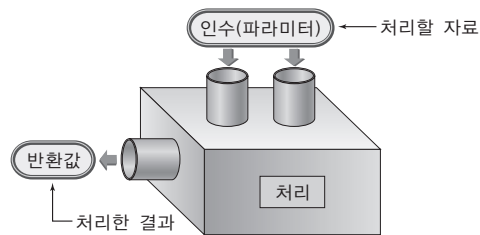


학습목표

- ▶ 함수의 개념에 대해 알 수 있다.
- ▶ 사용자 정의 함수를 작성할 수 있다.

1 함수의 개념

- ① 함수(function)란 특정한 작업을 수행하도록 설계된 독립적인 프로그램(코드 블록)이다.
- ② 프로그램 내에서 동일하게 반복되는 부분을 하나의 독립된 단위 프로그램으로 만들어 두고, 필요할 때마다 호출하여 사용하면 프로그램을 구조화할 수 있다.
- ③ C 언어에서 사용할 수 있는 함수는 C 언어 자체에 미리 준비되어 있는 라이브러리(library) 함수와 사용자가 직접 정의하여 사용하는 사용자 정의 함수로 구분한다.
- ④ 라이브러리 함수는 프로그래밍 언어가 만들어질 때 자체적으로 준비되어 있는 함수로 사용자가 만들지 않고 직접 이용할 수 있는 함수를 말한다.
- ⑤ 사용자 정의 함수는 사용자가 직접 정의하고 호출하여 사용하는 함수를 말한다.
- ⑥ 함수는 자기 자신을 다시 호출하여 사용하기에 하는데 이러한 함수를 되부름 함수라 하며, 되부름 함수를 보통 자기 호출(recursion)이라고 한다.



- ⑦ 함수는 전달받은 인수값을 이용하여 처리 작업을 수행한 후 결과값을 반환하는 경우도 있고, 반환하지 않는 경우도 있다.

예 함수를 사용하여 덧셈 연산을 하는 프로그램

프로그램 소스 코드	실행 결과
<pre>#include <stdio.h> int sum(int k, int m){ int s; s = k + m; return s; } void main(){ int a, b, t; a = 10, b = 20; t = sum(a, b); printf("%2d\n", t); }</pre>	30



EBS tip

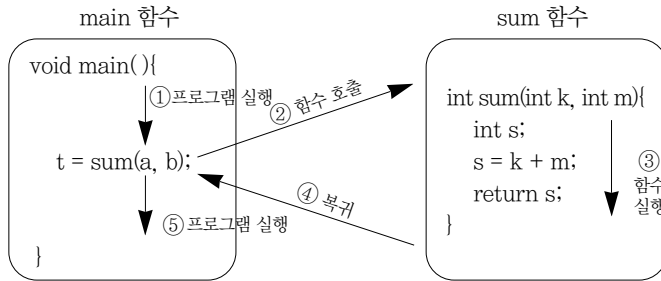
● 함수 사용 시 좋은 점

- 프로그램의 구조 파악 및 분석 용이
- 프로그램의 유지 보수에 대한 편리성 증가
- 프로그램에 대한 생산성 증가



[프로그램 해설]

• 함수의 실행 과정



- 함수의 이름은 sum이다.
- 프로그램을 실행하는 도중 sum 함수를 호출한다. 호출된 sum 함수로 이동한 후(②) 함수를 실행(③)한다. 함수 실행이 완료되면 호출한 프로그램으로 되돌아와서(④) 프로그램 실행을 완료(⑤)한다.
- a = 10, b = 20값을 sum 함수에 전달하면 k = 10, m = 20이 된다. sum 함수에서 변수 k와 m값을 더해 s에 저장한다. 함수에 의해 처리된 후, 반환값은 30이 된다.
- 따라서 프로그램의 실행 결과 출력값은 30이 된다.

2 사용자 정의 함수

- ① 사용자 정의 함수는 사용자가 직접 정의하고 작성하여 사용하는 함수를 말한다.
- ② 사용자 정의 함수의 구조는 다음과 같다.

```
자료형 함수명(인수 리스트){
  변수 선언;
  실행 문장;
  return;
}
```

- 자료형 : 함수의 실행 결과 반환되는 자료의 형을 지정하는 것으로 반환하는 값이 없으면 void를 사용한다.
- 함수명 : 변수명을 작성하는 것과 같은 방법으로 작성한다(단, 라이브러리 함수명과 같은 이름은 사용하지 않는다.).
- 인수 : 함수를 호출할 때 함수에 전달해 주는 값을 받기 위한 것으로, 반드시 괄호로 묶어 주어야 한다. 괄호 뒤에는 세미콜론(;)이 붙지 않는다. 인수가 없는 경우에는 인수 리스트를 생략하고 빈 괄호만 사용한다.
- 함수의 처리 문장은 블록({ })으로 묶어 지정한다.
- return : 함수의 처리 결과를 반환한다. 함수에서 값을 반환하지 않을 경우에는 return; 또는 return 문을 생략한다.

