

C언어 강의자료

문정욱



프로그래밍 연습 3

조건문, 반복문, 함수의 활용

If 문의 조건 변경

반대 조건의 사용

```
#include <stdio.h>

int main(void)
{
    int a;

    scanf("%d", &a);

    if( a == 1 ) {
        printf("a == one\n");
    }
    else {
        printf("a == other\n");
    }
    return 0;
}
```

```
#include <stdio.h>

int main(void)
{
    int a;

    scanf("%d", &a);

    if(  ) {
        printf("a == other\n");
    }
    else {
        printf("a == one\n");
    }
    return 0;
}
```

If 문의 조건 변경

복합 조건의 반대 조건

```
#include <stdio.h>

int main(void)
{
    int a, b;

    scanf("%d%d", &a, &b);

    if( a == 1 && b == 1 ) {
        printf("a and b == one\n");
    }
    else {
        printf("a or b != one\n");
    }
    return 0;
}
```

```
#include <stdio.h>

int main(void)
{
    int a, b;

    scanf("%d%d", &a, &b);

    if( a != 1 || b != 1 ) {
        printf("a or b != one\n");
    }
    else {
        printf("a and b == one\n");
    }
    return 0;
}
```

If 문의 조건 변경

중첩 if문을 단일 if문으로

```
#include <stdio.h>

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

    scanf("%d%d%d", &a, &b, &c);

    if( a == 1 ) {
        if( b == 1 ) {
            printf("a and b == one\n");
        }
        if( c == 1 ) {
            printf("a and c == one\n");
        }
    }
    return 0;
}
```

```
#include <stdio.h>

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

    scanf("%d%d%d", &a, &b, &c);

    if(  ) {
        printf("a and b == one\n");
    }
    if(  ) {
        printf("a and c == one\n");
    }

    return 0;
}
```

If 문의 조건 변경

else 대신 반대 조건 사용

```
#include <stdio.h>

int main(void)
{
    int a;

    scanf("%d", &a);

    if( a == 1 ) {
        printf("a == one\n");
    }
    else {
        printf("a == other\n");
    }
    return 0;
}
```

```
#include <stdio.h>

int main(void)
{
    int a;

    scanf("%d", &a);

    if( a == 1 ) {
        printf("a == one\n");
    }
    if(  ) {
        printf("a == other\n");
    }
    return 0;
}
```

중첩 if 문의 활용

■ 연습 (윤년 판별 2)

- 주어진 소스코드는 윤년(leap year) 판별 프로그램이다.
- 키보드로부터 년도를 의미하는 정수를 입력 받아 변수 y에 저장한다.
- 아래는 윤년 판별 조건이다.
 - ✓ y가 4의 배수이면 "leap year\n"를 출력하고, 그렇지 않으면 "not leap year\n"를 출력한다.
 - ✓ 위의 조건을 만족할 때, y가 100의 배수이면 "not leap year\n"를 출력하고, 그렇지 않으면 "leap year\n"를 출력한다.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

2014
not leap year.
계속하려면 아무 키나 누르십시오 . . .

입출력 결과

1988
leap year.
계속하려면 아무 키나 누르십시오 . . .

입출력 결과

1300
not leap year.
계속하려면 아무 키나 누르십시오 . . .

입출력 결과

1200
not leap year.
계속하려면 아무 키나 누르십시오 . . .

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

```
int main(void)
```

```
{
```

```
    int y;
```

```
    scanf("%d", &y);
```

```
    if( y%4==0 )
```

```
    else
```

```
        printf("not leap year.\n");
```

```
    return 0;
```

```
}
```

논리 오류: 1200년은 윤년이다.

