

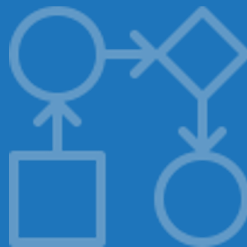
structure
C/C++



pointer



function



array[]



switch/case

for, while

프로그래밍 기초



malloc/free



if else

6장. 조건

- 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 개의 조건 중에 하나를 선택하는 구조

■ 반복

- 반복 몸체인 여러 문장을 반복하는 구조

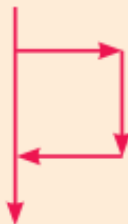
■ 분기 처리

- 정해진 위치로 이동하는 구조
- 반복이나 선택을 빠져 나감(break), 다음 반복을 수행(continue)

조건 선택

조건에 대한 선택 구문

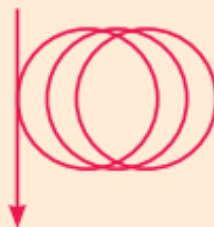
- if
- if else
- if else if
- nested if
- switch



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 $\leq 0.05 * \text{학생수}$

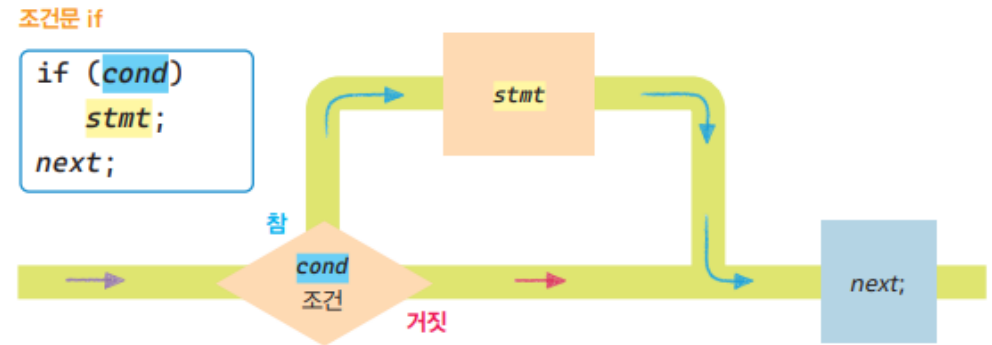
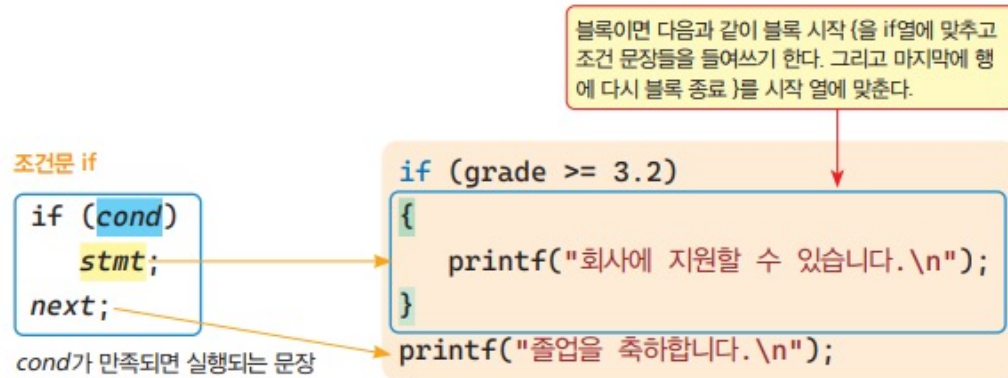
대학 B는 학과 석차가 상위 5% 이어야 장학금을 받을 수 있다고 한다.

