

9강

C_PROGRAMMING



함수

❖ 정의

- 독립적인 기능을 가지는 작은 프로그램

❖ 함수의 구조와 실행 내용을 구체적으로 명시한 문장들

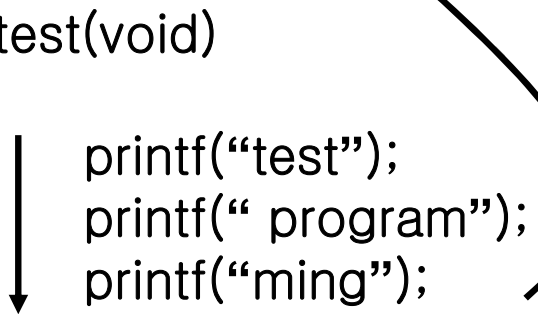
```
반환자료형  함수명(가인수 리스트){  
    함수내용  
}
```

❖ 사용 예

```
int func(int num1, int num2){  
    num1*=num2;  
    return num1;  
}
```

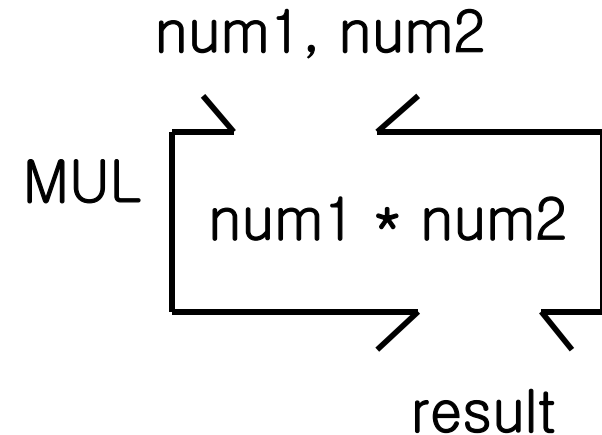
```
#include <stdio.h>  
void test(void);  
main()  
{  
    test();  
}
```

```
void test(void)  
{  
    printf("test");  
    printf(" program");  
    printf("ming");  
}
```



함수를 만드는 방법

- ❖ 함수가 수행하여야 할 기능을 결정한다.
- ❖ 전달받는 값(가 인수)들과 반환값을 결정한다.
- ❖ 함수의 이름과 가 인수들의 이름을 결정한다.
- ❖ 반환값의 자료형을 기술한다.
- ❖ 함수의 이름을 기술한다.
- ❖ 괄호 사이에 가 인수 리스트를 기술한다.
- ❖ 함수의 본체를 둘러싸는 괄호 {}를 기술한다.
- ❖ 함수의 본체 내부에서 요구되는 각종 선언문을 기술한다.
- ❖ 함수가 수행하여야 할 작업, 즉 실행문들을 기술한다.
- ❖ 필요하다면 return문을 기술한다.
- ❖ 머리부분에 기술된 반환값의 자료형과 return문에 의해 반환되는 자료의 형이 일치되는지 검사한다.
- ❖ 함수 원형 선언문이 필요하다면 기술한다.



함수를 만드는 방법

- ❖ 함수가 수행하여야 할 기능을 결정한다.
- ❖ 전달받는 값(가 인수)들과 반환값을 결정한다.
- ❖ 함수의 이름과 가 인수들의 이름을 결정한다.
- ❖ 반환값의 자료형을 기술한다.
- ❖ 함수의 이름을 기술한다.
- ❖ 괄호 사이에 가 인수 리스트를 기술한다.
- ❖ 함수의 본체를 둘러싸는 괄호 {}를 기술한다.
- ❖ 함수의 본체 내부에서 요구되는 각종 선언문을 기술한다.
- ❖ 함수가 수행하여야 할 작업, 즉 실행문들을 기술한다.
- ❖ 필요하다면 return문을 기술한다.
- ❖ 머리부분에 기술된 반환값의 자료형과 return문에 의해 반환되는 자료의 형이 일치되는지 검사한다.
- ❖ 함수 원형 선언문이 필요하다면 기술한다.

```
int mul(int num1, int num2){  
    int result;  
    result=num1*num2;  
    return result;  
}
```