중첩 if 문의 활용

■ 연습 (윤년 판별 3)


- 주어진 소스코드는 윤년(leap year) 판별 프로그램이다.
- 키보드로부터 년도를 의미하는 정수를 입력 받아 변수 y에 저장한다.
- 아래는 윤년 판별 조건이다.
 - ✓ y가 4의 배수이면 "leap year\n"를 출력하고, 그렇지 않으면 "not leap year\n"를 출력한다.
 - ✓ 위의 조건을 만족할 때, y가 100의 배수이면 "not leap year\n"를 출력하고, 그렇지 않으면 "leap year\n"를 출력한다.
 - ✓ 위의 조건을 만족할 때, y가 400의 배수이면 "leap year\n"를 출력하고, 그렇지 않으면 "not leap year\n"를 출력한다.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

```
1200
leap year.
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>

int main(void)
{
    int y;

    scanf("%d", &y);
    if( y%4==0 )
        if( y%100==0 )
            
        else
            printf("leap year.\n");
    else
        printf("not leap year.\n");
    return 0;
}
```


조건식의 변형

- 관계 연산자의 생략
 - 조건 수식에서 " $\neq 0$ "는 생략 가능하다.

cnd	if(cnd != 0)	if(cnd)
0	if(false)	if(false)
1	if(true)	if(true)

입출력 결과

```
0
false
false
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
1
true
true
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>

int main(void)
{
    int cnd;

    scanf("%d", &cnd);
    if( cnd !=0 ) // relational operator
        printf("true\n");
    else
        printf("false\n");

    if( cnd )
        printf("true\n");
    else
        printf("false\n");
    return 0;
}
```

조건식의 변형

■ 관계 연산자의 생략

- 조건 수식에서 "=="와 "!(...)" 가능하다.

cnd	if(cnd == 0)	if(!(cnd))
0	if(true)	if(true)
1	if(false)	if(false)

입출력 결과

```
0
false
false
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
1
true
true
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>

int main(void)
{
    int cnd;

    scanf("%d", &cnd);
    if( cnd == 0 ) // relational operator
        printf("false\n");
    else
        printf("true\n");

    if( !(cnd) ) // logical operator
        printf("false\n");
    else
        printf("true\n");
    return 0;
}
```

if 문의 활용

짝수/홀수 판별

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

```
int main(void)
```

```
{
```

```
    int a;
```

```
    scanf("%d", &a);
```

```
    if( /* the condition of odd */ )
```

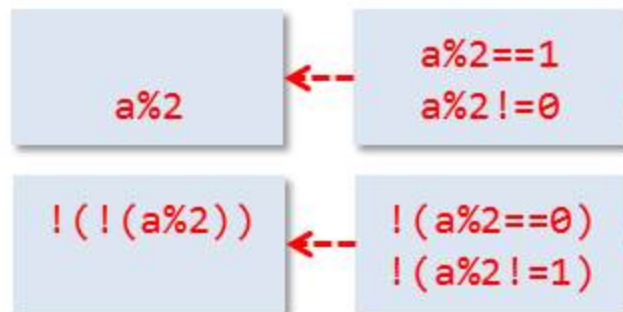
```
        printf("a == odd\n");
```

```
    else
```

```
        printf("a == even\n");
```

```
    return 0;
```

```
}
```



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

```
int main(void)
```

```
{
```

```
    int a;
```

```
    scanf("%d", &a);
```

```
    if( /* the condition of odd */ )
```

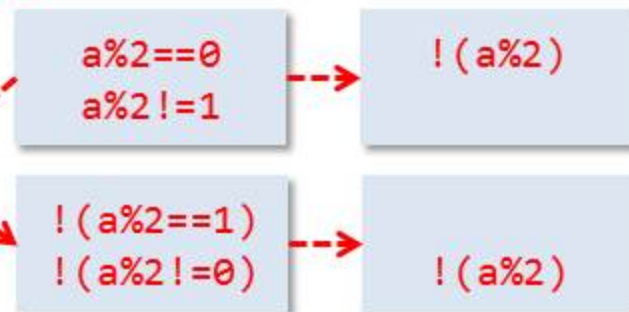
```
        printf("a == odd\n");
```

```
    if( /* the condition of even */ )
```

```
        printf("a == even\n");
```

```
    return 0;
```

```
}
```



반복문의 활용

■ 연습 (문자의 반복 출력 1)

- 주어진 소스코드는 반복문을 사용하여 문자를 반복 출력하는 프로그램이다.
- "*"를 10개 출력하라.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

계속하려면 아무 키나 누르십시오 . . .

