

---

# 컴퓨터 프로그래밍 개론

## 반복문

---

# 학습목차

---

I. 반복문 개요

II. while문

III. for문

IV. do~while

# 반복문 개요

---

▶ 컴퓨터는 같은 일을 몇 번이고 반복(loop)하며 필요하다면 몇만 번, 몇 억 번이라도 반복 가능

▶ 소스 코드에서 같은 동작을 하는 코드를 연속적으로 여러 번 작성하는 것은 비효율적임

▶ 반복문

▶ 어떤 행동을 반복적으로 수행하는 문장

▶ 반복을 종료하는 조건식에 따라 형태가 나뉨

▶ 반복문의 종류

▶ while문

▶ for문

▶ do ~ while문

# 반복문 개요

## ▶ 반복문의 종류

- ▶ 반복에는 횟수가 정해진 반복과 반복 횟수가 정해지지 않은 반복이 존재

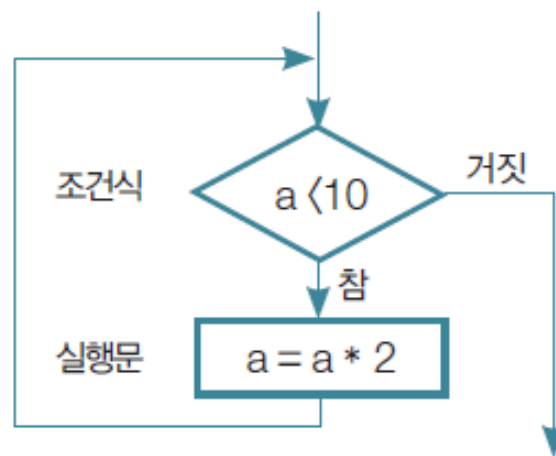
반복문 형식	실행 방식
<pre>while (조건식) {     실행문; }</pre>	조건식이 참인 동안 실행문을 반복한다. 최초 조건식이 거짓이면 실행문은 한 번도 실행되지 않는다.
<pre>for (초기식; 조건식; 증감식) {     실행문; }</pre>	초기식은 최초 한 번 실행한다. 조건식을 검사하여 참이면 실행문 → 증감식 → 조건식을 반복한다.
<pre>do {     실행문; } while (조건식);</pre>	실행문을 수행한 후에 조건을 검사한다. 조건식이 참인 동안 실행문을 반복한다. 실행문은 조건식과 관계없이 최소 한 번은 실행된다.

# while문

## ▶ while문

- ▶ 반복 횟수가 정해지지 않은 경우에 주로 사용
- ▶ while 문의 의미
  - ▷ '~하는 동안'
- ▶ 조건식을 먼저 검사
  - ▷ '조건식'이 '참'인 동안 실행문을 반복하여 처리
- ▶ 아래 예제는 변수 a가 10보다 작은 동안 두 배씩 반복적으로 늘리는 예제

```
while (a < 10) ← 조건식  
{  
    a = a * 2; ← 실행문  
}
```



# while문

## ▶ if문과 while문의 활용

- ▶ while문도 실행문이 하나일 경우 괄호 생략 가능

```
if (a != 0)
```

```
while (a < 10)
```

```
{
```

```
    a = a * 2;
```

```
    printf("a의 값 : %d\n", a);
```

```
}
```

} if문의 실행문이  
반복문 하나이므로  
if문의 중괄호는  
생략할 수 있다.

반복문 자체의 실행문은 두 문장이므로  
이 중괄호는 생략할 수 없다.

# while문

## ▶ while문의 활용

- ▶ while문은 뒤에 학습할 for문과 다르게 반복 횟수가 정해지지 않은 경우에 주로 사용
- ▶ 또한 반대로 반복 횟수를 모르는 경우, 이를 알아내기 위해서는 조건식만 가진 while문이 편리
- ▶ while문을 활용한 기하급수 계산

어떤 초등학생이 부모님께 부탁을 했다.

“이번 달은 1원, 다음 달은 2원, 다다음 달은 4원으로 지난달의 두배 씩 용돈을 주세요.”

부모님은 승락했다.

그 초등학생의 용돈이 100만원이 넘는 것은 몇 달째 일까?

# 실습문제 1

---

▶ while문을 이용하여 1부터 100까지의 합을 출력하는 프로그램을 작성하시오.