예제

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



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

```
SumFunc(int num)
{
    int i, sum=0;
    for( i=1;i<=num;i++)
        sum+=i;
    printf("%d\n", sum);
}
```

```
main()
{
    int n;
    scanf("%d", &n);
    SumFunc(n);
}
```

예제

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



```
#include <stdio.h>

int SumFunc(int num)
{
    int i, sum=0;
    for( i=1;i<=num;i++)
        sum+=i;
    return sum;
}

main()
{
    int ret, n;
    scanf("%d", &n);
    ret= SumFunc(n);
    printf("%d\n", ret);
}
```

예제 [함수 원형 선언문]

```
#include <stdio.h>
float avrg(int x,int y);
void show_avrg(int,int,float);

void main()
{
    int i, j; float f;
    i = 2; j = 3;
    f = avrg(i,j);
    show_avrg(i,j,f);
}
```

```
float avrg(int j,int k)
{
    int total; float f;
    total = j + k;
    f = total / 2.0f;    //(float)2.0
    return f;
}

void show_avrg(int a,int b,float c)
{
    printf("\n%d와 %d의 평균",a,b);
    printf("값은 %3.1f입니다. \n",c);
}
```

예제

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

```
char big(int x, int y)
```

```
{
```

```
    if(x > y) return 'Y';
```

```
    else if(x < y) return 'N';
```

```
    else return '=';
```

```
}
```

```
void main(void)
```

```
{
```

```
    int a = 100, b = 200;
```

```
    char size = big(a, b);
```

```
    if(size == 'Y')
```

```
        printf("Wn %d는(은) %d보다 크다. ", a,b);
```

```
    else if(size == 'N')
```

```
        printf("Wn %d는(은) %d보다 작다. ", a,b);
```

```
    else if(size == '=' )
```

```
        printf("Wn %d는(은) %d과(와) 같다.", a,b);
```

```
}
```

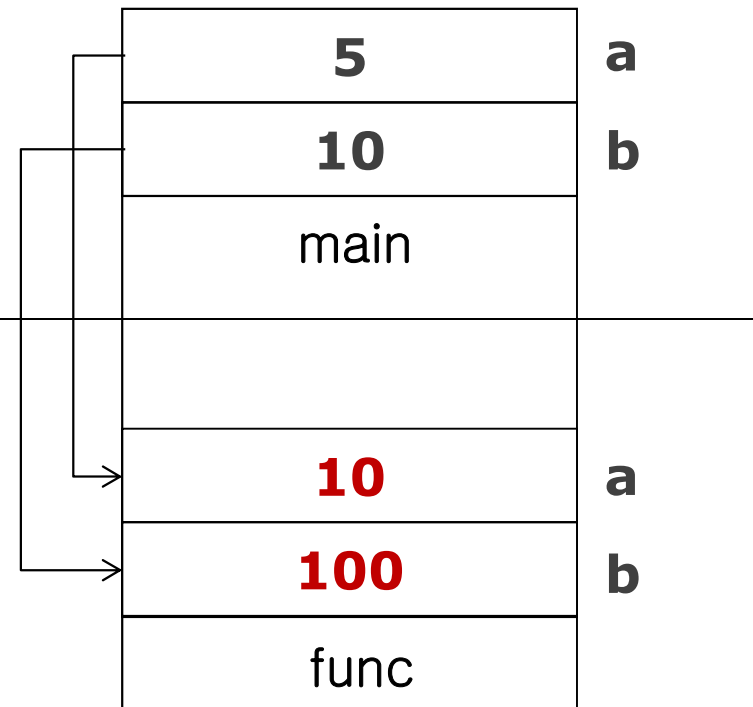

함수의 호출

❖ 값에 의한 호출

- 값을 받아 처리, 지역변수로 처리

```
main(){  
    int a=5, b=10;  
    func(a, b);  
    printf("a=%d, b=%d", a, b);  
}
```

```
func(int a, int b){  
    a+=5;  
    b*=10;  
    printf("a=%d, b=%d", a, b);  
}
```