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

```
int main(void)
```

```
{
```

```
    int i;
```

```
    for(  )
```

```
        printf("*");
```

```
    printf("\n");
```

```
    return 0;
```

```
}
```

반복문의 활용

산술 연산을 사용한 증감

```
#include <stdio.h>

int main(void)
{
    int i;

    for(i=0;i<10;i=i+1)
        printf("*");
    printf("\n");
    return 0;
}
```

대입 연산을 사용한 증감

```
#include <stdio.h>

int main(void)
{
    int i;

    for(i=0;i<10;i+=1)
        printf("*");
    printf("\n");
    return 0;
}
```


반복문의 활용

전치 증감 연산을 사용한 증감

```
#include <stdio.h>

int main(void)
{
    int i;

    for(i=0;i<10; ++i)
        printf("*");
    printf("\n");
    return 0;
}
```



후치 증감 연산을 사용한 증감

```
#include <stdio.h>

int main(void)
{
    int i;

    for(i=0;i<10;i++)
        printf("*");
    printf("\n");
    return 0;
}
```


반복문의 활용

■ 연습 (문자의 반복 출력 2)

- 주어진 소스코드는 반복문을 사용하여 문자를 반복 출력하는 프로그램이다.
- 변수 n 에 저장되어 있는 수만큼 "*"를 출력하라.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

계속하려면 아무 키나 누르십시오 . . .

```
#include <stdio.h>

int main(void)
{
    int n,i;

    n=10;
    for( ) printf("*");
    printf("\n");
    return 0;
}
```


반복문의 활용

■ 연습 (도형 만들기 1)

- 주어진 소스코드는 반복문을 사용하여 문자를 반복 출력하는 프로그램이다.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

```
*  
**  
***  
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>  
  
int main(void)  
{  
    int n,i;  
  
      
    for(i=0;i<n;++i) printf("*");  
    printf("\n");  
  
      
    for(i=0;i<n;++i) printf("*");  
    printf("\n");  
  
      
    for(i=0;i<n;++i) printf("*");  
    printf("\n");  
  
    return 0;  
}
```

반복문의 활용

```
#include <stdio.h>

int main(void)
{
    int n,i;

    
    for(i=0;i<n;++i) printf("*");
    printf("\n");

    
    for(i=0;i<n;++i) printf("*");
    printf("\n");

    
    for(i=0;i<n;++i) printf("*");
    printf("\n");

    return 0;
}
```

```
#include <stdio.h>

int main(void)
{
    int n,i;

    for() {
        for(i=0;i<n;++i) printf("*");
        printf("\n");
    }

    return 0;
}
```

반복문의 활용

```
#include <stdio.h>

int main(void)
{
    int n,i;

    for(  ) {
        for(i=0;i<n;++i) printf("*");
        printf("\n");
    }

    return 0;
}
```

```
#include <stdio.h>

int main(void)
{
    int n,j,i;

    for(j=0;j<3;++j) { // iteration = 3
        
        for(i=0;i<n;++i) printf("*");
        printf("\n");
    }

    return 0;
}
```

반복문의 활용

■ 연습 (도형 만들기 2)

- 주어진 소스코드는 반복문을 사용하여 문자를 반복 출력하는 프로그램이다.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

```
***
**
*
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>

int main(void)
{
    int n,j,i;

    for(j=0;j<3;++j) {
            
        for(i=0;i<n;++i) printf("*");
        printf("\n");
    }

    return 0;
}
```

반복문의 활용

■ 연습 (문자의 반복 출력 3)

- 주어진 소스코드는 반복문을 사용하여 변수 n1과 n2에 저장되어 있는 수만큼 각각 "."와 "*"를 출력하는 프로그램이다.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

...*****

계속하려면 아무 키나 누르십시오 . . .

