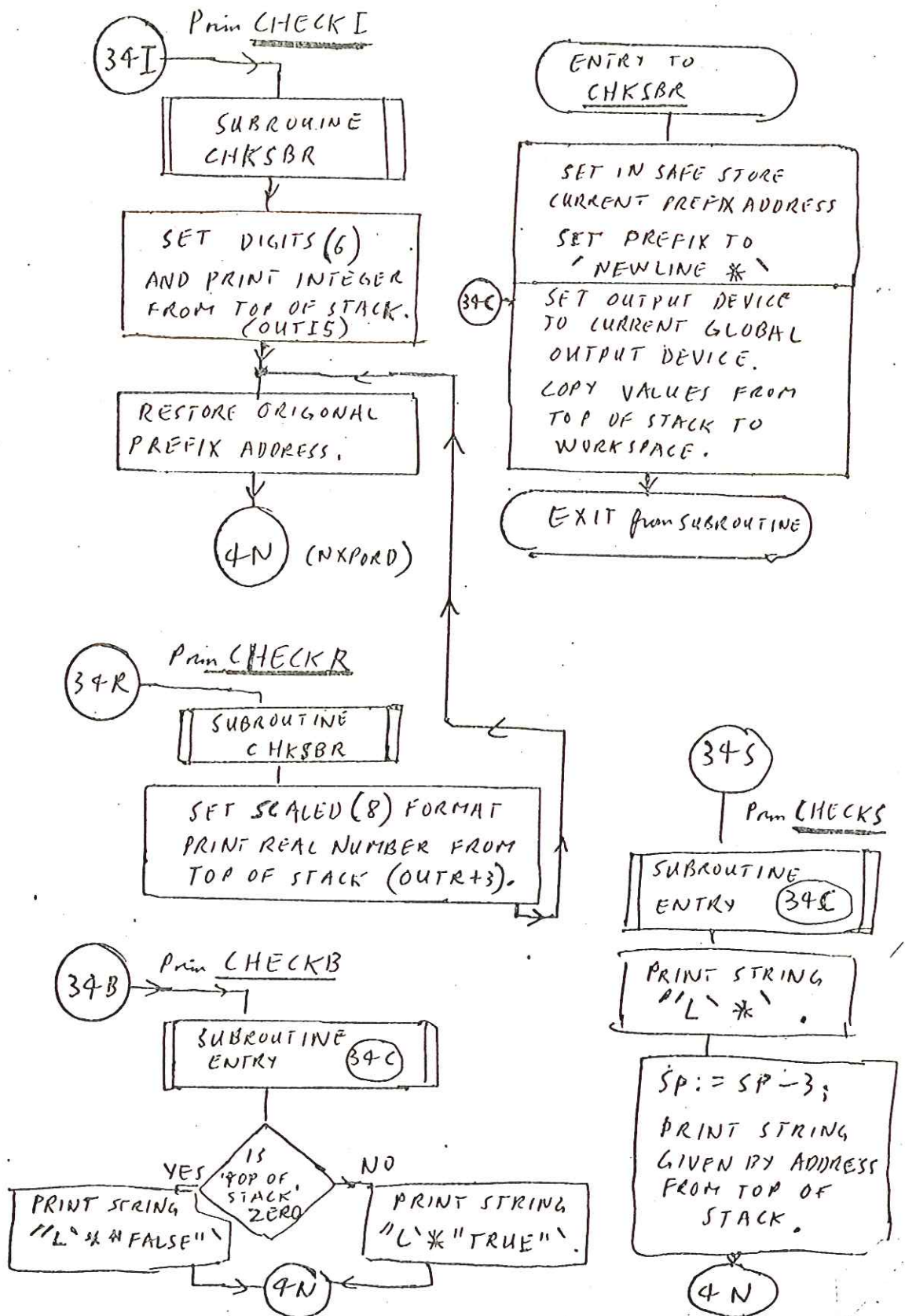
ADDRESS OF REAL := SP-3
