

C Language

C프로그래밍및실습

담당교수:최희식
dali3054@ssu.ac.kr

목차

- 1) 함수 개요
- 2) 함수 정의
- 3) 함수 호출과 반환
- 4) 함수와 변수의 종류
- 5) 함수에 배열 전달
- 6) 함수와 라이브러리

1) 함수 개요

- C언어에서 함수

- 어떤 특정한 일을 수행하는 **독립적인** 단위
- 함수의 예 : printf(), scanf(), main()

- 함수는 함수 정의부와 함수 호출부로 구성

- ✓ 함수 **정의부** : 함수를 구현하는 부분 (즉, 마술상자 내부 구현)
 - ✓ 예) main() 함수 구현
- ✓ 함수 **호출부** : 함수의 기능을 이용하는 부분 (즉, 마술상자 이용)
 - ✓ 예) printf() 함수 호출

2) 함수 정의

- 함수 정의

- 함수가 수행해야 할 기능 명세하기

- 구문

```
반환형    함수이름    (인자선언1, 인자선언2, ...)  
{  
    함수의 수행코드  
  
    return 문;        ⇒ 필요 시  
}
```

```
int add (int x, int y)  
{  
    int z;  
    z = x+y;  
  
    return z;  
}
```

2) 함수 정의

- 함수 정의 요소

- 함수 이름

- ✓ 함수 이름 규칙은 변수명 규칙과 동일 (식별자)
 - ✓ 의미있는 이름 사용 권장

- 함수 인자 or 매개 변수(parameter)

- ✓ 함수의 입력 데이터를 나타내는 변수들
 - ✓ 인자가 다수인 경우 콤마(,)로 구분
 - ✓ 인자가 없더라도 소괄호는 반드시 필요

- 반환형(return type)

- ✓ 함수는 수행된 후에 반환하는 결과의 자료형
 - ✓ 아무 결과도 반환하지 않으면 자료형에 void를 씀

- 참고) 위 세 구성 요소는 함수의 형태 명세 → 함수 헤더라 칭함

```
int add (int x, int y)
{
    int z;
    z = x+y;

    return z;
}
```

2) 함수 정의

▪ 함수 정의 요소

• 함수 몸체(body)

- ✓ 함수가 수행해야 할 작업을
`{ }` 안에 명세
- ✓ 함수는 제어 흐름에 따라 수행되다,
맨 마지막 문장까지 수행 되거나
`return` 문을 만나면 종료

```
int add (int x, int y)
{
    int z;
    z = x+y;

    return z;
}
```

✓ `return` 문

- ✓ 함수를 종료하고 함수의 결과를 반환하는 역할
- ✓ 반환형이 `void` 인 경우 보통 `return` 문 사용하지 않지만,
중간에 함수를 종료시키기 위해 쓰기도 함
 - » `return z;` ⇨ 변수 `z`에 저장된 값 반환
 - » `return 10+20;` ⇨ `10+20`, 즉 30을 반환
 - » `return;` ⇨ 아무 것도 반환하지 않음
(반환형이 `void` 인 경우 사용)

목차

- 1) 함수 개요
- 2) 함수 정의
- 3) 함수 호출과 반환**
- 4) 함수와 변수의 종류
- 5) 함수에 배열 전달
- 6) 함수와 라이브러리

[실습]

- 두 숫자를 입력한 후, 인자 값으로 함수를 호출하여 함수 영역에서 덧셈 합계를 출력하는 프로그램을 작성하시오.

[실행 결과]

15 20
합계=35

어떤 알파벳 문자 하나를 함수를 호출하여 대문자로 변환하여 결과 값을 반환하여 출력하는 함수 프로그램을 작성하시오.

```
char ch='k'
```

[실행 결과]

K

[실습]

- 함수를 호출하여 문자 ' a ' 는 한 번, 문자 ' b ' 는 두 번, ..., 문자 ' z ' 는 26번 까지 출력하는 프로그램을 작성하시오.
- 단, 각 문자별 한 줄에 하나씩 출력한다.