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

```
int main(void)
```

```
{
```

```
    int n1,n2,i;
```



```
    for(i=0;i<n1;++i) printf(".");
```

```
    for(i=0;i<n2;++i) printf("*");
```

```
    printf("\n");
```

```
    return 0;
```

```
}
```

반복문의 활용

■ 연습 (도형 만들기 3)

- 주어진 소스코드는 반복문을 사용하여 문자를 반복 출력하는 프로그램이다.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

```
..*
.**
***
```

계속하려면 아무 키나 누르십시오 . . .

n1	n2

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

```
int main(void)
```

```
{
```

```
    int n1,n2,i;
```

```
    
    
```

```
    for(i=0;i<n1;++i) printf(".");
```

```
    for(i=0;i<n2;++i) printf("*");
```

```
    printf("\n");
```

```
    
    
```

```
    for(i=0;i<n1;++i) printf(".");
```

```
    for(i=0;i<n2;++i) printf("*");
```

```
    printf("\n");
```

```
    
    
```

```
    for(i=0;i<n1;++i) printf(".");
```

```
    for(i=0;i<n2;++i) printf("*");
```

```
    printf("\n");
```

```
    return 0;
```

```
}
```

반복문의 활용

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

```
int main(void)
```

```
{
```

```
    int n1,n2,i;
```



```
    for(i=0;i<n1;++i) printf(".");
    for(i=0;i<n2;++i) printf("*");
    printf("\n");
```



```
    for(i=0;i<n1;++i) printf(".");
    for(i=0;i<n2;++i) printf("*");
    printf("\n");
```



```
    for(i=0;i<n1;++i) printf(".");
    for(i=0;i<n2;++i) printf("*");
    printf("\n");
```

```
    return 0;
```

```
}
```

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

```
int main(void)
```

```
{
```

```
    int n1,n2,j,i;
```

```
    for(j=0;j<3;++j) { // iteration = 3
```



```
        for(i=0;i<n1;++i) printf(".");
        for(i=0;i<n2;++i) printf("*");
        printf("\n");
```

```
    }
```

```
    return 0;
```

```
}
```

j	n1	n2

반복문의 활용

■ 연습 (도형 만들기 4)

- 주어진 소스코드는 반복문을 사용하여 문자를 반복 출력하는 프로그램이다.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

```
***
.**
..*
계속하려면 아무 키나 누르십시오 . . .
```

j	n1	n2

```
#include <stdio.h>

int main(void)
{
    int n1,n2,j,i;

    for(j=0;j<3;++j) {
        
        for(i=0;i<n1;++i) printf(".");
        for(i=0;i<n2;++i) printf("*");
        printf("\n");
    }
    return 0;
}
```

함수의 활용

■ 연습 (함수 만들기 1)

- 주어진 소스코드는 입력 받은 정수의 합을 구하는 프로그램이다.
- 키보드로 부터 정수 3개를 입력받아 변수 a, b, c에 저장한다.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

```
2 7 9
sum == 18
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
9 11 99
sum == 119
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
-2 -10 -30
sum == -42
계속하려면 아무 키나 누르십시오 . . .
```

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

```
 sum(    
{  
    return   
}
```

```
int main(void)
```

```
{  
    int a, b, c;  
  
    scanf("%d%d%d", &a, &b, &c);  
    printf("sum == %d\n", sum(a,b,c) );  
    return 0;  
}
```

함수의 활용

■ 연습 (함수 만들기 2)

- 주어진 소스코드는 2차 방정식의 근을 판별 (discriminant)하는 프로그램이다.
- 변수 a, b, c는 이차방정식 $ax^2+bx+c=0$ 의 x항의 개수를 의미한다.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

```
5 7.7 -6.5
two real roots.
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
2.0 4.0 2.0
one real root.
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
2. 3.5 9.
imaginary roots.
계속하려면 아무 키나 누르십시오 . . .
```

