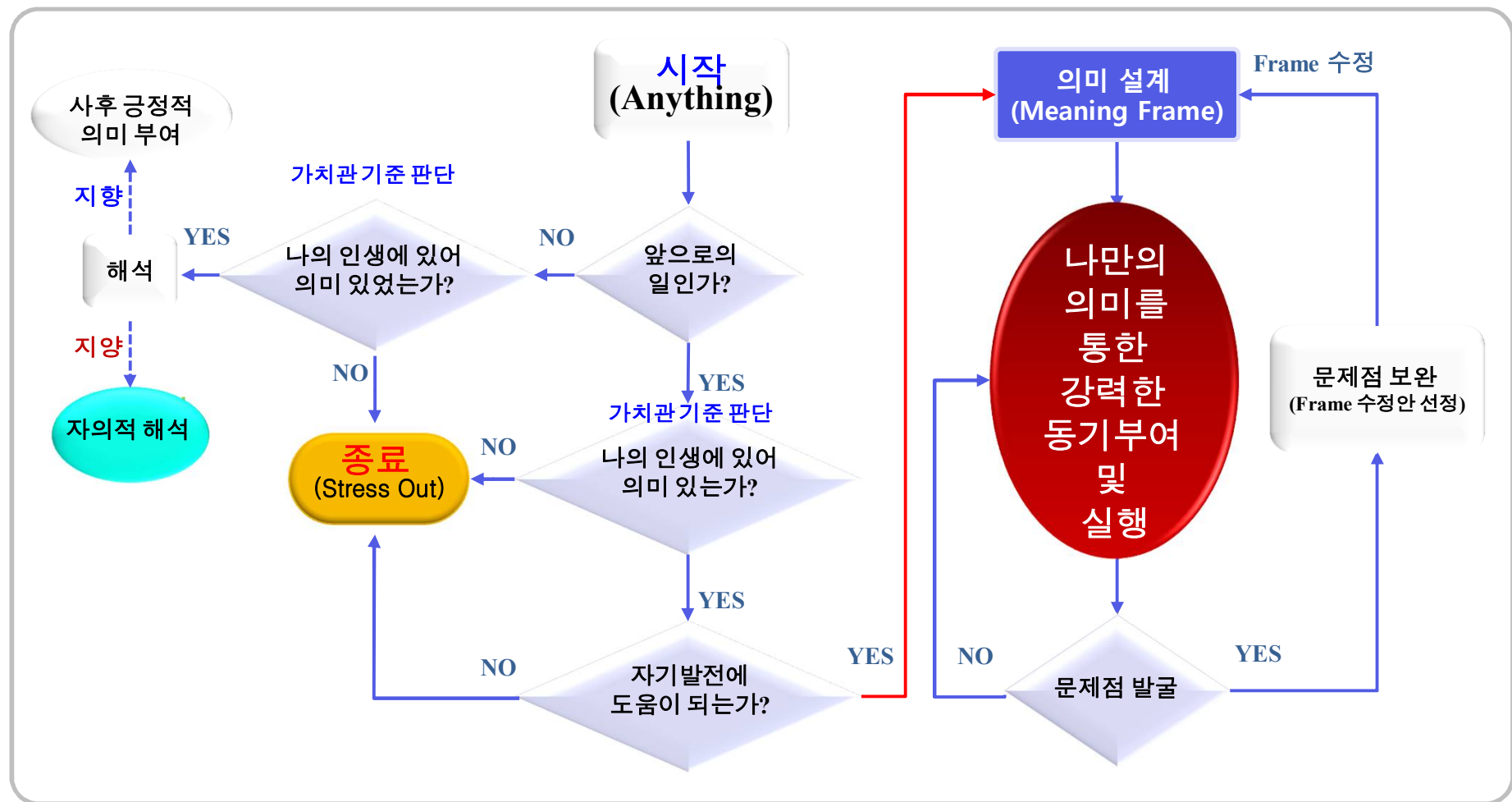

C 언어

조건문

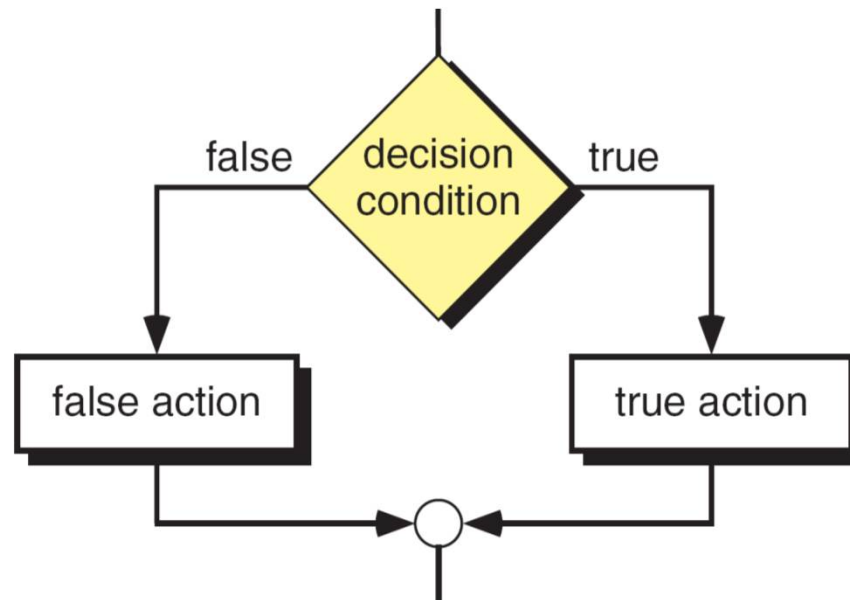
- **if ~ else**
- **switch ~ case**



Two-Way Selection

컴퓨터의 basic decision statement는 two-way selection 임
Decision condition의 결과에 따라 제어 흐름 바뀜
two-way decision의 로직

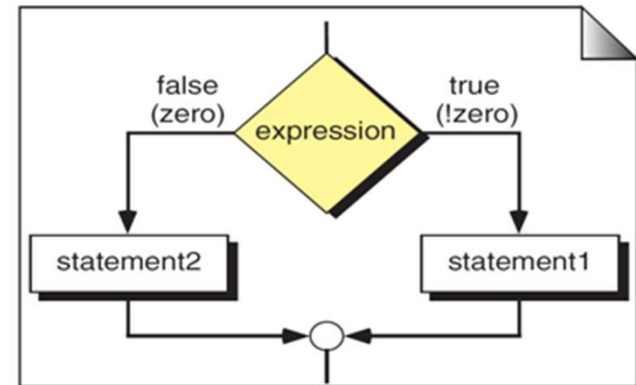
- condition이 **true**이면 true action을 취함
- condition이 **false**이면 false action을 취함



조건문 if, else

if (condition expression) statements

- “실행 조건”을 만족(true)하는 경우
 - ◆ “statement-1” 실행
- “실행 조건”을 불만족(false)하는 경우
 - ◆ “statement-2” 실행



(a) Logical Flow

