

# 프로그래밍 언어 실습 04

6주차

# 전달인자/반환 값 모두 있을 경우

- 예제1

```
#include <stdio.h>

int AbsoCompare(int num1, int num2);
int GetAbsoValue(int num);

int main()
{
    int num1, num2;
    printf("두 개의 정수 입력: ");
    scanf("%d %d", &num1, &num2);
    printf("%d와 %d중 절댓값이 큰 정수: %d #n", num1, num2, AbsoCompare(num1, num2));

    return 0;
}

int AbsoCompare(int num1, int num2)
{
    if(GetAbsoValue(num1) > GetAbsoValue(num2))
        return num1;
    else
        return num2;
}

int GetAbsoValue(int num)
{
    if(num < 0)
        return num * (-1);
    else
        return num;
}
```

# 전달인자/반환 값이 없을 경우

## • 예제2

```
#include <stdio.h>

int Add(int num1, int num2)
{
    return num1 + num2;
}

void ShowAddResult(int num)
{
    printf("덧셈결과 출력: %d \n", num);
}

int ReadNum(void)
{
    int num;
    scanf("%d", &num);
    return num;
}

void HowToUseThisProg(void)
{
    printf("두 개의 정수를 입력하시면 덧셈결과가 출력됩니다. \n");
    printf("자! 그럼 두 개의 정수를 입력하세요. \n");
}
```

```
int main()
{
    int result, num1, num2;
    HowToUseThisProg();
    num1 = ReadNum();
    num2 = ReadNum();
    result = Add(num1, num2);
    ShowAddResult(result);

    return 0;
}
```

# 둘 이상의 return문이 존재할 경우

## • 예제3

```
#include <stdio.h>

int AbsoCompare(int num1, int num2);
int GetAbsoValue(int num);

int main()
{
    int num1, num2;
    printf("두 개의 정수 입력: ");
    scanf("%d %d", &num1, &num2);
    printf("%d와 %d중 절댓값이 큰 정수: %d #n", num1, num2, AbsoCompare(num1, num2));

    return 0;
}

int AbsoCompare(int num1, int num2)
{
    if(GetAbsoValue(num1) > GetAbsoValue(num2))
        return num1;
    else
        return num2;
}

int GetAbsoValue(int num)
{
    if(num < 0)
        return num * (-1);
    else
        return num;
}
```

# 지역변수(Local Variable)

- 예제4

```
#include <stdio.h>

int SimpleFuncOne(void)
{
    int num = 10;    // 이후부터 SimpleFuncOne의 num 유효
    num++;
    printf("SimpleFuncOne num: %d \n", num);
    return 0;       // SimpleFuncOne의 num이 유효한 마지막 문장
}

int SimpleFuncTwo(void)
{
    int num1 = 20;   // 이후부터 num1 유효
    int num2 = 30;   // 이후부터 num2 유효
    num1++, num2++;
    printf("num1 & num2: %d %d \n", num1, num2);
    return 0;       // num1, num2가 유효한 마지막 문장
}

int main()
{
    int num = 17;    // 이후부터 main의 num 유효
    SimpleFuncOne();
    SimpleFuncTwo();
    printf("main num: %d \n", num);
    return 0;       // main의 num이 유효한 마지막 문장
}
```

# 전역변수

- 예제5

```
#include <stdio.h>

void Add(int val);
int num;    // 전역변수는 기본 0으로 초기화됨

int main(void)
{
    printf("num: %d \n", num);
    Add(3);
    printf("num: %d \n", num);
    num++;
    printf("num: %d \n", num);
    return 0;
}

void Add(int val)
{
    num += val;
}
```

# 지역변수와 이름이 동일할 경우

- 예제6

```
#include <stdio.h>

int Add(int val);
int num = 1;

int main()
{
    int num = 5;
    printf("num: %d \n", Add(3));
    printf("num: %d \n", num + 9);
    return 0;
}

int Add(int val)
{
    int num = 9;
    num += val;
    return num;
}
```

# static 변수

- 예제7

```
#include <stdio.h>

void SimpleFunc(void)
{
    static int num1 = 0;
    int num2 = 0;
    num1++, num2++;
    printf("static: %d, local: %d \n", num1, num2);
}

int main()
{
    int i;
    for(i = 0; i < 3; i++)
        SimpleFunc();
    return 0;
}
```



# 재귀함수(1)

- 예제8

```
#include <stdio.h>

void Recursive(int num)
{
    if(num <= 0)    // 재귀의 탈출조건
        return;    // 재귀의 탈출!
    printf("Recursive call! %d \#n", num);
    Recursive(num - 1);
}

int main()
{
    Recursive(3);
    return 0;
}
```

