

제 6 장 조건

- 01 제어문 개요
- 02 조건에 따른 선택 if 문
- 03 다양한 선택 switch 문



학습목표

- ▶ 제어문에 대하여 다음 용어를 이해하고 설명할 수 있다.
 - 순차적 실행과 제어문의 차이
 - 조건 선택, 반복, 분기 처리
- ▶ 조건에 따른 선택을 구현하는 if 문에 대하여 이해하고 구현할 수 있다.
 - if, if else, if else if, nested if 구문을 이해하고 샘플 소스 구현
 - 다양한 조건을 연산자로 구성
 - switch 문을 이해하고 성적처리 프로그램 구현
- ▶ 다양한 선택을 한 눈에 알아볼 수 있는 switch 문에 대하여 이해하고 구현할 수 있다.

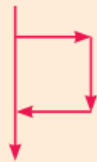
제어문 종류

- **조건 선택**
 - 두개 또는 여러 개 중의 하나를 선택하는 구조
- **반복**
 - 반복 몸체인 여러 문장을 여러 번 반복하는 구조
- **분기 처리**
 - 정해진 장소로 이동하는 구조

조건 선택

조건에 대한 선택 구문

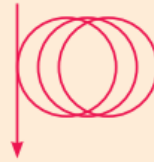
- if
- if else
- if else if
- nested if
- switch



반복 순환

반복 조건에 따라 일정 영역의 구문 반복

- for
- while
- do while



분기처리

지정된 영역으로 실행을 이동하는 구문

- break
- continue
- goto
- return



조건에 따른 선택 if 문

• 조건에 따른 결정

평균평점 ≥ 3.5

대학 A는 평균평점이 3.5는 넘어야 장학금을 받을 수 있다.



석차 $\leq 0.05 * \text{학생수}$

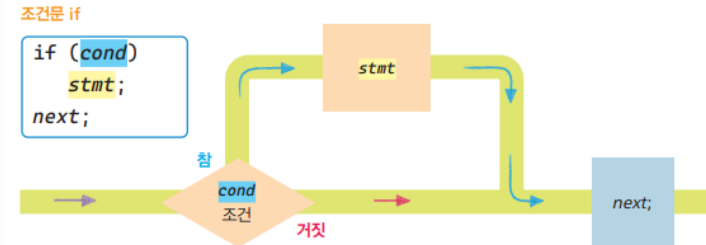
대학 B는 학과 석차가 상위 5% 이어야 장학금을 받을 수 있다고 한다.

| 조건 선택의 예 | 기준 변수 | 조건 표현의 의사코드 |
|---|----------------------------------|--|
| 온도가 30도 이상이면 "폭염 주의"를 출력 | 온도 temperature | 만일 (temperature ≥ 30) printf("폭염 주의"); |
| 낮은 혈압이 90이상이면 "고혈압 초기"로 진단 | 혈압 low_pressure | 만일 (low_pressure ≥ 90) printf("고혈압 초기"); |
| 속도가 40km와 60km 사이이면 "적정 속도"라고 출력 | 속도 speed | 만일 (40 \leq speed && speed \leq 60) printf("적정 속도"); |
| 운전면허 필기시험에서 60점 이상이면 "합격", 아니면 "불합격" 출력 | 시험 성적 point | 만일 (point ≥ 60) printf("면허시험 합격"); 아니면 printf("면허시험 불합격"); |
| 남성일 경우 체력 테스트에서 80점 이상이면 "합격"이고, 아니면 "불합격", 여성이면 70점 이상이면 "합격", 아니면 "불합격" | 성별 type 체력 점수 point | 만일 남성이면 (type == 1) 만일 (point ≥ 80) printf("남성: 합격"); 아니면 printf("남성: 불합격"); 아니고 만일 여성이면 (type == 2) 만일 (point ≥ 70) printf("여성: 합격"); 아니면 printf("여성: 불합격"); |

조건에 따른 선택 if 문장

• if 문

- 조건식이 0이 아니면(참) 문장을 실행
- 0이면(거짓) 문장을 실행하지 않음



블록이면 다음과 같이 블록 시작 {을 if열에 맞추고 조건 문장들을 들여쓰기 한다. 그리고 마지막에 행에 다시 블록 종료 }를 시작 열에 맞춘다.