```
if( 조건 )  
{  
    조건 만족 시 실행  
    일명 "이거"  
}  
else  
{  
    조건 불만족 시 실행  
    일명 "저거"  
}
```

"조건" 이 만족되는 경우 "이거"실행
만족되지 않으면 "저거" 실행

조건문 if, else

예제 프로그램 - if...else 문의 예

```
#include <stdio.h>
```

```
int main(void) {  
    int a;  
    int b;
```

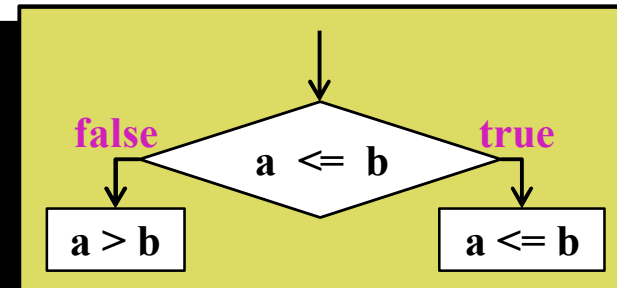
```
    printf("Please enter two integers: ");  
    scanf("%d %d", &a, &b);
```

```
    if(a <= b)  
        printf("%d <= %d\n", a, b);  
    else  
        printf("%d > %d\n", a, b);
```

```
    return 0;
```

```
}
```

사용자로부터 임의의
값을 입력 받는다.



