

Basic Computer Programming

Lecture 5

Electrical & Electronics Engineering
Chung-Ang University

Contents

- Understand and use if statements.
- Understand and use if-else statements.
- Understand and use overlapping if statements.
- Understand and use switch statements.
- Understand and use break, continue, and goto statements.

The program we are going to make in this chapter.



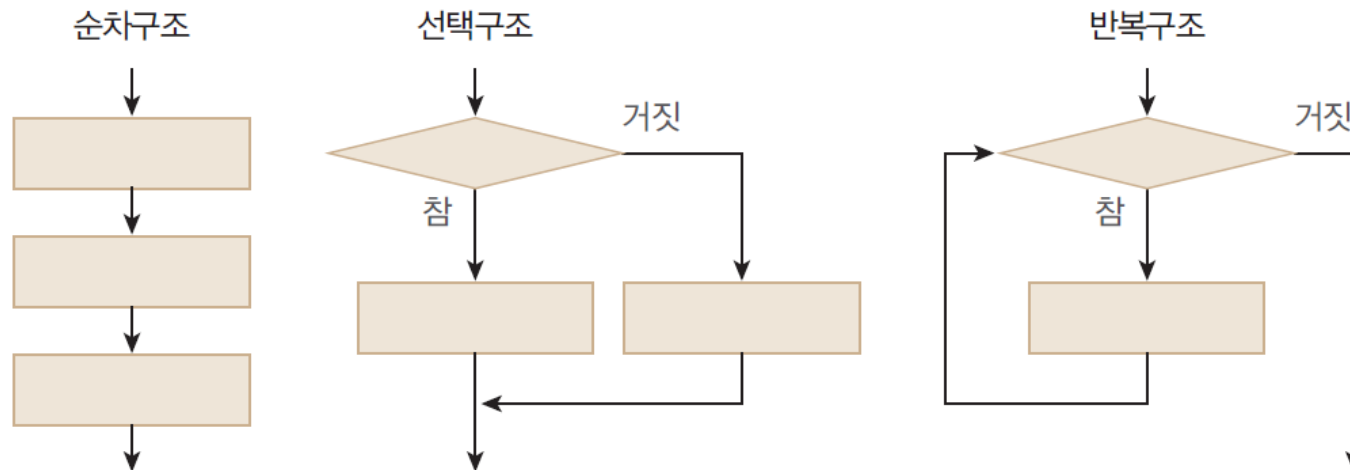
```
Microsoft Visual Studio 디버그 콘솔
연도를 입력하시오: 2024
2024년은 윤년입니다.
```



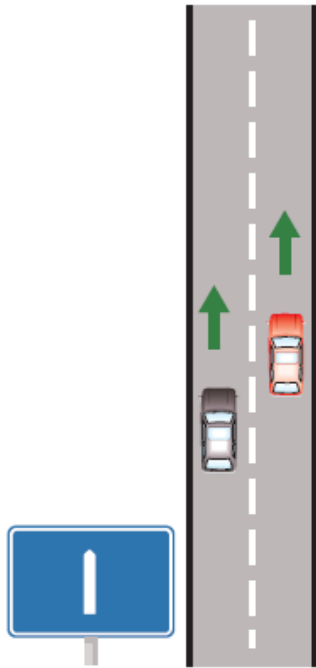
```
Microsoft Visual Studio 디버그 콘솔
가위, 바위, 보 게임에 오신 것을 환영합니다.
하나를 선택하세요(가위-0, 바위-1, 보-2): 0
사용자=0
컴퓨터=2
사용자 승리
```

Three basic control structures

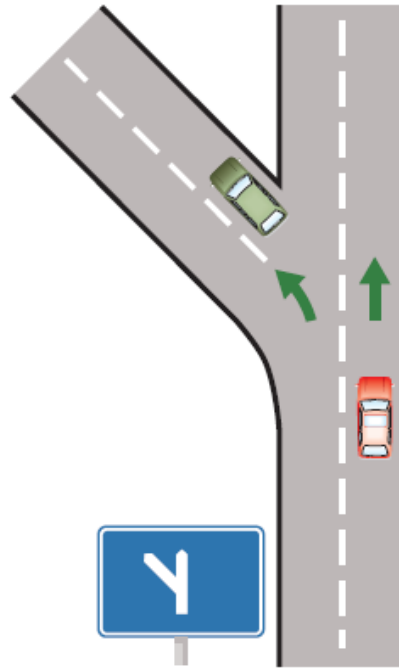
- Sequence - a structure in which instructions are sequentially executed.
- Selection - a structure executed by selecting one command from among commands.
- Iteration - a structure in which the same command is repeated and executed.



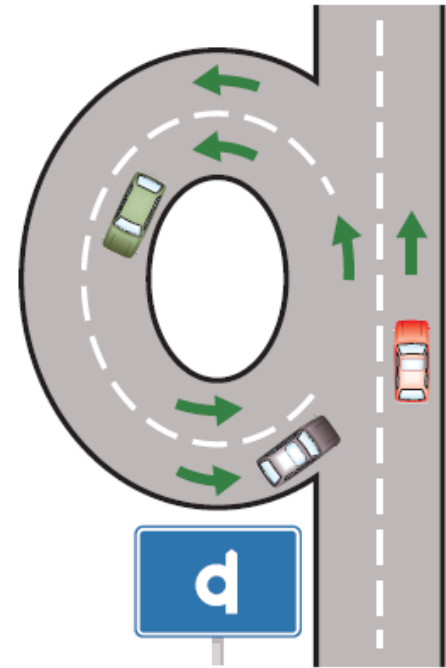
Similar to the road on which a car travels