조건문 if

```
if (cond)
    stmt;
next;
```

cond가 만족되면 실행되는 문장

```
if (grade >= 3.2)
{
    printf("회사에 지원할 수 있습니다.\n");
}
printf("졸업을 축하합니다.\n");
```

• 논리 오류와 문법 오류

```
if (grade >= 3.2);
    printf("회사에 지원할 수 있습니다.\n");

printf("졸업을 축하합니다.\n");
```

```
if grade >= 3.2
    printf("회사에 지원할 수 있습니다.\n");

printf("졸업을 축하합니다.\n");
```

현재 온도에 따른 폭염 주의 발령

실습예제 6-1

Prj01

01basicif.c

현재 온도에 따른 폭염 주의 발령

난이도: ★

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03
04 int main(void)
05 {
06     double temperature;
07
08     printf("현재 온도 입력: ");
09     scanf("%lf", &temperature);
10
11     if (temperature >= 30.0)
12     {
13         printf("폭염 주의보를 발령합니다.\n");
14         printf("건강에 유의하세요.\n");
15     }
16     printf("현재 온도는 섭씨 %.2f 입니다.\n", temperature);
17
18     return 0;
19 }
```

다음과 같이 블록 {의 시작을 조건식 오른쪽에 작성 하기도 함

```
if (temperature >= 32.0) {
    printf("폭염 주의보를 발령합니다.\n");
    printf("건강에 유의하세요.\n");
}
```

결과

현재 온도 입력: 29.3
현재 온도는 섭씨 29.30 입니다.

현재 온도 입력: 34.678
폭염 주의보를 발령합니다.
건강에 유의하세요.
현재 온도는 섭씨 34.68 입니다.

조건 만족 여부에 대한 선택 if else

- 조건을 만족하면 문장1을 실행
 - 조건을 만족하지 않으면 문장2를 실행하는 문장

조건문 if else

```
if (cond)
    stmt1;
else
    stmt2;
next;
```

```
if (n % 2 == 0)
    printf("짝수");
else
    printf("홀수");
printf("입니다.\n");
```

```
if (n % 2)
    printf("홀수");
else
    printf("짝수");
printf("입니다.\n");
```

| 조건식 | 설명 | 예 | |
|------------------|----------------------------------|---|---|
| (n != 0) (n) | n이 0이 아니어야 참인 연산식이므로 연산식 (n)과 같음 | <pre>if (n % 2 != 0) printf("홀수"); else printf("짝수");</pre> | <pre>if (n % 2) printf("홀수"); else printf("짝수");</pre> |
| (n == 0) (!n) | n이 0이어야 참인 연산식이므로 연산식 (!n)과 같음 | <pre>if (n % 2 == 0) printf("짝수"); else printf("홀수");</pre> | <pre>if (!(n % 2)) printf("짝수"); else printf("홀수");</pre> |

조건 if else로 짝수와 홀수 판정

실습예제 6-2

Prj02

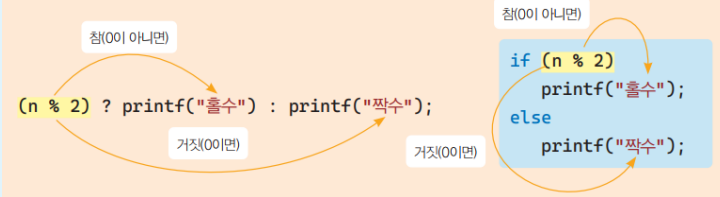
02evenifelse.c

조건 if else로 짝수와 홀수 판정

난이도: ★

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03
04 int main(void)
05 {
06     int n;
07     printf("정수 입력: ");
08     scanf("%d", &n);
09
10     if (n % 2) // if (n % 2 != 0)
11     {
12         printf("홀수\n");
13     }
14     else
15     {
16         printf("짝수\n");
17     }
18
19     //조건연산자 이용
20     (n % 2) ? printf("홀수\n") : printf("짝수\n");
21
22     return 0;
23 }
```

조건 문장 if는 이미 배운 조건연산자와 매우 유사해서 좀더 편리하게 활용될 수 있다. 즉 다음 오른쪽의 if 문은 좌측의 조건연산자를 이용한 연산식으로 간단히 대체될 수 있다.



if else 문에서 조건식에 따른 실행문이 하나이더라도 블록을 구성해도 좋음

```
if (n % 2 != 0)
    printf("홀수");
else
    printf("짝수");
```

else 앞의 블록에 세미콜론; 을 붙이면 문법 오류가 발생하니 주의가 필요하다.

조건 if else 문에서 if 블록 이후, else 앞에 세미콜론; 을 붙이면 문법 오류가 발생한다. 또한 else 바로 뒤에 세미콜론; 을 붙이면 문법 오류는 없으나 논리 오류가 발생해 예상치 못한 결과가 발생하니 주의가 필요하다.