# 재귀함수(2)

- 예제9

```
#include <stdio.h>

int Factorial(int n)
{
    if(n == 0)
        return 1;
    else
        return n * Factorial(n - 1);
}

int main()
{
    printf("1! = %d \n", Factorial(1));
    printf("2! = %d \n", Factorial(2));
    printf("3! = %d \n", Factorial(3));
    printf("4! = %d \n", Factorial(4));
    printf("9! = %d \n", Factorial(9));
    return 0;
}
```

# 문제 1

- 두 개의 정수를 입력 받아서 두 수 사이의 구구단을 출력하는 프로그램을 작성하라.  
(3과 5를 입력하건 5와 3을 입력하건 프로그램은 같은 결과를 출력해야함)

```
두개의 정수 입력 : 4 3
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

4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = 28
4 * 8 = 32
4 * 9 = 36

계속하려면 아무 키나 누르십시오 . . .
```

## 문제 2

- 프로그램 사용자로부터 초(second)를 입력 받은 후에, 이를 [시, 분, 초] 형태로 출력하는 프로그램을 작성하라.

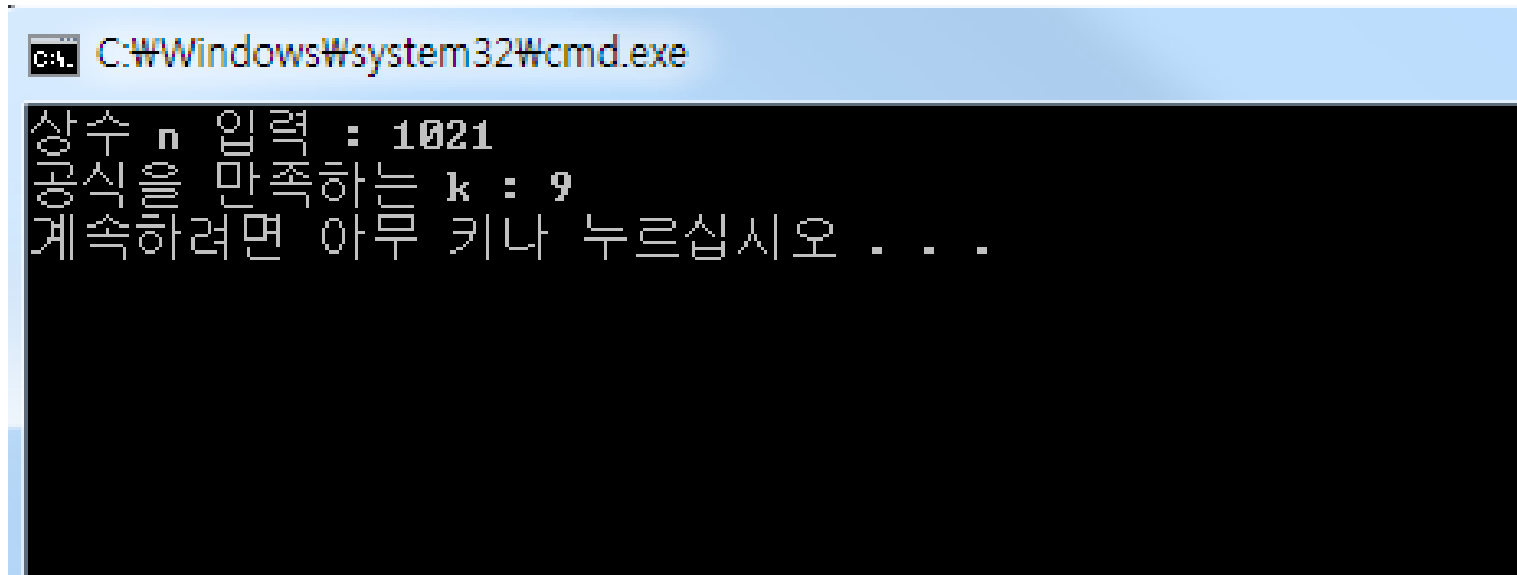
C:\Windows\system32\cmd.exe

```
초(second) 입력 : 3715  
[h:1, m:1, s:55]  
계속하려면 아무 키나 누르십시오 . . .
```

# 문제 3

- 프로그램 사용자로부터 숫자  $n$ 을 입력 받고 다음 공식이 성립하는  $k$ 의 최댓값을 계산해서 출력하는 프로그램을 작성하라.

$$2^k \leq n$$



```
C:\Windows\system32\cmd.exe
상수 n 입력 : 1021
공식을 만족하는 k : 9
계속하려면 아무 키나 누르십시오 . . .
```

Q & A