순차 구조



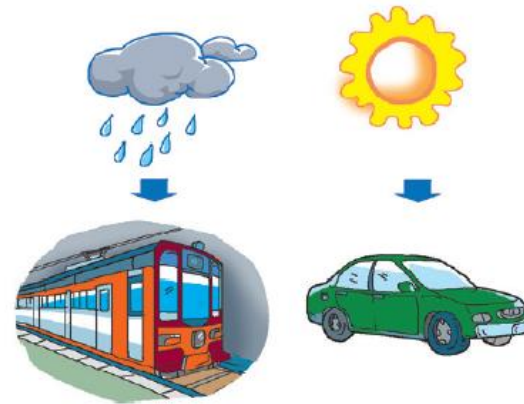
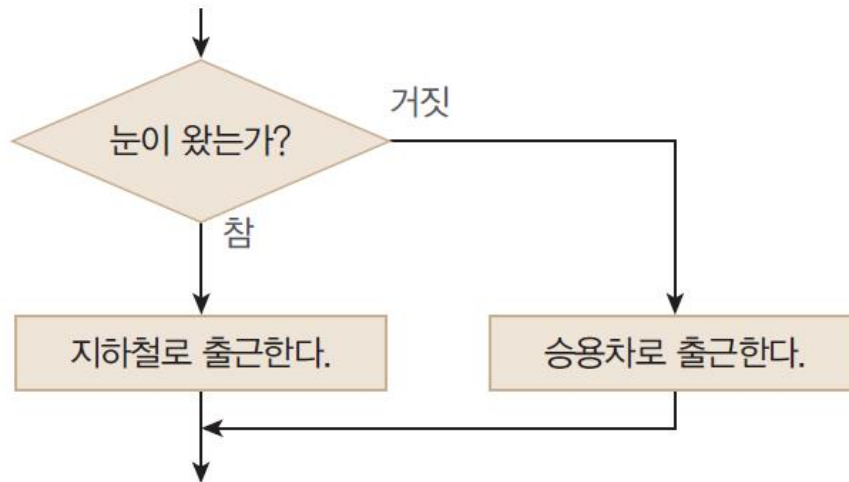
선택 구조



반복 구조

Why do we need a conditional statement?

- If only the sequential structure is used, the program will always repeat the same operation, so it is not very useful.
- You should be able to make different decisions depending on the situation.



if-else statement

Syntax 5.1 if-else 문

형식 `if(조건식) {`
 문장1;
`}`
`else {`
 문장2;
`}`

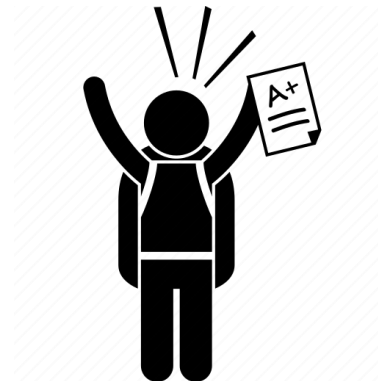
예 `if(number > 0) {`
 printf("양수입니다.\n");
`}`
`else {`
 printf("음수이거나 0입니다.\n");
`}`

설명 만약 조건식이 참이면 문장1이 실행된다. 그렇지 않으면 문장2가 실행된다.
조건에 따라 실행되는 문장이 하나이면 중괄호는 생략이 가능하다.
else 이하는 생략될 수 있다.

Example

- If the test score is 60 points or more, pass, and if the test score is less than 60 points, fail is displayed on the screen, write as follows.

```
if ( score >= 60 ) {  
    printf("합격입니다.\n");  
}  
else {  
    printf("불합격입니다.\n");  
}
```



Possible things

- If there is only one sentence executed according to the condition, the brackets may be omitted.

```
if ( score >= 60 )  
    printf("합격입니다.\n");  
else  
    printf("불합격입니다.\n");
```

- The else statement may be omitted.

```
if ( score >= 60 )  
    printf("합격입니다.\n");
```

Example #1

- After checking the temperature input from the user using the if-else statement, if it is above 0, it outputs "It's above 0" or "It's below 0 "

```
Microsoft Visual Studio 디버그 콘솔
온도를 입력하시오:-10
영하의 날씨입니다.
현재 온도는 -10도입니다.
```



Example #1

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

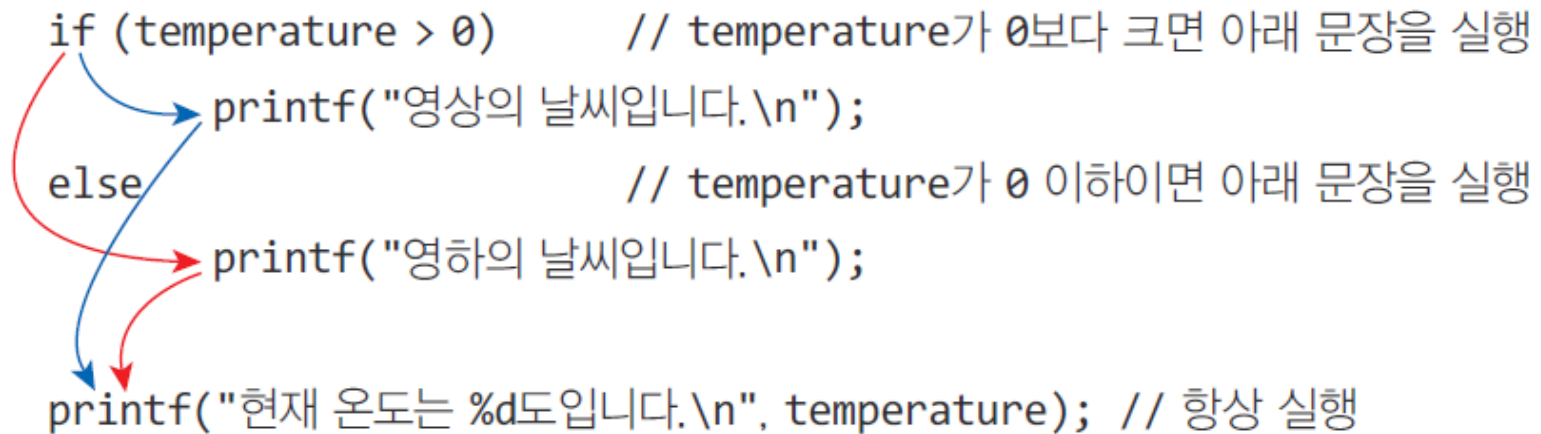
int main(void)
{
    int temperature;
    printf("온도를 입력하시오:");
    scanf("%d", &temperature);

    if (temperature > 0)           // temperature가 0보다 크면 아래 문장을 실행
        printf("영상의 날씨입니다.\n");
    else                           // temperature가 0 이하이면 아래 문장을 실행
        printf("영하의 날씨입니다.\n");

    printf("현재 온도는 %d도입니다.\n", temperature);    // 항상 실행
    return 0;
}
```

What statement will be executed after the if-else statement is executed?

