

개정3판

Visual
Studio
2017

쉽게 풀어쓴

C언어 EXPRESS

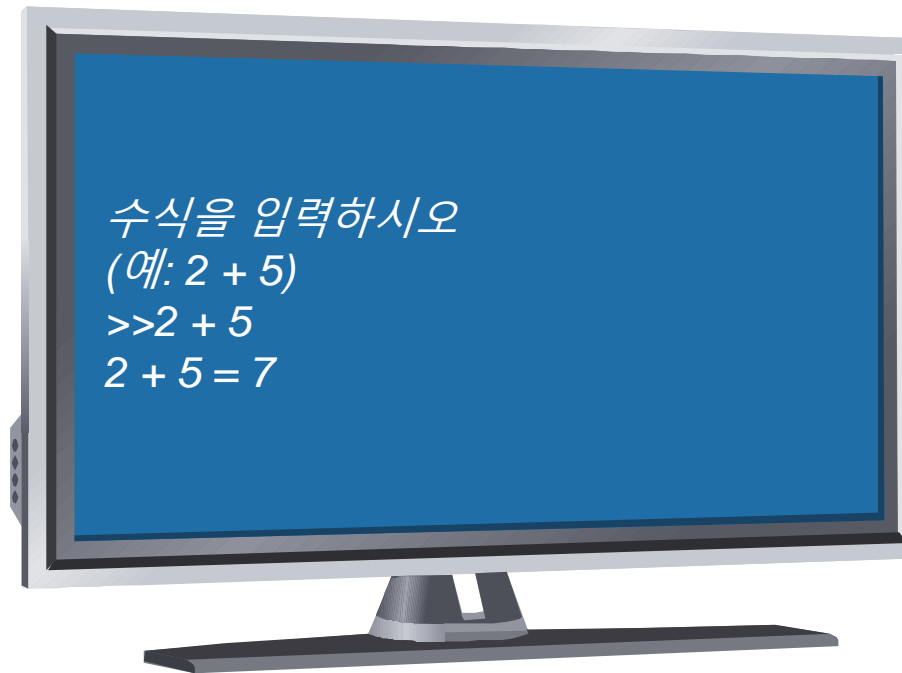
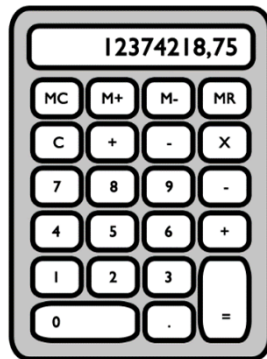


천인국 지음

제6장 조건문



실습: 산술 계산기





실습: 산술 계산기

```
#include <stdio.h>

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

    printf("수식을 입력하십시오");
    printf("(예: 2 + 5) ");
    printf(">>");

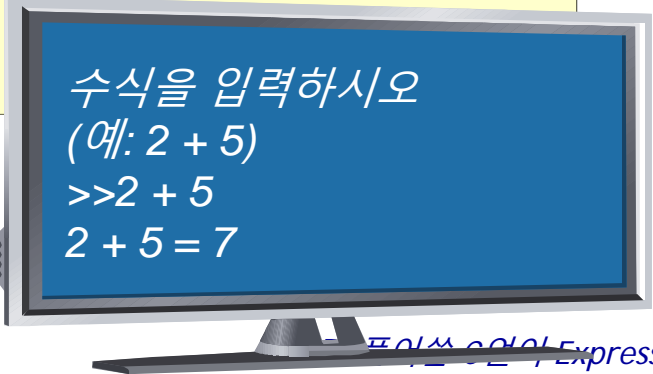
    scanf("%d %c %d", &x, &op, &y);
```



실습: 산술 계산기

```
if( op == '+' )
    result = x + y;
else if( op == '-' )
    result = x - y;
else if( op == '*' )
    result = x * y;
else if( op == '/' )
    result = x / y;
else if( op == '%' )
    result = x % y;
else
    printf("지원되지 않는 연산자입니다. ");

printf("%d %c %d = %d ", x, op, y, result);
return 0;
}
```