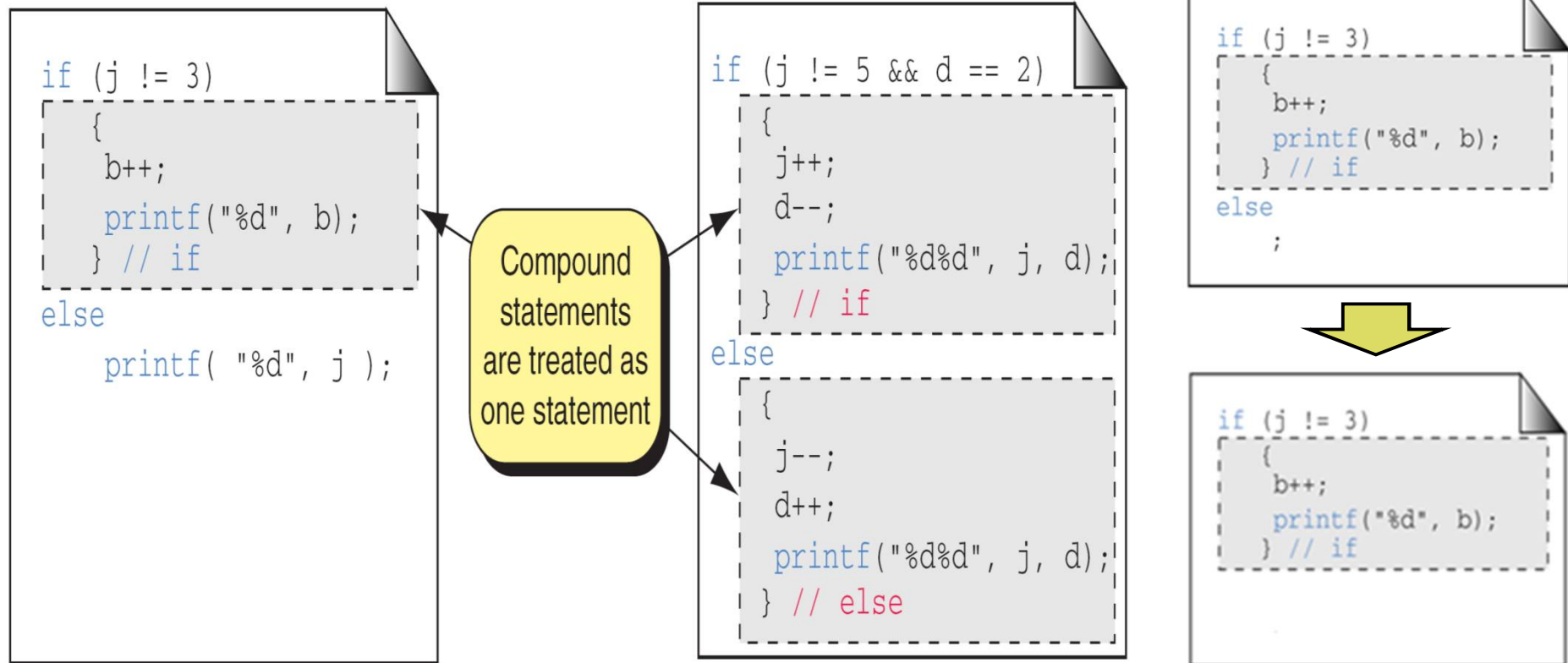
```
Please enter two integers: 21 5  
21 > 5
```

조건문 if, else

Conditional expression을 검사 후

- 수행하는 statement가 두 줄 이상 일 때는 { }로 묶어줌
- else부분에 들어갈 statement가 없을 경우
 - ◆ else 부분 생략 가능 (Null else statement)

조건문 if, else



<Compound statements in an if...else>

<Null else statement>

Nested if

예제 프로그램 – Nested if statement문의 예

```
#include <stdio.h>

int main(void){
    int a;
    int b;

    printf("Please enter two integers: ");
    scanf("%d %d",&a, &b);

    if(a <= b)
        if(a < b)
            printf("%d < %d\n",a,b);
        else
            printf("%d == %d\n",a,b);
    else
        printf("%d > %d\n",a,b);

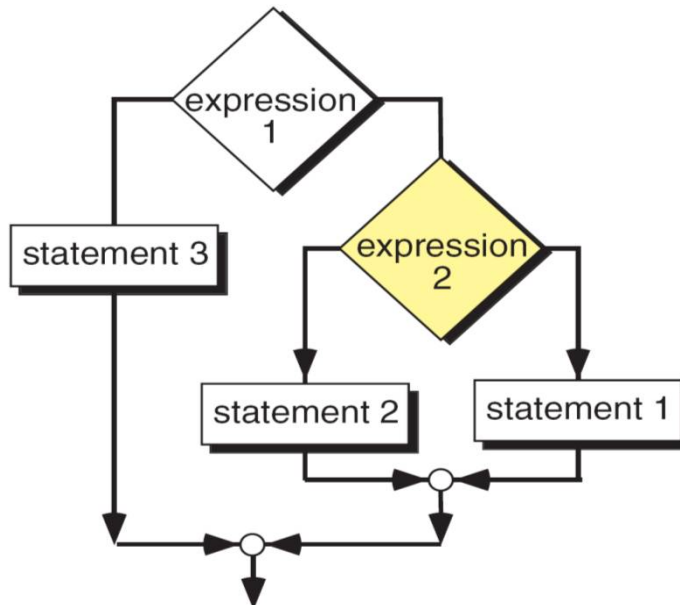
    return 0;
}
```

```
Please enter two integers: 4 7
4 < 7
```


