C 프로그래밍 및 실습

[12주차 2강]
함수(5)

학습 내용

8.5 함수에 배열 전달

8.6 함수와 라이브러리

학습 목표

8.5 함수에 배열을 전달하는 방법을 이해한다.

8.6 라이브러리가 무엇인지 이해한다.

C 프로그래밍 및 실습

8.5 함수에 배열 전달

8.6 함수와 라이브러리

배열의 개별 원소 전달

- 함수의 인자에서 사용된 배열 원소는 수식의 일부일 뿐
  - ✓배열 원소는 일반 변수와 동일하게 취급됨(7장)

```
void print_int( int x ) {    ⇨ 전달되는 값이 정수형이므로 int
   printf("전달된 값 : %d\n", x);
}
int main() {
   int a[3] = { 10, 20, 30 } ;

   print_int( a[1] );              ⇨ a[1]에 저장된 값이 전달됨
   print_int( a[0] + a[2] + 100 ); ⇨ 연산의 결과가 전달됨

   return 0;
}
```

실행 결과

```
전달된 값 : 20
전달된 값 : 140
```

배열 전체 전달 (일차원 배열)

✓형식인자에서 배열 크기 명시해도 의미 없음 → 보통 생략

```
void print_arr( int x[ ] ) { ⇨ 인자의 자료형은 배열로
   int i;                          (크기는 생략)
   printf("전달된 배열 :");
   for( i=0; i < 3 ; ++i )
      printf(" %d", x[i]);     ⇨ 배열의 원소 값 출력
}
int main() {
   int a[3] = { 10, 20, 30 } ;
   print_arr( a );            ⇨ 인자로 배열 전달
   return 0;                     (이름만 씀)
}
```

실행 결과

```
전달된 배열 : 10 20 30
```

배열 크기가 필요한 경우: 별도의 함수 인자로 전달

```
void print_arr( int x[ ], int n ) {  ⇨ 배열과 배열의 크기 전달
   printf("전달된 배열 :");
   for( int i=0; i < n ; ++i )  printf(" %d", x[i]); ⇨ 배열의 원소 출력
   printf("\n");
}
int main() {
   int a[3] = { 10, 20, 30 };
   int b[5] = { 11, 22, 33, 44, 55 };

   print_arr( a , 3 );    ⇨ 배열 a와 a의 크기 전달 (동일한 함수 호출)
   print_arr( b , 5 );    ⇨ 배열 b와 b의 크기 전달 (동일한 함수 호출)

   print_arr( a, sizeof(a)/sizeof(int) ); ⇨ sizeof 연산자를 이용하여
                                          배열 크기 자동 계산
   return 0;
}
```

배열 전체가 전달된 경우, 호출된 함수에서 배열의 값을 바꾸면?

- 호출한 원래 함수의 배열 값도 바뀜

```
void print_arr( int x[ ], int n ) { … }  ⇨ 배열 출력 함수 (생략)

void zero_arr_elem( int x[ ], int i ) {          ⇨ 배열이 전달됨
   x[i] = 0;                                  ⇨ x[i]의 원소 값을 0으로
}

int main() {
   int a[3] = { 11, 22, 33 };

   zero_arr_elem( a , 1 );                    ⇨ 배열 전달
   print_arr( a, sizeof(a)/sizeof(int) );

   return 0;
}
```

실행 결과

```
전달된 배열 : 11 0 33
```

배열 자체를 인자로 전달하기(정리)

- 배열 자체를 인자로 전달할 때는 일반 변수와 몇 가지 다른 점 존재
  - ✓ 형식 인자 형태
  - ✓ 배열 값 변경의 종속성

- 실제 함수 호출 과정은 일반 변수와 동일하나, 표면적으로 달라 보이는 것일 뿐

- 예외적으로 보이는 이유는 배열 자체를 전달한다는 것은
  배열의 주소(?)를 전달하기 때문
  - ✓ 이에 대해서는 9장 포인터 단원에서 학습

C 프로그래밍 및 실습

8.5 함수에 배열 전달

8.6 함수와 라이브러리

라이브러리

- 함수들을 구현해 모아 놓은 것
- 필요 시 함수를 호출하여 사용

표준 라이브러리

- C언어에서 정해놓은 표준 함수들로 구성: printf(), scanf() 등

표준함수 사용

- 함수의 형태와 기능만 알고 있으면 활용할 수 있음
- 어떻게 구현되어 있는 지는 몰라도 됨
- 다만, 호출하기 전에 함수 원형이 선언되어 있어야 함
- printf(), scanf() 함수의 원형 선언은 어디에 있을까?
  - ✓ 다음 장에서 계속

세종대학교

다음 코드를 컴파일하면?

```
#include <stdio.h>          // 이 부분 삭제

int main()
{
    printf("Hello, World!!\n");
    return 0;
}
```

✓ printf() 함수를 찾을 수 없다고 컴파일 오류 발생

- #include 문은 stdio.h 파일을 소스코드에 포함시키라는 의미
- printf() 함수의 원형은 stdio.h 파일에 선언되어 있음
- stdio.h 를 헤더파일이라 부름 (확장자 .h)

표준 함수와 헤더파일

- 표준 함수를 사용하기 위해서는 적절한 헤더 파일을 #include 문을 이용해 소스 코드에 포함시켜야 함

자주 사용되는 C 표준 헤더 파일 및 표준 함수

- 일부 함수는 뒷장에서 학습
- 자세한 내용은 개발 툴의 도움말이나 C 표준 문서 참고

헤더파일	포함된 함수의 기능	표준 함수
stdio.h	입력, 출력, 파일	printf, scanf, putc, getc, fopen 등
stdlib.h	숫자변환, 동적 할당	atoi, rand, srand, malloc, free 등
ctype.h	문자 검사 및 변환	isalnum, isalpha, islower, toupper 등
math.h	수학 함수	sin, asin, exp, log, pow, sqrt, abs 등
time.h	시간 처리	clock, time, difftime 등
string.h	문자열, 메모리 블록	strcpy, strcat, strcmp, strlen, memcpy 등

학습 정리

- 함수에 **배열을 전달**하기 위해서는 배열 이름과 배열 크기를 전달함
- 호출된 함수에서 배열의 값을 변경하면 호출한 함수의 배열 값도 바뀜
- 함수들을 구현해 모아 놓은 것을 **라이브러리**라 하고,
  다양한 표준 함수가 C 표준 라이브러리로 제공됨