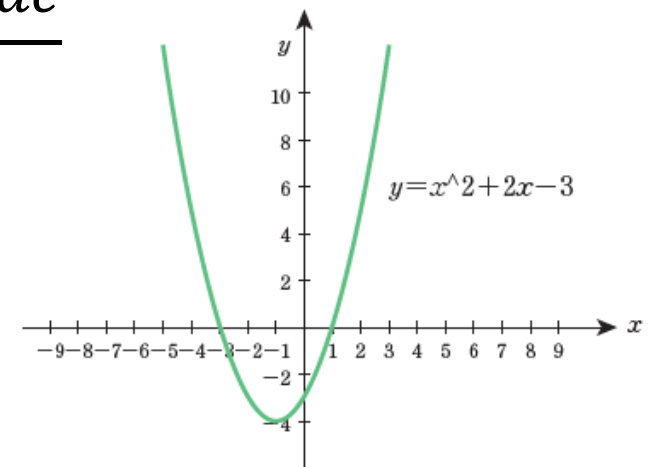
수식을 입력하십시오
(예: 2 + 5)
>>2 + 5
2 + 5 = 7



lab: 이차 방정식

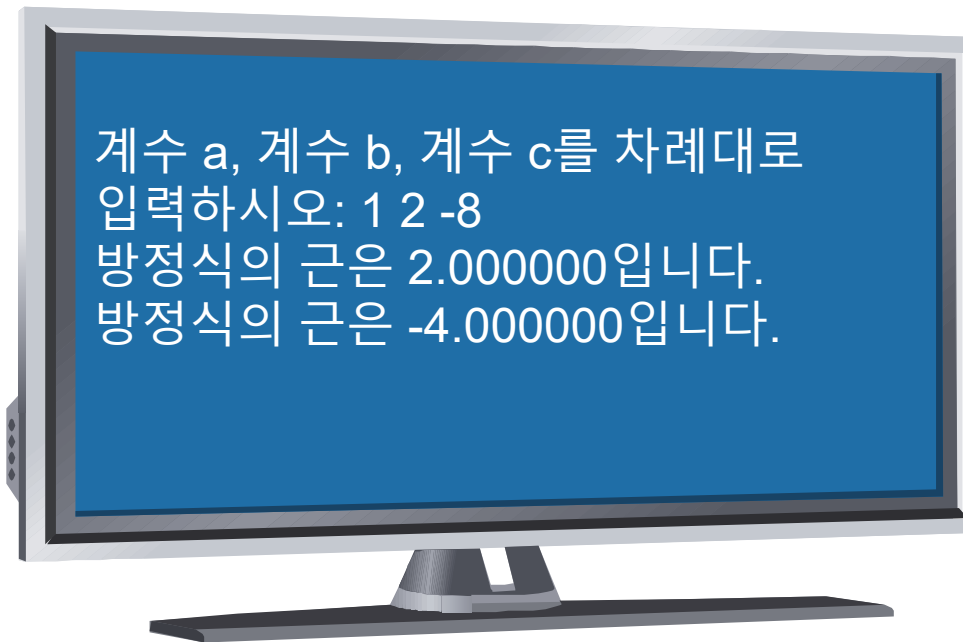
1. 사용자에게 이차 방정식의 계수 a , b , c 를 입력하도록 한다.
2. 만약 a 가 0이면 근은 $-c/b$ 이다.
3. 판별식 ($b^2 - 4ac$)가 음수이면 실근은 존재하지 않는다.
4. 위의 조건에 해당되지 않으면 다음과 같은 공식을 이용하여 실근을 구한다.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$





실행 결과





알고리즘

사용자로부터 a , b , c 를 읽는다.

if $a == 0$ 이면

일차 방정식의 근을 구한다.

실근을 출력한다.

else

판별식을 계산한다.

if 판별식 ≥ 0

근의 공식을 이용하여 실근을 구한다.