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 문장

- 조건식이 참이면(0이 아님) stmt 문장을 실행
- 거짓(0)이면, stmt 문장을 실행하지 않음



■ 논리 오류와 문법 오류

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

printf("졸업을 축하합니다.\n");
```

논리 오류

조건식 다음에 세미콜론(;) 사용
- 무조건 printf() 문장 수행

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

printf("졸업을 축하합니다.\n");
```

문법 오류

조건식에 괄호를 사용하지 않음
- 에러 발생

현재 온도에 따른 폭염 주의 발령

<01basicif.c>

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

```
int main(void)  
{
```

```
    double temperature;
```

```
    printf("현재 온도 입력: ");  
    scanf("%lf", &temperature);
```

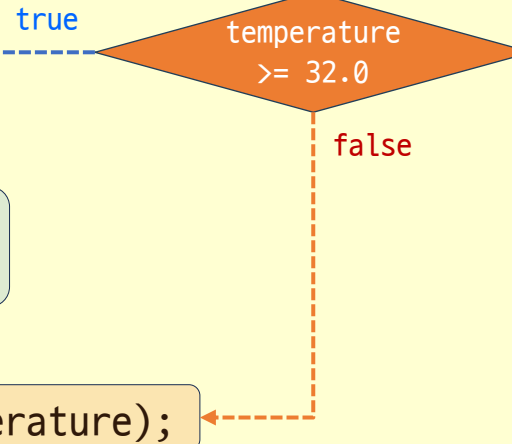
```
    if (temperature >= 32.0)  
    {  
        printf("폭염 주의보를 발령합니다.\n");  
        printf("건강에 유의하세요.\n");  
    }
```

2개 이상의 문장인 경우,
반드시 { ... }로 묶음

```
    printf("현재 온도는 섭씨 %.2f 입니다.\n", temperature);
```

```
    return 0;
```

```
}
```



실행 결과 #1

현재 온도 입력: 29.3
현재 온도는 섭씨 29.30 입니다.

실행 결과 #2

현재 온도 입력: 39
폭염 주의보를 발령합니다.
건강에 유의하세요.
현재 온도는 섭씨 39.00 입니다.

조건 만족 여부에 대한 선택 if else

- 조건을 만족하면(조건식이 true이면) stmt1을 실행
 - 조건을 만족하지 않으면(조건식이 false이면) stmt2를 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");
```

if(n % 2)
- n % 2의 값이 0이 아닌 경우

권장하는 코딩 스타일

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>

if(!(n % 2))
- n % 2의 값이 0인 경우

if else로 짝수와 홀수 판정 (실습)

<02evenifelse.c>

```
#include <stdio.h>

int main(void)
{
    int n;
    printf("정수 입력: ");
    scanf("%d", &n);

    if (n % 2 != 0) // if(n % 2)
    {
        printf("홀수\n");
    }
    else
    {
        printf("짝수\n");
    }

    //조건연산자 이용
    (n % 2) ? printf("홀수\n") : printf("짝수\n");

    return 0;
}
```

실행 결과 #1


정수 입력: 5
홀수
홀수

실행 결과 #2

정수 입력: 6
짝수
짝수


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

연속된 조건: if ... else if ... else

■ 연속적인 조건

```
if(조건식1) {  
    문장1;  
}
```

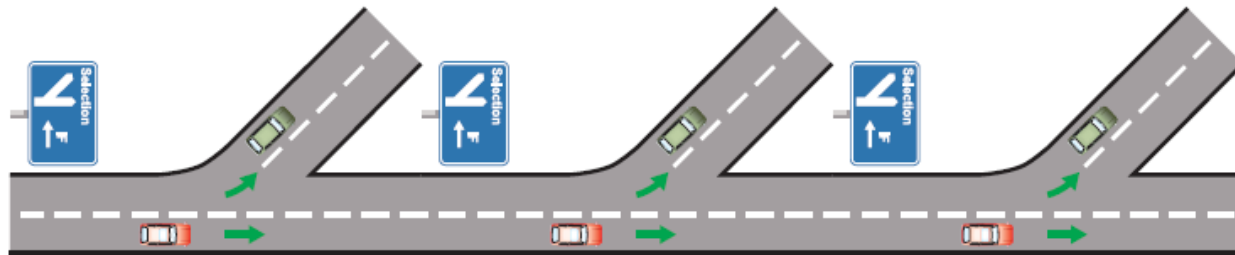
조건식1이 true이면, 문장1만 수행