```
if (temperature > 0)      // temperature가 0보다 크면 아래 문장을 실행
    printf("영상의 날씨입니다.\n");
else                      // temperature가 0 이하이면 아래 문장을 실행
    printf("영하의 날씨입니다.\n");
printf("현재 온도는 %d도입니다.\n", temperature); // 항상 실행
```



Example #2

- Let's write a program that tells you whether the integer input from the user is odd or even.



```
Microsoft Visual Studio 디버그 콘솔
정수를 입력하시오: 23
입력된 정수는 홀수입니다.
```

Example #2

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
int main(void)
{
    int number;
    printf("정수를 입력하시오:");
    scanf("%d", &number);

    if (number % 2 == 0) // number를 2로 나눈 나머지가 0이면 짝수이다.
        printf("입력된 정수는 짝수입니다.\n");
    else // 그렇지 않으면 홀수이다.
        printf("입력된 정수는 홀수입니다.\n");

    return 0;
}
```

Check



참고

짝수와 홀수를 구분할 때, if 문을 2개를 사용하는 방법과 if-else 문을 사용하는 방법 중에서 어떤 것이 더 효율적일까?

여러 개의 if 문을 사용하는 방법	if-else 문을 사용하는 방법
<pre>if (number % 2 == 0) // ① printf("짝수\n"); if (number % 2 == 1) // ② printf("홀수\n");</pre>	<pre>if (number % 2 == 0) // ① printf("짝수\n"); else printf("홀수\n");</pre>
모든 값에 대하여 항상 문장 ①과 문장 ②가 실행되어야 하므로 비효율적이다.	입력 값에 대하여 문장 ①만 실행되므로 효율적이다.

Example #3

- For example, what should I do to print out that I can get a scholarship at the same time as I pass the exam if I get more than 60 points?



The image shows a screenshot of the Microsoft Visual Studio debugger console window. The title bar is orange and contains the text "Microsoft Visual Studio 디버그 콘솔" (Microsoft Visual Studio Debug Console). The console area is white and contains the following Korean text: "성적을 입력하시오:90", "합격입니다.", and "장학금도 받을 수 있습니다.". The text is displayed in a monospaced font. On the right side of the console, there is a vertical scrollbar with up and down arrow buttons.

```
Microsoft Visual Studio 디버그 콘솔
성적을 입력하시오:90
합격입니다.
장학금도 받을 수 있습니다.
```


Compound statements

- Compound statements
 - Grouping sentences using brackets,
 - It is also called a block.
 - You can enter instead of a single sentence.

```
if( score >= 60 )  
{  
    printf("합격입니다.\n");  
    printf("장학금도 받을 수 있습니다.\n");  
}
```

조건식이 참이면 2개의
문장이 묶여서 실행된다.

Example #3

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

int main(void)
{
    int score;

    printf("성적을 입력하시오:");
    scanf("%d", &score);
    if (score >= 60)    // 성적이 60점 이상이면
    {
        printf("합격입니다.\n");
        printf("장학금도 받을 수 있습니다.\n");
    }
    else                // 그렇지 않으면
    {
        printf("불합격입니다.\n");
        printf("다시 도전하세요.\n");
    }
    return 0;
}
```

Notice



경고

초보자가 저지르기 쉬운 하나의 실수가 다음과 같이 들여쓰기를 하면 `score`가 90보다 크거나 같을 때 두 개의 문장이 실행될 거라고 믿는 것이다.

```
if( score >= 90 )  
    printf("합격입니다.\n");  
    printf("장학금도 받을 수 있습니다.\n");
```

그러나 컴파일러는 첫 번째 `printf` 문장만 조건에 따라서 실행하고 두 번째 `printf` 문장은 조건에 상관 없이 무조건 실행하게 된다. 즉 다음과 같은 코드와 실행 결과는 동일하다.

```
if( score >= 90 )  
    printf("합격입니다.\n");  
printf("장학금도 받을 수 있습니다.\n");
```

따라서 조건에 따라 두 개의 문장을 묶어서 실행하는 경우라면 반드시 중괄호로 묶어서 블록으로 만들어야 한다는 것을 기억하자.

Notice



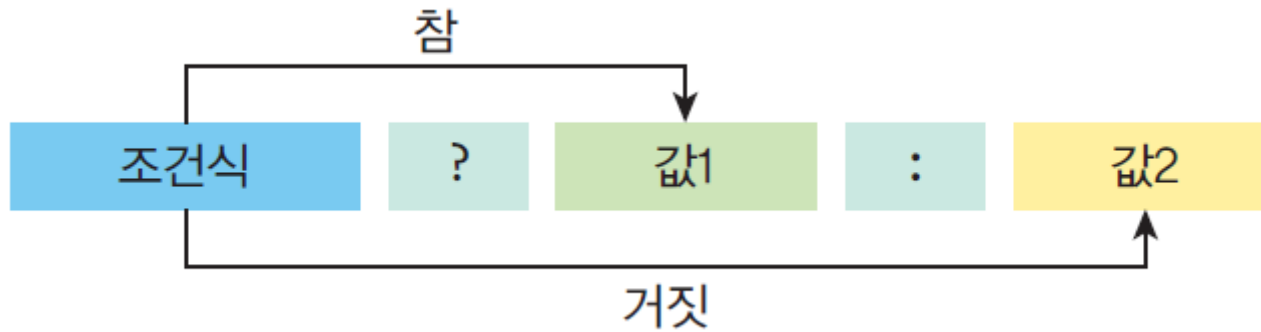
경고

다음과 같이 `if` 문장의 조건식 뒤에 세미콜론을 찍으면 안된다. `if` 문장은 조건식과 문장이 합쳐서 하나의 문장을 이룬다. 아래와 같이 작성하면 `if` 문은 `if(x > 0);`로 끝나고 `printf` 문장은 조건에 관계없이 실행된다.