실근을 출력한다.

else

실근은 없다는 메시지 출력



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

int main(void)
{
    double a, b, c, dis;

    printf("계수 a를 입력하시오: ");
    scanf("%lf", &a);

    printf("계수 b를 입력하시오: ");
    scanf("%lf", &b);

    printf("계수 c를 입력하시오: ");
    scanf("%lf", &c);
```



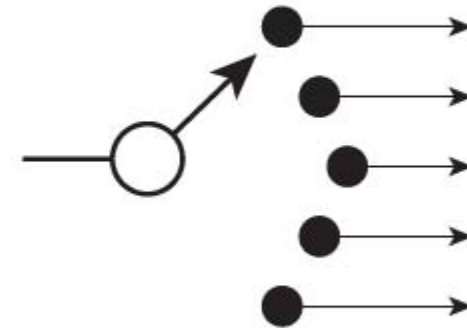
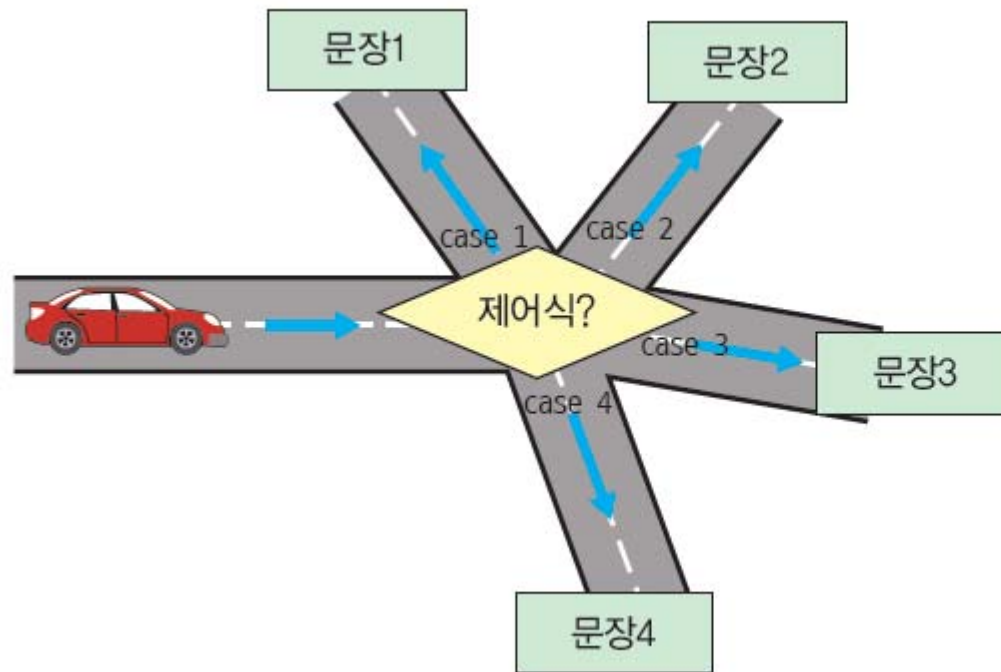

```
if( a == 0 )  
    printf("방정식의 근은 %f입니다.", -c/b);  
  
else  
{  
    dis = (b*b - 4.0*a*c);  
    if( dis >= 0 )  
    {  
        printf("방정식의 근은 %f입니다. \n", (-b+sqrt(dis))/(2.0*a));  
        printf("방정식의 근은 %f입니다. \n", (-b-sqrt(dis))/(2.0*a));  
    }  
    else  
        printf("실근이 존재하지 않습니다");  
}  
  