정상 실행

```
if (n % 2)
{
    printf("홀수\n");
}
else
{
    printf("짝수\n");
}
```

문법 오류

```
if (n % 2)
{
    printf("홀수\n");
};
else
{
    printf("짝수\n");
}
```

논리 오류

```
if (n % 2)
{
    printf("홀수\n");
}
else
{
    printf("짝수\n");
}
```

else 문에 실행할 문장이 없이 종료되고, 이후 블록은 무조건 실행되는 논리 오류가 발생함

결과

정수 입력: 5

홀수

홀수

정수 입력: 6

짝수

짝수

반복된 조건에 따른 선택 if else if

이 조건식은 첫 if의 조건식인 (point >= 90)이 만족되지 않고 체크되는 것이므로 결국 (!(point >= 90) && (point >= 80))이므로 80 이상에서 90 미만인 조건 (90 > point && point >= 80)이 만족된다.

```
if (point >= 90)
    printf("A\n");
else if (point >= 80)
    printf("B\n");
else if (point >= 70)
    printf("C\n");
else if (point >= 60)
    printf("D\n");
else
    printf("F\n");
```

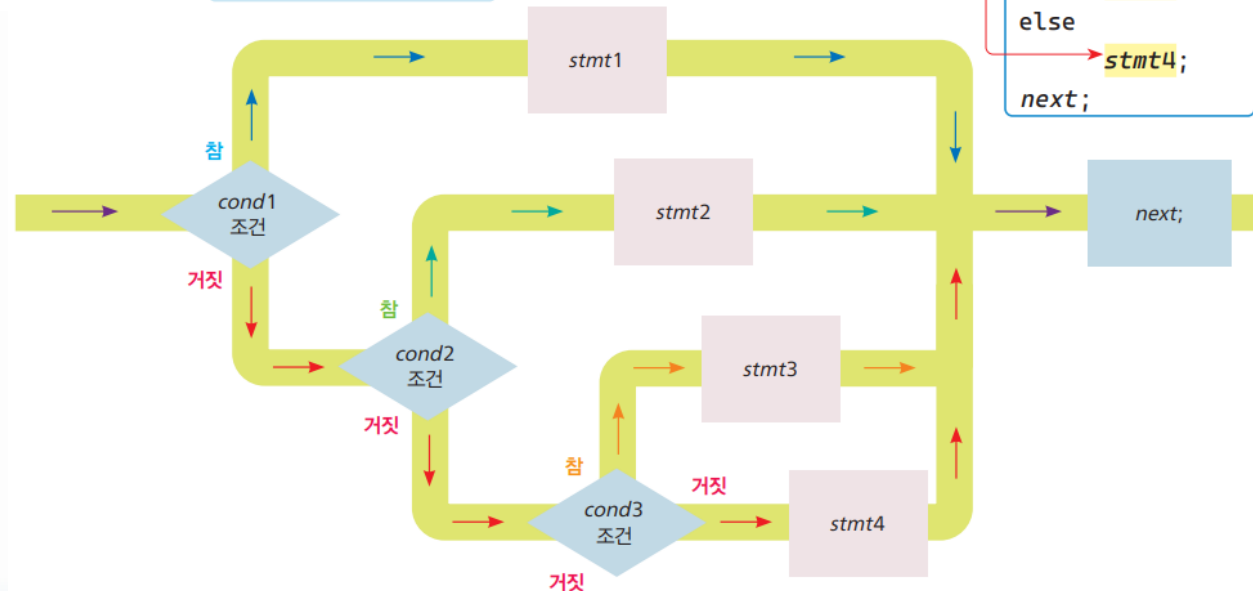
필요하면 이와 같이 블록
사용이 가능하다.

```
if (point >= 90)
{
    printf("A\n");
}
else if (point >= 80)
{
    printf("B\n");
}
else if (point >= 70)
{
    printf("C\n");
}
else if (point >= 60)
{
    printf("D\n");
}
else
{
    printf("F\n");
}
```

조건문 if else if

```
if (cond1)
    stmt1;
else if (cond2)
    stmt2;
else if (cond3)
    stmt3;
else
    stmt4;
next;
```

여러 개의 문장중
단 하나만 실행된다.