```
if( x > 0 );  
    printf("양수입니다.\n");
```

Conditional operators "?"

- Conditional operators consist of "?" and ":" symbols. The condition is before the symbol "?". If the condition is true, the value_1 is returned. If the condition is false, the value_2 is returned.



```
absolute_value = (x > 0) ? x : -x; // 절대값 계산
```

```
big = (x > y) ? x : y;           // 최대값 계산
```

```
small = (x < y) ? x : y;        // 최소값 계산
```

Example

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

int main(void)
{
    int x, y;

    printf("첫 번째 수=");
    scanf("%d", &x);
    printf("두 번째 수=");
    scanf("%d", &y);

    printf("큰 수=%d \n", (x > y) ? x : y);
    printf("작은 수=%d \n", (x < y) ? x : y);
}
```

Microsoft Visual Studio 디버그 콘솔

```
첫 번째 수=2
두 번째 수=3
큰 수=3
작은 수=2
```

Interim check



중간점검

1. 중괄호로 묶은 여러 개의 문장을 무엇이라고 하는가?
2. C에서 참과 거짓은 어떤 정수로 표시되는가?
3. if 문안의 조건식으로 많이 사용되는 수식의 종류는 무엇인가?
4. if 문이 끝나면 어떤 문장이 실행되는가?
5. 조건에 따라서 실행되어야 하는 문장이 두개 이상이면 어떻게 하여야 하는가?
6. 변수 n의 값이 100보다 크거나 같으면 "large", 100보다 작으면 "small"을 출력하는 if-else 문을 작성하라.

Logic operators

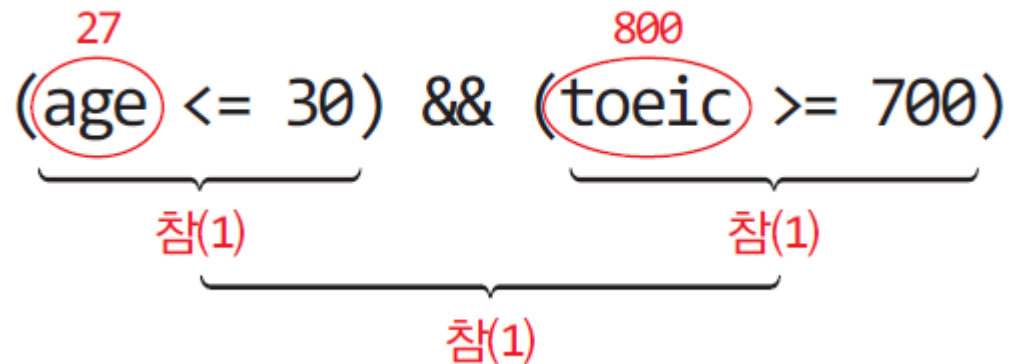
- Logic operators combine several conditions to determine whether they are true or false.

연산	의미
$x \ \&\& \ y$	AND 연산, x와 y가 모두 참이면 참, 그렇지 않으면 거짓
$x \ \ y$	OR 연산, x나 y중에서 하나만 참이면 참, 모두 거짓이면 거짓
$!x$	NOT 연산, x가 참이면 거짓, x가 거짓이면 참



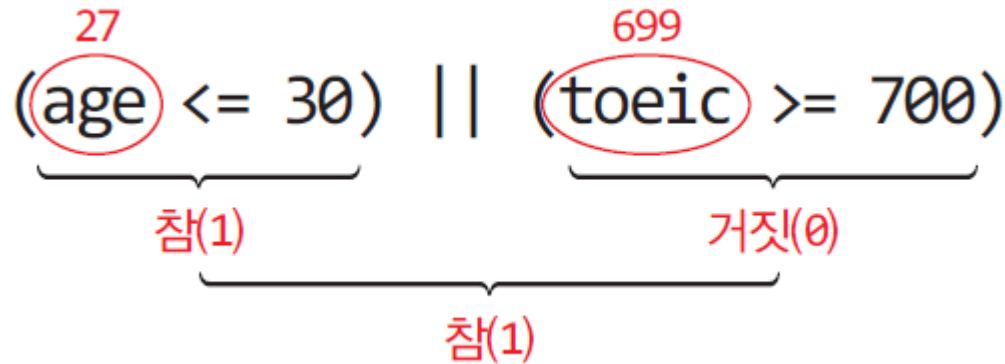
AND operator

- Suppose that the conditional sentence is under 30 years old and the TOEIC score is over 700.



OR operator

- Suppose that the conditional sentence is under 30 years old or the TOEIC score is over 700.



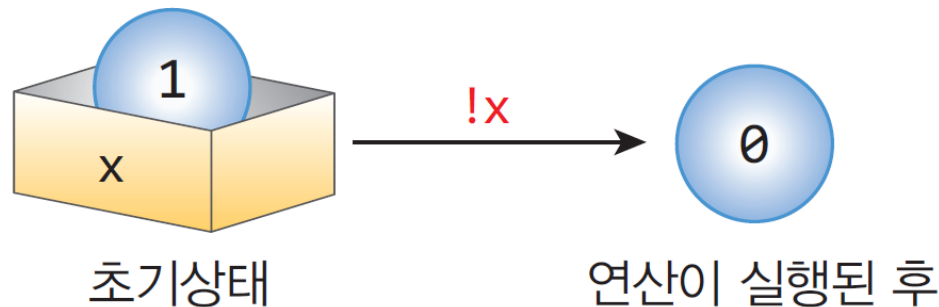
Use of multiple logical operators

- For example, the question "x is one of 1, 2, 3" can be written as follows.

```
if ((x == 1) || (x == 2) || (x == 3)) {  
    printf("x는 1, 2, 3 중의 하나입니다. \n");  
}
```

NOT operator

- The NOT operator makes the result value of the operation false if the value of the operand is true, and makes the result value of the operation true if the value of the operand is false.



