

1. 다음 포인터 변수들의 정보를 작성하세요.(첫번째 작성한 내용을 참조할 것)

	타입	값 / 주소값	개수	메모리크기	차수
int * data[2];	Int	주소값	2	8	2
char* p1;					
char **p2;					
int *p3[5];					
double *p4;					

2. 주석에 있는 코드를 작성하시오.

```
//정수를 1개 저장할 수 있는 num 이라는 이름의 변수 선언
//num에 사용자 입력을 통해 값 저장
//정수의 주소를 저장할 수 있는 pnum이라는 이름의 변수 선언
//pnum 에 num의 주소를 저장
//num의 값과 num의 주소값 출력
//pnum의 값과 pnum의 주소값 pnum의 역참조값을 출력
//pnum의 역참조 문법을 이용해서 num의 값은 50으로 수정
//num의 값 출력

// 문자를 5개 저장할 수 있는 arr 배열변수를 선언하고, 선언과 동시에 a,b,c,d, e 의 문자로 초기화
//문자의 주소를 저장할 수 있는 pch1 이라는 이름의 변수 선언
//문자의 주소를 저장할 수 있는 pch2 이라는 이름의 변수 선언
//pch1에 arr 배열의 주소를 저장
//pch2에 arr 배열의 인덱스 2의 주소를 저장
//arr 와 pch1를 이용하여 arr에 있는 b 문자를 각각 출력
//arr 와 pch2를 이용하여 arr에 있는 a 문자를 각각 출력
//pch1을 이용하여 arr에 저장된 d 를 대문자 D로 변경
//pch2를 이용하여 arr에 저장된 e 를 소문자 E로 변경
//arr를 이용하여 배열 전체 문자를 출력(for문 사용)
```

3. 메모리 그림을 그리고, 출력 결과를 적으시오

```
int a[] = {2, 3, 4, 5, 6, 8};
1)    printf( "%d\n", sizeof(a) );           //
2)    printf( "%d\n", sizeof (a[0]) );       //
3)    printf( "%d\n", (*a + 5) );             //
4)    printf( "%d\n", *(a + 4) );             //
5)    printf( "%d\n", *a );                   //
```

4. 메모리 그림을 그리고, 출력 결과를 적으시오.

```
int arr[5] = {10,20,30,40,50};
int *ip = arr;
1)    printf("%d\n", arr);                    //출력 100 이라고 가정
2)    printf("%d\n", ip);                     //
3)    printf("%d\n", ip + 2 );                //
4)    printf("%d\n", ip[2]);                  //
5)    printf("%d\n", *(arr+1));               //
```

5. 아래 코드를 작성하시오.(순차적으로 함수를 작성하고, main 에서 호출하는 코드)

정수 10 개를 저장할 수 있는 arr 라는 이름의 변수를 함수 밖에 선언한다.

```
ex)
#include <stdio.h>

//여기에 선언

void main()
{
}
```

반복문을 이용하여 arr 의 모든 값들을 0 으로 초기화 하는 함수를 구현한다.(함수명 initarr)

그리고, main 함수에서 해당 함수를 호출한다.

```
#include <stdio.h>

//여기에 선언

//여기에 initarr 함수의 선언부

void main()
{
    //여기서 initarr 함수 호출
}

//여기에 initarr 함수의 구현부
```

printarr 라는 이름의 함수를 만들고, 배열의 모든 값들을 한 줄로 출력한다.

그리고, main 함수에서 initarr 함수의 호출부 아래에서 printarr 함수를 호출한다.

Insertarr 라는 이름의 함수를 만들고, 사용자로부터 저장인덱스, 저장값을 입력받아 arr 의 인덱스에 저장값을 저장한다.

그리고, printarr 함수 아래에서 호출하고, 다시 printarr 함수를 호출하여 결과를 확인한다.

deletearr 라는 이름의 함수를 만들고, 사용자로부터 배열의 인덱스를 입력받고, arr 의 인덱스에 있는 값을 0 으로 변경한다.

그리고 printarr 함수를 호출하여 결과를 확인한다.

마지막으로 main 함수에 아래와 같은 코드를 구현하여 실행시킨다.

```
void main() {
    int idx;
    while(1) {
        printf("[1]입력 [2]전체출력 [3] 삭제 [4] 종료\n");
        scanf_s("%d", &idx);
        switch(idx)
        {
            case 1: Insertarr(); break;
            case 2: printarr(); break;
            case 3: deletearr(); break;
            case 4: return;
        }
    }
}
```