



# 12 주차 과제

\* 디지털 문서로 작성하여 제출 합니다.

한글문서(hwp), 워드 문서(doc), 파워포인트(ppt), 등으로 작성하여 제출.

\* 다음 페이지 설문은 자유롭게 작성하여 제출 합니다.

\* 문제와 풀이를 작성합니다.

## 프로그램 작성 문제는

1. 문제분석( 분석, 설계, 알고리즘 등을 글로 설명 .또는 ,순서도 등으로 작성하여 설명 )

- 최소 글로 2줄 이상 작성하면 됩니다.

2. 프로그램 소스 + 각 라인 주석 설명

3. 실행화면 캡처. 첨부.

4. 프로그램 동작 설명.

- 작성 프로그램은 이렇게 이렇게 동작 한다. 라고 설명 하면 됩니다.

- 최소 글로 2줄 이상 작성하면 됩니다.

\* 제출할 파일 이름은 아래의 규칙을 따릅니다..

예) 11주차 과제의 경우 : 해당주차\_본인이름.hwp

11\_홍길동.hwp

여러 번 제출 하였을 경우 해당주차\_본인이름\_제출회차.hwp

11\_홍길동\_3.hwp



# 레포트

1

빈칸 채우기.

```
#define SIZE 3
int get_array_sum(int *p, int n);
int main(void)
{
    int expenses[SIZE] = { 100, 200, 200 };
    printf("%d\n", get_array_sum(expenses, SIZE));
}
int get_array_sum(int *p, int n)
{
    int i, result=0;
    for(i=0; i<n; i++)
        result += p[i];
    return result;
}
```

배열의 이름은 배열의 주소

포인터를 통하여 배열의 주소를 받는다.

포인터를 배열처럼 사용할 수 있다.

- ▶ 메모리는 \_\_\_\_\_를 기준으로 주소가 매겨진다.
- ▶ 포인터는 \_\_\_\_\_을 저장할 수 있는 변수이다.
- ▶ 변수 x의 주소를 추출하려면 \_\_\_\_\_라고 하면 된다.
- ▶ \*p의 의미는 \_\_\_\_\_이다.
- ▶ int형 포인터 p가 가리키는 위치에 100을 저장하는 문장은 \_\_\_\_\_이다.
- ▶ 포인터가 아무것도 가리키고 있지 않는 경우에는 \_\_\_\_\_값을 넣어두는 편이 좋다.
- ▶ 배열 a에서 a는 \_\_\_\_\_의 주소이다.
- ▶ p가 포인터라면 p[2]는 수식 \*(p+\_\_\_\_)와 같다.
- ▶ \*p++의 의미는 \_\_\_\_\_이다.
- ▶ 사칙 연산 중에서 포인터에 대하여 적용할 수 있는 연산에는 \_\_\_\_\_, \_\_\_\_\_이 있다.
- ▶ int형 포인터 p가 80번지를 가리키고 있었다면 (p+1)은 \_\_\_\_\_번지를 가리킨다.
- ▶ 함수 호출시 인수 전달 방법 중에서 기본적인 방법은 "\_\_\_\_\_에 의한 호출"이다.

```

#define SIZE 3
int get_array_sum(int *p, int n);
int main(void)
{
    int expenses[SIZE] = { 100, 200, 200 };
    printf("%d\n", get_array_sum(expenses, SIZE));
}
int get_array_sum(int *p, int n)
{
    int i, result=0;
    for(i=0; i<n; i++)
        result += p[i];
    return result;
}

```

배열의 이름은 배열의 주소

포인터를 통하여 배열의 주소를 받는다.

포인터를 배열처럼 사용할 수 있다.

- ▶ 메모리는 \_\_\_\_\_를 기준으로 주소가 매겨진다.
- ▶ 포인터는 \_\_\_\_\_을 저장할 수 있는 변수이다.
- ▶ 변수 x의 주소를 추출하려면 \_\_\_\_\_라고 하면 된다.
- ▶ \*p의 의미는 \_\_\_\_\_이다.
- ▶ int형 포인터 p가 가리키는 위치에 100을 저장하는 문장은 \_\_\_\_\_이다.
- ▶ 포인터가 아무것도 가리키고 있지 않는 경우에는 \_\_\_\_\_값을 넣어두는 편이 좋다.
- ▶ 배열 a에서 a는 \_\_\_\_\_의 주소이다.
- ▶ p가 포인터라면 p[2]는 수식 \*(p+\_\_\_\_)와 같다.
- ▶ \*p++의 의미는 \_\_\_\_\_이다.
- ▶ 사칙 연산 중에서 포인터에 대하여 적용할 수 있는 연산에는 \_\_\_\_\_, \_\_\_\_\_이 있다.
- ▶ int형 포인터 p가 80번지를 가리키고 있었다면 (p+1)은 \_\_\_\_\_번지를 가리킨다.
- ▶ 함수 호출시 인수 전달 방법 중에서 기본적인 방법은 "\_\_\_\_\_에 의한 호출"이다.

