8. Flow Chart

전산 프로그램(High Level Language)에서 주로 사용되는 방식이다.
공장자동화 분야에서는 거의 사용되지 않는 방식이지만 End User들이 요청하는 경우도 있으니 Flow Chart가 뭔지는 알고 있자.
예를 제시한 전산 프로그램(High Level Language)은 Qbasic으로 작성한 것이다.
왜냐하면 Modicon PLC Program이 공장자동화에서 가장 기초가 되는 Program이듯이, 전산 프로그램(High Level Language)에서 가장 기본이 되는 Program은 Basic Program이라고 생각하기 때문이다.

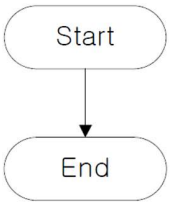
전산 Program에 High Level Language에는 여러 종류가 있다. Basic, Fortran, Cobol, C, Pascal, Assembler, Java, Pyton…..
하지만 같은 방식으로 Program을 처리하지 않는다.
작성한 Program의 처리 방법에 따라 Interpreter 방식과 Compiler 방식이 있다.

Interpreter : 한줄 한줄씩 각 줄마다 바로 바로 실행하는 방식이다. (Basic, Pyton)
Compiler : 전체 완성된 Program을 한꺼번에 처리하는 방식이다. (Fortran, Cobol, C++, Assebbler, …)

사람들이 착각하는 것이 Assembler가 기계어라고 착각하는 것이다.
기계어에 가깝게 Program하는 방식이지 기계어는 아니다.
기계는 Binary Code만 인식한다.
더 길이 파해치면 Binary Code의 값은 인식하지 못한다.
전기가 통했는지 통하지 않았는지의 결과만 인식한다.
결국 Assember로 작성한 Source Program을 Compile하면 확장자가 .exe로된 실행 File이 만들어지며, 이 만들어진 실행 File을
ExetoBin을 사용하여 기계어(Binary Code)로 바꿔야 하며, 이를 다시 ROM(Read Only Memory) Writer를 이용하여 CPU Memory 영역에 Program을 Write해야 한다.
아직도 PCB Board로 제어하는 업체들은 이런 방법을 사용하여 제어를 하고 있다.
Assembler로 작성한 Program을 기계어로 바꾸면 기계어와 같은 방식으로 처리하기 때문에 처리 속도가 빠르다는 장점이 있다.
그래서 처리 속도를 높이기 위해 C언어에서 Assembler를 부분적으로 사용하는 경우가 있다.
그렇다고 모든 언어가 Assembler를 지원하지는 않는다.

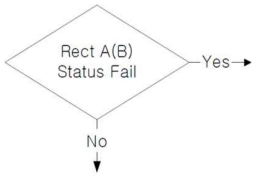
프로그램을 처음 배우는 사람들이 Basic이나 Pyton이 쉬운 이유가 Interpreter 방식이므로 작성된 Program을 곧바로 실행하여 문제점를 빨리 찾을 수 있기 때문이다.
Modicon PLC에서 사용하는 Software인 Modsoft도 Interpreter 방식이라 Online으로 Program 수정과 동시에 결과를 확인해 볼 수 있기 때문이다.
다른 PLC Program에서는 Online으로 Program 수정했더라도 Update를 해야만 변경한 Program이 반영되므로 그때서야 결과를 확인해 볼 수 있다.

High Level Language에서 Compiler 방식을 사용하는 Program은 모두 작성한 후에 Compile하여 실행하기 때문에 오류가 발생하는 문제를 찾기가 어렵다.
물론 사람에 따라 Compiler 방식이 쉽다고 얘기하는 사람도 있을 것이다.



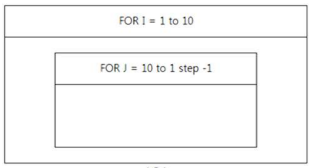
시작과 끝을 명시한다.

조건문(IF문)

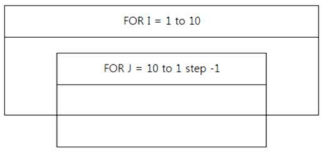


인쇄 (Print문)

반복 Roof문(FOR~NEXT문)

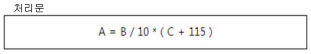


(O)



(X)

처리문



Subroutine Call & Return

A = B 수학에서는 A와 B가 같다는 말이지만 전산 Program에서는 다르게 표현한다.
B에 저장되어 있는 값을 A에 넣는걸 말한다. (B → A)

```
IF (조건)
  THEN
    A
  ELSE
    B
ENDIF
```

IF 명령어의 기본 구성이다.
(조건)은 0(False),1(True)로 표현이 가능한 조건이어야만 한다. A=B(x:틀린 조건)가 아니고 A==B(o:맞는 조건) 맞는 조건문의 표현 방식이다.
그 이유는 A=B는 연산문이고, A==B 조건문의 표현 방식이기 때문이다.
(조건)이 1(True)이면 THEN 이하 조건을 수행하고, (조건)이 0(False)이면 ELSE 이하 조건을 수행한다.