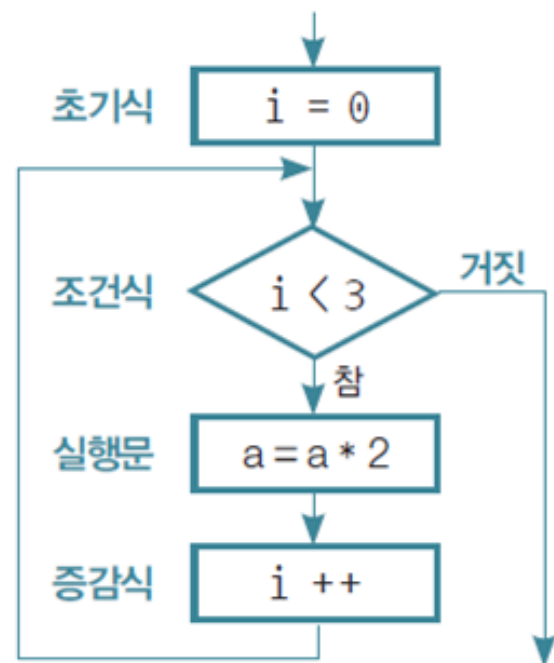
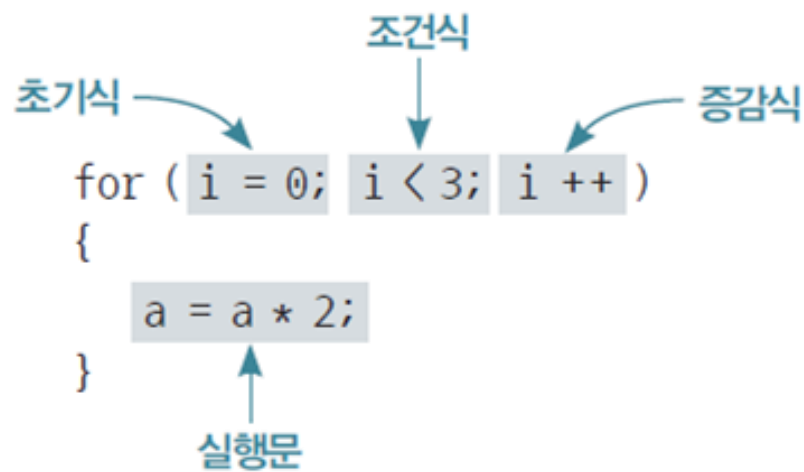
▶ 출력결과 : 5050



# for 문

## ▶ for문

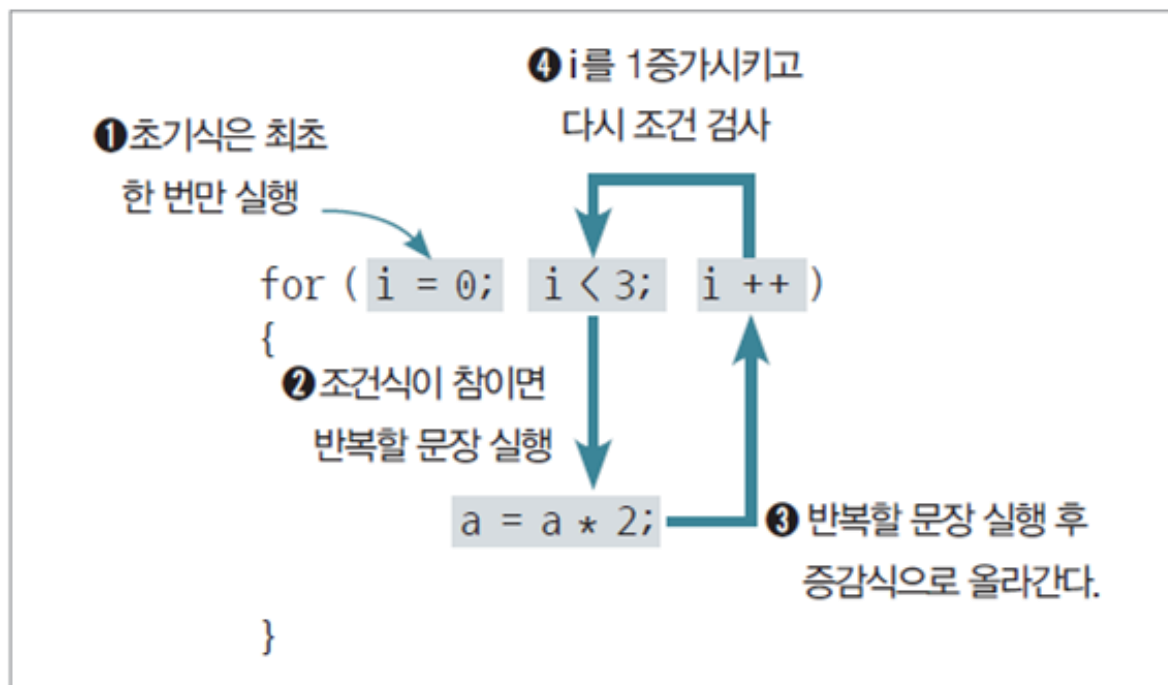
- ▶ while문을 확장한 구문
- ▶ 반복 횟수가 정해진 경우에 주로 사용
  - ▷ 실행문을 원하는 횟수만큼 반복할 때 사용
- ▶ 초기식, 조건식, 증감식으로 반복 횟수 제어하며 블록 안의 문장 반복
  - ▷ 초기식에 사용하는 변수를 카운트 변수라고 하고 주로 i, j, k를 사용
- ▶ 아래 예제는 a값 두 배 증가시키는 문장을 3번 반복하는 예제