```
}  
else if(조건식2) {  
    문장2;  
}
```

```
}  
else if(조건식3) {  
    문장3;  
}
```

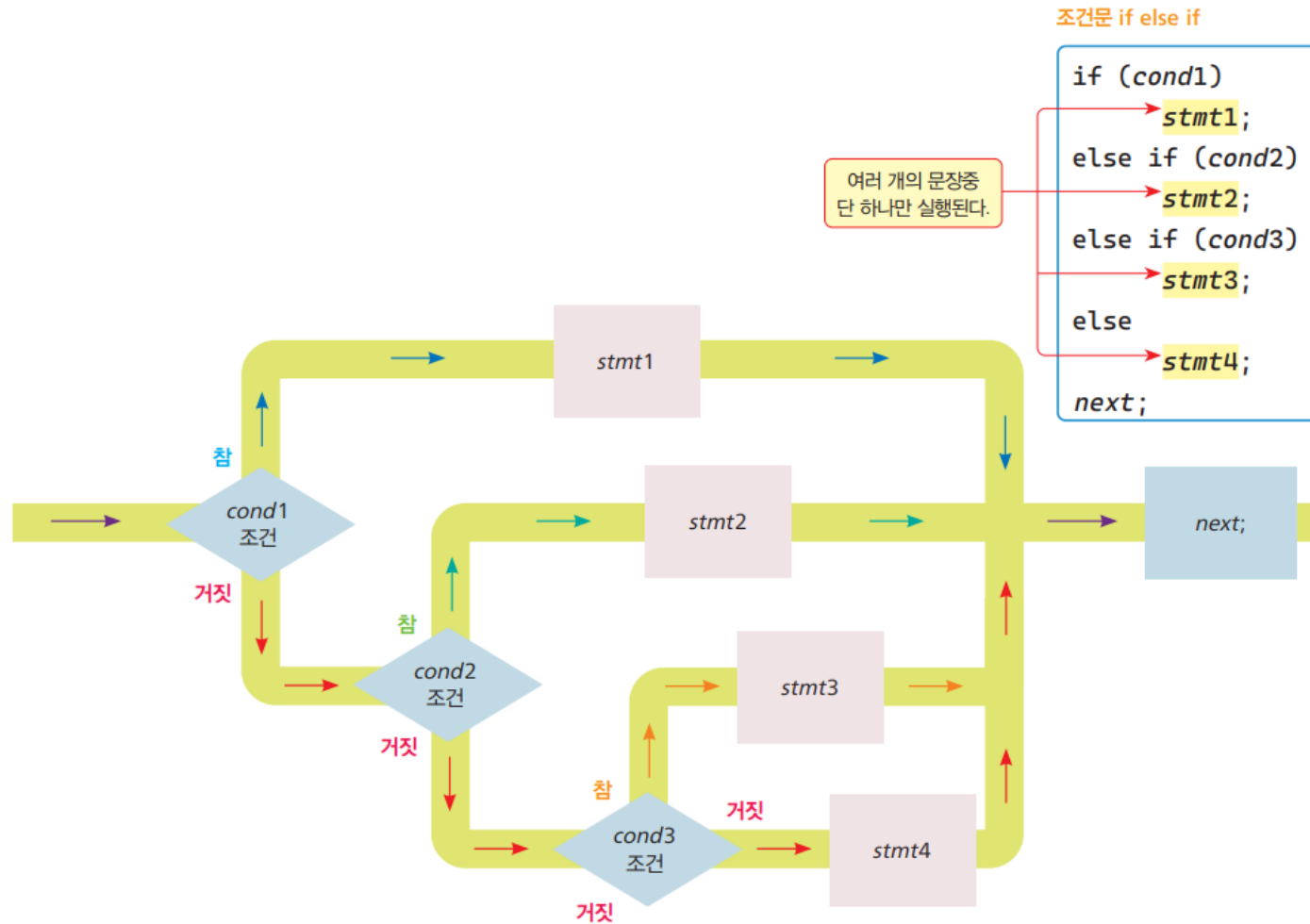
```
}  
else {  
    문장4;  
}
```

조건식1, 2, 3이 모두 false 이면,
else 블록(문장4) 수행



연속된 조건 if else if

- if ... else if ... else 제어 흐름



조건 if else 문으로 평균평점에 따른 결과 출력

<03gradeif.c>