Example

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

int main(void)
{
    int number;

    printf("정수를 입력하시오: ");
    scanf("%d", &number);

    if (number >= 0 && number <= 100)
        printf("입력한 정수가 0에서 100 사이에 있습니다. \n");
    else
        printf("입력한 정수가 0에서 100 사이가 아닙니다. \n");

    return 0;
}
```

Microsoft Visual Studio 디버그 콘솔

정수를 입력하시오: 98
입력한 정수가 0에서 100 사이에 있습니다.

Skipped operations

- For AND operators, if the value of the first operand is false among several operands, no other operand is calculated.



```
if( (2 > 3) && (++x < 5) ) {  
    ...  
}
```

Not operated

Check



참고: 참과 거짓

관계 수식이나 논리 수식은 수식이 거짓이면 0을 생성하고 참이면 1을 생성한다. 하지만 피연산자의 경우에는 0이면 거짓이고 0이 아닌 값은 무조건 참으로 간주한다.

따라서 NOT 연산자의 경우와 같이 !100 하면 100의 값을 참으로 간주하여 여기에 NOT을 적용하여 0값이 생성된다. 수식의 결과로는 항상 0 아니면 1만 생성되지만 피연산자로서 참과 거짓을 분류할 때는 0이면 거짓이고 나머지는 모두 참으로 간주한다.



경고

수학에서처럼 다음과 같이 수식을 만들면 프로그래머가 전혀 예상하지 못한 결과가 생성된다.

$(2 < x < 5)$

위의 수식은 x가 2보다 크고 5보다 작으면 참이고 그렇지 않으면 거짓일 거라고 생각된다. 하지만 C언어는 다음과 같이 연산을 진행한다.

$((2 < x) < 5)$

Interim check



중간점검

1. 2개의 정수를 입력받아서 큰 수에서 작은 수를 뺀 결과를 출력하는 프로그램을 작성해보자. 예를 들어서 2과 10를 입력하면 8을 출력하면 된다.
2. 정수를 입력받아서 2의 배수인 동시에 3의 배수인지를 검사하는 프로그램을 작성해보자.

Lab: coin flipping

- Each time a program is executed, the front and back must appear randomly as if flipping a real coin.



Sol: coin flipping

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

int main(void)
{
    printf("동전 던지기 게임을 시작합니다.\n");
    srand(time(NULL)); // 현재 시간을 난수의 씨앗값으로 한다.

    int coin = rand() % 2;
    if (coin == 0)
        printf("앞면입니다.\n");
    else
        printf("뒷면입니다.\n");
    return 0;
}
```

Lab: leap year determination program

- The observatory adjusts this deviation by inserting a leap year once every four years. Let's write a program that determines whether it is leap year using the if statement.



A screenshot of a Microsoft Visual Studio debug console window. The title bar is orange and contains the text "Microsoft Visual Studio 디버그 콘솔" (Microsoft Visual Studio Debug Console). The console area is white and contains two lines of Korean text: "연도를 입력하시오: 2024" (Enter the year: 2024) and "2024년은 윤년입니다." (2024 is a leap year). The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

- Leap year must meet the following conditions.
 - The year can be divided by four.
 - Excluding the years that can be divided by 100.
 - The year that can be divided by 400 is leap year.

Sol: leap year determination program

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

int main(void)
{
    int year;

    printf("연도를 입력하시오: ");
    scanf("%d", &year);

    if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0)
        printf("%d년은 윤년입니다.\n", year);
    else
        printf("%d년은 윤년이 아닙니다.\n", year);

    return 0;
}
```

Sequential if

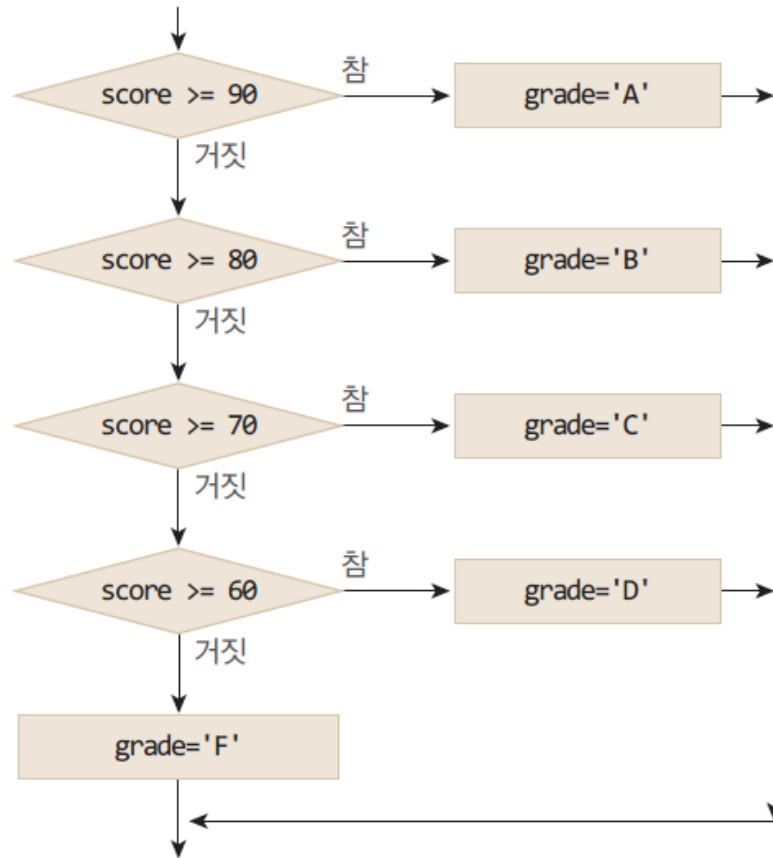
- Sequential if-else statements allow different code execution for three or more conditions.

Syntax 5.2 연속적인 if-else 문

```
형식  if (조건식1) {  
        // 조건식1이 참인 경우 실행할 명령문  
    }  
    else if (조건식2) {  
        // 조건식2이 참인 경우 실행할 명령문  
    }  
    ...  
    else {  
        // 모든 조건식이 거짓 일 때 실행될 명령문  
    }
```

Grade assignment program

- For example, depending on the student's score, I would like to assign grades as shown in the following procedure. Examine grades with several conditional formulas and execute sentences that meet the conditions.



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