Nested if

Nested if Statements

- if 문 내부에 다른 if문이 중첩(nested)되어 나오는 문장
 - ◆ 조건 내에 다른 조건을 넣고 싶을 때 유용한 표현
- 다음은 nested if문의 Logic 흐름과 실제 표기법임



(a) Logic flow

```
if (expression 1)
    if (expression 2)
        statement 1
    else
        statement 2
else
    statement 3
```

(b) Code

Nested if

- if...else 구문에서 여러 개의 if문이 나타난 경우
else문이 생략되면 의미가 불분명해 질 수 있음
 - ◆ 컴퓨터는 if문에 가장 가까운 else를 하나의 쌍으로 처리
 - ◆ Dangling else problem 발생
 - [해결책] 의도한 if문을 compound statement { }로 처리

하나의
if..else쌍으
로 인식

```
if(alpha == 3)
    if(beta == 4)
        printf("alpha = 3, beta = 4\n");
else
    printf("alpha beta not valid\n");
```



Compound
statement
{ }

```
if(alpha == 3)
{
    if(beta == 4)
        printf("alpha = 3, beta = 4\n");
}
else
    printf("alpha beta not valid\n");
```

실습예제2

if~else를 이용하여 정수 3개를 입력 받고 그 중 가장 큰 수를 출력하는 프로그램을 작성하시오

- INPUT

- ◆ 세수를 입력하세요. 1, 4, 5

- OUTPUT

- ◆ 입력 받은 수중 가장 큰 수는 5입니다.

- 실행결과예시



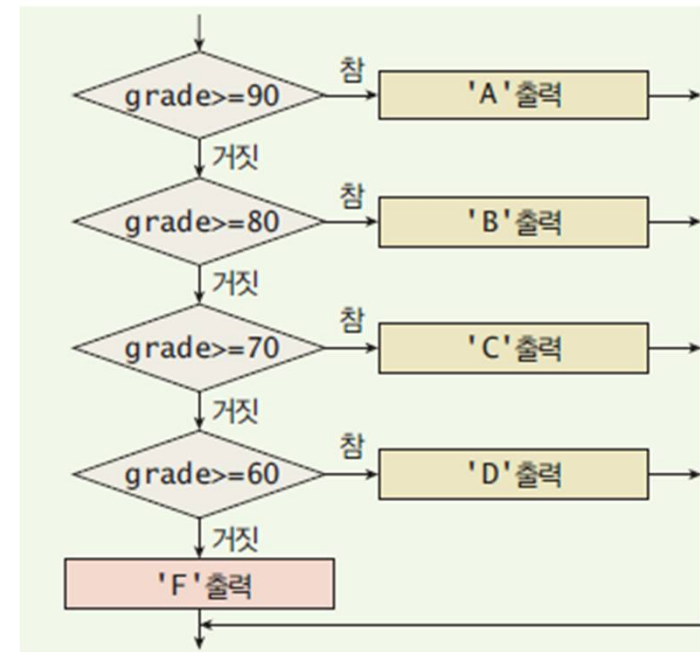
실습예제2

하나의 점수를 입력 받고, 입력 받은 점수에 해당하는 학점을 출력하는 프로그램을 작성하시오.

