



학습자가 강의 저작물을 다운로드·캡처 받아 **외부로 유출하는 행위**는 저작권자의  
이용허락 없이 저작물을 복제·공중송신 또는 배포 하는 것으로 **저작권 침해 행위**에 해당함.

# C 프로그래밍

(001/002)

## 제 4 강

신 한 대 학 교  
소프트웨어융합학과  
교수 송 진 희

# C 프로그래밍

## 제 4 강

- 형(Type) 변환
- 제어문
  - 1) 조건문
    - if()문
    - switch()문
  - 2) 분기문 : go to 문

# 학습 목표

- 구분한 자료형을 연산 처리에서 활용하고, 연산 수식에 자료 형 변환을 적용할 수 있다.
- if()문을 이용한 조건 처리를 표현할 수 있다.
- if-else()문을 이용한 조건 처리를 표현할 수 있다.
- if-else if()문을 이용한 조건 처리를 표현할 수 있다.
- switch-case()문을 이용한 조건 처리를 표현할 수 있다.

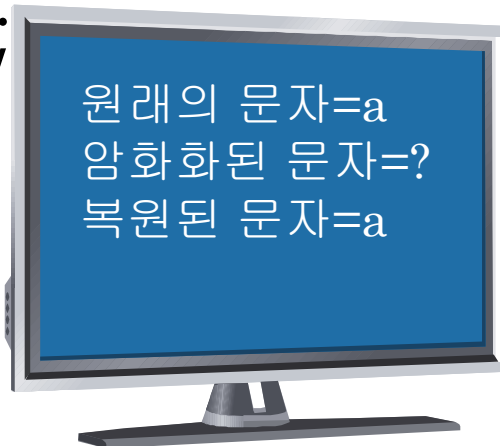
# 3강 - 정리 요약

- 문자에 대한 암호화 방법의 예

- 하나의 문자  $x$ 를 암호화
- $x = x \wedge \text{key};$

- 복호화

- $x = x \wedge \text{key};$



```
#include <stdio.h>
int main(void)
{
    char data = 'a';
    char key = 0xff;
    char encrypted_data, orig_data;

    printf("원래의 문자=%c\n", data);

    encrypted_data = data ^ key;
    printf("암호화된 문자=%c\n", encrypted_data);

    orig_data = encrypted_data ^ key;
    printf("복원된 문자=%c\n", orig_data);

    return 0;
}
```

# 3강 - 정리 요약

## ○ 연산자의 종류와 우선순위

➤ 단항연산자 > 산술 연산자 > 관계연산자 > 비트연산자 > 논리 연산자

연산	연산자	C 수식
덧셈	+	$x + y$
뺄셈	-	$x - y$
곱셈	*	$x * y$
나눗셈	/	$x / y$
나머지	%	$x \% y$

연산	의미
$x == y$	x와 y가 같은가?
$x != y$	x와 y가 다른가?
$x > y$	x가 y보다 큰가?
$x < y$	x가 y보다 작은가?
$x >= y$	x가 y보다 크거나 같은가?
$x <= y$	x가 y보다 작거나 같은가?

연산	의미
$x \&\& y$	AND 연산
$x \ \ y$	OR 연산
$!x$	NOT 연산

## ○ 기타 연산자

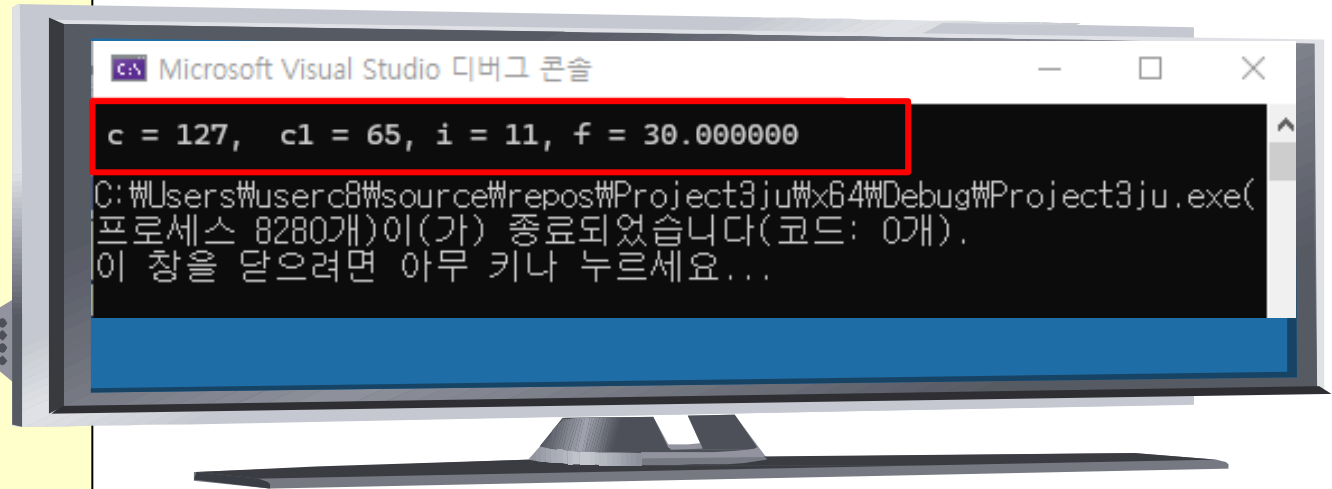
➤ 증감 연산자(++ , --), 복합 연산자(+=, \*=, ..), 조건 연산자(? :)

# 자동 형 변환(1)

○정수 연산 시 char형, short형은 자동적으로 int형으로

```
#include <stdio.h>
int main(void)
{
    char c, c1;
    int i;
    float f;

    c = 127;           // 0~127은 그대로 출력( $2^8 - 1$ )
    c1 = 'A';          // 문자 A의 아스키 코드 값 출력
    i = 1.23 + 10;      // 11.23에서 소수점 이하 버림
    f = 10 + 20;        // 합계 30을 실수 형으로 변환
    printf("c = %d, c1 = %d, i = %d, f = %f \n", c, c1, i, f);
    return 0;
}
```



# 자동 형 변환(2)

○수식에 서로 다른 자료형이 혼합 사용되면, 큰 자료형으로 변환

➤ double > float > int > short

○명시적인 형 변환은 변수나 데이터 앞에 ( )를 사용해서  
형 지정

Syntax

형변환

예

자료형

(int)1.23456

(double) x

(long) (x+y)