조건 if else 문으로 평균평점에 따른 적정 구문 출력

실습예제 6-3

Prj03 03gradeif.c 조건 if else 문으로 평균평점에 따른 적정 구문 출력 난이도: ★

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03
04 int main(void)
05 {
06     double gpa;
07
08     printf("평균평점 입력: ");
09     scanf("%lf", &gpa);
10
11     if (gpa >= 4.3)
12         printf("최우등\n");
13     else if (gpa >= 3.8)
14         printf("우등\n");
15     else if (gpa >= 3.0)
16         printf("우수\n");
17     else
18         printf("3.0 미만\n");
19
20     return 0;
21 }
```

| 조건 | 단독 조건식 | 출력 |
|-------------------|-------------------------|--------|
| 평균평점 >= 4.3 | gpa >= 4.3 | 최우등 |
| 4.3 > 평균평점 >= 3.8 | gpa < 4.3 && gpa >= 3.8 | 우등 |
| 3.8 > 평균평점 >= 3.0 | gpa < 3.8 && gpa >= 3.0 | 우수 |
| 3.0 > 평균평점 | gpa < 3.0 | 3.0 미만 |

else if에 의해 if (gpa < 4.3 && gpa >= 3.8)의 의미가 됨

결과

| | |
|---------------------|------------------------|
| 평균평점 입력: 4.3 최우등 | 평균평점 입력: 3.9 우등 |
| 평균평점 입력: 3.3 우수 | 평균평점 입력: 2.7 3.0 미만 |

중첩된 if else 문으로 자동차 면허 합격 여부 판정

실습예제 6-4

Prj04

04nestedif.c

중첩된 if else 문으로 자동차 면허 합격 여부 판정

난이도: ★

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03
04 int main(void)
05 {
06     int type, point;
07
08     printf("번호를 선택: 1(1종면허), 2(2종면허): ");
09     scanf("%d", &type);
10     printf("필기시험 점수 입력: ");
11     scanf("%d", &point);
12
13     if (type == 1)
14     {
15         if (point >= 70)
16             printf("1종면허 합격\n");
17         else
18             printf("1종면허 불합격\n");
19     }
20     else if (type == 2)
21     {
22         if (point >= 60)
23             printf("2종면허 합격\n");
24         else
25             printf("2종면허 불합격\n");
26     }
27
28     return 0;
29 }
```

결과

번호를 선택: 1(1종면허), 2(2종면허): 1
필기시험 점수 입력: 67
1종면허 불합격

번호를 선택: 1(1종면허), 2(2종면허): 2
필기시험 점수 입력: 58
2종면허 불합격

번호를 선택: 1(1종면허), 2(2종면허): 1
필기시험 점수 입력: 77
1종면허 합격

번호를 선택: 1(1종면허), 2(2종면허): 2
필기시험 점수 입력: 63
2종면허 합격



TIP [코딩 주의] 조건식에서 등호 연산자 ==를 대입 연산자 =으로 잘못 코딩하는 경우

위 소스에서 다음과 같이 조건식 (type == 1)을 실수로 (type = 1)로 잘못 코딩하면 (type = 1)의 결과는 대입 값인 1이므로 if 조건을 항상 실행하는 논리 오류가 발생하니 주의가 필요하다. 초보자들에게 자주 발생하는 오류이니 조심하자.

```
if (type = 1)
{
    ...
}
else if (type == 2)
{
    ...
}
```



TIP [코딩 주의] 조건식에서 등호 ==를 사용한 연산식에서 실수를 사용하는 경우의 문제

다음 코드의 결과는 '이상해요'가 출력된다. 변수 sum에는 float나 double의 문제로 실제 5.1이 아닌 5.1보다 조금 큰 실수가 저장된다. 그러므로 float나 double과 같은 실수를 관계 연산식이나 특히 등호 ==나 부등호 !=를 사용하는 경우 원하는 값의 결과가 발생할 수 있으니 가급적 사용하지 말자.

```
double a = 4.7, b = 0.4;
double sum = a + b;

if (sum == 5.1)
{
    printf("%s\n", "좋아요.");
}
else
{
    printf("%s\n", "이상해요.");
}
printf("%.20f ", sum);
```

이상해요.
5.100000000000000053291

else의 if를 적절히 조합하여 나이에 따른 선택

실습예제 6-5

Prj05

05ageifelse.c

else의 if를 적절히 조합하여 나이에 따른 선택

난이도: ★

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03
04 int main(void)
05 {
06     int age = 0;
07
08     printf("당신의 나이는? ");
09     scanf("%d", &age);
10
11     if (age >= 20)
12     {
13         if (age >= 65)
14             printf("당신은 어르신입니다.\n");
15         else
16             printf("당신은 성인입니다.\n");
17     }
18     else
19     {
20         printf("당신은 미성년자입니다.\n");
21     }
22
23     return 0;
24 }
```

int age = 30; 들어쓰기와 논리 오류

```
if (age >= 20)
if (age >= 65)
    printf("당신은 어르신입니다.\n");
else
    printf("당신은 미성년자입니다.\n");
```

들어쓰기 뿐만 아니라 논리적으로도 오류가 있는 else 문장

20세 미만이면 미성년자이므로 블록 {}이후에 else가 있어야 논리적으로 맞는 문장

```
int age = 30;
if (age >= 20)
{
    if (age >= 65)
        printf("당신은 어르신입니다.\n");
    }
else
    printf("당신은 미성년자입니다.\n");
}
```

20세 미만이면 미성년자이므로 논리적으로 오류가 있는 else 문장

int age = 30; 논리 오류