SUBROUTINE NEGR1
(sheet 11)

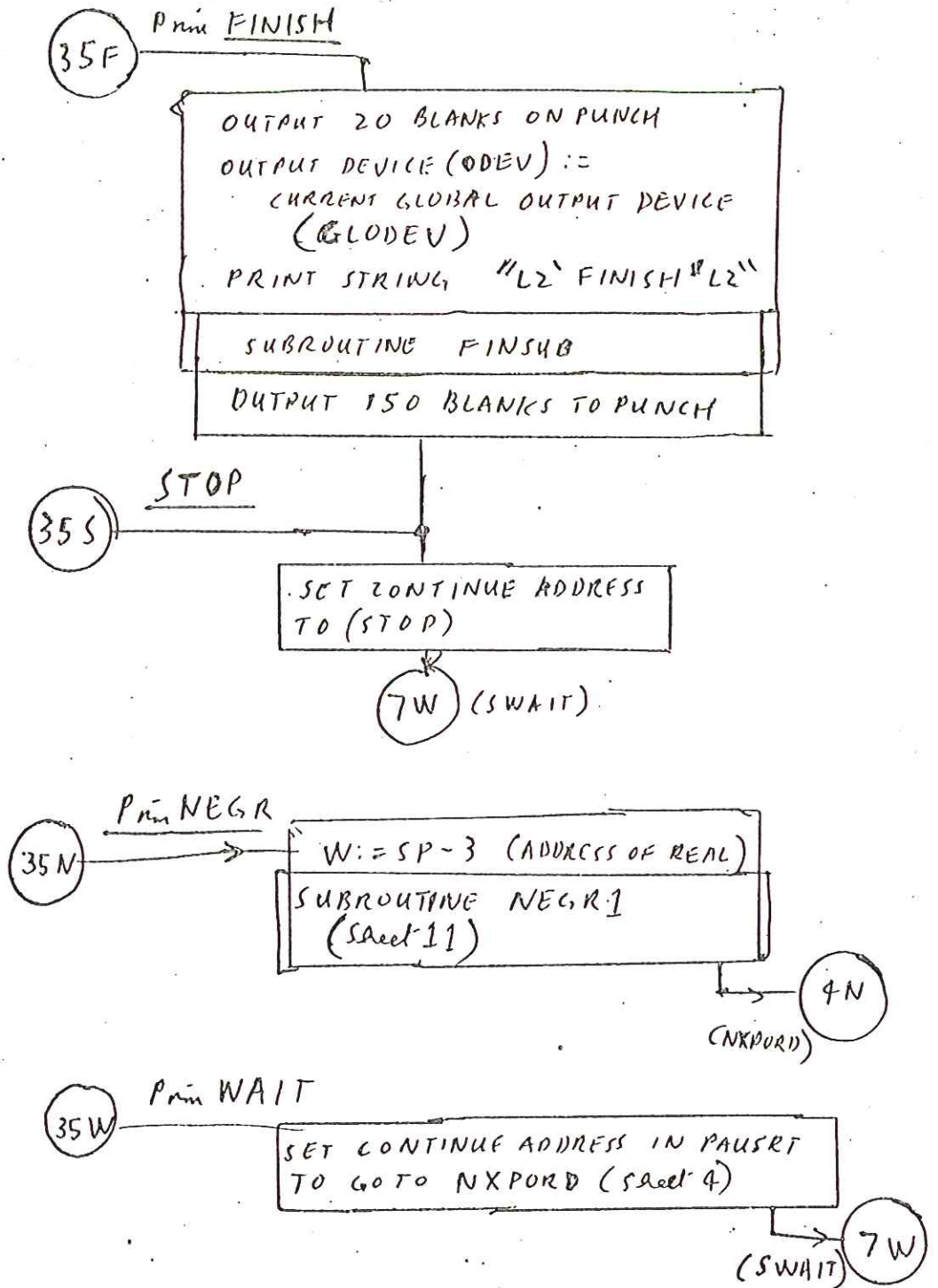
4N