return 0;  
}
```

```
계수 a를 입력하시오: 1  
계수 b를 입력하시오: -4  
계수 c를 입력하시오: 3  
방정식의 근은 3.000000입니다.  
방정식의 근은 1.000000입니다.
```



switch 문

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





switch 문

Syntax: switch 문

문법

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

제어식의 값이 c1이면 실행된다.

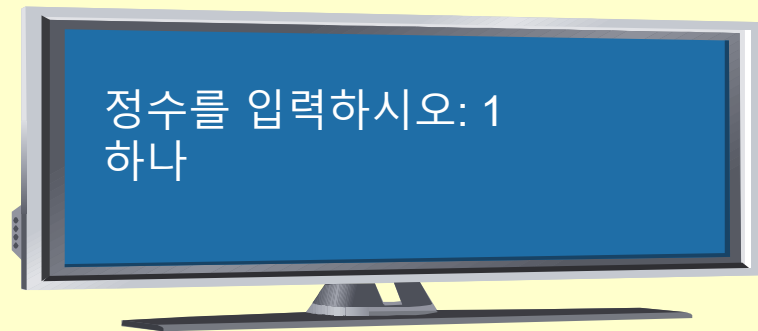
제어식의 값이 c2이면 실행된다.

일치하는 값이 없으면 실행된다.



예제

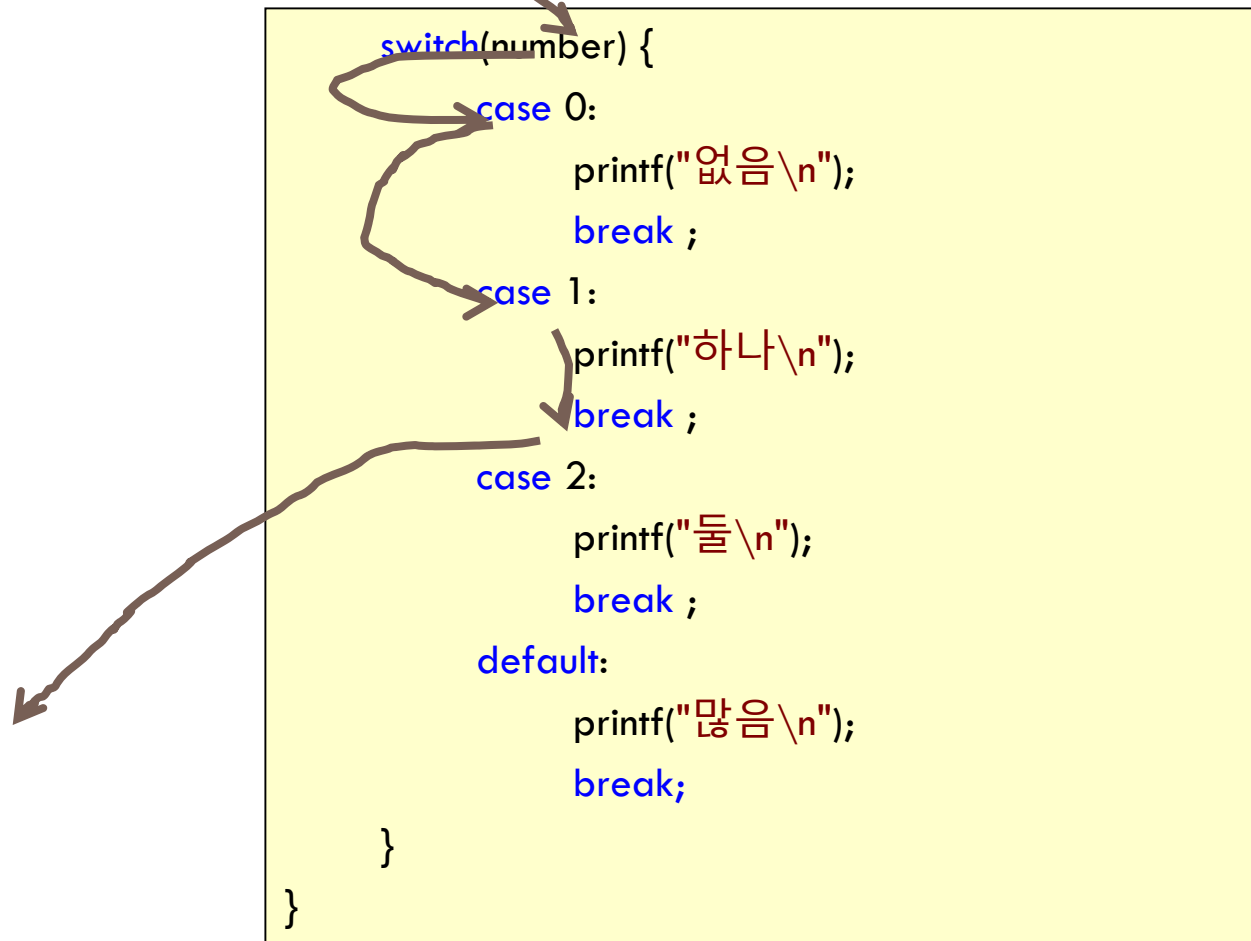
```
int main(void)
{
    int number;
    printf("정수를 입력하시오:");
    scanf("%d", &number);
    switch(number) {
        case 0:
            printf("없음\n");
            break ;
        case 1:
            printf("하나\n");
            break ;
        case 2:
            printf("둘\n");
            break ;
        default:
            printf("많음\n");
            break;
    }
}
```





사용자가 1을 입력하는 경우

1





break가 생략되는 경우

1

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

break를
만날 때까지
계속 문장을
실행합니다.





의도적인 break 생략