입력 받는 정수는 0~100까지이고 90점 이상이면 A, 80점 이상이면 B, 70점 이상이면 C, 60점 이상이면 D, 나머지는 F

■ 실행결과예시

Score : **95**
Grade : A



Else if문

예제 프로그램 – else if 문의 예

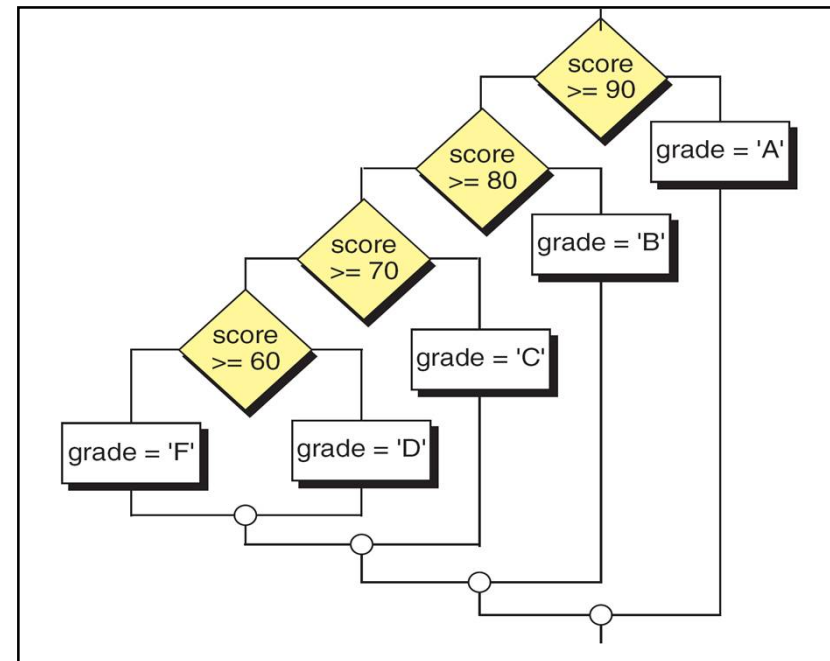
```
#include <stdio.h>

int main(void) {
    int score=0;
    char grade;

    printf("Enter the test score(0-100); ");
    scanf("%d",&score);

    if(score >= 90)
        grade='A';
    else if(score >= 80)
        grade='B';
    else if(score >= 70)
        grade='C';
    else if(score >= 60)
        grade='D';
    else
        grade='F';

    printf("The grade is %c\n", grade);
    return 0;
}
```



```
Enter the test score (0-100): 91
The grade is A
```

Else if문

다중판단을 할 경우에 사용 되는 형태
(if...else if...else)

- if의 expression을 검사하여 참일 경우
 - ◆ 바로 아래의 statement1 실행
- expression이 거짓일 경우
 - ◆ 다음의 else if 로 넘어가서 조건 검사
- 마지막 else는 위의 조건들이 모두 참이 아닐 경우에 statement4 실행
- 맞는 조건을 찾은 다음에는 나머지 조건검사 수행 하지 않음

```
If(expression)  
    statement1  
else if(expression)  
    statement2  
else if(expression)  
    statement3  
else  
    statement4
```

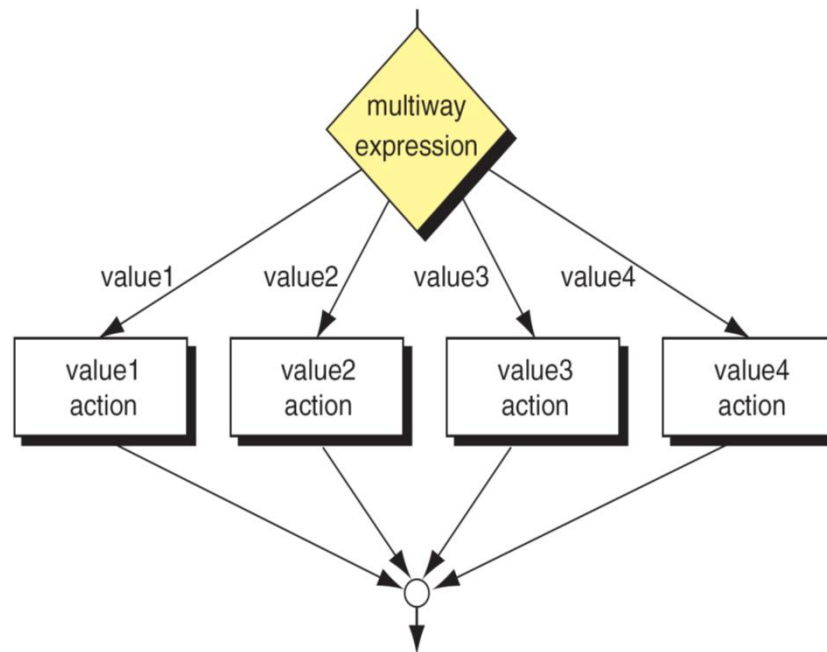
Multiway Selection

Multiway selection에는 크게 switch 문과 else if 문이 존재

Switch문

- switch 문은 여러 개의 선택을 처리하는 구문
- switch 이후의 괄호 ()사이의 표현 식의 값 중에서 case의 값과 일치하는 것을 처리
 - ◆ Expression과 value는 반드시 정수나 정수수식이어야 한다.

switch 문 로직



switch문

switch문의 구조

```
switch ( n )  
{
```

case 1 :

n이 1인 경우 실행되는 영역
break;

case 2 :

n이 2인 경우 실행되는 영역
break;