# for 문

## ▶ for문의 실행 순서

- ▶ 아래 예제 코드에서 사용한 카운트 변수인 `i`는 for문을 사용하기 전에 선언해야 함 → `int i;`
- ▶ 초기식은 반복 횟수에 관계 없이 단 한 번만 실행됨
- ▶ 조건식은 루프의 종료 조건을 설정
- ▶ 증감식은 카운트 변수의 증감을 실행



# for 문

## ▶ for문 사용시 주의점

- ▶ 초기식, 조건식, 증감식은 반복 횟수 쉽게 알 수 있도록 작성

```
for (i = 0; i < 3; i++)           // i의 값이 0, 1, 2일 때 3번 반복
for (i = 2; i <= 4; i++)          // i의 값이 2, 3, 4일 때 3번 반복
for (i = 0; i < 7; i += 3)        // i의 값이 0, 3, 6일 때 3번 반복
for (i = 3; i > 0; i--)           // i의 값이 3, 2, 1일 때 3번 반복
```

- ▶ 반복 횟수 세는 변수를 반복문 안에서 바꾸지 말 것

- ▶ 10번 실행되어야 할 printf 문이 5번 실행

```
for(i = 1; i <= 10; i++){
    printf("%02d\n",i);
    i += 1; //i의 값을 2씩 증가
}
```

# for 문

## ▶ for문과 while 문 비교

- ▶ while문의 초기식은 while 문의 위에 위치
- ▶ while문의 증감식은 while문 내부에 위치
- ▶ 반복 횟수가 정해진 경우라면 for 문의 사용이 더 효율적임

```
i = 0;
while ( i < 10 )
{
    printf("Be happy!\n");
    i++;
}
```

```
for ( i = 0; i < 10; i++ )
{
    printf("Be happy!\n");
}
```

## 실습문제 2

---

▶ [실습문제 1]의 while문을 for문으로 변경하여 작성하시오.

▶ 출력결과 : 5050

# for 문

## ▶ for문의 여러 가지 변형(1)

### ▶ 초깃값이 없는 경우

```
#include <stdio.h>
int main( )
{
    int i = 1, factorial = 1;
    for( ; i<=10; i++)           // 초깃값이 없는 경우
    {
        factorial = factorial * i;
    }
    printf("1부터 10까지의 곱 : %d\n", factorial);

    return 0;
}
```

# for 문

## ▶ for문의 여러 가지 변형(2)

### ▶ 조건이 없는 경우

```
#include <stdio.h>
int main( )
{
    int i, factorial = 1;
    for(i = 1; ; i++)           // 조건값이 없는 경우
    {
        if (i > 10)           // 무한 반복문을 탈출하기 위한 break 문
            break;

        factorial = factorial * i;
    }
    printf("1부터 10까지의 곱 : %d\n", factorial);
    return 0;
}
```

# for 문

## ▶ for문의 여러 가지 변형(3)

### ▶ 증감값이 없는 경우

```
#include <stdio.h>
int main( )
{
    int i, factorial=1;
    for(i = 1; i <= 10; )    // 증감값이 없는 경우
    {
        factorial = factorial * i;
        i++;                // 증감값을 설정
    }
    printf("1부터 10까지의 곱 : %d\n", factorial);
    return 0;
}
```