```
int main(void)
{
    int score;

    char grade;

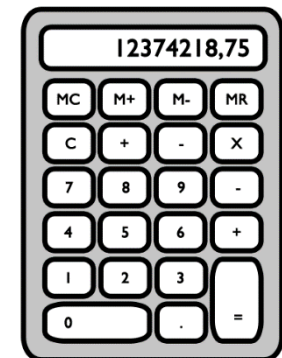
    printf("성적을 입력하시오: ");
    scanf("%d", &score);

    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("학점 %c \n", grade);
    return 0;
}
```

Calculator program

- Take two operands and create a program that can do +, -, *, / operations.




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

```
int main(void)
{
    char op;
    int x, y;

    printf("수식을 입력하시오: ");
    scanf("%d %c %d", &x, &op, &y);

    if (op == '+')
        printf("%d \n", x + y);
    else if (op == '-')
        printf("%d \n", x - y);
    else if (op == '*')
        printf("%d \n", x * y);
    else if (op == '/')
        printf("%d \n", x / y);
    else
        printf("지원되지 않는 연산자입니다. \n");

    return 0;
}
```

Lab: Find the largest of the three integers.

- Let's write a program that finds the largest of the three integers received from the user.



The screenshot shows the 'Microsoft Visual Studio 디버그 콘솔' (Debug Console) window. It contains two lines of Korean text: '3개의 정수를 입력하시오: 10 20 30' (Enter 3 integers: 10 20 30) and '가장 큰 정수는 30이다.' (The largest integer is 30). The window has a standard Windows title bar with minimize, maximize, and close buttons.

```
Microsoft Visual Studio 디버그 콘솔
3개의 정수를 입력하시오: 10 20 30
가장 큰 정수는 30이다.
```

Sol: Find the largest of the three integers.

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

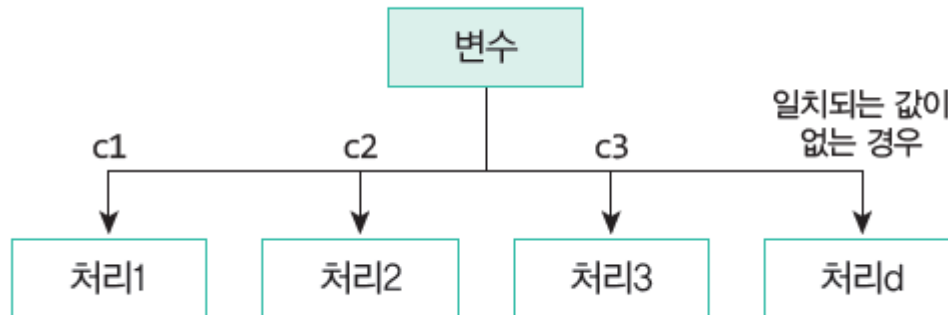
int main(void)
{
    int a, b, c, largest;

    printf("3개의 정수를 입력하시오: ");
    scanf("%d %d %d", &a, &b, &c);

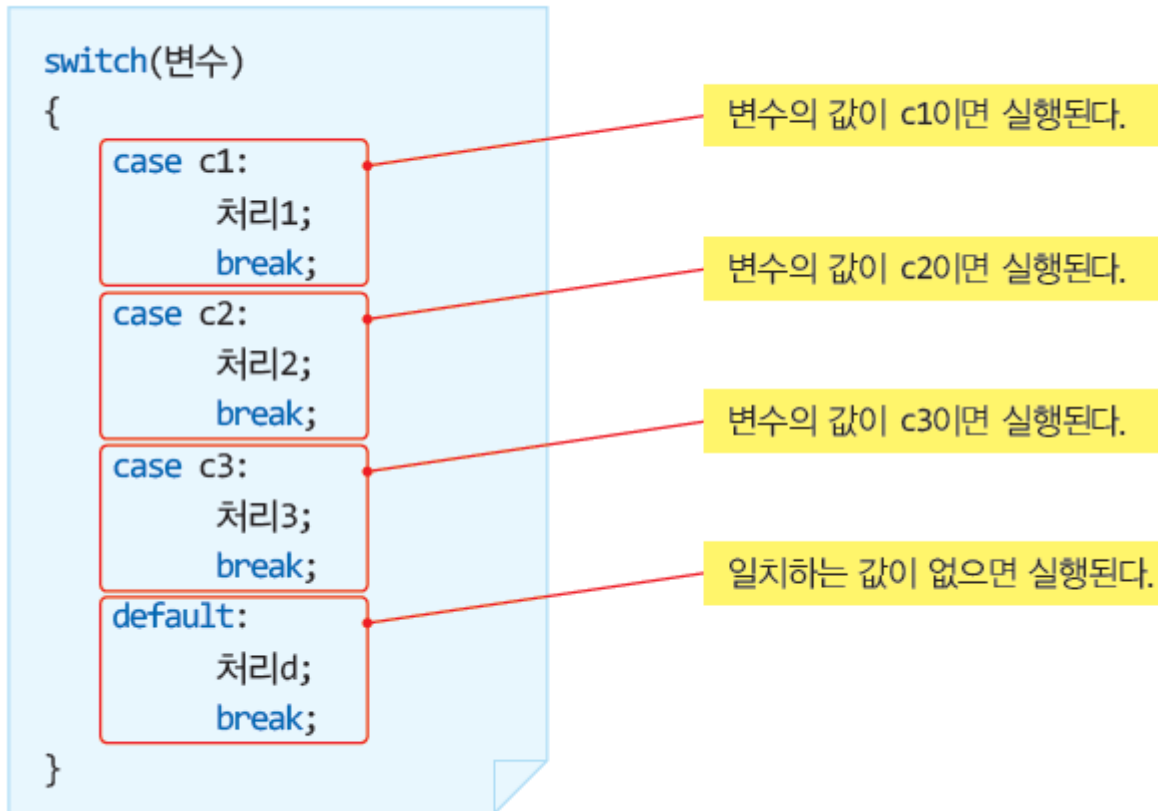
    largest = a;           // 일단 a가 최대값이라고 가정
    if (largest < b) largest = b;      // b가 최대값보다 크면 b를 최대값으로 한다.
    if (largest < c) largest = c;      // c가 최대값보다 크면 c를 최대값으로 한다.
    printf("가장 큰 정수는 %d이다. \n", largest);
    return 0;
}
```