```
#include <stdio.h>

int main(void)
{
    double gpa;

    printf("평균평점 입력: ");
    scanf("%lf", &gpa);

    if (gpa >= 4.3)
        printf("최우등\n");
    else if (gpa >= 3.8)
        printf("우등\n");
    else if (gpa >= 3.0)
        printf("우수\n");
    else
        printf("3.0 미만\n");

    return 0;
}
```

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

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

실행 결과 #1

평균평점 입력: 4.3
최우등

실행 결과 #2

평균평점 입력: 3.9
우등

실행 결과 #3

평균평점 입력: 3.3
우수

실행 결과 #4

평균평점 입력: 2.7
3.0 미만

반복된 조건에 따른 선택 예제 (실습 및 디버깅)

<03gradeif2.c>

```
#include <stdio.h>

int main()
{
    int point = 98;

    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");

    return 0;
}
```

연속된 if문

실행 결과 A

<03gradeif3.c>

```
#include <stdio.h>

int main()
{
    int point = 98;

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

    return 0;
}
```

각각 독립된 if문

실행 결과

A
B
C
D

중첩된 if 문 예제 (실습)

<04nestedif.c>

```
#include <stdio.h>

int main(void)
{
    int type, point;

    printf("번호를 선택: 1(1종면허), 2(2종면허): ");
    scanf("%d", &type);
    printf("필기시험 점수 입력: ");
    scanf("%d", &point);
```

```
    if (type == 1)
```

```
    {
```

```
        if (point >= 70)
            printf("1종면허 합격\n");
```

```
        else
            printf("1종면허 불합격\n");
```

```
    }
```

```
    else if (type == 2)
```

```
    {
```

```
        if (point >= 60)
            printf("2종면허 합격\n");
```

```
        else
            printf("2종면허 불합격\n");
```

```
    }
```

```
    return 0;
```

```
}
```

중첩된 if 문장

중첩된 if 문장

■ 중첩된 if

- if문 내부에 if문이 존재

실행 결과 #1

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

실행 결과 #2

번호를 선택: 1(1종면허), 2(2종면허): 2
필기시험 점수 입력: 65
2종면허 합격

Tip. if 조건식에서 주의할 점

- 조건식에서 등호 연산자(==) 대신 **대입 연산자(=)**를 사용한 경우
 - 항상 조건식을 만족

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

- 조건식에서 **실수와 값을 비교하는 등호 연산자(==)** 사용
 - float나 double에는 실제 5.1보다 큰 값이 저장
 - 계산 오차(부동소수점 오차)가 발생
 - 원하지 않는 결과가 발생할 수 있음
 - 가급적 사용하지 말 것

<04doubleif.c>

```
#include <stdio.h>

int main()
{
    double a = 4.7, b = 0.4;
    double sum = a + b;

    if (sum == 5.1)
    {
        printf("%s\n", "Good");
    }
    else
    {
        printf("%s\n", "Strange!");
    }
    printf("%.20f\n", sum);
    return 0;
}
```

Strange!
5.100000000000000053291

블록 표시와 else

■ 블록이 없는 경우

- else 문장이 어느 if문의 소속인지 판단하기 어려움
- else는 같은 블록 내에서 else가 없는 가장 근접한 상위 if문에 소속된 else로 해서
 - else의 혼란을 방지하기 위해 블록을 이용하는 것을 권고

```
int main(void)
{
    int age = 30;

    if (age >= 20)
        if (age >= 65)
            printf("Older than 65.\n");
        else
            printf("Younger than 20.\n");

    return 0;
}
```

논리적 오류 발생

마지막의 else 문장은 if(age>=65)의 else로 인식

```
05ageifelse1.c:7:8: warning: suggest explicit braces to avoid ambiguous 'else' [-Wdangling-else]
   7 |     if (age >= 20)
     |
```

Younger than 20.