❖ 주소에 의한 호출

재귀함수

- ❖ 자기 자신을 호출하므로써 직접 재귀와 간접 재귀가 있다
- ❖ 직접 재귀는 자기 자신을 직접 호출하는 것이고 간접 재귀는 자신을 호출한 함수를 다시 호출하는 것이다.

❖ 예제

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

```
void main()  
{  
    printf(" recursive function ");  
    main();  
}
```

예제

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

```
void main(void)
```

```
{
```

```
    static int a = 0;
```

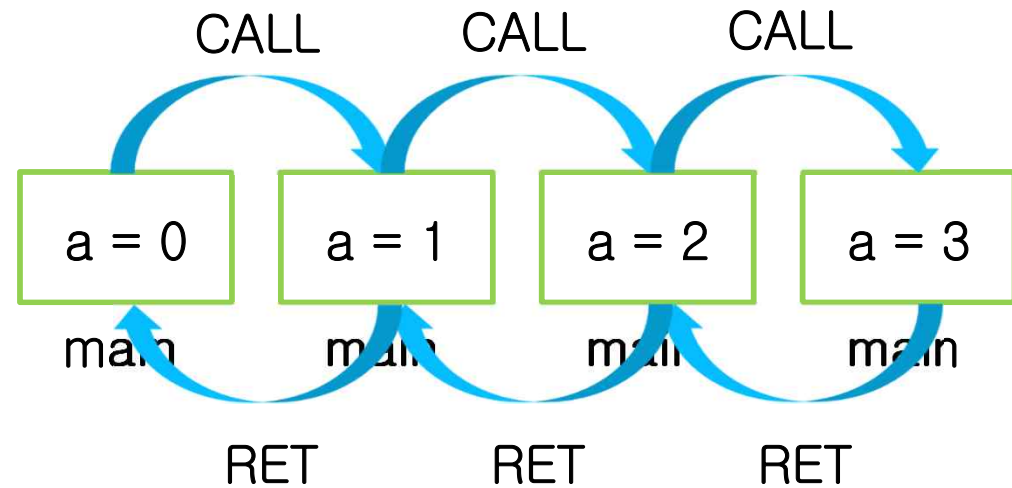
```
    if(a >= 3) return;
```

```
    else a = a + 1;
```

```
    printf(" recursive function ");
```

```
    main();
```

```
}
```



예제

```
#include <stdio.h>
long fact(long n)
{
    if(n==1L) return 1L;
    else return (n * fact(n-1));
}
void main(void)
{
    long x = 5L;
    printf("Wn %ld!의 값은 %ld이다. Wn", x, fact(x));
}
```

```
fact(5)
5 * fact(5-1)
  4 * fact(4-1)
    3 * fact(3-1)
      2 * fact(2-1)
        return 1
```

문제

- ❖ 다음의 기능을 가진 함수를 만들고 main 함수에서 호출 및 결과 확인.
- ✓ 입력 받은 값이 짝수인지 홀수인지 판별 후 결과 출력.
- ✓ 3의 배수를 판별하는 함수를 만들고 이 함수를 이용하여 3의 배수만 출력하시오.
- ✓ 절대값을 구하는 함수를 만들고 결과 출력.
- ✓ 거꾸로 수를 반환하는 함수를 만들고 입력 받은 수에 대한 결과 출력.
- ✓ 입력 받은 문자를 대/소문자 변환하는 함수 만들고 확인.