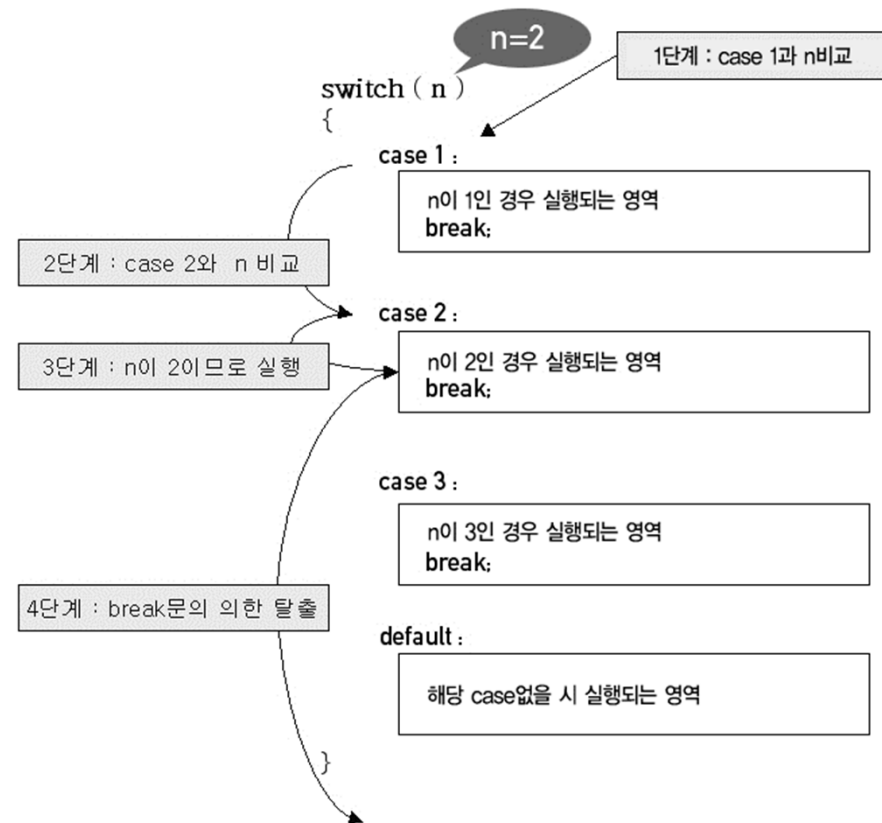
case 3 :

n이 3인 경우 실행되는 영역
break;

default :

해당 case 없을 시 실행되는 영역

```
}
```



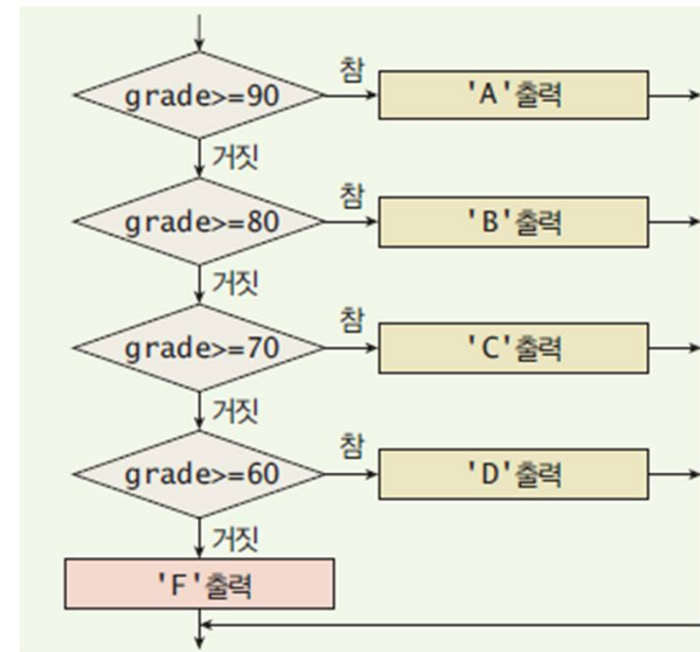
실습예제

하나의 점수를 입력 받고, 입력 받은 점수에 해당하는 학점을 출력하는 프로그램을 작성하시오.