```
int main(void)
{
    int age = 30;

    if (age >= 20)
    {
        if (age >= 65)
            printf("Older than 65.\n");
    }
    else
    {
        printf("Younger than 20.\n");
    }

    return 0;
}
```

블록 표시

else의 if를 적절히 조합하여 나이에 따른 선택 (실습)

```
#include <stdio.h>

int main(void)
{
    int age = 0;

    printf("당신의 나이는? ");
    scanf("%d", &age);

    if(age >= 65)
    {
        printf("age >= 65\n");
    }
    else if(age >= 20)
    {
        printf("20 <= age < 65\n");
    }
    else
    {
        printf("age < 20\n");
    }

    return 0;
}
```

<05ageifelse.c>

실행 결과

당신의 나이는? 20
20 <= age < 65

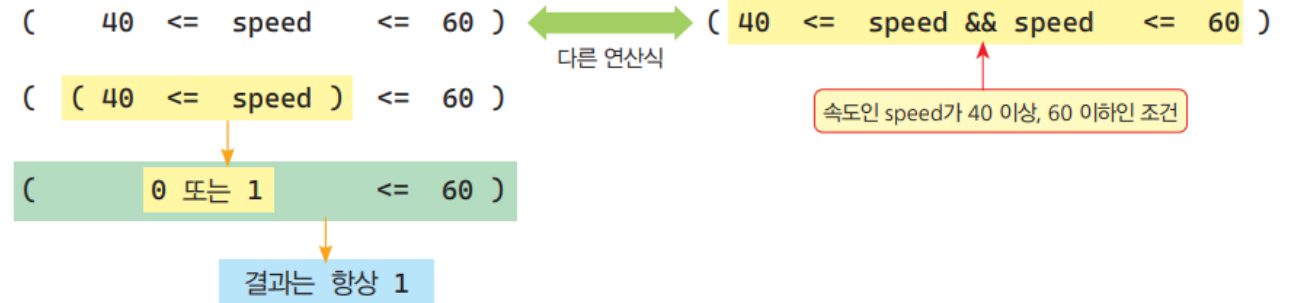
당신의 나이는? 65
age >= 65

당신의 나이는? 19
age < 20

다양한 if 문의 이용과 조건연산자

■ 속도 speed가 40km와 60km 사이라는 조건식

- `if (40 <= speed && speed <= 60)`
- ❌ • `if (40 <= speed <= 60)`
 - 잘못된 조건식: Python에서 가능



■ 조건연산자의 기능은 if문으로 가능

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 x 와 z 의 최대값
 - 거짓이면 y 와 z 의 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 }
```