수식

// int형으로 변환

// double형으로 변환

// long형으로 변환



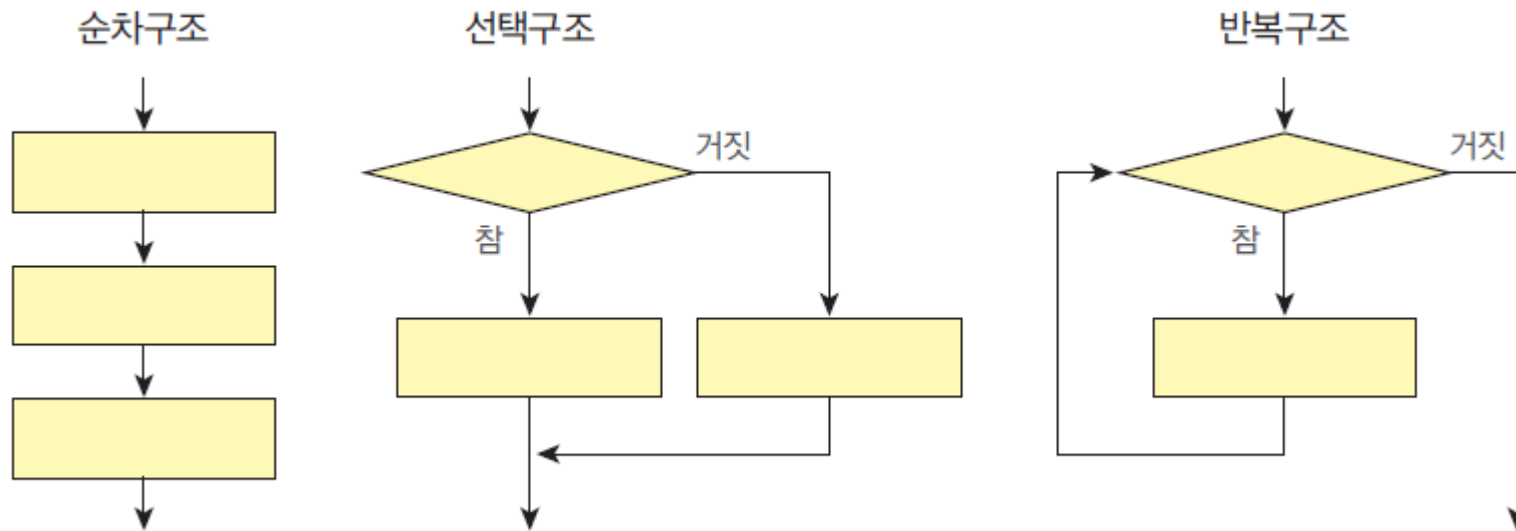
# 6장. 조 건 문

# 제어문(Control Statement)

## ○ 특정 조건 또는 반복에 의해 명령문의 실행 순서를 변경하는 것

➤ If() / switch() / for() / while() / do-while() / goto()

## ○ 제어 구조



# if문의 구조

## ○ 주어진 조건에 따라 수행하는 명령문

➤ if()문 / if()~else문 / if~else if()문

- 1) if() 문 : 주어진 조건이 참(True)일 경우에만 명령을 실행
- 2) if()~else 문 : 주어진 조건이 참(True)일 경우에는 if()와 else 사이에 기술된 명령을 실행, 조건이 거짓(false)이면  
else  
이후에 기술된 명령문 실행
- 3) if()~else if()문 : 주어진 조건이 참(True)일 경우에는 if()와  
else if() 사이에 기술된 명령을 실행, 조건  
이 거짓  
(false)이면 다시 조건을 확인하는 명령문  
실행

# if문의 구조

## ○if()문의 형식과 순서도

Syntax if 문

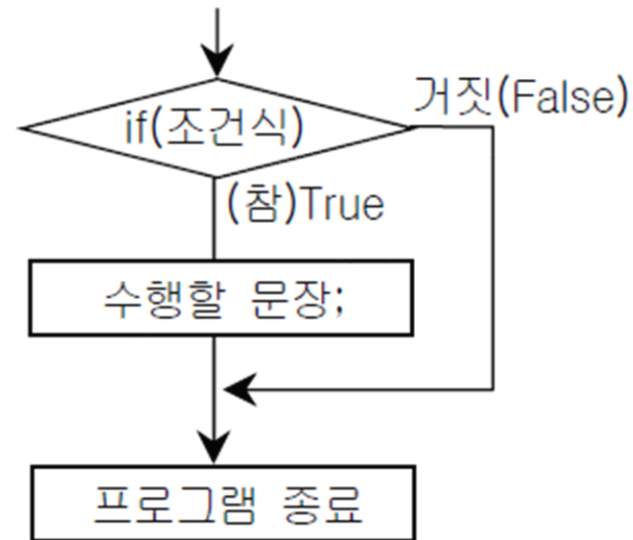
예

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

조건식

조건식이 참인 경우에만 문장이 실행된다.

if(조건식)  
수행할 문장;



# 예제

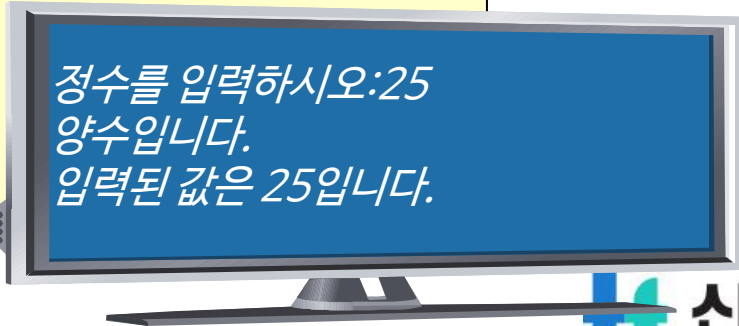
```
#include <stdio.h>
int main(void)
{
    int number;

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

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

    printf("입력된 값은 %d입니다. \n", number);

    return 0;
}
```



정수를 입력하시오:25  
양수입니다.  
입력된 값은 25입니다.

## ○복합문(compound statement)

- 중괄호를 사용하여 문장들을 그룹핑 하는 것(2개 이상의 실행문으로 구성)
- 블록(block)이라고도 한다.

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

- 조건식이 참이면 2개의 문장이 차례로 실행
- 조건식이 거짓이면 해당 블록을 실행하지 않음

# 실수 비교

## 참고사항

실수와 실수를 비교할 때는 다음과 같은 문장을 사용하는 것은 문제가 될 수 있다.

```
if (result == expectedResult) { ... }
```

위의 비교는 참이 되기 힘들다. 왜냐하면 0.2와 같은 단순한 값은 정확하게 표현되지만 복잡한 값은 정확하게 표현되지 않기 때문이다. 따라서 부동소수점 수 2개가 같은지를 판별하려면 다음과 같이 오차를 감안하여서 비교하여야 한다. 즉 2개의 숫자가 오차 이내로 아주 근접하면 같은 것으로 판정하는 방법이다.