```
if (age >= 20)
{
    if (age >= 65)
        printf("당신은 어르신입니다.\n");
    else
        printf("당신은 미성년자입니다.\n");
}
```

블록 {}은 없어도 오류는 없으나 소스의 가독성을 높이기 위해 삽입

블록 종료로 이 else는 조건식 (age >= 20)이 거짓인 경우 실행됨

블록 {}은 없어도 오류는 없으나 소스의 가독성을 높이기 위해 삽입

결과

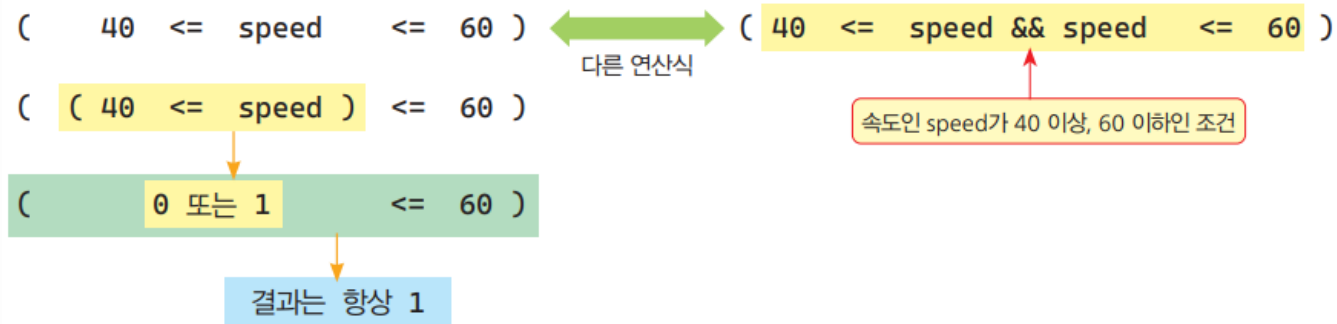
당신의 나이는? 18
당신은 미성년자입니다.

당신의 나이는? 66
당신은 어르신입니다.

당신의 나이는? 22
당신은 성인입니다.

다양한 if 문의 이용과 조건연산자

- 속도 speed가 40km와 60km 사이라는 조건식



| 구현 내용 | 조건연산자 | if |
|--------------|---|--|
| 두 수의 최대값 구하기 | <code>max = x > y ? x : y;</code> | <pre>if (x > y) max = x; else max = y;</pre> |
| 두 수의 최소값 구하기 | <code>min = x > y ? y : x;</code> | <pre>if (x > y) min = y; else min = x;</pre> |
| 절대값 구하기 | <code>abs = x >= 0 ? x : -x;</code> | <pre>if (x >= 0) abs = x; else abs = -x;</pre> |
| 홀수와 짝수 구하기 | <code>a % 2 ? printf("홀수") : printf("짝수");</code> | <pre>if (a % 2) printf("홀수"); else printf("짝수");</pre> |

LAB 표준입력으로 받은 세 정수의 최대값을 출력

- 먼저 조건식 $x > y$ 이 참이면 x와 z의 최대값
 - 거짓이면 y와 z의 최대값을 구해 출력

Lab 6-1 lab1max.c 난이도: ★★

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03
04 int main(void)
05 {
06     int x, y, z;
07     printf("세 정수를 입력: ");
08     scanf("%d %d %d", &x, &y, &z);
09
10     if (x > y)
11     {
12         if ( )
13             printf("최대 수: %d\n", x);
14         else
15             printf("최대 수: %d\n", z);
16     }
17     else
18     {
19         if ( )
20             printf("최대 수: %d\n", y);
21         else
22             printf("최대 수: %d\n", z);
23     }
24
25     return 0;
26 }
```