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

```

disc(
{
    return
}

```

```
int main(void)
```

```

{
    double a, b, c, d;

    scanf("%lf%lf%lf", &a, &b, &c);
    d = disc(a, b, c);
    if( d>0.0 )
        printf("two real roots.\n");
    if( d==0.0 )
        printf("one real root.\n");
    if( d<0.0 )
        printf("imaginary roots.\n");
    return 0;
}

```

if-else와 개별 if의 차이

else 대신 반대 조건 사용

```
#include <stdio.h>

int main(void)
{
    int a, b;

    scanf("%d%d", &a, &b);

    if( a == 1 ) {
        printf("a == one\n");
    }
    else if( b == 1 ) {
        printf("a != one, b == one\n");
    }
    return 0;
}
```

```
#include <stdio.h>

int main(void)
{
    int a, b;

    scanf("%d%d", &a, &b);

    if(  ) {
        printf("a == one\n");
    }
    if(  ) {
        printf("a != one, b == one\n");
    }
    return 0;
}
```

if-else와 개별 if의 차이

■ 연습 (배수 판별 1)

- 주어진 소스코드는 배수 판별 프로그램이다.
- 키보드로부터 정수를 입력 받아 변수 a에 저장한다.
- 아래는 배수 판별 조건이다.
 - ✓ a가 2와 3의 배수이면 "a == multiple of 2 and 3\n"를 출력한다.
 - ✓ a가 2의 배수이면 "a == multiple of 2\n"를 출력한다.
 - ✓ 단, 위의 메시지는 위의 순서대로 동시에 출력될 수 있다. (동시 실행)
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

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

```
int main(void)
```

```
{
```

```
    int a;
```

```
    scanf("%d", &a);
```

```
        printf("a == multiple of 2 and 3\n");
```

```
        printf("a == multiple of 2\n");
```

```
    return 0;
```

```
}
```

입출력 결과

```
6
a == multiple of 2 and 3
a == multiple of 2
계속하려면 아무 키나 누르십시오 . . .
```

동시 실행
(비배타적 조건)

입출력 결과

```
4
a == multiple of 2
계속하려면 아무 키나 누르십시오 . . .
```


if-else와 개별 if의 차이

■ 연습 (배수 판별 2)

- 주어진 소스코드는 배수 판별 프로그램이다.
- 키보드로부터 정수를 입력 받아 변수 a에 저장한다.
- 아래는 배수 판별 주거이다.
 - ✓ a가 2와 3의 배수이면 "a == multiple of 2 and 3\n"를 출력한다.
 - ✓ a가 2의 배수이면 "a == multiple of 2\n"를 출력한다.
 - ✓ 단, 위의 메시지는 동시에 출력될 수 없으며 두 조건을 모두 만족할 경우 첫 번째 메시지만 출력한다. (동시 실행 불가)
- 주어진 입출력 결과들 참고하여 소스코드를 완성하라.

입출력 결과

```
6
a == multiple of 2 and 3
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
4
a == multiple of 2
계속하려면 아무 키나 누르십시오 . . .
```

동시 실행 안됨
(배타적 조건)

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

```
int main(void)
```

```
{
```

```
    int a;
```

```
    scanf("%d", &a);
```

```
        printf("a == multiple of 2 and 3\n");
```

```
        printf("a == multiple of 2\n");
```

```
    return 0;
```

```
}
```

- ✓ a가 2와 3의 배수이면 "a == multiple of 2 and 3\n"를 출력한다.
- ✓ 만일 위 조건을 만족하지 않고 a가 2의 배수이면 "a == multiple of 2\n"를 출력한다. (배타적 조건)

※ 배타적 조건: 여러 개의 조건들이 동시에 참이 될 수 없을 때 각 조건은 다른 조건에 대해 배타적 조건이라 한다.