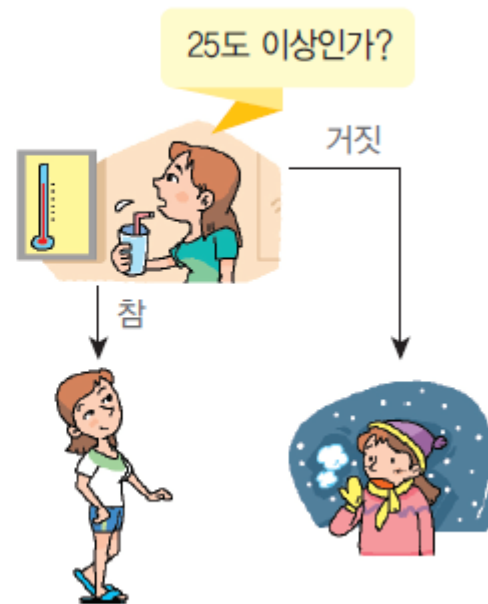
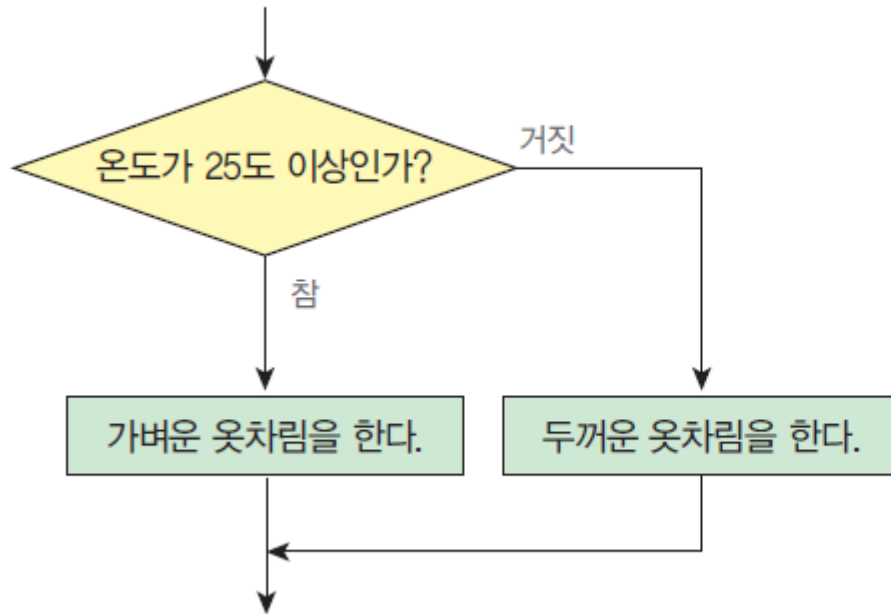
```
if (fabs(result - expectedResult) < 0.00001) { ... }
```

`fabs()` 함수는 실수의 절대값을 계산하여서 반환한다.

오차가 무시할 만 하면  
같은 것으로 인정

`(0.2 == 0.21)`

# if-else 문



## Syntax if-else 문

예

조건식  
`if( number > 0 )`

`printf("양수입니다.\n");`

`else`

`printf("양수가 아닙니다.\n");`

만약 조건식이 참이면  
문장1이 실행된다.

그렇지 않으면 문장2가 실행  
된다.



# if-else 문

```
if ( score >= 60 )
```

```
    printf("합격입니다.\n");
```

```
else
```

```
    printf("불합격입니다.\n");
```

score가 60이상이면 실행

score가 60미만이면 실행

```
if ( score >= 60 )
```

```
{
```

```
    printf("합격입니다.\n");
```

```
    printf("장학금도 받을 수 있습니다.\n");
```

```
}
```

```
else
```

```
{
```

```
    printf("불합격입니다.\n");
```

```
    printf( " 다시 도전하세요.\n");
```

```
}
```

score가 60이상이면 실행

score가 60미만이면 실행

# 복잡한 조건식도 가능

## ○학점 결정 코드

```
if( score >= 80 && score < 90 )  
    grade = 'B';
```

## ○공백 문자와 줄바꿈, 탭 문자의 개수를 세는 코드

```
if( ch == ' ' || ch == '\n' || ch == '\t' )  
    white_space++;
```

# if-else 문의 스타일

## 스타일

if-else 문은 보통 다음의 2가지 중의 하나의 스타일을 이용하는 것이 좋다. 이 책에서는 주로 첫 번째 방법을 사용하지만 지면이 부족할 때는 두 번째 방법도 사용하였다.

복합문은 들여쓰기를 하는 편이 읽기가 쉬워진다.

```
if( expression )
{
    → statement11;
    statement12;
    ...
}
else
{
    → statement21;
    statement22;
    ...
}
```

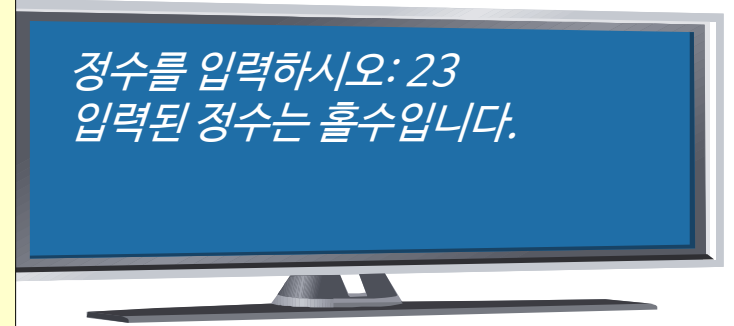
공간의 절약을 위하여 이런 형태로 작성하기도 한다.

```
if( expression ){
    statement11;
    statement12;
    ...
}
else {
    statement21;
    statement22;
    ...
}
```

# 예제 #1

○키보드에서 입력 받은 정수가 홀수인지 짝수인지를 말해주는 프로그램을 작성하여 보자. 홀수와 짝수는 어떻게 구별할 수 있는가?

```
// if-else 문을 이용하여 홀수와 짝수를 구분한다.  
#include <stdio.h>  
  
int main(void)  
{  
    int number;  
  
    printf("정수를 입력하시오:");  
    scanf("%d", &number);  
  
    if( number % 2 == 0 )  
        printf("입력된 정수는 짝수입니다.\n");  
    else  
        printf("입력된 정수는 홀수입니다.\n");  
  
    return 0;  
}
```



# 예제 #2

- 사용자로부터 두 개의 정수를 입력 받아서 정수 간의 나눗셈을 실행한다. 나눗셈을 하기 전에 분모가 0인지를 if 문을 이용하여 검사한다.

```
// 나눗셈을 하기 전에 분모가 0인지를 if-else 문을 이용하여 검사
#include <stdio.h>

int main(void)
{
    int n, d, result;

    printf("분자와 분모를 입력하십시오: ");
    scanf("%d %d", &n, &d);
    if( d == 0 )
    {
        printf("0으로 나눌 수는 없습니다.\n");
    }
    else
    {
        result = n / d;
        printf("결과는 %d입니다.\n", result);
    }
    return 0;
}
```



분자와 분모를 입력하십시오: 5 4  
결과는 1입니다.