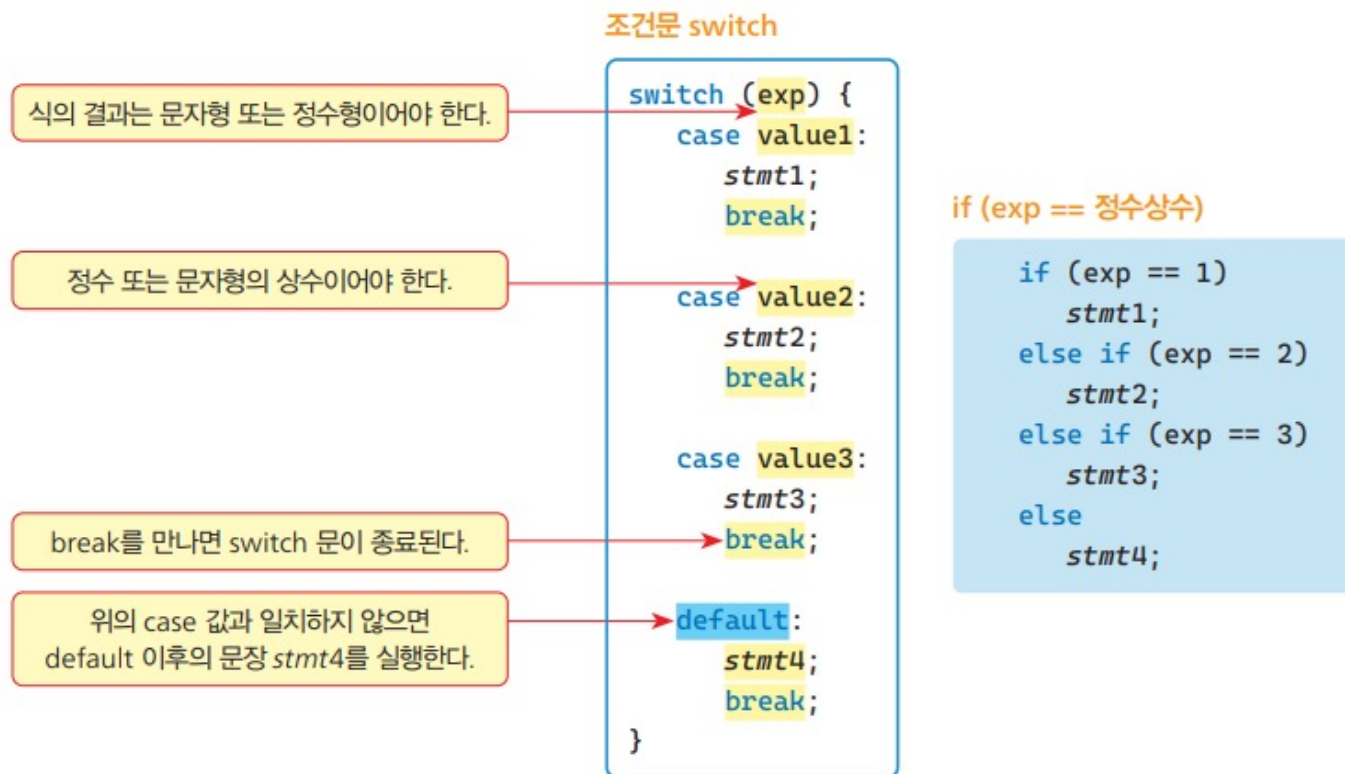
정답

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

switch 문장 개요

■ switch 문

- 연산식의 결과값에 따라 여러 경로 중에서 하나를 선택하는 구문
- if else가 여러 번 계속 반복되는 구문을 좀 더 간략하게 구현
- 주어진 연산식의 값이 정수형 또는 문자형
- break가 없으면 다음 case가 계속 실행됨