입력 받는 정수는 0~100까지이고 90점 이상이면 A, 80점 이상이면 B, 70점 이상이면 C, 60점 이상이면 D, 나머지는 F

■ 실행결과예시

Score : **95**
Grade : A



switch문

예제 프로그램 – switch문의 예

```
#include <stdio.h>

char scoreToGrade (int score);

int main(void)
{
    int score;
    char grade;

    printf("Enter the test score (0-100): ");
    scanf("%d", &score);

    grade = scoreToGrade(score);
    printf("The grade is : %c\n", grade);

    return 0;
}
```

```
char scoreToGrade(int score)
{
    char grade;
    int temp;

    temp = score / 10;
    switch(temp)
    {
        case 10 :
        case 9 : grade='A'; break;
        case 8 : grade='B'; break;
        case 7 : grade='C'; break;
        case 6 : grade='D'; break;
        default : grade='F';
    }
    return grade;
}
```

```
Enter the test score (0-100): 89
The grade is : B
Enter the test score (0-100): 97
The grade is : A
```

Switch문

실행순서

- switch문의 표현식(expression) 계산
- 계산된 값과 일치하는 상수 값(constant)을 갖는 case의 값을 위에서부터 찾음
- 해당 case내부의 문장(statement) 실행
- case 내부에서 break를 만나면 switch문 종료
- 일치된 case문을 만나지 못하여 default를 만나면 default내부의 문장 실행

Default : 반드시 있어야 하지는 않지만 선택되지 않은 모든 경우를 나타낸다.

```
switch (expression)
{
    case constant-1: statement
                    :
                    statement
    case constant-2: statement
                    :
                    statement
    case constant-n: statement
                    :
                    statement
    default         : statement
                    :
                    statement
} // end switch
```

실습예제

Switch를 이용하여 각 달의 일수를 출력하는 프로그램을 작성하시오
2월은 29일까지 4,6,9,11월은 30일까지 나머지 달은 31일까지

- INPUT

- ◆ 일수를 알고 싶은 달은 : 4

- OUTPUT

- ◆ 월의 날수는 30일 입니다.

- 실행결과예시



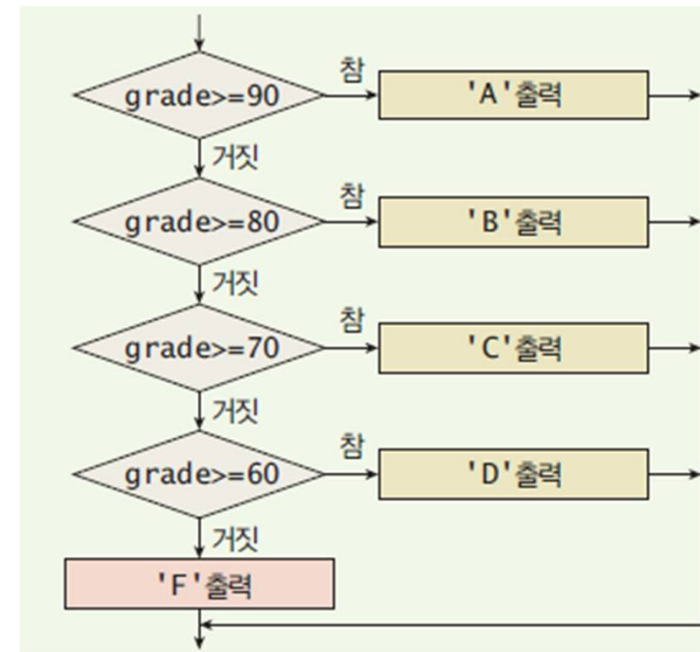
실습예제

Switch를 이용하여 하나의 점수를 입력 받고, 입력 받은 점수에 해당하는 학점을 출력하는 프로그램을 작성하시오.