# 예제 #3

## ○윤년인지 아닌지를 판단하는 프로그램을 작성

- 연도가 4로 나누어 떨어지면서 100으로 나누어 떨어지지 않은 연도
- 400으로 나누어 떨어지는 연도

```
// 윤년 판단 프로그램
#include <stdio.h>

int main(void)
{
    int year;

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

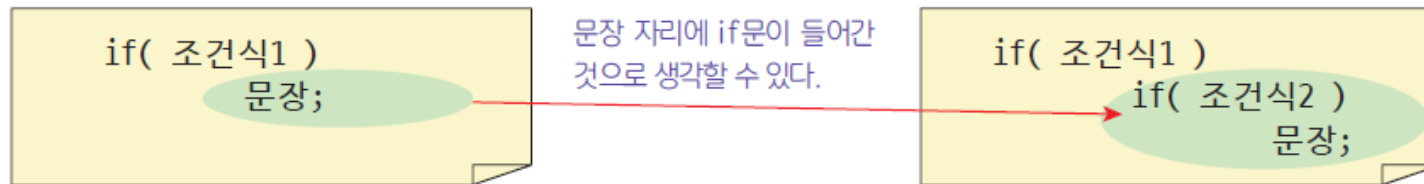
    if((year % 4 == 0 && year % 100 != 0) || year % 400 == 0)
        printf("%d년은 윤년입니다.\n", year);
    else
        printf("%d년은 윤년이 아닙니다.\n", year);

    return 0;
}
```



# 중첩 if문

## ○if 문에 다시 if 문이 포함



```
if( score >= 80 )  
    if( score >= 90 )  
        printf("당신의 학점은 A입니다.\n");
```

```
if( score >= 80 )  
    if( score >= 90 )  
        printf("당신의 학점은 A입니다.\n");  
    else  
        printf("당신의 학점은 B입니다.\n");
```

# if() ~ else if()문

## Syntax

### 연속적인 if 문

#### 문법

```
if( 조건식1 )
```

```
    문장1;
```

```
else if( 조건식2 )
```

```
    문장2;
```

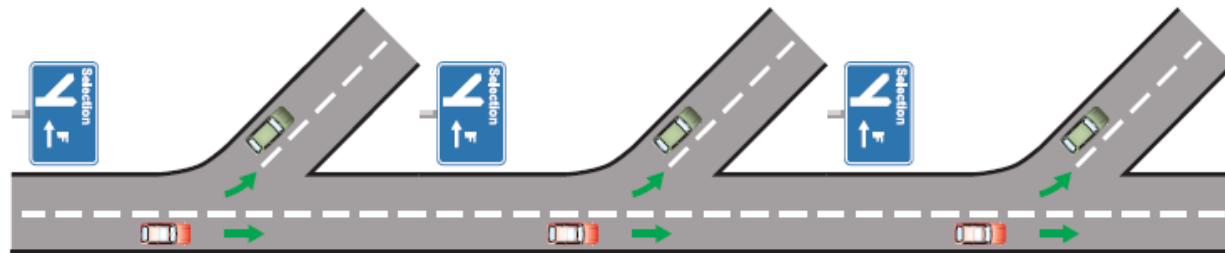
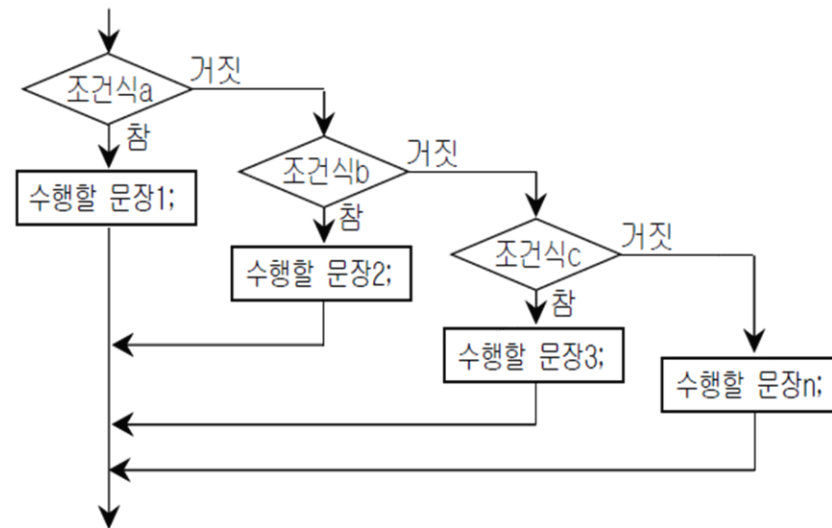
```
else if( 조건식3 )
```

```
    문장3;
```

```
else
```

```
    문장4;
```

만약 조건식1이 참이면 문장1이 실행된다.  
그렇지 않고 조건식2가 참이면 문장2가 실행된다.  
그렇지 않고 조건식3이 참이면 문장3이 실행된다.  
그렇지 않으면 문장4가 실행된다.