```
switch(number)
{
    case 0:
        printf("없음\n");
        break;
    case 1:
        printf("하나\n");
        break;
    case 2:
    case 3:
        printf("두서너개\n");
        break;
    default:
        printf("많음\n");
        break;
}
```

2개의 경우를
하나로 묶어서
처리하기
위하여 이러한
기법을 사용





default 문

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;  
}
```

어떤
case문과도
일치되지
않는 경우에
선택

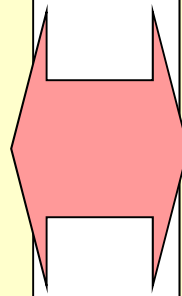




switch 문과 if-else 문

```
int main(void)
{
    int number;
    scanf("%d", &number);

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



```
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;
}
```



switch 문에서 주의할 점

```
switch(number)
{
    case x:                                // 변수는 사용할 수 없다.
        printf("x와 일치합니다. ");
        break;

    case (x+2):                            // 변수가 들어간 수식은 사용할 수 없다.
        printf("수식과 일치합니다. ");
        break;

    case 0.001:                            // 실수는 사용할 수 없다.
        printf("실수");
        break;

    case "001":                            // 문자열은 사용할 수 없다.
        printf("문자열");
        break;
}
```



예제

```
// 달의 일수를 계산하는 프로그램
#include <stdio.h>

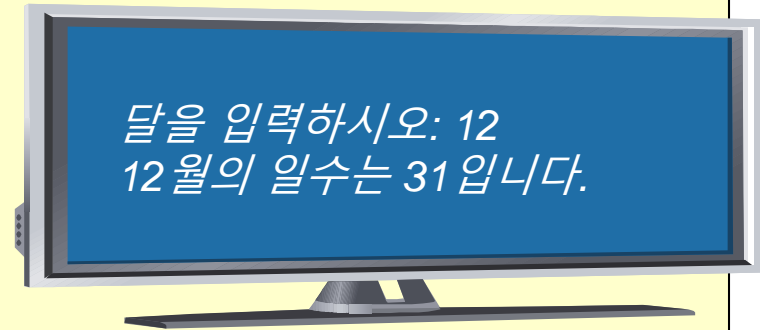
int main(void)
{
    int month, days;

    printf("달을 입력하시오: ");
    scanf("%d", &month);
```



예제

```
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;
}
```





실습: 산술 계산기

```
#include <stdio.h>

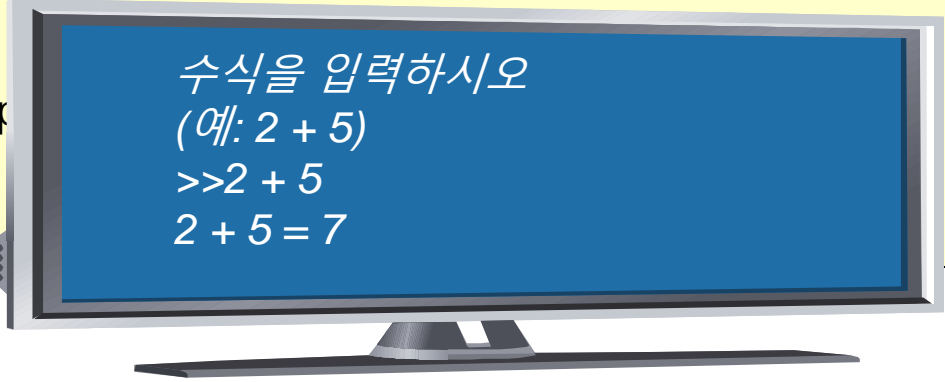
int main(void)
{
    char op;
    int x, y, result;

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



실습: 산술 계산기

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



수식을 입력하시오
(예: 2 + 5)
>>2 + 5
2 + 5 = 7



도전문제

- 위의 프로그램은 단순히 산술 연산자만을 처리한다. 비트 연산자(&, |, ^)을 추가하여 보자. 비트 연산자인 경우에는 16진수로 입력값과 결과값을 출력하여 보자.





