

## ภาคผนวก B : PL/I-80 STATEMENTS

รูปแบบของคำสั่ง PL/I-80 ข้างล่างนี้เรียงลำดับตามตัวอักษร และตามด้วยเลขบอกหัวข้อที่ตรงกันภายในวงเล็บ ทุกคำสั่งอาจจะมี label prefixes นำหน้าได้ แต่ในรูปแบบที่เขียนต่อไปนี้ จะไม่ใส่ label prefixes ไว้ ยกเว้นคำสั่งนั้นต้องมี label prefixes จึงจะใส่ไว้

## B.1 คำสั่ง ALLOCATE (6.5)

ALLOCATE based-variable SET (pointer-variable);

ตัวอย่าง

DCL A CHAR(16) BASED(P),

P POINTER;

ALLOCATE A SET(P);

## B.2 คำสั่ง ASSIGNMENT (7)

variable = expression;

ตัวอย่าง

B = C\*D;

SUBSTR(A,5,2) = 'FF';

## B.3 คำสั่ง BEGIN (2.3)

BEGIN;

## B.4 คำสั่ง CALL (8.11)

CALL procname [(arg1,...,argN)];

ตัวอย่าง

CALL P1;

CALL MAX3(A,B,C);

## B.5 คำสั่ง CLOSE (9.6)

CLOSE FILE (fname);

ตัวอย่าง

```
CLOSE FILE (INP);
```

```
CLOSE FILE (OUT);
```

## B.6 คำสั่ง DECLARE (5.1)

```
DECLARE [DCL [level] name [attribute-list] ...
        [, [level] name [attribute-list]]];
```

ตัวอย่าง

```
DCL A FIXED;
```

```
DCL 1 B,
```

```
2 C_NAME CHAR(20),
```

```
2 D_ADDRESS,
```

```
3 STREET CHAR(20),
```

```
3 CITYST CHAR(20),
```

```
3 ZIP CHAR(5);
```

```
DCL ZZ(10) FIXED;
```

```
DCL A FIXED ENTTERNAL;
```

## B.7 คำสั่ง DO (2.2, 8.3)

```
DO [control-var] spec;
```

เมื่อ spec อาจมีรูปแบบอย่างใดอย่างหนึ่งข้างล่างนี้

```
[start-exp [TO end-exp] [BY incr-exp]] [WHILE (cond)]
```

```
[start-exp [BY incr-exp] [TO end-exp]] [WHILE (cond)]
```

```
[start-exp [REPEAT (repeat-exp)]] [WHILE (cond)]
```

ตัวอย่าง

```
DO I=0;
```

```
DO WHILE (A<B);
```

```
DO I = 1 TO 10;
```

```
DO K = 10 TO 0 BY -2 WHILE (A<B);
```

```
DO P = START REPEAT P-> NEXT WHILE(P~=NULL);
```

B.8 คำสั่ง END (2.2, 2.3, 8.12)

```
END[label];
```

ตัวอย่าง

```
END;
```

```
END FIRST;
```

B.9 คำสั่ง FORMAT (10.9)

```
label: FORMAT(format-list);
```

ตัวอย่าง

```
L1: FORMAT(A(5));
```

```
L2: FORMAT(F(3),X(2),F(5,2),SKIP,A(10));
```

```
L3: FORMAT(10B4(2));
```

B.10 คำสั่ง FREE (6.3)

```
FREE [pointer-variable ->] based-variable;
```

ตัวอย่าง

```
FREE A;
```

```
FREE P->A;
```

B.11 คำสั่ง GET EDIT (10.10)

```
GET [FILE(fname)][SKIP[(nl)]] [EDIT(input-list)
```

```
(format-list)];
```

ตัวอย่าง

```
GET EDIT (A,B,C)(3F5,2));
```

```
GET FILE (INP) EDIT ((D(I) DO I = 1 TO 5))(A);
```

B.12 คำสั่ง GET LIST (10.2)

```
GET [FILE(fname)][SKIP[(nl)]] [LIST(input-list)];
```

ตัวอย่าง

```
GET LIST (P,Q,R);
```

B.13 คำสั่ง GO TO (8.1)

```
GOTO|GO TO label-constant|label-variable;
```

ตัวอย่าง

```
GO TO LABEL_5;
```

```
GO TO TAB(K);
```

B.14 คำสั่ง IF (8.2)

```
IF cond THEN action [ELSE [action2]];
```

ตัวอย่าง

```
IF A=2 THEN B=A**2;
```

```
ELSE;
```

```
IF I>K THEN L=L+1;
```

```
ELSE L=L+3;
```

B.15 คำสั่ง EMPTY (8.2)

```
;
```

ตัวอย่าง

```
;
```

```
ELSE;
```

B.16 คำสั่ง ON (8.5)

```
ON condition ON-unit
```

ตัวอย่าง

```

ON ENDFILE(INP)

    BEGIN;

    PUT LIST('END OF INPUT');

    STOP;

    END;

ON ERROR PUT LIST (ONCODE(.));