# 학점 결정 예제

○ 학생들의 성적을 받아서 학점(A, B, C, D, F)을 출력하는 프로그램을 작성하여 보자.

```
#include <stdio.h>

int main(void)
{
    int score;

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

    if (score >= 90)
        printf("합격: 학점A\n");
    else if (score >= 80)
        printf("합격: 학점B\n");
    else if (score >= 70)
        printf("합격: 학점C\n");
    else if (score >= 60)
        printf("합격: 학점D\n");
    else
        printf("불합격: 학점F\n");
    return 0;
}
```



# 문자 분류 예제

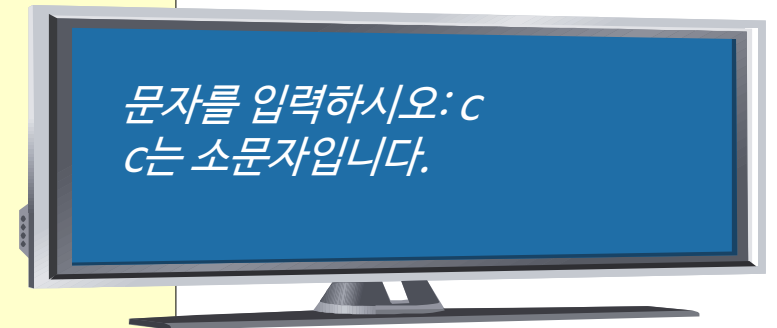
- 키보드에서 문자를 받아서 문자들을 대문자(A-Z), 소문자(a-z), 숫자(0-9), 그 외의 문자들로 구분하여 보자.
- 문자를 입력 받는 함수인 **getchar( )**를 사용

```
// 문자들을 분류하는 프로그램
#include <stdio.h>
int main(void)
{
    char ch;
    printf("문자를 입력하시오: ");

    ch = getchar(); //한 개 글자 입력 받음

    if(ch >= 'A' && ch <= 'Z')
        printf("%c는 대문자입니다.\n", ch);
    else if(ch >= 'a' && ch <= 'z')
        printf("%c는 소문자입니다.\n", ch);
    else if(ch >= '0' && ch <= '9')
        printf("%c는 숫자입니다.\n", ch);
    else
        printf("%c는 기타 문자입니다.\n", ch);

    return 0;
}
```



# [문제 1]

## 1. 컵의 사이즈를 정수로 입력 받아

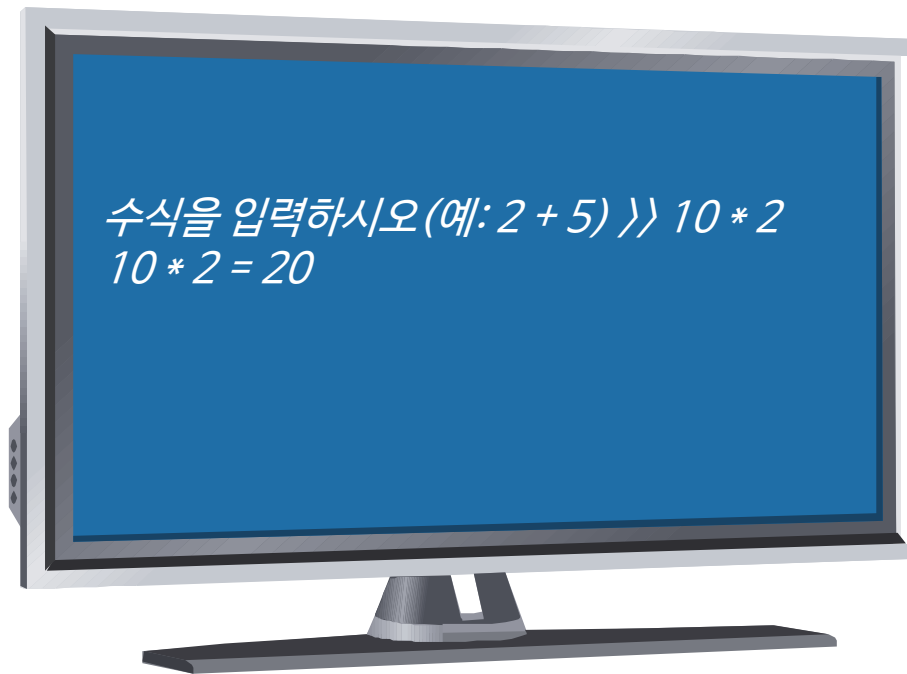
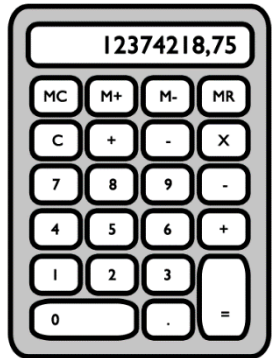
- ‘100ml’ : “Small”
- ‘100ml 이상 200ml 미만’ : “Medium”
- ‘200ml 이상’ : “Large”

의 문자열이 출력되는 프로그램을 작성하시오.  
(연속적인 if-else 문 사용)



# [문제2] 산술 계산기

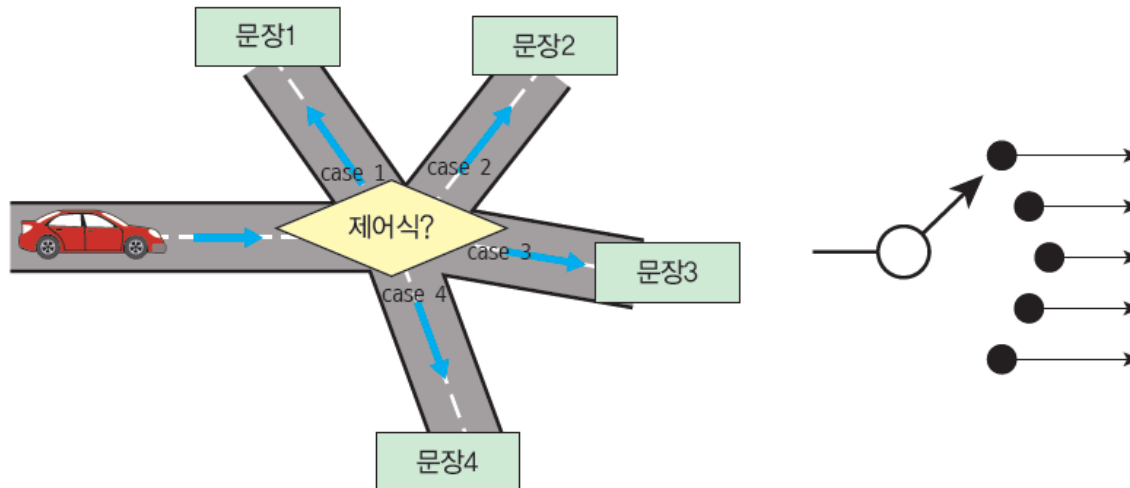
○ 두 개의 숫자와 연산자를 입력받아 계산 결과를 출력하는 프로그램을 작성하시오.