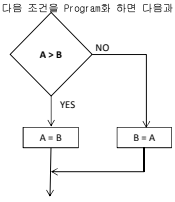
Print 명령어

```
PRINT "*"              PRINT "*";
PRINT "*"              PRINT "*"
```

세미콜론(;)이 있는가 없는가의 차이점이다.
하지만 결과는 확연히 다르다?

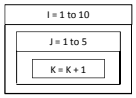
```
*                      **
*
```

다음 조건을 Program화 하면 다음과 같다.




```
IF A > B
  THEN
    A = B
  ELSE
    B = A
ENDIF
```

다음의 반복 Roof문을 Program화 하면 다음과 같다.




```
FOR I = 1 to 10
  FOR J = 1 to 5
    K = K + 1
  NEXT J
NEXT I
```

For~Next문을 다양한 활용법을 알아보자.
다음에 나오는 문제들을 스스로 풀면서 For~Next문의 다양한 활용을 배우기 바란다.

다음 문제는 Basic으로 만든 Program이다. 특별한 테크닉을 말하기 위함이 아니다. 단지 사고의 확장성을 키우라는 의미다.

◈ 8-1 For ~ Next문을 사용하여 다음과 같이 출력되도록 만들어라.

```
**********

FOR I = 1 to 10        FOR I = 10 to 1 Step -1
  PRINT "*";             PRINT "*";
NEXT I                 NEXT I
```

◈ 8-2 For ~ Next문을 사용하여 다음과 같이 출력되도록 만들어라.

```
**********
**********

FOR I = 1 to 2
  FOR J = 1 to 10
    PRINT "*";
  NEXT J
  PRINT
NEXT I
```

◈ 8-3 For ~ Next문을 사용하여 다음과 같이 출력되도록 만들어라.

```
**********
**********
**********
**********
**********

FOR I = 1 to 5
  FOR J = 1 to 10
    PRINT "*";
  NEXT J
  PRINT
NEXT I
```

◈ 8-4 For ~ Next문을 사용하여 다음과 같이 출력되도록 만들어라.

```
*
**
***
****
*****

FOR I = 1 to 5
  FOR J = 1 to I
    PRINT "*";
  NEXT J
  PRINT
NEXT I
```

◈ 8-5 For ~ Next문을 사용하여 다음과 같이 출력되도록 만들어라.

```
*****
****
***
**
*

FOR I = 5 to 1 Step -1
  FOR J = 1 to I
    PRINT "*";
  NEXT J
  PRINT
NEXT I
```

◈ 8-6 For ~ Next문을 사용하여 다음과 같이 출력되도록 만들어라. *과 * 사이에는 공배가 있음을 유의하라.

```
    *
   * *
  * * *
 * * * *
* * * * *

FOR I  = 1 to 5
  PRINT SPACE$(5-I);
  FOR J = 1 to (I-1)
    PRINT "*";SPACE$(1);
  NEXT J
  PRINT "*"
NEXT I
```

◈ 8-7 For ~ Next문을 사용하여 다음과 같이 출력되도록 만들어라. *과 * 사이에는 공배가 있음을 유의하라.

```
* * * * *
 * * * *
  * * *
   * *
    *

FOR I  = 5 to 1 STEP -1
  PRINT SPACE$(5-I);
  FOR J = 1 to (I-1)
    PRINT "*";SPACE$(1);
  NEXT J
  PRINT "*"
NEXT I
```

◈ 8-8 1-10까지의 합을 다음과 같이 출력하도록 Program을 만드시오.

```
1+2+3+4+5+6+7+8+9+10=55

FOR I = 1 to 9
  X = X + I
  PRINT I;
  PRINT "+";
NEXT I
PRINT I"=";
X = X + I
PRINT X
```

◈ 8-9 한 학급에 10명의 학생이 있다.
이번에 시험을 봤는데 각 학생의 점수가 아래에 있다.

```
A(0) = 94
A(1) = 88
A(2) = 45
A(3) = 65
A(4) = 98
A(5) = 32
A(6) = 92
A(7) = 90
A(8) = 77
A(9) = 89
```

성적이 높은 순서에서 낮은 순서로 정렬하라.

```
A(0) = 94
A(1) = 88
A(2) = 45
A(3) = 65
A(4) = 98
A(5) = 32
A(6) = 92
A(7) = 90
A(8) = 77
A(9) = 89
FOR I = 0 to 9
  FOR J = (I+1) to 9
    IF A(J) > A(I) THEN
      Memory = A(I)
      A(I) = A(J)
      A(J) = Memory
    ENDIF
  NEXT J
  PRINT A(I)
NEXT I
```