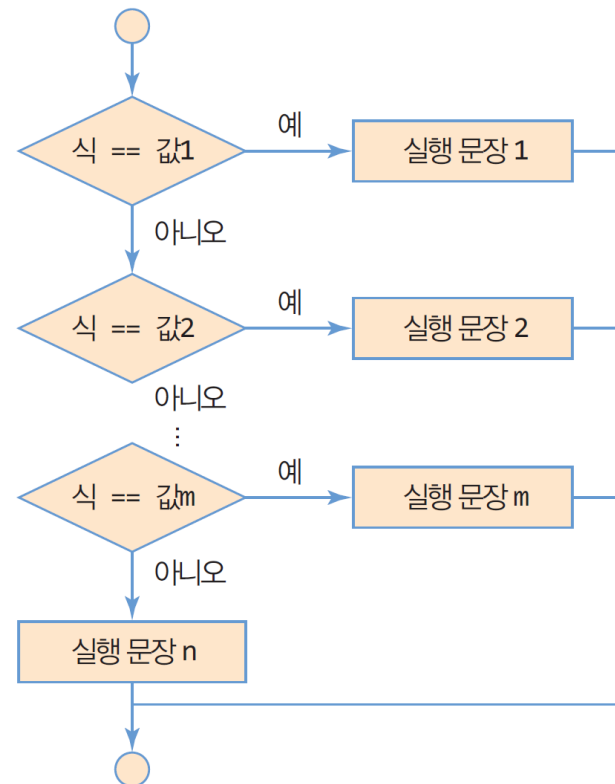


default의 위치

■ default

- default는 선택적: 생략 가능
 - case 의 비교 값과 일치하는 것이 없으면 default 내부의 문장을 실행
- default의 위치는 제한이 없음
 - 일반적으로 마지막에 위치

```
switch (식) {  
  case 값1:  
    실행 문장 1;  
    break;  
  case 값2:  
    실행 문장 2;  
    break;  
  ...  
  case 값m:  
    실행 문장 m;  
    break;  
  default:  
    실행 문장 n;  
}
```



switch로 두 실수의 사칙연산 수행 (실습)

```
#include <stdio.h>
int main(void)
{
    double x, y, result;
    int op;

    printf("두 실수 입력: ");
    scanf("%lf %lf", &x, &y);
    printf("연산종류 번호선택 1(+), 2(-), 3(*), 4(/): ");
    scanf("%d", &op);

    switch (op) {
    case 1:
        printf("%.2f + %.2f = %.2f\n", x, y, x + y);
        break;
    case 2:
        printf("%.2f - %.2f = %.2f\n", x, y, x - y);
        break;
    case 3:
        printf("%.2f * %.2f = %.2f\n", x, y, x * y);
        break;
    case 4:
        printf("%.2f / %.2f = %.2f\n", x, y, x / y);
        break;
    default:
        printf("번호를 잘못 선택했습니다.\n");
        break; // 마지막 default의 break는 생략가능
    }
    return 0;
}
```