if-else와 개별 if의 차이

If 문

```
#include <stdio.h>

int main(void)
{
    int a;

    scanf("%d", &a);

    if( a%2==0 && a%3==0 )
        printf("a == multiple of 2 and 3\n");
    if( a%2==0 )
        printf("a == multiple of 2\n");
    return 0;
}
```

동시 실행
(비배타적 조건)

입출력 결과

```
6
a == multiple of 2 and 3
a == multiple of 2
계속하려면 아무 키나 누르십시오 . . .
```

If-else 문

```
#include <stdio.h>

int main(void)
{
    int a;

    scanf("%d", &a);

    if( a%2==0 && a%3==0 )
        printf("a == multiple of 2 and 3\n");
    else if( a%2==0 )
        printf("a == multiple of 2\n");
    return 0;
}
```

동시 실행 안됨
(배타적 조건)

입출력 결과

```
6
a == multiple of 2 and 3
계속하려면 아무 키나 누르십시오 . . .
```


if-else와 개별 if의 차이

비배타적 조건 1

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

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

```
    scanf("%d", &a);
```

```
    if( a%2==0 && a%3==0 )
        printf("a == multiple of 2 and 3\n");
```

```
    if( a%2==0 )
        printf("a == multiple of 2\n");
    return 0;
```

```
}
```

입출력 결과

```
6
a == multiple of 2 and 3
a == multiple of 2
계속하려면 아무 키나 누르십시오 . . .
```

비배타적 조건일 경우
if문의 순서가 바뀌면
결과에 영향을 준다.

비배타적 조건 2

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

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

```
    scanf("%d", &a);
```

```
    if( a%2==0 )
        printf("a == multiple of 2\n");
```

```
    if( a%2==0 && a%3==0 )
        printf("a == multiple of 2 and 3\n");
    return 0;
```

```
}
```

입출력 결과

```
6
a == multiple of 2
a == multiple of 2 and 3
계속하려면 아무 키나 누르십시오 . . .
```

if-else와 개별 if의 차이

if-else의 사용

```
#include <stdio.h>

int main(void)
{
    int a;

    scanf("%d", &a);

    if( a%2==0 && a%3==0 )
        printf("a == multiple of 2 and 3\n");
    else if( a%2==0 )
        printf("a == multiple of 2\n");
    return 0;
}
```

개별 if의 사용

```
#include <stdio.h>

int main(void)
{
    int a;

    scanf("%d", &a);

    if(  )
        printf("a == multiple of 2 and 3\n");

    if(  )
        printf("a == multiple of 2\n");

    return 0;
}
```

배타적 조건

배타적 실행

if-else와 개별 if의 차이

배타적 조건 1

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

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

```
    int a;
```

```
    scanf("%d", &a);
```

```
    if( a%2==0 && a%3==0 )
        printf("a == multiple of 2 and 3\n");
```

```
    if( a%2==0 && a%3!=0 )
        printf("a == multiple of 2\n");
```

```
    return 0;
```

```
}
```

배타적 조건일 경우
if문의 순서가 바뀌어도
결과에 영향을 주지 않는다.

입출력 결과

```
6
a == multiple of 2 and 3
계속하려면 아무 키나 누르십시오 . . .
```

배타적 조건 2

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

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

```
    int a;
```

```
    scanf("%d", &a);
```

```
    if( a%2==0 && a%3!=0 )
        printf("a == multiple of 2\n");
```

```
    if( a%2==0 && a%3==0 )
        printf("a == multiple of 2 and 3\n");
```

```
    return 0;
```

```
}
```

입출력 결과

```
4
a == multiple of 2
계속하려면 아무 키나 누르십시오 . . .
```

if-else와 개별 if의 차이

- 배타적 조건, 배타적 실행
 - if-else는 배타적 조건을 내포하고 있다.
 - 배타적 조건은 배타적 실행을 유발한다.
(즉, 둘 중에 하나만 실행됨)

```
#include <stdio.h>

int main(void)
{
    int a;

    scanf("%d", &a);

    if( a%2==0 && a%3==0 )
        printf("a = multiple of 2 and 3\n");
    else if( a%2==0 )
        printf("a = multiple of 2\n");
    return 0;
}
```