# switch ~ case문

## ○제어식의 값에 따라서 여러 경로 중에서 하나를 선택할 수 있는 제어 구조

- 다중 if~else문은 주어진 조건이 많아질수록 조건식을 여러 번 비교하는 단점을 보완하기 위해 사용
- 각각의 'case'문을 수행한 다음에 빠져나올 때는 'break;'문을 선언
- 조건에 일치하는 case문이 없을 경우에는 'default:'를 수행
- case(상수)로 선언(상수 : 문자 상수와 정수 상수만 허용)



# switch ~ case문

## Syntax switch 문

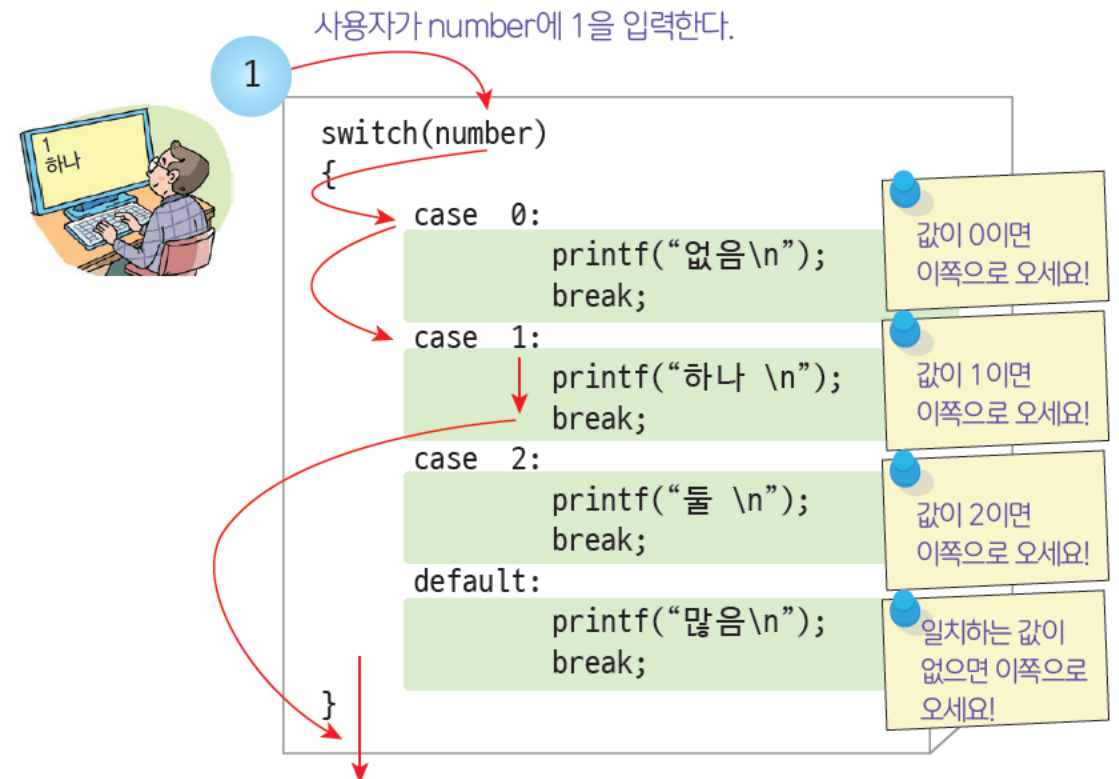
### 문법

```
switch(제어식)
{
    case c1:
        문장1;
        break;
    case c2:
        문장2;
        break;
    ...
    default:
        문장d;
        break;
}
```