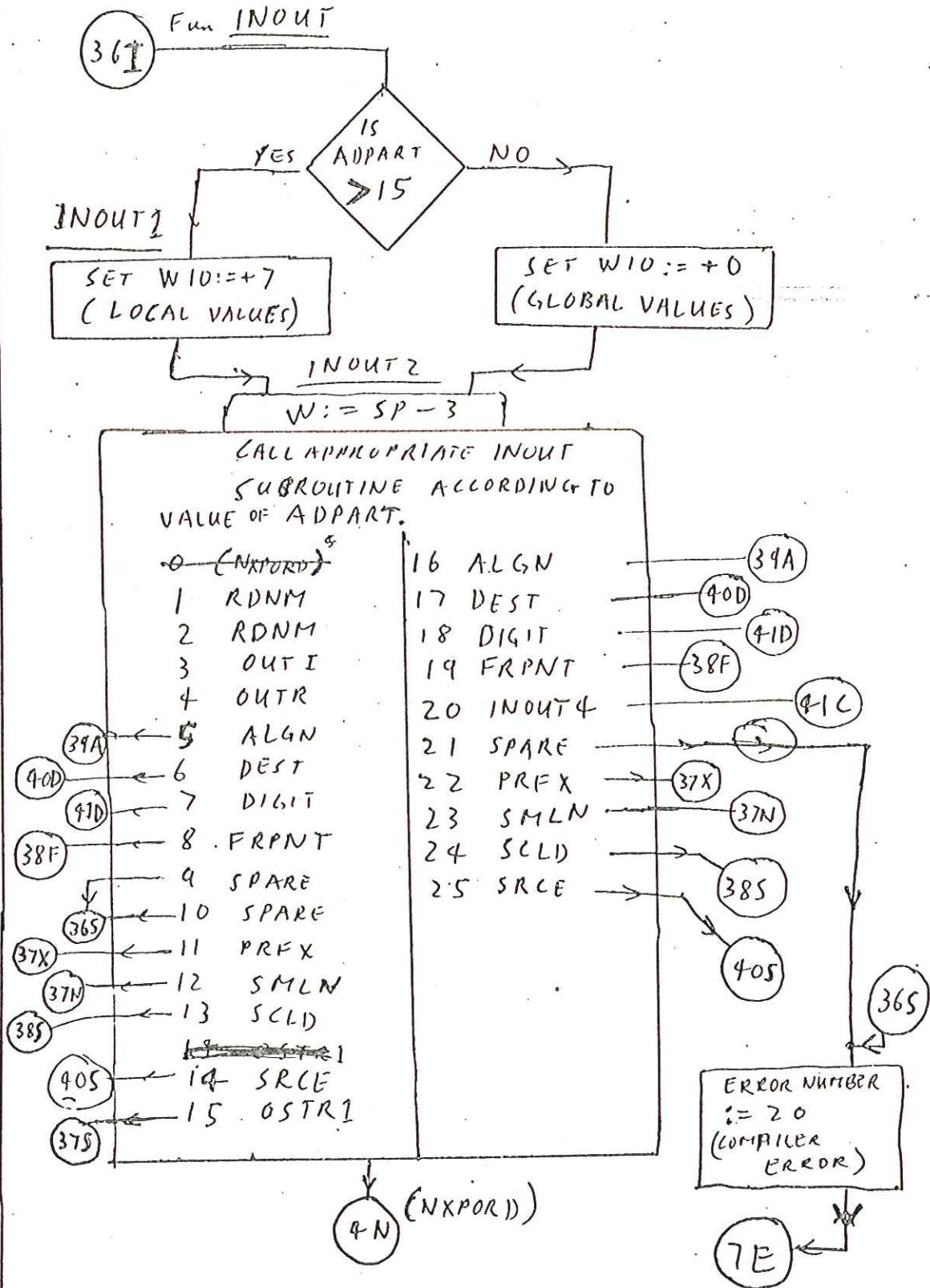


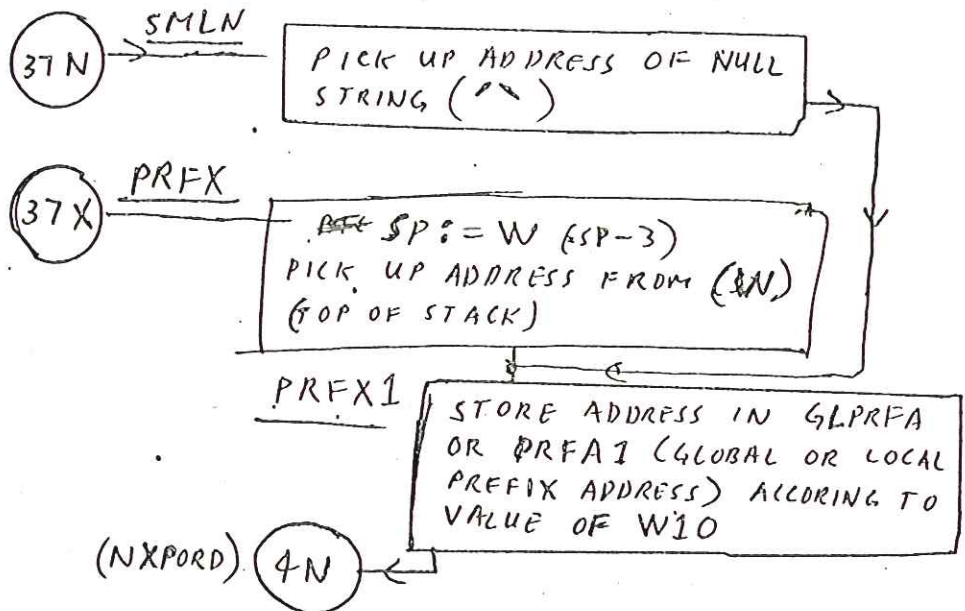