# for 문

## ▶ for문의 여러 가지 변형(4)

### ▶ 무한 루프

▷ 아래 코드를 실행시키면 콘솔 창을 강제로 종료시켜야 함

▷ 윈도우의 X 버튼을 눌러 종료시킬 때까지 반복(또는 ctrl + c)

```
#include <stdio.h>
int main( )
{
    int i = 1;
    for( ; ; )
    {
        printf("%d\\n", i);
        i++;
    }
    return 0;
}
```

# for 문

---

## ▶ 무한 루프

- ▶ 널리 사용되는 기술
- ▶ 같은 동작을 반복하는 프로그램
- ▶ 넓은 의미로 보면 모든 애플리케이션들은 무한 루프로 구성

## 실습문제 3

---

▶ for문을 이용하여 1부터 100까지의 범위(100포함)에서 짝수의 합을 출력하는 프로그램을 작성하시오.

▶ 출력결과 : 2550

# do~while

## ▶ do ~ while문

### ▶ 반복할 문장을 일단 수행 후 조건 검사

▷ do~while문은 어떤 값이 입력되더라도 최소한 한 번은 do 안의 문장 실행

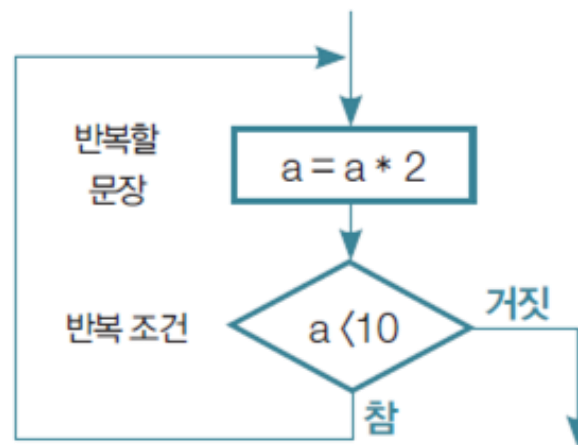
▷ do~while문의 마지막에 위치한 조건식 뒤에 세미콜론이 반드시 있어야 함

### ▶ 아래 예제는 a값이 10보다 작은 동안 두 배씩 늘리는 do ~ while문 예제

▷ 실행문(반복할 문장)이 하나인 경우 괄호 생략가능

▷ 세미콜론을 누락한 후 오류를 찾기 위해 고생하는 경우가 종종 발생

```
do  
{  
    a = a * 2; ← 반복할 문장  
} while (a < 10); ← 반복 조건
```



# do~while

## ▶ while 문과 do while 문 비교

### ▶ 변수 값을 입력 받는 경우

▷ while문은 입력되는 값 따라 반복 여부 결정

▷ do ~ while문은 어떠한 상황이더라도 최소한 한 번은 do 안의 문장을 실행

---

```
scanf("%d", &a);           // 어떤 값이 입력되더라도
do
{
    a = a * 2;              // 한 번은 실행된다.
} while ( a < 10 );
```

---

# do~while

## ▶ 입력 검사

- ▶ 반드시 한 번은 실행되는 do ~ while문의 특징은 입력 검사 시에 유용하게 작용
- ▶ 예를 들어 원의 면적을 구하는 경우 반지름의 값이 음수일 경우 문제가 발생
  - ▷ 물론 원의 면적을 구하는 공식이 반지름 X 반지름 X 원주율이므로 결과는 양수이겠지만 크기는 음수일 수 없으므로 오류라고 가정

```
int r = 0;
double s = 0.0;
printf("반지름 : ");
scanf("%d",&r);
while(r<0){
    printf("반지름 : ");
    scanf("%d",&r);
}
s = r * r * 3.14;
printf("원의 면적은 %1f입니다.", s);
```

```
int r = 0;
double s = 0.0;

do{
    printf("반지름 : ");
    scanf("%d",&r);
}while(r<0);

s = r * r * 3.14;
printf("원의면적은 %1f입니다.", s);
```

- ▶ while문은 scanf를 두 번 사용

- ▷ 동일한 데이터를 입력받기 위하여 scanf함수를 두 번 사용하는 것은 비효율적임

# 반복문 활용

## ▶ 중첩 반복문

### ▶ 반복할 문장이 반복문

- ▷ 하나의 실행 문장으로 다른 반복문 안에 겹쳐서 사용할 수 있음
- ▷ break 사용해 반복 끝내거나 continue로 반복 문장의 일부 건너뛸 수 있음

#### 중첩 반복문 예

```
for ( i=0; i<10; i+ + )  
{  
    for ( j=0; j<10; j+ + )  
    {  
        반복할 문장;  
    }  
}
```

---

i-for문이 10번 반복되고 j-for문이  
10번 반복되므로 반복할 문장은  
100번 반복된다.