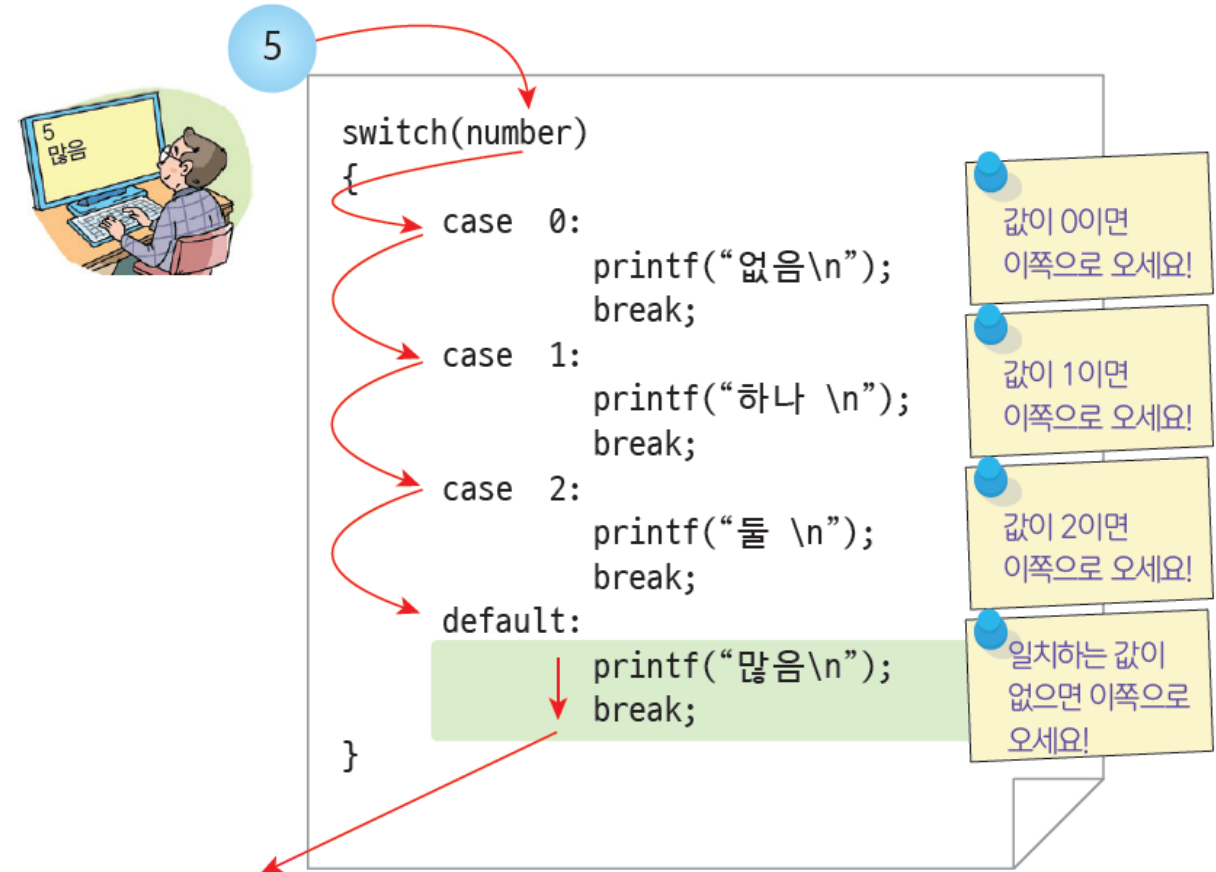
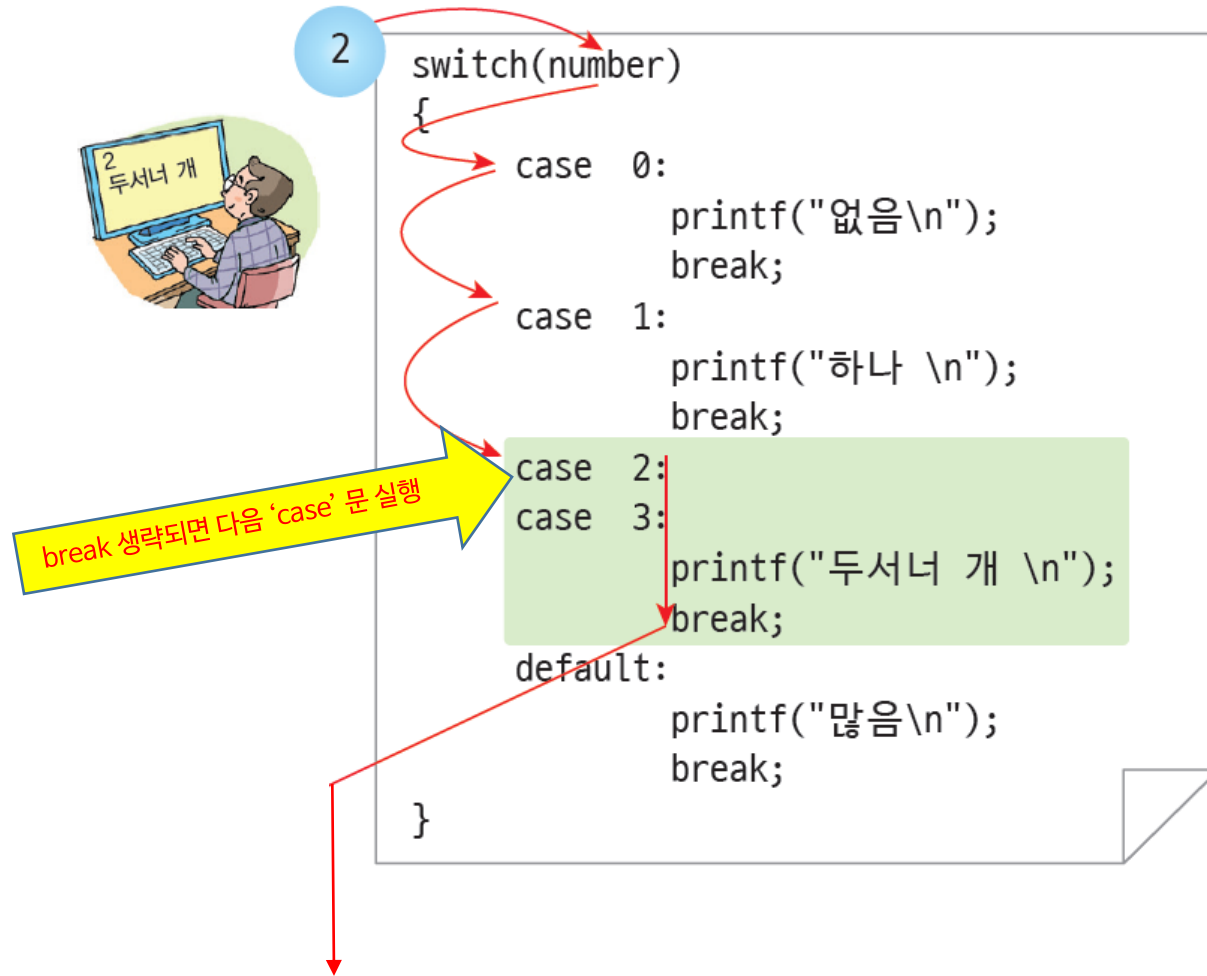
제어식의 값이 c1이면 실행된다.

제어식의 값이 c2이면 실행된다.

일치하는 값이 없으면 실행된다.



# Break 생략 / default 문



# switch 문의 유의 사항

```
switch(number)
{
    case x:
        printf("x와 일치합니다. %n ");
        break;
    case (x+2):
        printf("수식과 일치합니다. %n ");
        break;
    case 0.001:
        printf("실수 %n ");
        break;
    case 'a':
        printf("문자 %n ");
        break;
    case "001":
        printf("문자열 %n ");
        break;
}
```

// 변수는 사용할 수 없다.

// 변수가 들어간 수식은 사용할 수 없다.

// 실수는 사용할 수 없다.

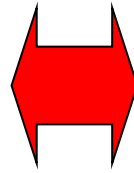
// 문자는 사용할 수 있다.

// 문자열은 사용할 수 없다.



# switch 문과 if-else 문

```
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");
```

# 정수의 범위를 나타낼 때

```
switch (score) {  
    case 100:  
    case 99:  
    case 98:  
    ...  
    case 90:  
        printf("A학점입니다.\n");  
        break;  
    ...  
}
```



```
if( score >= 90 && score <= 100 )  
    printf("A학점입니다.\n");
```

정수의 범위도 표현할 수 있으나 번거롭다.

# 정수의 범위를 나타낼 때

```
int iscore;
...
iscore = score/10;           // 정수 나눗셈의 경우, 나머지는 없어진다.
switch (iscore) {
    case 9: grade = 'A'; break; // 90-100은 A 학점
    case 8: grade = 'B'; break; // 80-89은 B 학점
    case 7: grade = 'C'; break; // 70-79은 C 학점
    case 6: grade = 'D'; break; // 60-69은 D 학점
    default: grade = 'F'; break; // 59점 이하는 F 학점
}
```

**Q** switch 문과 if/else 체인 중에서 어떤 것이 더 효율적인가?

차이는 미소하다. 하지만 switch 문은 간략한 점프 테이블로 효율적으로 구현이 가능하도록 설계되었다. 따라서 대부분의 경우 switch를 사용하는 것이 좋다. 코드가 간결하고 아마 약간은 효율적이다.

# 예제

○ 각 달의 일수를 출력하는 프로그램을 작성하자. 즉, 달을 입력하면 그 달의 일수를 출력한다.