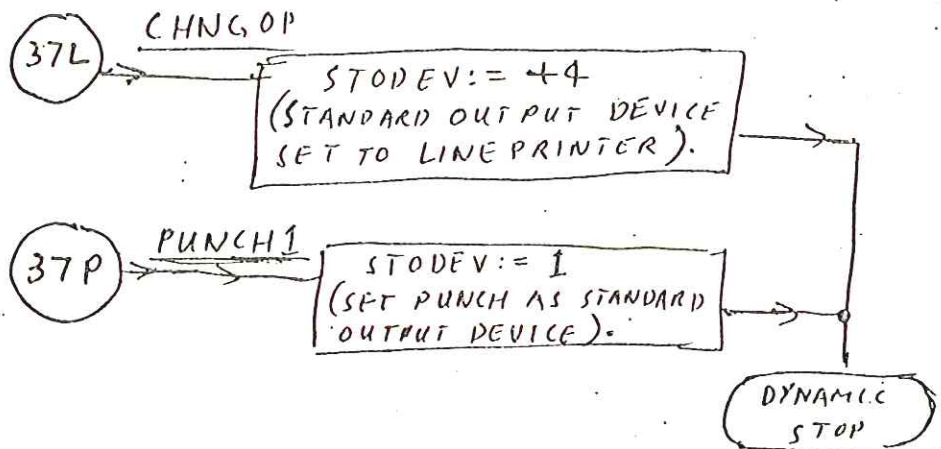