중간 점검

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");  
        break;  
}
```





goto문

- 조건없이 어떤 위치로 점프
- 사용하지 않는 것이 좋음





goto 문

```
goto label;
```

```
...  
...  
...  
...
```

```
label:
```

```
문장 #1;  
문장 #2;  
문장 #3;
```

전향 참조

```
label:
```

```
문장 #1;  
문장 #2;  
문장 #3;  
...  
...  
...  
...
```

```
goto label;
```

후향 참조



예제

```
// 구구단출력프로그램
```

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

```
int main(void)
```

```
{
```

```
    int i = 1;
```

```
loop:
```

```
    printf("%d * %d = %d \n", 3, i, 3 * i);
```

```
    i++;
```

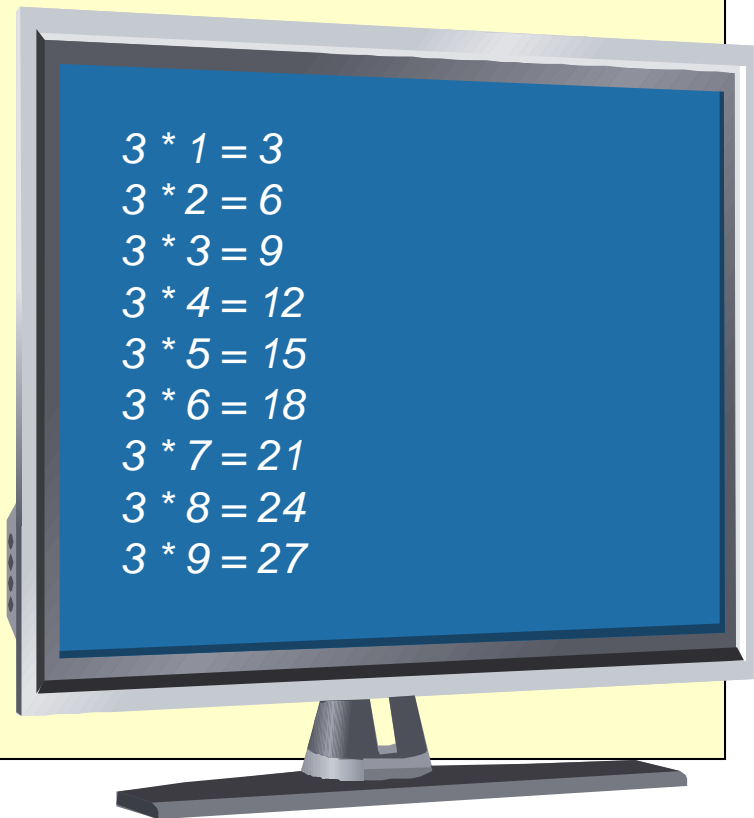
```
    if( i == 10.) goto end;
```

```
    goto loop;
```

```
end:
```

```
    return 0;
```

```
}
```

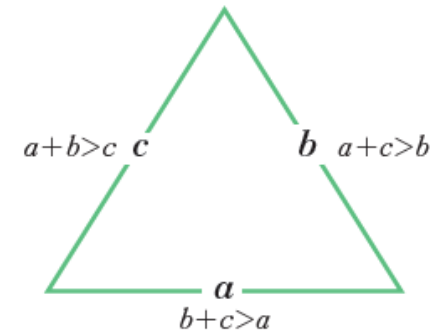


```
3 * 1 = 3  
3 * 2 = 6  
3 * 3 = 9  
3 * 4 = 12  
3 * 5 = 15  
3 * 6 = 18  
3 * 7 = 21  
3 * 8 = 24  
3 * 9 = 27
```



mini project: 올바른 삼각형 구별하기

삼각형의 3변을 입력하시오: 8 10 3
올바른 삼각형





예제

```
#include <stdio.h>

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

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

    if ((a + b) > c && (b + c) > a && (a + c) > b) {
        printf("올바른 삼각형\n");
    }
    else {
        printf("올바르지 않은 삼각형\n");
    }

    return 0;
}
```



Q & A