● 정의된 함수의 호출

④ 함수명(인수 리스트);

예 plus(a,b);

⑥ 변수=함수명(인수 리스트);

예 c=plus(a,b);

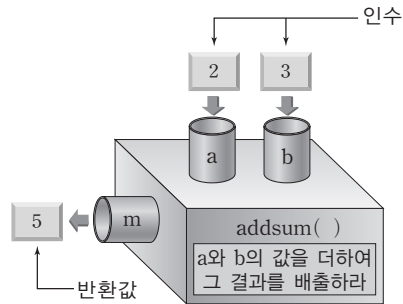
함수를 호출할 때 함수에 넘겨 줄 인수가 없는 경우에는 인수 리스트를 생략하고, 빈 괄호만 사용한다.

함수의 실행값을 반환하지 않는 경우에는 ④ 방식을, 반환하는 경우에는 ⑥ 방식을 주로 사용한다.



● 두 수의 합을 구하는 사용자 정의 함수

```
int addsum(int a, int b){
    int m;
    m=a+b;
    return m;
}
함수명 : addsum
매개 변수 : a, b
반환값의 자료형 : int
```



예 큰 수와 작은 수의 차를 구하는 사용자 정의 함수

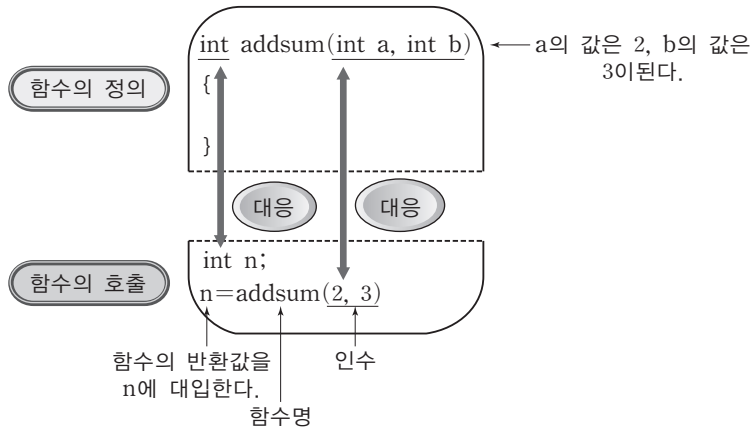
프로그램 소스 코드	실행 결과
<pre>#include <stdio.h> int dn(int a, int b){ if (a > b) return a - b; else return b - a; } void main(){ int k; k = dn(5, 3); printf("%2d", k); }</pre>	2

[프로그램 해설]

- dn() 함수는 전달받은 두 변수값의 크기를 비교하여 큰 값에서 작은 값을 빼는 함수이다.
- 조건문을 이용하여 두 수의 크기를 비교한 후 조건을 만족하면 a-b를, 만족하지 않으면 b-a 값을 돌려준다.
- 메인 함수에서 a와 b에 각각 5와 3을 전달해 주고, 매개 변수에서는 5와 3을 각각 전달받게 된다.
- 조건문에서 $5 > 3$ 의 조건을 만족하므로 $5-3$ 한 결과값 2를 돌려준다.
- 따라서 출력되는 값은 2가 된다.

3 값의 전달

- ① 인수(parameter)는 함수를 호출할 때 자료를 전달하기 위한 것으로 함수를 호출하는 쪽에서 본 매개 변수를 실인수라고 하고, 호출되는 함수 쪽에서 본 매개 변수를 가인수라 한다.
- ② 실인수가 여러 개일 경우에는 콤마(,)로 구분하여 나열하며, 이 값들은 호출되는 함수의 가인수에 순서대로 전달된다.
- ③ 실인수와 가인수의 자료형은 반드시 일치해야 한다.
- ④ 최종값을 전달받기 위한 자료형도 함수에서 결과값을 되돌려 주는 자료형과 일치해야 한다.