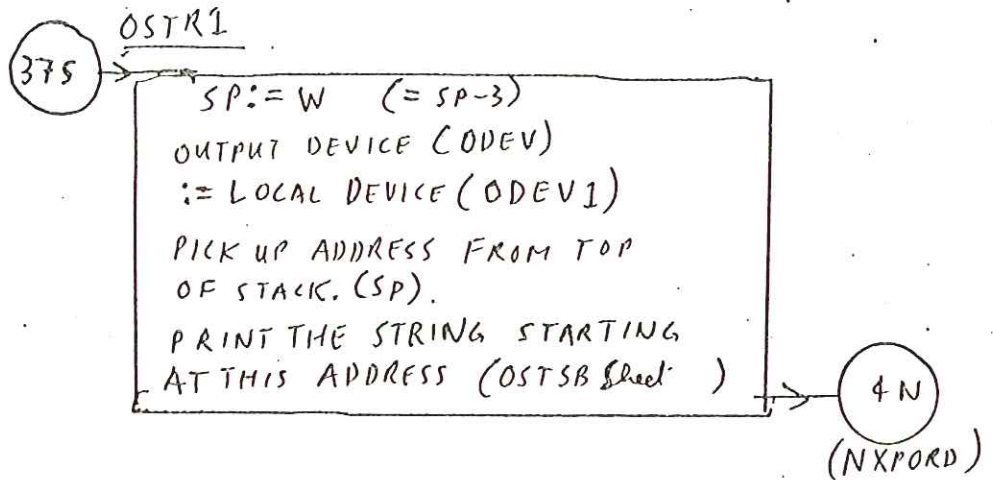


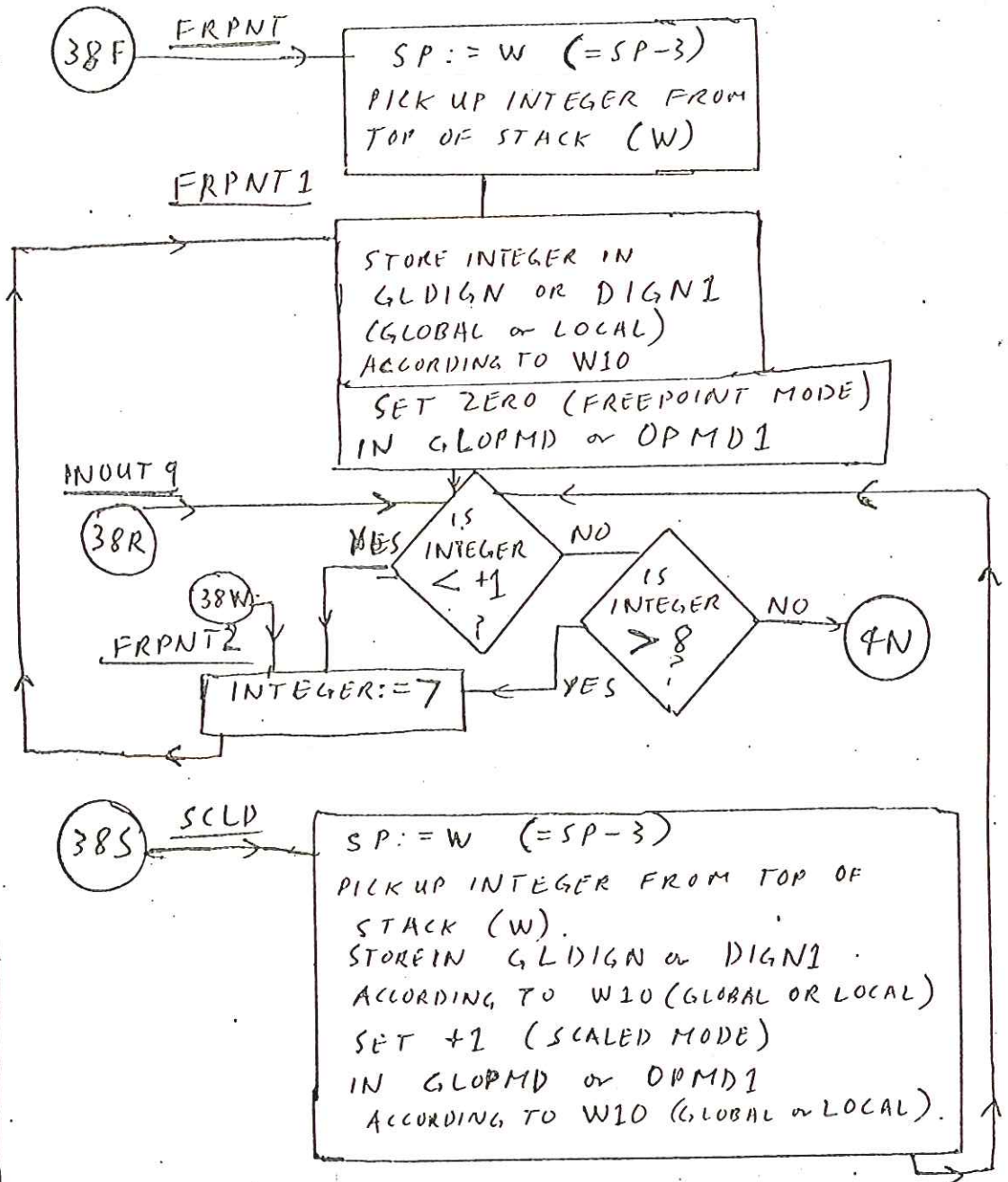