// 달의 일수를 계산하는 프로그램

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

```
int main(void)
```

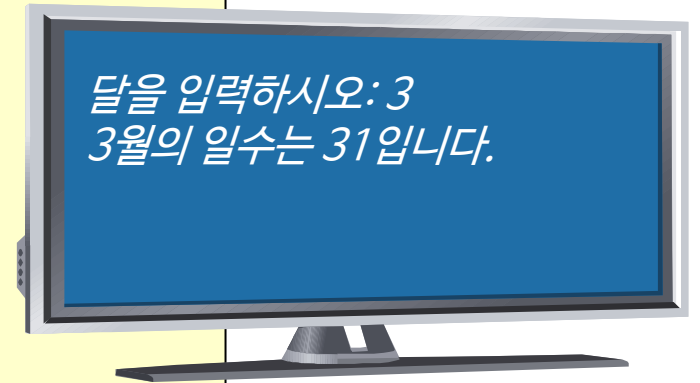
```
{
```

```
    int month, days;
```

```
    printf("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;
    }
```



# [ 문제 3 ]

1. 변수 fruit의 값이 각각 1이면 '사과', 2이면 '배', 3이면 '바나나', 그 이외는 '과일'이 출력되도록 switch 문을 사용해서 프로그램을 작성하시오.

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

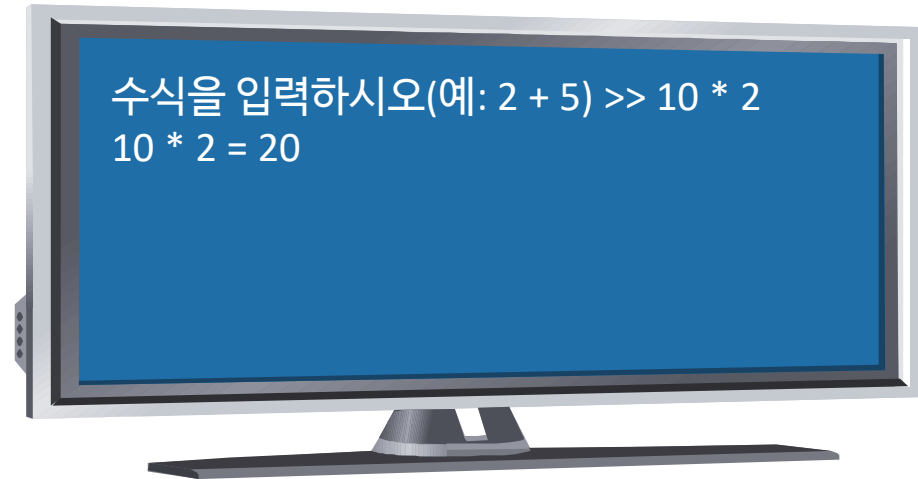
# [문제 4] 산술 계산기

○ [문제2]의 산술 계산기 예제를 switch 문을 이용하여 완성하시오.

```
// 간단한 산술 계산기 프로그램
#include <stdio.h>

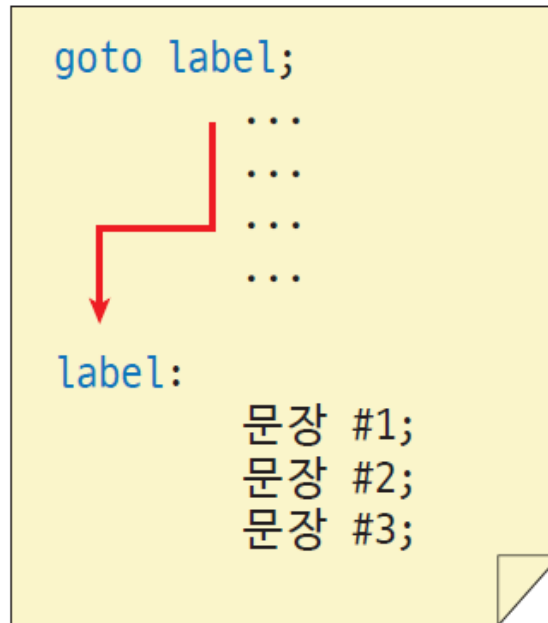
int main(void)
{
    char op;
    int x, y, result;

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

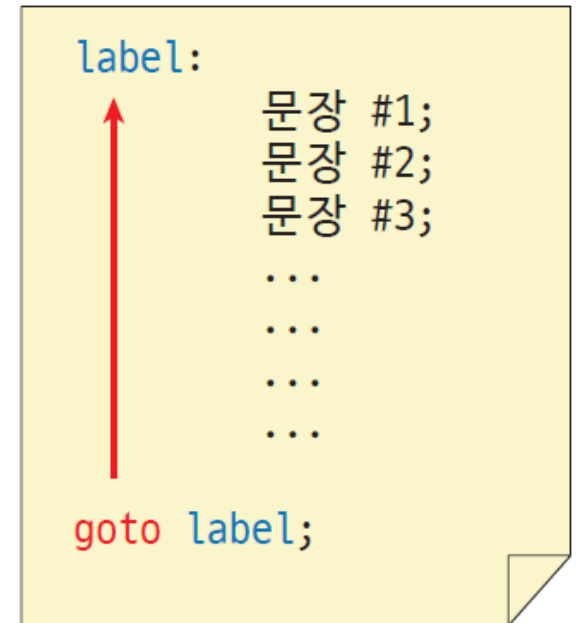


# goto문

- 조건없이 어떤 위치로 점프
- 사용하지 않는 것이 좋음



전향 참조



후향 참조

# 예제

```
// 구구단출력프로그램
#include <stdio.h>

int main(void)
{
    int i = 1;

loop:
    printf("%d * %d = %d \n", 3, i, 3*i);
    i++;

    if(i == 10) goto end;
    goto loop;

end:
    return 0;
}
```



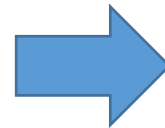


# [문제 5] 소득세 계산

○ 다음 조건에 따른 소득세를 계산해서 출력하는 프로그램을 작성하시오.

- 1) 산출세액 = 과세표준 \* 세율 - 누진공제액  
$$= 35,000,000 * 15\% - 1,080,000 = 4,170,000$$
- 2) 과세표준별 세율 및 누진 공제표

세율표		
과세표준	세율	누진공제
0 ~ 11,999,999 :	6%	0
12,000,000 ~ 45,999,999 :	15%	1,080,000
46,000,000	24%	5,220,000
88,000,000	35%	14,900,000
150,000,000	38%	19,400,000
300,000,000	40%	25,400,000
500,000,000	42%	35,400,000
1,000,000,000	45%	65,400,000



과세 표준: 35000000  
소득세율: 15%  
(공제전) 소득세: 5250000  
누진공제액: 1080000  
산출세액: 4170000

# 4강 – 정리 요약

- 수식에 서로 다른 자료형이 혼합 사용되면, 큰 자료형으로 자동 변환
  - double > float > int > short
- 명시적인 형 변환은 변수나 데이터 앞에 ( )를 사용해서 형 지정 가능
  - float x;  
int k;  
x = (float)k \* 2.5 ;
- 주어진 조건에 따라 수행하는 명령문
  - if()문 / if()~else문 / if~else if()문
  - switch() ~ case 문