정답

```
10     if (x > z)
19     if (y > z)
```

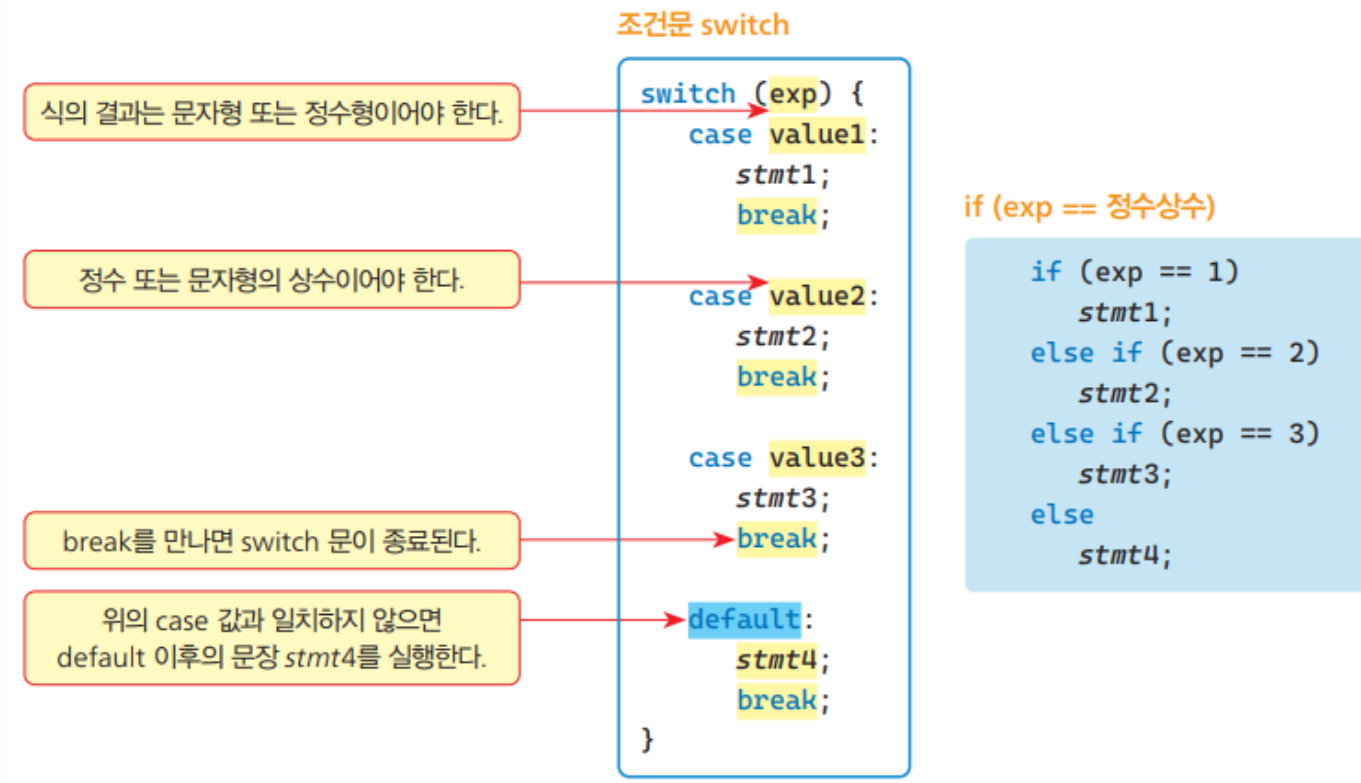
Microsoft Visual Studio 디버그 콘솔

```
세 정수를 입력: 30 15 18
최대 수: 30
```

switch 문장 개요

- switch 문

- 연산식의 결과값에 따라 여러 경로 중에서 하나를 선택하는 선택하는 구문



switch로 두 실수의 사칙연산 수행

실습예제 6-6

Prj06

06arithswitch.c

switch를 사용하여 두 실수의 사칙연산 수행

난이도 ★

```

01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03
04 int main(void)
05 {
06     double x, y, result;
07     int op;
08
09     printf("두 실수 입력: ");
10     scanf("%lf %lf", &x, &y);
11     printf("연산종류 번호선택 1(+), 2(-), 3(*), 4(/): ");
12     scanf("%d", &op);
13
14     switch (op)
15     {
16     case 1:
17         printf("%.2f + %.2f = %.2f\n", x, y, x + y);
18         break;
19     case 2:
20         printf("%.2f - %.2f = %.2f\n", x, y, x - y);
21         break;
22     case 3:
23         printf("%.2f * %.2f = %.2f\n", x, y, x * y);
24         break;
25     case 4:
26         printf("%.2f / %.2f = %.2f\n", x, y, x / y);
27         break;
28     default:
29         printf("번호를 잘못 선택했습니다.\n");
30         break; //생략 가능
31     }
32
33     return 0;
34 }
35

```

만일 op가 2라면 case 2: 내부의 문장을 실행하고 종료된다.

case 1, 2, 3, 4 내부의 break 문은 반드시 필요하다.

case 1, 2, 3, 4 내부의 break 문은 반드시 필요하다.