입력 받는 정수는 0~100까지이고 90점 이상이면 A, 80점 이상이면 B, 70점 이상이면 C, 60점 이상이면 D, 나머지는 F

■ 실행결과예시

Score : **95**
Grade : A



Switch문

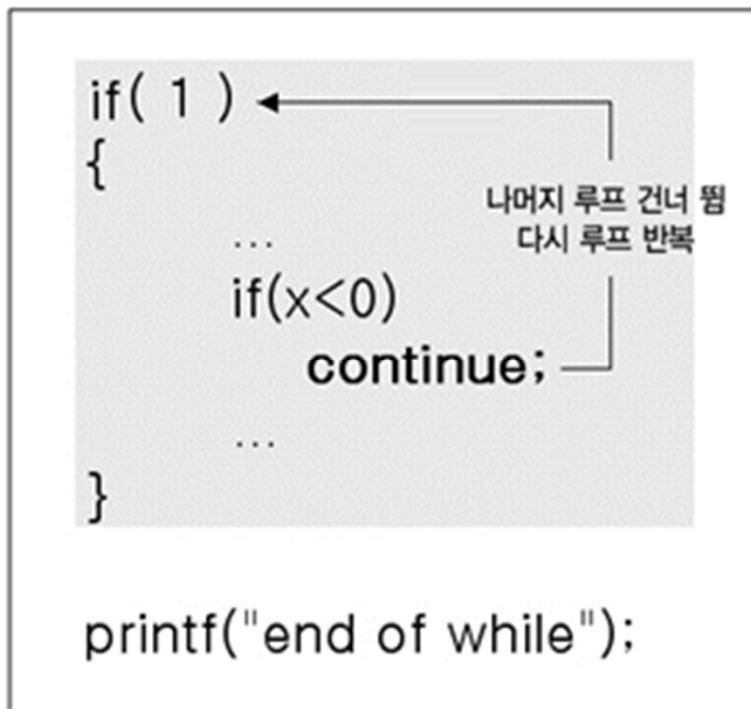
break문

- switch 문에서 break 문을 만나면 무조건 switch 문을 종료
- 그러나 switch 문의 case 문 내부에 break 문이 없다면
 - ◆ 일치하는 case 문을 실행하고,
 - ◆ 계속해서 break 문을 만나기 전까지 무조건 다음 case 문 내부의 문장을 실행
- break 이후 실행되는 곳은 지금 빠져 나온 structure의 바로 뒷 부분
- break 문을 주로 사용하는 용도
 - ◆ Loop으로부터의 탈출
 - ◆ switch structure에서 나머지 실행문장을 건너 뛰기 위함

break vs continue

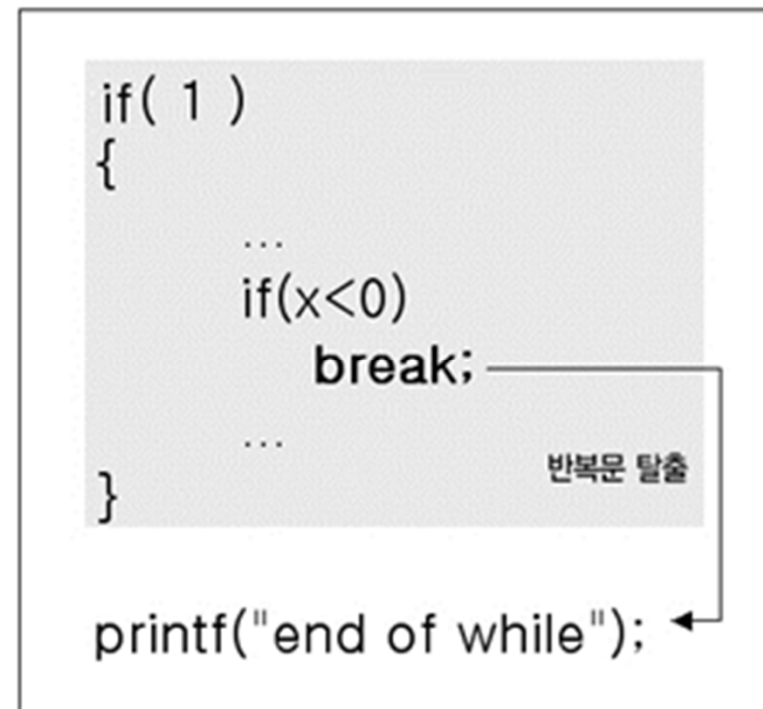
다음으로 **continue!**(생략)

- 다음 번으로 넘어갈 때 사용



이제 그만 **break!**(탈출)

- 반복문을 빠져 나올 때 사용



Switch vs Else if

switch문과 else if문 모두 조건에 따라 프로그램의 흐름을 분기시키는 목적으로 사용

switch문에는 비교연산이 올 수 없음

switch문으로 구현된 내용은 else if문으로 구현 가능

- 반대의 경우는 구현이 안될 경우가 생김
 - ◆ 예 : 비교연산

분기의 횟수가 많을 수록 switch문이 else if문에 비해 간결한 코드 구현이 가능

삼항 연산자

조건 연산자(삼항 연산자)

- if~else문을 간결히 표현하는데 사용될 수 있다.

조건 ? A : B

조건이 true인 경우 A를 반환
조건이 false인 경우 B를 반환

$X = (y < 0) ? 10 : 20;$

"y<0"이 true면 10이 반환되어 x에 대입
"y<0"이 false면 20이 반환되어 x에 대입

$X = (y > 0) ? a * b : a / b;$

"y>0"이 true면 a*b이 연산결과 x에 대입
"y>0"이 false면 a/b이 연산결과 x에 대입

실습예제

다섯 개의 정수를 입력 받고, 이 숫자들을 역순으로 출력하는 프로그램을 작성하시오.

- 실행결과예시

```
input : 88 34 1 -345 23  
output : 23 -345 1 34 88
```

실습예제

키보드에서 영문자 하나를 읽어서 모음과 자음을 구분하고 개수도 세는 프로그램을 작성하여 보자.

- 실행결과예시