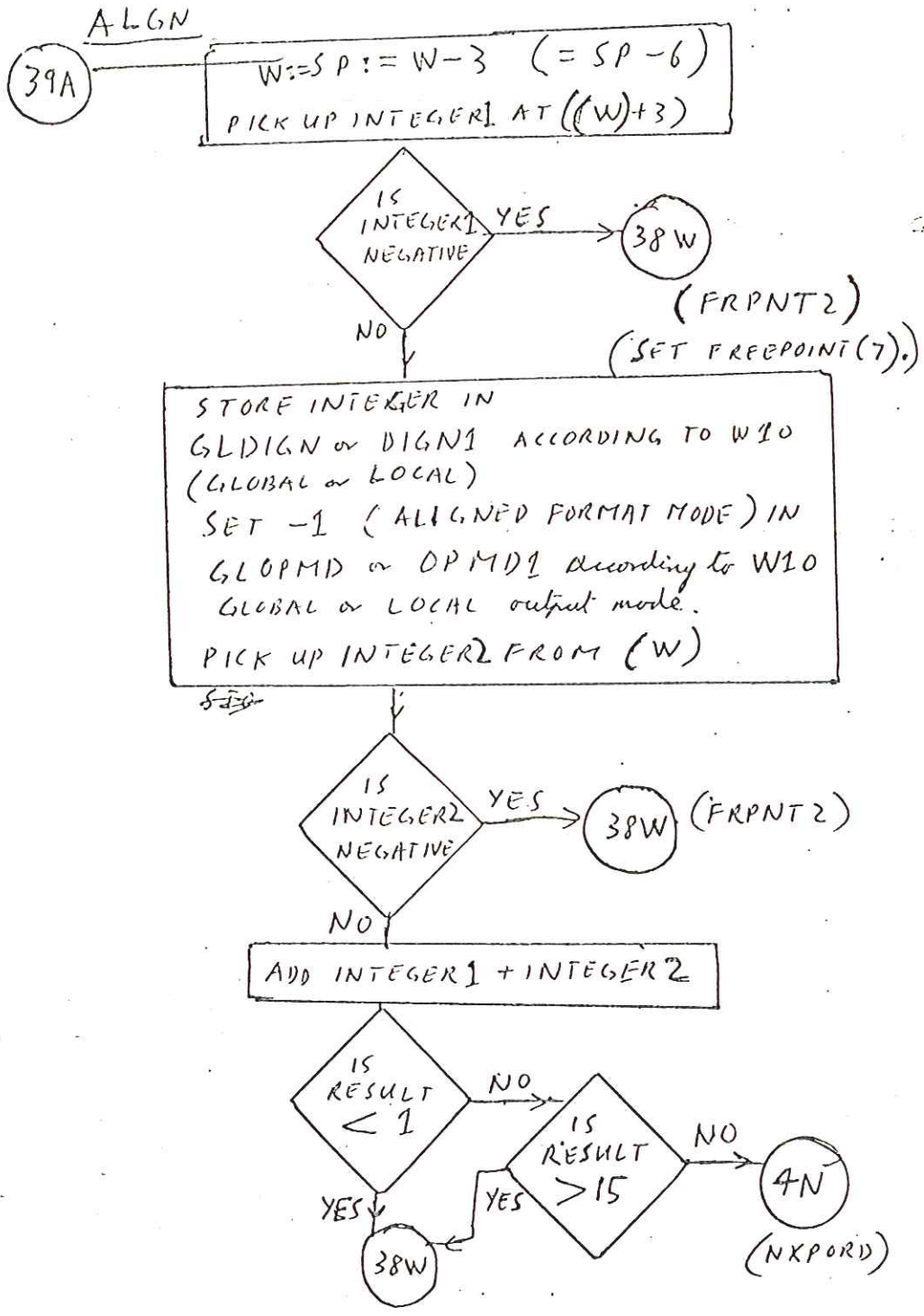


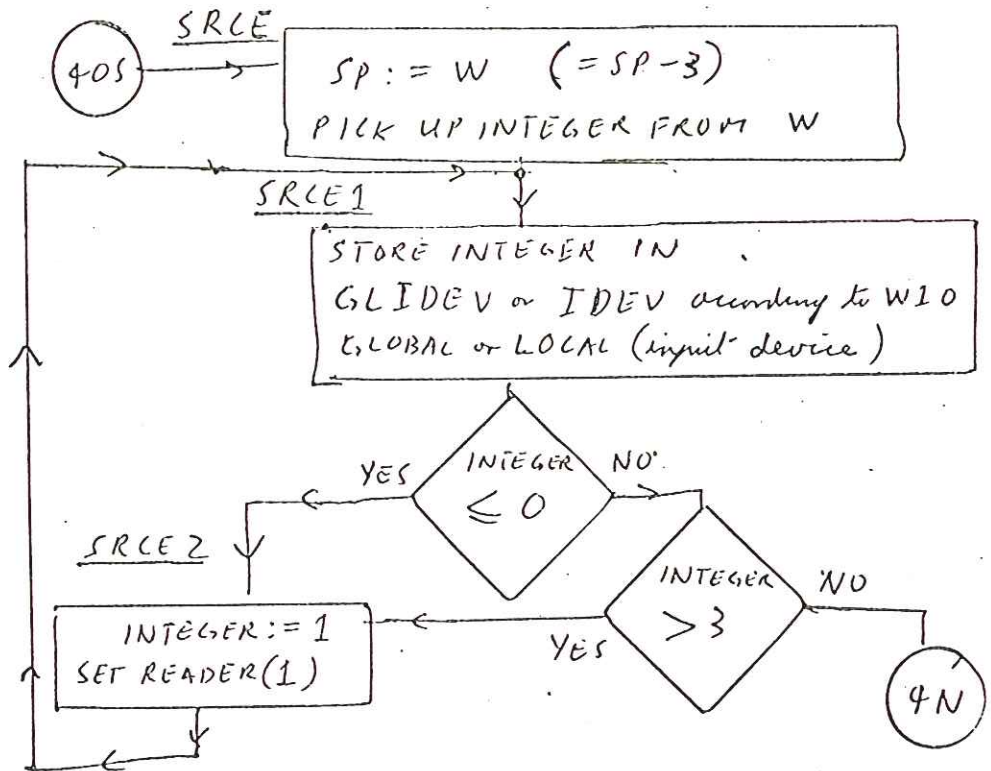