switch statement

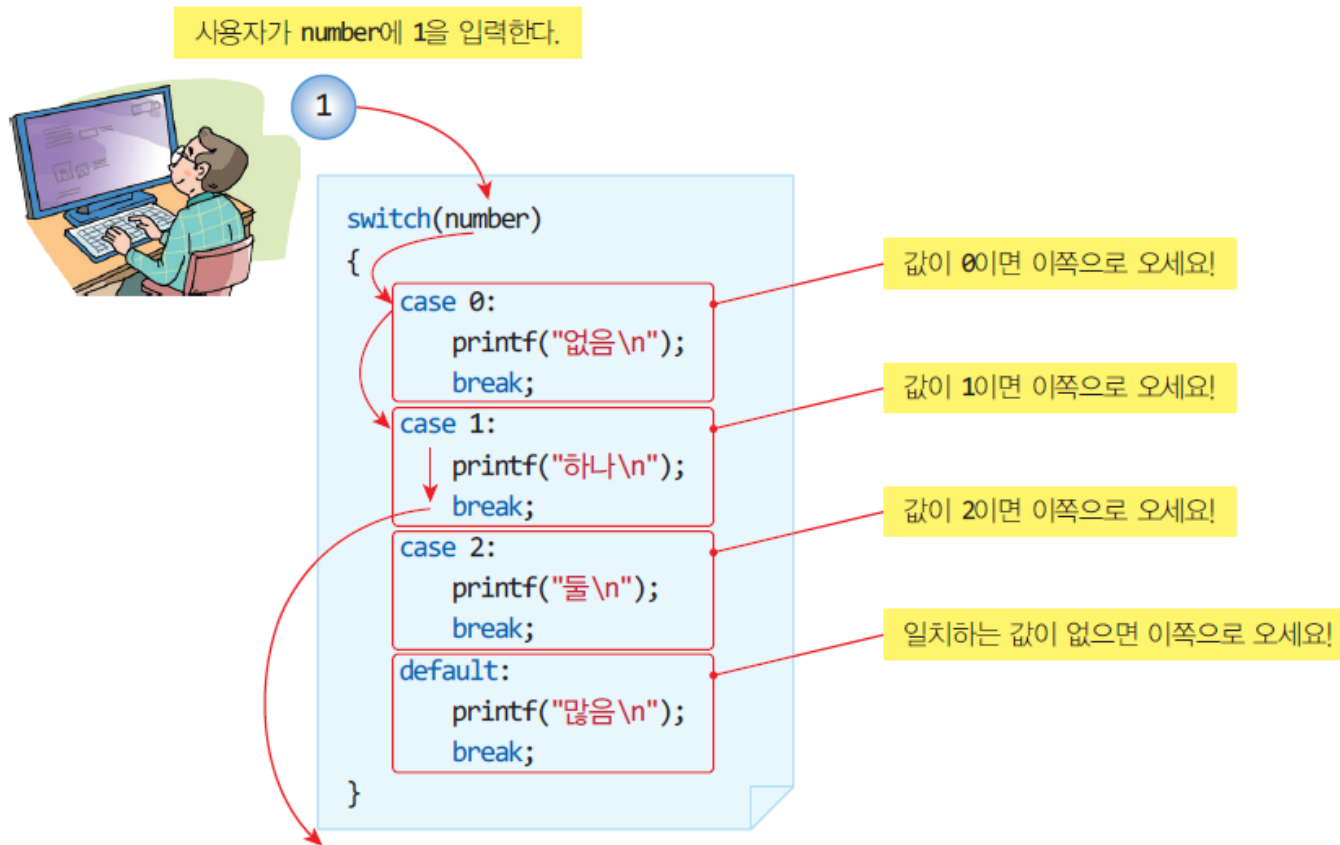
- Use to choose one of the many things



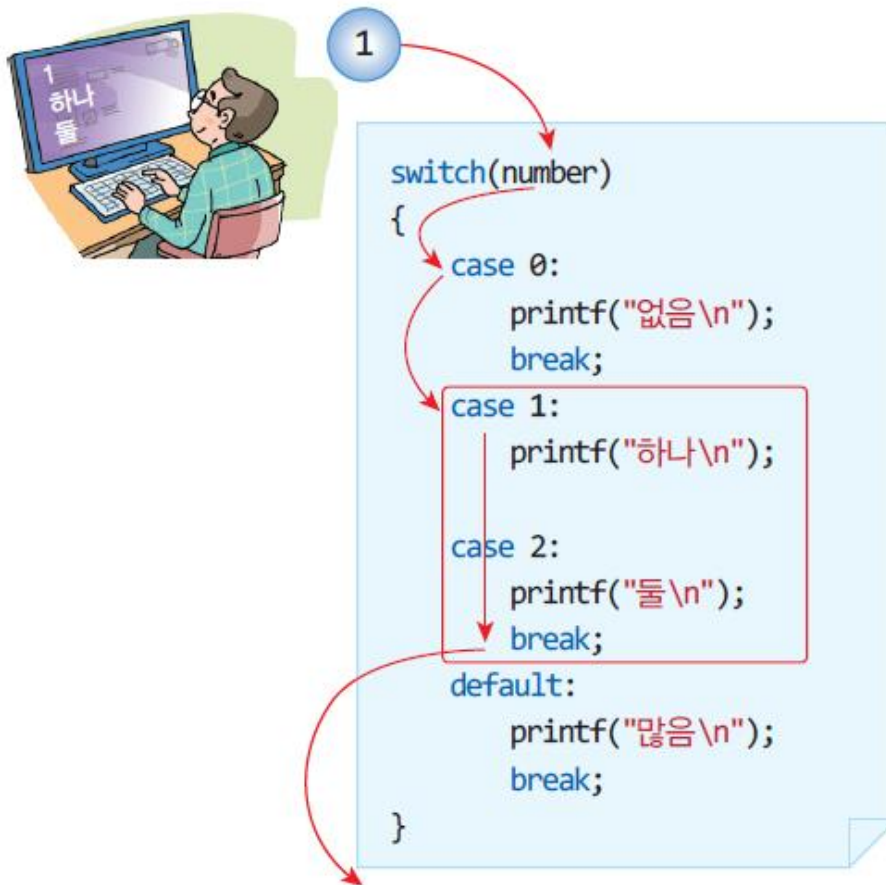
switch statement



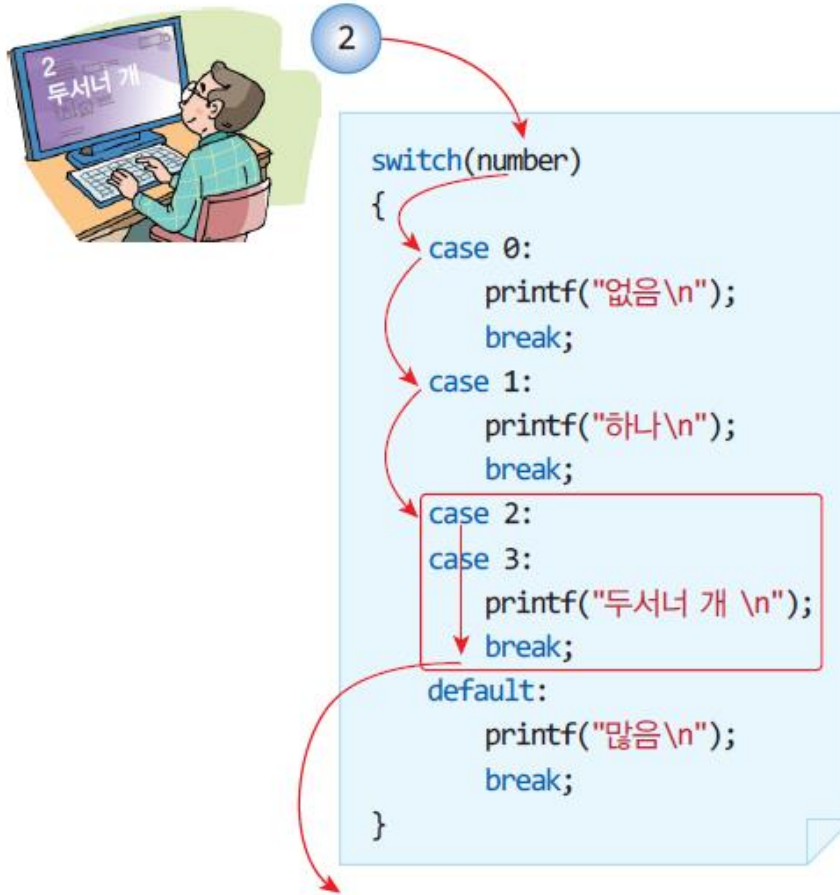
Example



If there is no "break"



Intentional break omission



default statement

- The default statement is selected and executed if it does not match any case statement.



5

```
switch(number)
{
    case 0:
        printf("없음\n");
        break;
    case 1:
        printf("하나\n");
        break;
    case 2:
        printf("둘\n");
        break;
    default:
        printf("많음\n");
        break;
}
```

값이 0이면 이쪽으로 오세요!

값이 1이면 이쪽으로 오세요!

값이 2이면 이쪽으로 오세요!

일치하는 값이 없으면 이쪽으로 오세요!

switch and if-else statements

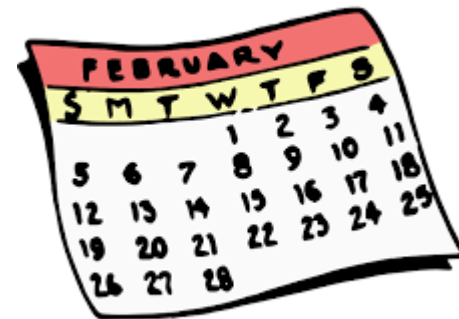
```
switch(number)
{
    case 0:
        printf("없음\n");
        break;
    case 1:
        printf("하나\n");
        break;
    case 2:
        printf("둘\n");
        break;
    default:
        printf("많음\n");
        break;
}
```



```
if( number == 0 )
    printf("없음\n");
else if( number == 1 )
    printf("하나\n");
else if( number == 2 )
    printf("둘\n");
else
    printf("많음\n");
```