결과

두 실수 입력: 3.765 6.987
연산종류 번호선택 1(+), 2(-), 3(*), 4(/): 1
3.77 + 6.99 = 10.75

두 실수 입력: 3.986 4.826
연산종류 번호선택 1(+), 2(-), 3(*), 4(/): 3
3.99 * 4.83 = 19.24

두 실수 입력: 4.82 3.987
연산종류 번호선택 1(+), 2(-), 3(*), 4(/): 2
4.82 - 3.99 = 0.83

두 실수 입력: 87.354 6.98
연산종류 번호선택 1(+), 2(-), 3(*), 4(/): 4
87.35 / 6.98 = 12.51

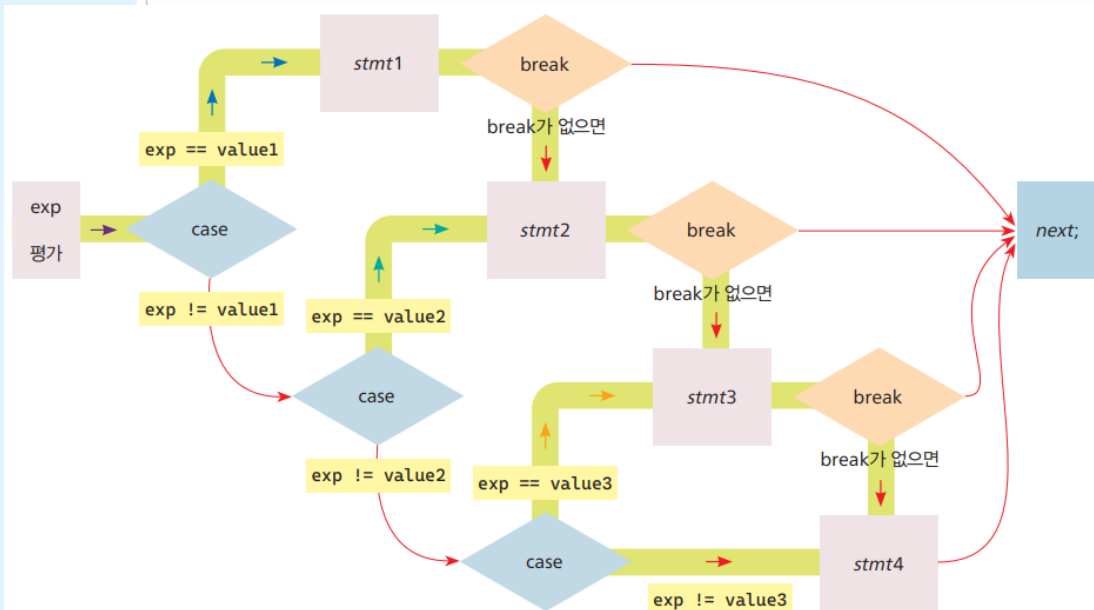


그림 6-20 switch 문의 제어흐름

월에 따른 사계절 출력

실습예제 6-7

Prj07

07seasonswitch.c

월에 따른 사계절 출력

난이도: ★

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03
04 int main(void)
05 {
06     int month;
07     printf("월(month)을 입력: ");
08     scanf("%d", &month);
09
10     switch (month)
11     {
12         case 4: case 5:
13             printf("%d월은 봄입니다.\n", month);
14             break;
15         case 6: case 7: case 8:
16             printf("%d월은 여름입니다.\n", month);
17             break;
18         case 9: case 10: case 11:
19             printf("%d월은 가을입니다.\n", month);
20             break;
21         case 12: case 1: case 2: case 3:
22             printf("%d월은 겨울입니다.\n", month);
23             break;
24
25         default:
26             printf("월(month)을 잘못 입력했습니다.\n");
27     }
28
29     return 0;
30 }
```

month가 6, 7, 8이면 이 case로
들어와 여름이 출력된다.

```
switch ( month )
{
    case 4 : case 5 :
        printf("%d월은 봄입니다.\n", month);
        break;

    case 6 : case 7 : case 8 :
        printf("%d월은 여름입니다.\n", month);
        break;

    ...

    default :
        printf("월(month)을 잘못 입력하셨습니다.\n");
}
```

```
case 4, 5 : //오류발생
...
break;
```

```
case 6, 7, 8 : //오류발생
...
break;
```

결과

월(month)을 입력: 11
11월은 가을입니다.
월(month)을 입력: 5
5월은 봄입니다.

월(month)을 입력: 1
1월은 겨울입니다.
월(month)을 입력: 7
7월은 여름입니다.

switch 연산식의 활용

• 점수에 따른 성적 부여

실습예제 6-8

Prj08

08scoreswitch.c

점수에 따른 성적 부여

난이도: ★★