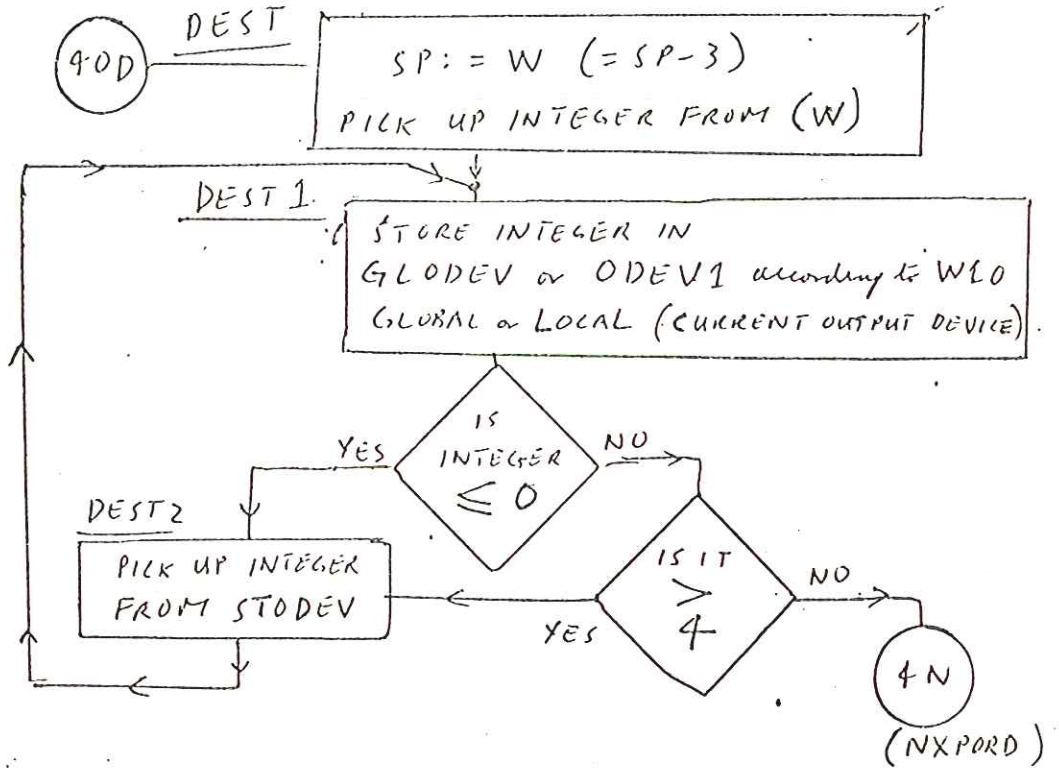












DIGIT