```

## B.17 คำสั่ง OPEN (9.3)

```
OPEN FILE(fname)[file-attributes];
```

ตัวอย่าง

```

OPEN FILE (INP) INPUT;

OPEN FILE (SYSPRINT) OUTPUT;

```

## B.18 คำสั่ง PROCEDURE (8.12)

```

procname: PROCEDURE|PROC[(parml,...,parmN)]

    [OPTIONS(MAIN)][RETURNS(attribute-list)][RECURSIVE];

```

ตัวอย่าง

```

P1: PROC(A,B,C);

P2: PROCEDURE(ST) RETURNS(FLOAT);

P3: PROC(N) RETURNS(FIXED BIN) RECURSIVE;

P4: PROCEDURE OPTIONS(MAIN);

```

## B.19 คำสั่ง PUT EDIT (10.11)

```

PUT [FILE(fname)][SKIP[n1]][PAGE[(p)]]

    [EDIT(output-list)(format-list)];

```

ตัวอย่าง

```
PUT EDIT (P,Q,R)(F(5,2),X(3)2E(10,2));
```

```
PUT EDIT ((Y(I) DO I=1 TO 10))(A);
```

## B.20 คำสั่ง PUT LIST (10.3)

```
PUT [FILE(fname)][SKIP[(n1)]] [PAGE [(p)]]
```

```
[LIST (output-list)];
```

ตัวอย่าง

```
PUT LIST (A,B,C);
```

```
PUT FILE (F) LIST ((Z(I) DO I =1 TO 10));
```

## B.21 คำสั่ง READ (สำหรับแฟ้มข้อมูลชนิด stream) (10.13)

```
READ [FILE(fname)] INTO(v);
```

ตัวอย่าง

```
DCL (VV,S) CHAR(200) VAR;
```

```
READ INTO (VV);
```

```
READ FILE (INP) INTO (S);
```

## B.22 คำสั่ง READ (สำหรับเรคคอร์ดชนิด sequential) (11.1)

```
READ FILE(fname) INTO (X);
```

ตัวอย่าง

```
READ FILE(INP) INTO(XX);
```

## B.23 คำสั่ง READ (ที่มี KEYTO) (11.4)

```
READ FILE(fname) INTO(x) KEYTO(keyto);
```

ตัวอย่าง

```
READ FILE(INP) INTO(Z) KEYTO(IKEY);
```

B.24 คำสั่ง READ (ที่มี KEY) (11.3)

```
READ FILE(fname) INTO(x) KEY (ikey);
```

ตัวอย่าง

```
READ FILE(INP) INTO (STRUC) KEY (IKEY);
```

B.25 คำสั่ง RETURN (8.13)

```
RETURN [(exp)];
```

ตัวอย่าง

```
RETURN;
```

```
RETURN (X);
```

```
RETURN (A**2);
```

B.26 คำสั่ง REVERT (8.7)

```
REVERT condition;
```

ตัวอย่าง

```
REVERT ERROR;
```

```
REVERT ENDFILE;
```

B.27 คำสั่ง SIGNAL (8.6)

```
SIGNAL condition;
```

ตัวอย่าง

```
SIGNAL ERROR;
```

```
SIGNAL ENDFILE;
```

B.28 คำสั่ง STOP (8.15)

```
STOP;
```

B.29 คำสั่ง WRITE (กับแฟ้มข้อมูลชนิด stream) (10.14)

```
WRITE [FILE(fname)] FROM (v);
```

ตัวอย่าง

```
DCL (A,B) CHAR(200) VAR;
WRITE FILE(OUTPUT) FROM (A);
WRITE FROM (B);
```

B.30 คำสั่ง WRITE (สำหรับเรคคอร์ดชนิด sequential) (11.2)

```
WRITE FILE(fname) FROM (c);
```

ตัวอย่าง

```
WRITE FILE (OUTP) FROM (A);
WRITE FILE (F) FROM (STRUC);
```

B.31 คำสั่ง WRITE (แบบ KEYFROM) (11.5)

```
WRITE FILE(fname) FROM(c) KEYFROM(ikey)
```

ตัวอย่าง

```
WRITE FILE(KP) FROM(REC) KEYFROM(IKEY);
```