---

# 반복문 활용

## ▶ 분기문

### ▶ break문의 기능

▷ 반복문을 종료

▷ 반복문과 switch문에서 사용

### ▶ continue의 기능

▷ 반복문을 생략

▷ while문, for문, do~while문과 같은 반복문의 내부에서 사용

▷ switch문에는 사용할 수 없음

## 분기문 사용 예

```
while ( 1 )  
{  
    if ( 조건식 1 ) break;  
    if ( 조건식 2 ) continue;  
    반복할 문장;  
}
```

---

조건식 1이 참이면 반복문을 끝낸다.

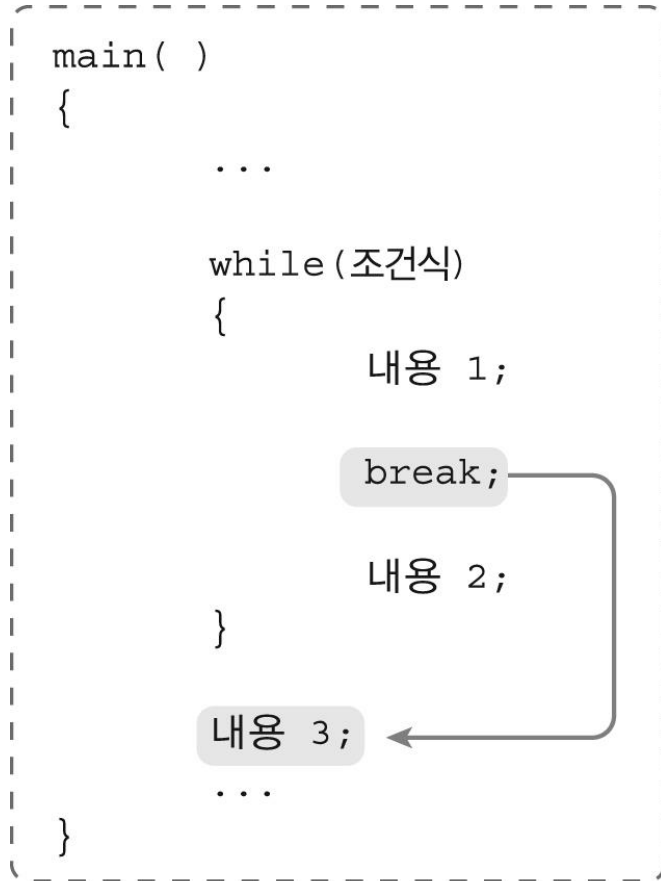
조건식 2가 참이면 반복할 문장을  
건너뛰고 처음부터 다시 반복한다.