배타적 조건
내포

배타적 실행

중첩 if 문이나 연속 if 문일 경우

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

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

    scanf("%d", &a);

    if( a%2==0 && a%3==0 )
        printf("a == multiple of 2 and 3\n");
    else if( a%2==0 )
        printf("a == multiple of 2\n");
    return 0;
}
```

배타적 조건일 경우
개별 if문으로 전환한 후
if문을 바꿔야 실수를 줄인다.

입출력 결과

```
2
a == multiple of 2
계속
```

입출력 결과

3

계속

입출력 결과

```
6
a == multiple of 2 and 3
계속하려면 아무 키나 누르십시오 . . .
```

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

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

    scanf("%d", &a);

    if( a%2==0 )
        printf("a == multiple of 2\n");
    else if( a%2==0 && a%3==0 )
        printf("a == multiple of 2 and 3\n");
    return 0;
}
```

어떤 경우에도
수행되지 않음

입출력 결과

```
2
a == multiple of 2
계속
```

입출력 결과

3

계속

입출력 결과

```
6
a == multiple of 2
계속하려면 아무 키나 누르십시오 . . .
```


If-else 문에서의 조건 순서의 중요성

```
if( a%2==0 && a%3==0 )  
    printf("a = multiple of 2 and 3\n");  
else if( a%2==0 )  
    printf("a = multiple of 2\n");
```



```
if( a%2==0 && a%3==0 )  
    printf("a = multiple of 2 and 3\n");  
if( a%2==0 && a%3!=0 )  
    printf("a = multiple of 2\n");
```

두 소스코드에 있는 if문의 조건은 순서만 바뀌어 유사한 듯 보이나 실제로는 완전히 다르다.
이를 확인하기 위해서는 개별 if문으로 변환하는 것이 가장 좋다.

```
if( a%2==0 )  
    printf("a = multiple of 2\n");  
else if( a%2==0 && a%3==0 )  
    printf("a = multiple of 2 and 3\n");
```



```
if( a%2==0 )  
    printf("a = multiple of 2\n");  
if( FALSE )  
    printf("a = multiple of 2 and 3\n");
```

if 문의 활용

■ 연습

- 아래 결과를 출력하도록 소스 코드를 완성하라.

입출력 결과

```
12
a == multiple of 2 and 3
```

입출력 결과

```
8
a == multiple of 2
```

입출력 결과

```
21
a == multiple of 3
```

입출력 결과

```
5
a == other
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>

int main(void)
{
    int a;

    scanf("%d", &a);

    
    printf("a == multiple of 2 and 3\n");
    
    printf("a == multiple of 2\n");
    
    printf("a == multiple of 3\n");
    
    printf("a == other\n");

    return 0;
}
```


if 문의 활용

■ 연습

- 아래 결과를 출력하도록 소스 코드를 완성하라.

입출력 결과

```
12
a == multiple of 2 and 3
a == multiple of 2
a == multiple of 3
```

입출력 결과

```
8
a == multiple of 2
```

입출력 결과

```
21
a == multiple of 3
```

입출력 결과

```
5
a == other
계속하려면 아무 키나 누르십시오 . . .
```

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

```
int main(void)
```

```
{
```

```
    int a;
```

```
    scanf("%d", &a);
```

```
        printf("a == multiple of 2 and 3\n");
```

```
        printf("a == multiple of 2\n");
```

```
        printf("a == multiple of 3\n");
```

```
        printf("a == other\n");
```

```
    return 0;
```

```
}
```

if 문의 활용

■ 윤년(leap year) 판별

- 윤년 판별 조건

- ✓ y가 4의 배수이면 "leap year\n"를 출력하고, 그렇지 않으면 "not leap year\n"를 출력한다.
- ✓ 위의 조건을 만족할 때, y가 100의 배수이면 "not leap year\n"를 출력하고, 그렇지 않으면 "leap year\n"를 출력한다.
- ✓ 위의 조건을 만족할 때, y가 400의 배수이면 "leap year\n"를 출력하고, 그렇지 않으면 "not leap year\n"를 출력한다.

