실 습 문 제 제11장

1. [기본적인 포인터 사용법]

```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

(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.

2. [포인터와 배열] 배열 원소들의 주소를 화면에 출력하여 보자.

```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;
}
```

(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?

3. [포인터 연산] 포인터 연산은 기존의 산술 연산과는 조금 다르다. 실습을 통하여 살펴보자.

```
#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;
}
```

(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);
```

4. **[배열과 함수] 배열이 함수의 인수로 전달될 때에는, 배열을 가리키는 포인터가 전달되는 것이나 마찬가지이다.**

```
#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;
}
```

(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);
```