A program that prints out the number of days of the month when a month is given.

```
Microsoft Visual Studio 디버그 콘솔
일수를 알고 싶은 달을 입력하시오: 5
5월의 일수는 31입니다.
```



```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
int main(void)
{
    int month, days;
    printf("일수를 알고 싶은 달을 입력하시오: ");
    scanf("%d", &month);

    switch (month)
    {
        case 2:
            days = 28;
            break;
        case 4:
        case 6:
        case 9:
        case 11:
            days = 30;
            break;
        default:
            days = 31;
            break;
    }
    printf("%d월의 일수는 %d입니다.\n", month, days);
    return 0;
}
```

Interim check



중간점검

1. case 절에서 break 문을 생략하면 어떻게 되는가?
2. 변수 fruit의 값이 각각 1, 2, 5일 때, 다음의 코드의 출력을 쓰시오.

```
switch(fruit) {  
    case 1:    printf("사과");  
               break;  
    case 2:    printf("배");  
    case 3:    printf("바나나");  
               break;  
    default:   printf("과일");  
}
```

Mini Project: Rock-paper-scissors game.

- Let's write a game of rock-paper-scissors in which computers and people compete.

```
Microsoft Visual Studio 디버그 콘솔
가위, 바위, 보 게임에 오신 것을 환영합니다.
하나를 선택하세요(가위-0, 바위-1, 보-2): 0
사용자=0
컴퓨터=2
사용자 승리
```



Q & A