<06arithswitch.c>

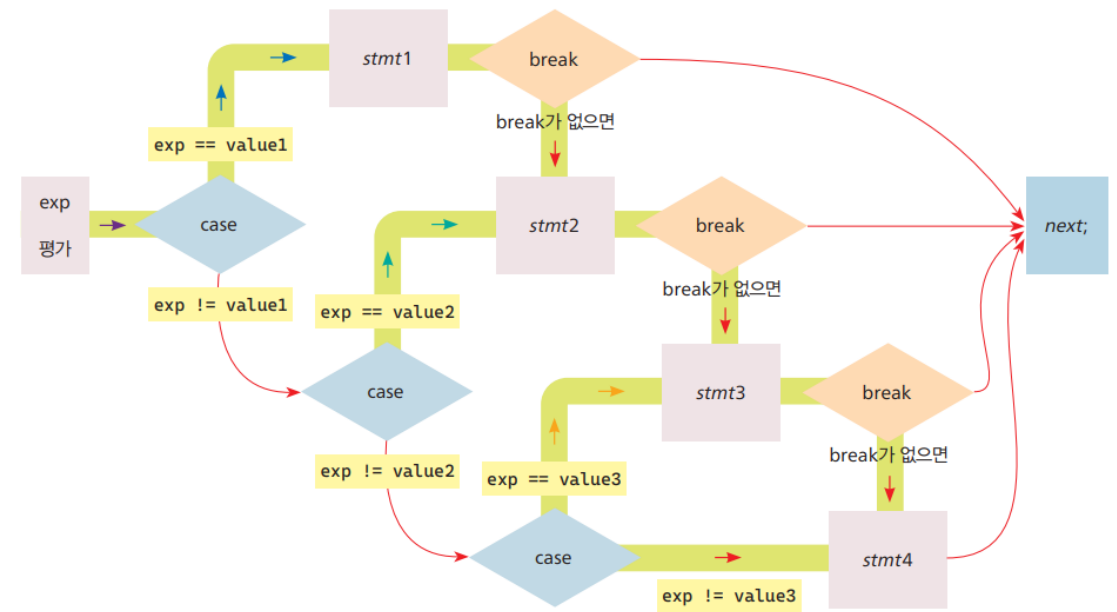


그림 6-20 switch 문의 제어흐름

월에 따른 사계절 출력 (실습)

```
#include <stdio.h>

int main(void)
{
    int month;
    printf("월(month)을 입력: ");
    scanf("%d", &month);

    switch (month)
    {
        case 4: case 5:
            printf("%d월은 봄입니다.\n", month);
            break;
        case 6: case 7: case 8:
            printf("%d월은 여름입니다.\n", month);
            break;
        case 9: case 10: case 11:
            printf("%d월은 가을입니다.\n", month);
            break;
        case 12: case 1: case 2: case 3:
            printf("%d월은 겨울입니다.\n", month);
            break;
        default:
            printf("월(month)을 잘못 입력했습니다.\n");
            break;
    }
    return 0;
}
```

case 4, 5: 오류 발생



<07seasonswitch.c>

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

실행 결과

월(month)을 입력: 8
8월은 여름입니다.

switch 연산식의 활용

<08scoreswitch.c>

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

int main(void)
{
    int score;
    printf("점수 입력: ");
    scanf("%d", &score);

    switch (score / 10)
    {
        case 10:
        case 9:
            printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'A');
            break;
        case 8:
            printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'B');
            break;
        case 7:
            printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'C');
            break;
        case 6:
            printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'D');
            break;
        default:
            printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'F');
            break;
    }
    return 0;
}
```

정수의 나누기 결과는
정수라는 것을 활용

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

실행 결과

점수 입력: 95
점수가 95 점으로 성적이 A 입니다.

점수 입력: 58
점수가 58 점으로 성적이 F 입니다.

실습 예제 6-9: default 위치

<09scoreswitch2.c>

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

```
int main(void)
{
```

```
    int score;
    printf("점수(0에서 100사이) 입력: ");
    scanf("%d", &score);
```

```
    if (score < 0 || score > 100)
    {
        printf("점수 입력이 잘못되었습니다.\n");
        return 0;
    }
```

```
switch (score / 10)
{
```

```
    default:
```

```
        printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'F');
        break;
```

```
    case 10:
```

```
    case 9:
```

```
        printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'A');
        break;
```

```
    case 8:
```

```
        printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'B');
        break;
```

```
    case 7:
```

```
        printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'C');
        break;
```

```
    case 6:
```

```
        printf("점수가 %d 점으로 성적이 %c 입니다.\n", score, 'D');
        break;
```

```
}
```

```
    return 0;
```

```
}
```

default가 맨 처음에 나옴
- 가능하지만 권장하지 않음
- 코드의 흐름을 이해하기 어려움

switch/case 문자형 예제 (실습)

<10switchchar.c>

```
#include <stdio.h>

int main()
{
    char op;
    int x, y, result;

    printf("수식을 입력하시오(예 2 + 5): \n");
    printf(">> ");
    scanf("%d %c %d", &x, &op, &y);
```

실행 결과 1

수식을 입력하시오(예 2 + 5):

>> 10 + 5

10 + 5 = 15

실행 결과 2

수식을 입력하시오(예 2 + 5):

>> 2 * 4

2 * 4 = 8

```
switch (op)
{
    case '+':
        result = x + y;
        break;
    case '-':
        result = x - y;
        break;
    case '*':
        result = x * y;
        break;
    case '/':
        result = x / y;
        break;
    case '%':
        result = x % y;
        break;
    default:
        printf("지원되지 않는 연산자입니다. \n");
        break;
}
printf("%d %c %d = %d\n", x, op, y, result);
return 0;
}
```

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



Questions?