■ 판별 조건식

$(y \% 4 == 0 \text{ and } y \% 100 == 0 \text{ and } y \% 400 == 0) \text{ or } (y \% 4 == 0 \text{ and } y \% 100 != 0)$

```
#include <stdio.h>

int main(void)
{
    int y;

    scanf("%d",&y);

    if(y%4==0)
        if(y%100==0)
            if(y%400==0)
                printf("leap year\n");
            else
                printf("not leap year\n");
        else
            printf("leap year\n");
    else
        printf("not leap year\n");

    return 0;
}
```

if 문의 활용

윤달 판별 조건

```
(y%4==0 and y%100==0 and y%400==0) or (y%4==0 and y%100!=0)
```

```
y%4==0 and ( (y%100==0 and y%400==0) or y%100!=0 )
```

```
y%4==0 and ( (y%100==0 or y%100!=0) and (y%400==0 or y%100!=0))
```

```
y%4==0 and ( TRUE and (y%400==0 or y%100!=0))
```

```
y%4==0 and (y%400==0 or y%100!=0)
```

if 문의 활용

```
#include <stdio.h>

int main(void)
{
    int y;

    scanf("%d",&y);

    if(y%4==0)
        if(y%100==0)
            if(y%400==0)
                printf("leap year\n");
            else
                printf("not leap year\n");
        else
            printf("leap year\n");
    else
        printf("not leap year\n");

    return 0;
}
```

```
#include <stdio.h>

int main(void)
{
    int y;

    scanf("%d",&y);

    if(  )
        printf("leap year\n");
    else
        printf("not leap year\n");

    return 0;
}
```

for문의 분석

```
#include <stdio.h>

int main(void)
{
    int s,i;

    s=0;
    for(i=1; i<10; i=i+1) {
        s=s+2*i-1;
    }
    printf("%d\n",s);
    return 0;
}
```

for문 분석 테이블

variable (i)	condition (i<10)	statement (s = s + 2*i-1)

for문 분석 결과

Analysis	Value
The value of i in for-stat.	
The value of i after for-stat.	
The number of iteration	

for문의 분석

```
#include <stdio.h>

int main(void)
{
    int s,i;

    s=0;
    for(i=23; i>0; i=i-2) {
        s=s+i;
    }
    printf("%d\n",s);
    return 0;
}
```

for문 분석 테이블

variable (i)	condition (i>0)	statement (s = s + i)

for문 분석 결과

Analysis	Value
The value of i in for-stat.	
The value of i after for-stat.	
The number of iteration	

for문의 분석

■ 연습

- 다음 for문 분석 테이블과 동일한 동작을 하도록 프로그램을 완성하라.

variable (i)	condition (?)	statement (s = s + 2*i-1)
1	true	1 = 0 + 2*1-1
2	true	4 = 1 + 2*2-1
3	true	9 = 4 + 2*3-1
...
9	true	? = ? + 2*9-1
10	false	no execution

```
#include <stdio.h>

int main(void)
{
    int s,i;

    s=0;
    for(  ) {
        s=s+2*i-1;
    }
    printf("%d\n",s);
    return 0;
}
```

for문의 분석

■ 연습

- 다음 for문 분석 결과와 동일한 동작을 하도록 프로그램을 완성하라.

Analysis	Value
The value of i in for-stat.	23~1, -2
The value of i after for-stat.	-1
The number of iteration	12

```
#include <stdio.h>

int main(void)
{
    int s,i;

    s=0;
    for(  ) {
        s=s+i;
    }
    printf("%d\n",s);
    return 0;
}
```