```
01 #define _CRT_SECURE_NO_WARNINGS
02
03 #include <stdio.h>
04
05 int main(void)
06 {
07     int score;
08
09     printf("점수 입력: ");
10     scanf("%d", &score);
11
12     switch (score / 10) {
13     case 10: case 9:
14         printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'A');
15         break;
16     case 8:
17         printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'B');
18         break;
19     case 7:
20         printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'C');
21         break;
22     case 6:
23         printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'D');
24         break;
25     default:
26         printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'F');
27     }
28
29     return 0;
30 }
31 }
```

점수의 나누기 결과는 정수라는 것을 활용한다.

100점도 있으므로 case 10이 필요하며, 90점 이상이고 100점 사이면 실행되는 case이다.

break를 생략하면 다음 case 내부 문장을 수행하므로 반드시 필요하다.

표 6-6 점수에 따른 성적처리를 위한 연산값

| 점수 예 | 점수 범위 | (score / 10) 연산값 | 성적처리 |
|-----------------|-----------------|------------------|--------|
| 100, 98, 95, 90 | 90 <= 점수 <= 100 | 9 또는 10 | 'A' 부여 |
| 80, 85, 88, 89 | 80 <= 점수 < 90 | 8 | 'B' 부여 |
| 80, 85, 88, 89 | 70 <= 점수 < 80 | 7 | 'C' 부여 |
| 80, 85, 88, 89 | 60 <= 점수 < 70 | 6 | 'D' 부여 |
| 30, 55, 58, 59 | 점수 < 60 | 그 외 | 'F' 부여 |

결과

점수 입력: 100
점수가 100 점으로 성적이 A 입니다.

점수 입력: 75
점수가 75 점으로 성적이 C 입니다.

점수 입력: 88
점수가 88 점으로 성적이 B 입니다.

점수 입력: 62
점수가 62 점으로 성적이 D 입니다.

default의 위치

- switch문 주의점

- 연산식 결과는 정수형 또는 문자형
- 각 case 뒤에 나오는 식은 상수식
 - 상수식에는 변수와 const 상수 사용 불가능
 - 리터럴 상수와 매크로 상수의 연산식은 사용 가능

- default

- 선택적으로 없거나 하나이며
- 어디에 위치해도 모든 case 처리를 하지 않은 경우 실행
- 다른 case가 뒤에 있다면 break가 필요

실습예제 6-9

Prj09 09scoreswitch2.c 잘못된 점수도 고려하여 점수에 따른 성적 부여

난이도: ★★

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03
04 int main(void)
05 {
06     int score;
07     printf("점수(0에서 100사이) 입력: ");
08     scanf("%d", &score);
09     if (score < 0 || score > 100)
10     {
11         printf("점수 입력이 잘못되었습니다.\n");
12         return 0;
13     }
14
15     switch (score / 10)
16     {
17         default:
18             printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'F');
19             break;
20
21         case 10: case 9:
22             printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'A');
23             break;
24         case 8:
25             printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'B');
26             break;
27         case 7:
28             printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'C');
29             break;
30         case 6:
31             printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'D');
32             break;
33     }
34
35     return 0;
36 }
```

!!(0 <= score && score <= 100)와 같으며, 점수가 음수이거나 100을 초과하면 조건을 만족한다.

| 결과 | 점수(0에서 100사이) 입력: 101 점수 입력이 잘못되었습니다. | 점수(0에서 100사이) 입력: 94 점수가 94 점으로 성적이 A 입니다. |
|----|---|---|
| | 점수(0에서 100사이) 입력: 65 점수가 65 점으로 성적이 D 입니다. | 점수(0에서 100사이) 입력: 55 점수가 55 점으로 성적이 F 입니다. |

LAB 표준입력으로 받은 정수에 대응하는 열거 상수로 switch문 활용

Lab 6-2

lab2shapeswitch.c

난이도: ★★

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03
04 int main(void)
05 {
06     enum shape {TRIANGLE = 1, RECTANGLE};
07     int input;
08     double width, height;
09
10     printf("삼각형[1], 사각형[2] 중의 번호 하나를 선택 >> ");
11     scanf("%d", &input);
12     printf("넓이와 높이를 입력 >> ");
13     scanf("%lf %lf", &width, &height);
14     printf("가로: %.2lf 세로: %.2lf \n", width, height);
15
16     switch (input)
17     {
18     case :
19         printf("삼각형 면적: %.2lf\n", width * height / 2);
20         break;
21     case RECTANGLE:
22         printf("사각형 면적: %.2lf\n", );
23         break;
24
25     default:
26         printf("잘못된 입력\n");
27     }
28
29     return 0;
30 }
```

정답

```
18     case TRIANGLE:
22         printf("사각형 면적: %.2lf\n", width * height);
```

Microsoft Visual Studio 디버그 콘솔

```
삼각형[1], 사각형[2] 중의 번호 하나를 선택 >> 1
넓이와 높이를 입력 >> 3.47 8.21
가로: 3.47 세로: 8.21
삼각형 면적: 14.24
```

감사합니다.