- [실행 결과]

```
a
bb
ccc
dddd
eeee
ffffff
gggggg
hhhhhhh
iiiiiii
jjjjjjj
kkkkkkkk
lllllllll
mmmmmmmmmm
nnnnnnnnnnnn
oooooooooooo
ppppppppppppppp
qqqqqqqqqqqqqqq
rrrrrrrrrrrrrrrrr
ssssssssssssssss
ttttttttttttttttt
uuuuuuuuuuuuuuuuu
vvvvvvvvvvvvvvvvvvv
wwwwwwwwwwwwwwww
xxxxxxxxxxxxxxxxxxx
yyyyyyyyyyyyyyyyyy
zzzzzzzzzzzzzzzzzzz
```

[실습]

- 두 숫자를 입력한 후, 인자 값으로 함수를 호출하여 함수 영역에서 덧셈 합계 결과를 호출부 영역에 반환하여 출력하는 프로그램을 작성하시오.

[실행 결과]

15 20
합계=35

3) 함수 호출과 반환

- 함수 원형 선언 (함수 선언)

- add() 함수 정의를 main() 함수 뒤에 작성하면?

```
int main()
{
    . . .
    c = add(3,4);
    . . .
}

int add (int x, int y)
{
    . . .
}
```

오류 발생
add()가 정의
되지 않았습니다.

- 컴파일 오류 or 경고 발생
- 함수 정의를 함수 호출 위치보다 앞에 작성 해주어야 함
- 다수의 함수를 정의해 사용할 때, 함수 호출 순서에 맞추어 함수를 배치하는 것은 불편

3) 함수 호출과 반환

- 함수 원형 선언 (함수 선언)
 - 이 문제를 해결하는 방법은?

```
int add (int x, int y); ← 함수 원형 선언

int main(){
    . . .
    c = add(3,4); ← 정상 동작
    . . .
}

int add (int x, int y){ ← 함수 정의
    . . .
}
```

- 함수의 형태를 표현하는 **함수 원형**을 앞 부분에 **선언**
- 인자가 두 개의 int형 변수이고, 반환형이 int형인 add() 함수가 어딘가 정의되어있다는 것을 알려주는 (선언) 역할
- 용어 '**선언**' 과 '**정의**'를 혼동하지 말자

3) 함수 호출과 반환

- 함수 원형 선언 (함수 선언)

- 함수의 헤더와 동일한 형태를 가지는데, **마지막에 세미콜론(;)**을 붙여줌

반환형 함수이름 (인자선언1, 인자선언2,...) ;

- 함수의 형태를 지정해 주는 것이므로, 인자 선언에서 변수명은 무시됨
 - ✓ 인자 이름을 생략해도 되고, 심지어 함수 정의에 사용된 변수명과 다른 변수명 명시하는 것도 가능
 - ✓ 아래 세 선언은 모두 동일

```
int add(int x, int y);  
int add(int a, int b);  
int add(int, int);    → 권장 형태
```

[실습]

- 함수 원형을 정의한 후, 두 숫자를 입력한 후, 인자 값으로 함수를 호출하여 함수 영역에서 덧셈 합계 결과를 호출부 영역에 반환하여 출력하는 프로그램을 작성하시오.

[실행 결과]

15 20
합계=35

[실습]

- 함수 원형을 정의한 후, 사용자로 부터 팩토리얼 값을 입력받은 후, 인자 값으로 함수를 호출하여 계산 된 결과 값을 반환하여 [실행 결과]와 팩토리얼 값을 출력하시오.

[실행 결과]

5	← 입력
5!=120	← 출력

[실습]

두 개의 양의 정수 A과 B를 입력 받아, A부터 B까지의 합을 계산하는 프로그램을 작성하시오.

- $A \leq B$ 인 입력만 들어온다고 가정하시오.
- sum 함수 정의 및 사용
- 인자는 하나의 정수 n, 반환형은 int

[실행 결과]

4 8	← 입력
30	← 출력

[실습]

- 양의 정수를 반복해서 입력 받아(종료 조건 시까지), 예시와 같이 트리 모양을 출력하는 프로그램을 작성하시오.
 - 종료조건: 0 또는 음수 입력
 - print_triangle 함수
 - ✓ 인자는 하나의 정수 x, 반환형은 void
 - ✓ 높이가 x인 삼각형 출력

[실행 결과]

3 -2 ← 입력

* ← 출력

**

[실습]

- 사용자로부터 팩토리얼 값을 입력받은 후, 인자 값으로 함수를 호출하여 계산 된 결과 값을 반환하여 [실행 결과]와 팩토리얼 모양 계산 표시와 결과 값을 출력하시오.

[실행 결과]

5	← 입력
$5*4*3*2*1=120$	← 출력

감사합니다.



감사합니다.