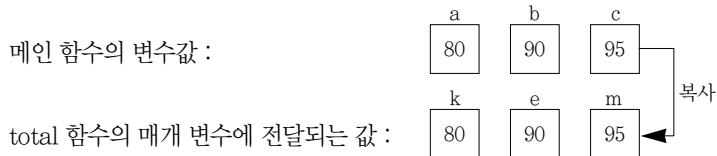
- 호출한 함수에서 전달하는 값은 2, 3으로 정수값이다. 따라서 addsum 함수에서 값을 전달받기 위한 매개 변수 a, b는 정수형으로 선언해 준다. 또한 addsum 결과값은 정수형이므로 결과값을 저장할 변수도 역시 정수형이 되어야 하므로 변수 n은 정수형으로 선언되었다.

예 값을 전달받아 합을 구하는 사용자 정의 함수

프로그램 소스 코드	실행 결과
<pre>#include <stdio.h> int total(int k, int e, int m){ int tot; tot = k + e + m; return tot; } void main(){ int a, b, c; int sum; a = 80, b = 90, c = 95; sum = total(a, b, c); printf("%2d", sum); }</pre>	265

[프로그램 해설]

- 사용자 정의 함수 total을 정의한다.
- 프로그램을 실행하면서 total 함수에 값을 전달한다. 전달되는 값은 80, 90, 95가 된다.



- 변수 a, b, c의 기억 공간과 변수 k, e, m의 기억 공간은 전혀 다르다. 즉 메인 함수에서 total 함수에 값을 복사만 한다.
- total 함수는 k, e, m의 값을 더하여 결과값을 tot에 저장한다.
- total 함수의 값을 메인 함수에 전달한다. 함수의 결과값을 정수형으로 반환하고 있으므로 반환되는 값을 저장하기 위한 변수 sum도 정수형으로 선언된다.

- 함수에 있는 매개 변수에 단지 값을 복사하여 전달하는 방법을 값에 의한 전달(Call By Value)이라고 한다. 값에 의한 전달 방법의 경우 실인수와 매개 변수의 기억 공간이 전혀 별개이므로 가인수의 변화는 실인수에 영향을 미치지 못한다.

예 매개 변수 값의 변화

프로그램 소스 코드	실행 결과
<pre>#include <stdio.h> void re(int a, int b){ a = a + 10; b = b + 10; printf("re 함수의 a,b 값\n"); printf("a:%2d b:%2d\n", a, b); } void main(){ int a, b; a = 30, b = 40; re(a, b); printf("main 함수의 a, b 값\n"); printf("a : %2d b : %2d", a, b); }</pre>	<pre>re 함수의 a,b 값 a:40 b:50 main 함수의 a,b 값 a : 30 b : 40</pre>

[프로그램 해설]

- main 함수에서 re 함수에 30, 40을 전달하면 re 함수의 a, b 값은 30, 40이 된다.
- re 함수에서 a와 b에 각 10을 더해 주므로 a와 b의 값은 40과 50이 된다. 하지만 main 함수에서 a, b의 값을 출력해 보면 30과 40이다.
- main 함수에서 매개 변수에 값을 단지 복사하여 전달했으므로 함수 re에서 변수값의 수정은 main 함수에 있는 실인수의 값에 전혀 영향을 미치지 않는다.

예 함수를 이용한 배열값의 변화

프로그램 소스 코드	실행 결과
<pre>#include <stdio.h> int a[5] = {1, 1, 0, 0, 1}; void ta(int x); void td(); void main(){ int m; td(); ta(4); for(m = 0; m <= 4; m++) printf("%2d", a[m]); } void ta(int x){ a[0] = x; } void td(){ int k; for(k = 4; k >= 1; k--) a[k] = a[k-1]; }</pre>	<pre>4 1 1 0 0</pre>

[프로그램 해설]

- td() 함수를 호출하면 td() 함수는 배열 a의 요소를 첨자 4부터 1까지의 요소를 하나씩 뒤로 미루어 준다.
- 함수 ta()에 의해 ta(4)는 a[0] 요소가 4로 대체된다.
- 따라서 실행 결과는 4 1 1 0 0이 된다.



● 함수는 정의하는 위치에 따라 다음과 같이 2가지 경우가 있다.

- ① 사용자 정의 함수가 main 함수 이후에 나오는 경우 선언을 해줌.

형식 1

```
자료형 함수명( ); //함수의 선언
void main(void) {
    :
    함수 호출(실인수);
    :
}
자료형 함수명(가인수) {
    함수 내부 변수 선언
    :
    함수의 본체
    :
    return(되돌리는 값);
}
```

- ② 사용자 정의 함수가 main 함수보다 먼저 나오는 경우에는 선언하지 않음.