---

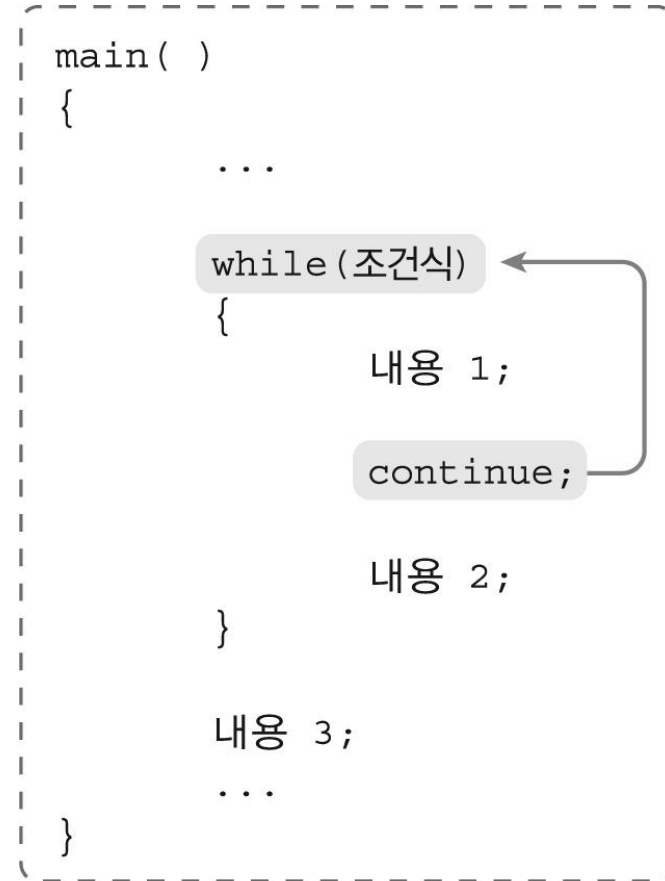


# 반복문 활용

## ▶ break 와 continue 비교



반복문에서 **break**를 만나면 무조건  
반복문을 종료하고 다음 문장을 수행



반복문에서 **continue**를 만나면 무조건  
조건식을 수행

# 반복문 활용

## ▶ break 와 continue 비교 예제

```
int i = 1;
for(;i <= 20;i++){
    if(i % 5 == 0) break;
    printf("%d\n", i);
}
```

```
int i = 1;
for(;i <= 20;i++){
    if(i % 5 == 0) continue;
    printf("%d\n", i);
}
```

## 실습문제 4

▶ do~while문을 이용하여 반복적으로 양수를 입력 받아 총합을 구하는 프로그램을 작성하시오.

▶ 음수를 입력하면 반복을 종료하고 입력 받은 모든 양수의 합을 출력

▶ 프로그램을 시작하자마다 음수가 입력되면 반복을 끝내고 0을 출력

▶ 실행결과

```
양수입력: 5  
양수입력:10  
양수입력:15  
양수입력:-1  
누적된 값 : 30
```

# 반복문 활용

## ▶ 중첩 반복문의 활용

### ▶ 한 줄에 별을 5개씩 3줄 출력하는 예제

```
int i, j;
for(i = 0; i < 3; i++){
    for(j = 0; j < 5; j++){
        printf("*");
    }
    printf("\n");
}
```

```
*****
*****
*****
```

# 실습문제 5

▶ 중첩 반복문을 이용하여 아래의 실행결과와 같이 출력하는 코드를 작성하시오.

▶ 맨 윗 줄은 5칸 들여쓰기, 그 다음 줄은 4칸 들여쓰기...

▶ 실행결과

 Microsoft Visual Studio 디버그 콘솔

```
    1
   12
  123
 1234
12345
```

## 실습문제 6

▶ 아래와 같이 구구단이 출력되도록 프로그램을 작성하시오.

▶ 2 ~ 9단까지 출력되도록 하시오.

▶ %2d 서식문자를 사용하여 구구단의 몫이 가지런히 출력되도록 하시오.

```
Microsoft Visual Studio 디버그 콘솔

2 x 1 = 2  2 x 2 = 4  2 x 3 = 6  2 x 4 = 8  2 x 5 = 10  2 x 6 = 12  2 x 7 = 14  2 x 8 = 16  2 x 9 = 18
3 x 1 = 3  3 x 2 = 6  3 x 3 = 9  3 x 4 = 12  3 x 5 = 15  3 x 6 = 18  3 x 7 = 21  3 x 8 = 24  3 x 9 = 27
4 x 1 = 4  4 x 2 = 8  4 x 3 = 12  4 x 4 = 16  4 x 5 = 20  4 x 6 = 24  4 x 7 = 28  4 x 8 = 32  4 x 9 = 36
5 x 1 = 5  5 x 2 = 10  5 x 3 = 15  5 x 4 = 20  5 x 5 = 25  5 x 6 = 30  5 x 7 = 35  5 x 8 = 40  5 x 9 = 45
6 x 1 = 6  6 x 2 = 12  6 x 3 = 18  6 x 4 = 24  6 x 5 = 30  6 x 6 = 36  6 x 7 = 42  6 x 8 = 48  6 x 9 = 54
7 x 1 = 7  7 x 2 = 14  7 x 3 = 21  7 x 4 = 28  7 x 5 = 35  7 x 6 = 42  7 x 7 = 49  7 x 8 = 56  7 x 9 = 63
8 x 1 = 8  8 x 2 = 16  8 x 3 = 24  8 x 4 = 32  8 x 5 = 40  8 x 6 = 48  8 x 7 = 56  8 x 8 = 64  8 x 9 = 72
9 x 1 = 9  9 x 2 = 18  9 x 3 = 27  9 x 4 = 36  9 x 5 = 45  9 x 6 = 54  9 x 7 = 63  9 x 8 = 72  9 x 9 = 81

C:\Users\choe\source\HelloWorld\Debug\HelloWorld.exe(21344 프로세스)이(가) 0 코드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

---

**Q & A**

---