

프로그래밍을 통한 논리적 사유 연습





10장. 함수

(인자 값, 재귀 함수)

배열을 함수의 인자로 전달

- 다음 1차원 배열 원소의 합을 구하는 방법

```
int a[ ] = {10, 5, 15, 20, 30};
```

① 반복 문

```
int i, sum = 0;
for(i=0; i<5; i++)
    sum += i;
```

② 사용자가 작성한 함수 SumArray()

```
int SumArray(int Array[], int Size)
{
    ① 배열    ② 원소개수
    int1, result = 0;
    for(i=0; i<Size; i++) ③ 내용
        result += Array[i];

    return result; ④ 반환값
}
```

배열을 함수의 인자로 전달

- SumArray() 함수

① 배열: 배열 명은 가인자이기 때문에 식별자 작성 규칙에 따라 임의로 작성

```
int SumArray(int Array[4], ..... ) // 이 경우는 배열 원소의 개수가 4개인
{
    .....
    // 배열만 인자로 사용될 수 있다.
}
```

② 원소 개수 : 일반적으로 배열 명에는 빈 대괄호인 []를 사용. 실제 배열의 원소 개수는 두 번째 인자로 표시. 변수 Size가 배열의 원소 개수를 나타냄

③ 내용 : 인자로 사용된 배열과 원소의 개수를 이용해서 필요한 기능을 작성

④ 반환 값: 함수에서 계산된 결과값을 반환

배열을 함수의 인자로 전달

- 앞의 SumArray()같이 배열을 인자로 사용하는 함수의 호출

```
int main( )  
{  
    Int Sum, a[] = {10, 5, 15, 25, 7};  
    Sum = SumArray(a, 5);           // 함수 호출  
  
    ...  
    return 0;  
}
```

배열 부분에 배열명만 기록!

배열을 함수의 인자로 전달

예제) 배열을 인자로 사용하는 함수

```
01 #include <stdio.h>
02
03 int SumArray(int Array[ ], int size); // 함수 정의
04
05 int main(void)
06 {
07     int a[ ] = {10, 5, 15, 25, 7};
08     int Sum;
09     Sum = SumArray(a, 5); // 함수 호출
10     printf("배열 원소의 합 : %d\n", Sum);
11     return 0;
12 }
13
14 int SumArray(int Array[ ], int size) // 함수 정의
15 {
16     int result = 0, i;
17     for(i = 0; i < size; i ++){
18         result += Array[i];
19     }
20     return result;
21 }
```

배열 원소의 합 : 62

재귀 함수

- 재귀호출(Recursion) : 함수에서 그 함수를 다시 호출하는 것
- 재귀함수 : 재귀호출을 구현한 함수

$$N! = N * (N-1) * (N-2) * (N-3) * \dots 1$$

$$N! = N * (N-1)!$$

$$(N-1)! = (N-1) * (N-2)!$$

.....

재귀 함수

$$\text{Factorial}(3) = 3 * \text{Factorial}(2)$$

$$\text{Factorial}(2) = 2 * \text{Factorial}(1)$$

$$\text{Factorial}(1) = 1 * \text{Factorial}(0)$$

$$\text{Factorial}(0) = 1$$

$$\text{Factorial}(3) = 3 * 2 = 6$$

$$\text{Factorial}(2) = 2 * 1 = 2$$

$$\text{Factorial}(1) = 1 * 1 = 1$$

재귀 함수

- 다음 factorial() 함수를 보면, 함수에서 자기 자신을 다시 호출

```
int factorial(int n)
{
    ....
    return n * factorial(n-1);
}
```

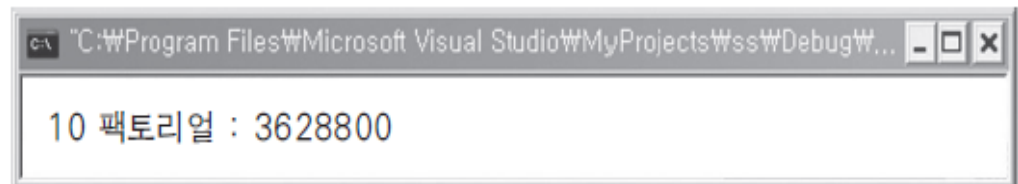
- factorial() 함수는 n값이 1일 때나 0일 때, 즉 1!이나 0!일 때 값이 1이므로 다음과 같이 종료 조건을 작성

```
int factorial(int n)
{
    if(n<=1) return 1;           // 종료 조건
    else return n * factorial(n-1);
    ....
}
```

재귀 함수

예제) 팩토리얼을 구하는 예제

```
01 #include <stdio.h>
02 int factorial(int n);
03
04 int main(void)
05 {
06     int fact_num;
07     fact_num = factorial(10);
08
09     printf("10 팩토리얼 : %d\n", fact_num);
10
11     return 0;
12 }
13
14 int factorial(int n)
15 {
16     if (n <= 1)
17         return 1;
18     else
19         return (n * factorial(n-1));
20 }
```



재귀 함수

예제) 사용자가 입력한 숫자까지의 합을 재귀함수로 구하는 예제

```
01 #include <stdio.h>
02
03 int sum(int a);
04 int main(void)
05 {
06     int input, s = 0;
07     printf("양의 정수를 입력하세요 : ");
08     scanf("%d", &input);
09
10     s = sum(input);
11     printf("재귀함수를 이용한 1부터 %d까지 합 : %d\n", input, s);
12     return 0;
13 }
14
15 int sum(int a)
16 {
17     if (a <= 1)
18         return 1;
19     else
20         return a + sum(a-1);
21 }
```