형식 2

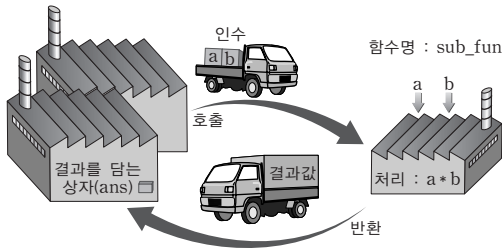
```
자료형 함수명(가인수) {
    함수 내부 변수 선언
    :
    함수의 본체
    :
    return(되돌리는 값);
}
void main(void) {
    :
    함수 호출(실인수)
    :
}
```



기출 모의고사

정답 및 해설 p. 14

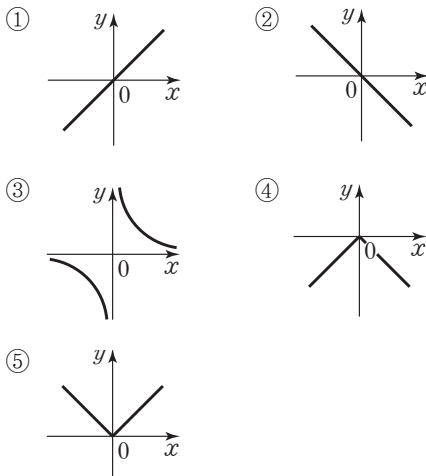
- 1 2006학년도 대수능
그림은 함수의 호출과 반환 과정을 나타낸 것이다. 이 과정을 명령문으로 표현할 때 옳은 것은?



- ① sub_fun();
- ② sub_fun(ans);
- ③ sub_fun(a*b);
- ④ ans = sub_fun(a,b);
- ⑤ ans = sub_fun(a*b);

- 2 2010년 9월 시행 평가원 모의평가
다음은 사용자 정의 함수 Fun_A()를 정의하고 이를 호출하는 프로그램의 일부이다. 모든 실수 x에 대해 점선 (가) 부분을 실행하여 얻은 y와 x의 관계를 그래프로 표현할 때 옳은 것은?

사용자 정의 함수	<pre>double Fun_A(double a){ if(a > 0) { a = a * (-1); } return a; }</pre>
프로그램	<pre>... y = Fun_A(x); (가) ...</pre>



- 3 2007년 6월 시행 평가원 모의평가
다음 프로그램의 입력값이 123일 때 출력값은?

```
# include <stdio.h>

int rev(int x) {
    int y=0;

    while(x>0) {
        y=(y*10)+(x%10);
        x=x/10;
    }
    return y;
}

void main() {
    int a, b;

    scanf("%d", &a);
    b=rev(a);
    printf("%d", b);
}
```

- ① 123 ② 131 ③ 132
- ④ 231 ⑤ 321

[4~5] 다음은 어떤 수까지의 합을 구하는 프로그램이다.
물음에 답하시오.

```
#include <stdio.h>
int in_f(int num){
    static int k=0;
    int cnt;
    for(cnt=1; cnt<=num; cnt++){
        k=k+cnt;
    }
    return k;
}
void main(){
    int n;
    int ss;
    for(n=1; n<=5; n++){
        ss=in_f(n);
        printf("%d ", ss);
    }
}
```

4 이 프로그램에서 사용자 정의 함수와 변수의 범위에 대한 설명으로 옳지 않은 것은?

- ① 변수 k는 정적 변수이다.
- ② 함수 in_f()는 1번 호출된다.
- ③ 변수 ss와 cnt는 지역 변수이다.
- ④ 함수 in_f()의 인수는 하나이다.
- ⑤ 함수 in_f() 반환값의 자료형은 int 형이다.

5 이 프로그램의 출력 결과는?

- ① 1 2 3 4 5 ② 1 3 6 10 15
- ③ 1 4 10 20 35 ④ 1 5 10 15 25
- ⑤ 5 5 5 5 5

6 다음 프로그램의 실행 결과는?

```
#include <stdio.h>
void digit(int k, int m);
void main(){
    int k, m;
    k = 30, m = 50;
    printf("%2d %2d\n", k, m);
    digit(k,m);
    printf("%2d %2d\n",k, m);
}
void digit(int k, int m){
    int tmp;
    tmp = k;
    k = m;
    m = tmp;
    printf("%2d %2d\n", k, m);
}
```

- | | | |
|---------------------------|---------------------------|---------------------------|
| ① 30 50
50 30
30 50 | ② 30 50
50 30
50 30 | ③ 30 50
50 50
30 30 |
| ④ 50 30
30 50
50 30 | ⑤ 50 30
50 30
30 50 | |