

Chapter 11

DSLAB

Example

- 과제에서 표현하는 Code 캡처와 실행화면 캡처에 대한 설명입니다.
 - Cygwin terminal 을 캡처 후 이미지를 함께 보내시면 됩니다.
 - 과제 번호와 알파벳으로 이름을 저장하여 코드와 같이 압축하여 보내주세요.
 - ex) 8_1_a.png
 - Alt + Printscreen (insert 위에 있는 키) 를 누르면 현재 포커스 되어있는 프 로그 램의 화면을 캡처할 수 있습니다.
 - 그림판에 붙여넣기를 한 뒤 저장하면 됩니다.

➤ Code 캡처

```

1 #include <stdio.h>
2 int all_files;
3 static int this_file;
4 extern void sub();
5
6 int main(void){
7     sub();
8     printf("%d\n", all_files);
9     return 0;
10 }
    
```

➤ 실행화면 캡처

```

DSLab@DESKTOP-10J0DED ~
$ ls
8_5f.exe 8_6.exe 8_8.exe a.exe linkage1.c test.c
8_6.c 8_8.c 8_section link.exe linkage2.c test.exe

DSLab@DESKTOP-10J0DED ~
$ ./8_8.exe
f() 호출 전 x=10, y=20
f() x=30, y=40
f() 호출 후 x=10, y=20

DSLab@DESKTOP-10J0DED ~
$
    
```

실습 문제 1

➤ 기본적인 포인터 사용법 (File: 11_1.c)

```
#include <stdio.h>

int main(void)
{
    int i = 100;
    int *pi = NULL;

    pi = &i;    // ①

    // ②
    printf("i = %d\n", i);
    printf("i = %d\n", *pi);

    printf("i의 주소= %d\n", &i);
    printf("i의 주소= %d\n", pi);

    return 0;
}
```

➔ (a) 위의 프로그램을 컴파일하여 실행하고 그 결과를 기록하라.
실행결과

[Blank area for recording execution results]

- ➔ (b) 문장 ①을 주석 처리하고 전체 프로그램을 실행하여 보라. 어떤 결과가 얻어지는가? 다시 원상태로 복구하라.
- ➔ (c) *pi = 200;을 ②번 위치에 삽입하고 실행하여 보라. 어떤 결과가 얻어지는가?
- ➔ (d) pi를 double형 포인터로 선언하고 실행하여 보라. 어떤 결과가 얻어지는가? 포인터와 변수의 자료형이 일치하지 않으면 어떻게 되는가?
- ➔ (e) double형 변수와 포인터에 대해서도 똑같은 실습을 하여 보라.

과제 제출

- (a) 실행화면 캡처
- (e) 11_1.c (code)

실습 문제 2

➤ 포인터와 배열 (File: 11_2.c)

```
#include <stdio.h>
#define SIZE 10

int main(void)
{
    int array[SIZE];
    int i;
    for(i = 0; i < SIZE; i++)
        printf("%d번째 원소의 주소: %d\n", i, &array[i]);

    return 0;
}
```

· 배열 원소들의 주소를 화면에 출력하여 보자.

- (a) 위의 프로그램을 컴파일하여 실행하고 그 결과를 기록하라. 결과에서 알 수 있는 것은 무엇인가?

실행결과



- (b) array를 %d 형식 지정자로 출력하여 보자. 어떤 배열 원소의 주소와 일치하는가?
- (c) 배열 array를 char형, float형, double형으로 차례대로 변경하여 결과를 기록하라. 어떤 차이가 있는가?

과제 제출

(a, b) 실행화면 캡처

(c) 11_2.c (code)

실습 문제 3

➤ 포인터 연산 (File: 11_3.c)

```
#include <stdio.h>

int main(void)
{
    int array[10];
    int *pi = &array[3];

    printf("pi-1 = %d\n", pi - 1);
    printf("pi = %d\n", pi );
    printf("pi+1 = %d\n", pi + 1);

    return 0;
}
```

과제 제출

(a, c, d) 실행화면 캡처

(c) 11_3.c (code)

· 포인터 연산은 기존의 산술 연산과는 조금 다르다. 실습을 통하여 살펴보자.

- ➔ (a) 위의 프로그램을 컴파일하여 실행하고 그 결과를 기록하라. 결과에서 알 수 있는 것은 무엇인가?

실행결과



- ➔ (b) pi - 2와 pi + 2도 예측하여 보고 실습을 통하여 확인하라.
- ➔ (c) pi++와 pi--도 예측하여 보고 실습을 통하여 확인하라.
- ➔ (c) 배열과 포인터를 모두 char형으로 변경하고 실습을 반복하라. 어떤 결과가 얻어지는가?
- ➔ (d) 배열과 포인터를 모두 double형으로 변경하고 실습을 반복하라. 어떤 결과가 얻어지는가?
- ➔ (e) 다음과 같은 연산들을 테스트하여 보라.

```
*p++;
(*p)++;
*--p;
--(*p);
```

실습 문제 4

➤ 배열과 함수 (File: 11_4.c)

```

#include <stdio.h>
#define SIZE 5
int get_largest(int array[],int n);

int main(void)
{
    int i;
    int score[SIZE];

    for(i = 0; i < SIZE; i++)
    {
        printf("정수를 입력하시오:");
        scanf("%d", &score[i]);
    }

    printf("최대값은 %d입니다\n", get_largest(score, SIZE));

    return 0;
}

int get_largest(int array[], int n)
{
    int i;
    int largest = array[0];

    for(i = 1; i < n; i++)
        if( array[i] > largest ) // ①
            largest = array[i]; // ②

    return largest;
}

```

· 배열이 함수의 인수로 전달될 때에는, 배열을 가리키는 포인터가 전달되는 것이나 마찬가지이다.

➡ (a) 위의 프로그램을 컴파일하여 실행하고 그 결과를 기록하라.
실행결과

➡ (b) ①과 ②를 배열의 인덱스가 아닌 포인터 표기법으로 바꾸어서 작성하여 보라. 즉 array[i] 대신에 *(array+i)를 사용하여 수정하여 보라.

➡ (c) 함수의 원형 정의와 함수 헤더를 다음과 같이 변경하여 실행하여 보라. 어떤 차이가 있는가?

```
int get_largest(int *array, int n);
```

과제 제출

(a) 실행화면 캡처

(b) Code 캡처

(c) 11_4.c (code)

과제 제출

▪ 과제 제출 & 포맷

- E-mail: dslab.pp@gmail.com
- A반: 목요일, 금요일 (11:00~13:00) 602호
- B반: 목요일, 금요일 (15:30~17:30) 402호
- 메일 제목: [프프#반] 학번_이름_실습5
 - [프프#B] 201111339_김민우_실습5
- 과제 파일을 메일 제목과 동일하게 압축하여 제출
 - [프프#A] 201111339_김민우_실습5.zip

▪ 제출 일자

- 4월 12일 수요일 23:59 (기한 엄수)

▪ 과제 제출 파일 List (마지막 항목까지 수행한 파일 제출)

- Code(c 파일): (11_1, 11_2, 11_3, 11_4).c
- 이미지 파일: 실습 문제의 Code 및 실행화면 캡처 이미지
- C 파일과 이미지 파일을 압축하여 포맷에 맞게 명명하여 제출