- 바이트
- 주소
- &x
- \*p = 100;
- NULL
- 첫 번째 원소
- 2
- p가 가리키는 위치에서 값을 추출한 후에 p를 증가시키는 것
- +, -
- 84
- 값



# 레포트

2

02 다음 코드의 빈칸에 주석에 알맞은 문장을 넣으시오.

```
char code;
```

```
_____ ; // char형 포인터 p선언
```

```
_____ ; // 포인터에 변수 code의 주소 대입
```

```
_____ ; // 포인터를 통하여 변수 code에 'a' 대입하기
```



## 레포트

01 다음에서 인덱스에 의한 배열 요소 참조는 포인터에 의한 참조로 바꾸고 반대로 포인터에 의한 참조는 인덱스에 의한 참조로 바꾸시오. (a)번은 예이다.

(a) list[6]                      ~~\*(list+6)~~

(c) \*(cost + 8)                      \_\_\_\_\_

(b) name[3]

(d) message[0]

02 다음 코드의 빈칸에 주석에 알맞은 문장을 넣으시오.

```
char code;
```

```
_____ ;    // char형 포인터 p선언
```

```
_____ ;    // 포인터에 변수 code의 주소 대입
```

```
_____ ;    // 포인터를 통하여 변수 code에 'a' 대입하기
```

03 int a[]={10, 20, 30, 40, 50}으로 정의되었다고 가정하자. \*(a+2)의 값은?

① 10

② 20

③ 30

④ 40

1.

(a) \*(list+6)

(b) \*(name+3)

(c) cost[8]

(d) \*(message+0)

2.

```
char *p;
```

```
p = &code;
```

```
*p = 'a';
```

3. (3)



## 레포트

3

04 아래 문장이 실행되었다고 가정하자. 다음 중 다른 문장들과 실행 결과가 다른 것은?

```
int i;  
int *p = &i;
```

① `i = i + 1;`

② `i++;`

③ `*p++;`

④ `*p = *p + 1;`

05 다음 프로그램의 출력은?

```
int x = 6;  
int *p = &x;  
printf("%d\n", --(*p));  
printf("%d\n", (*p)++);
```

06 다음 프로그램의 출력은?

```
int *p = (int *)1000;  
double *q = (double *)2000;  
printf("%d\n", p+2);  
printf("%d\n", q+1);
```



## 레포트

04 아래 문장이 실행되었다고 가정하자. 다음 중 다른 문장들과 실행 결과가 다른 것은?

(3)

```
int i;  
int *p = &i;
```

① `i = i + 1;`

② `i++;`

③ `*p++;`

④ `*p = *p + 1;`

05 다음 프로그램의 출력은?

```
int x = 6;  
int *p = &x;  
printf("%d\n", --(*p));  
printf("%d\n", (*p)++);
```

5  
5

계속하려면 아무 키나 누르십시오 . . .

06 다음 프로그램의 출력은?

```
int *p = (int *)1000;  
double *q = (double *)2000;  
printf("%d\n", p+2);  
printf("%d\n", q+1);
```

1008  
2008

계속하려면 아무 키나 누르십시오 . . .





## 레포트

02 2개의 정수의 합과 차를 동시에 반환하는 함수를 작성하고 테스트하라. 포인터 매개 변수를 사용한다.

```
void get_sum_diff(int x, int y, int *p_sum, int *p_diff) {  
    ...  
}
```

### 실행결과

```
C:\WINDOWS\system32\cmd.exe  
원소들의 합=300  
원소들의 차=-100
```

**HINT** 함수 매개 변수에 포인터를 사용하면 2개 이상의 값을 반환할 수 있다. 본문에서 직선의 기울기와 절편을 반환하는 예제를 참고하라.





## 레포트

02 2개의 정수의 합과 차를 동시에 반환하는 함수를 작성하고 테스트하라. 포인터 매개 변수를 사용한다.

```
void get_sum_diff(int x, int y, int *p_sum, int *p_diff) {  
    ...  
}
```

실행결과

```
C:\WINDOWS\system32\cmd  
원소들의 합=300  
원소들의 차=-100  
<
```

**HINT** 함수 매개 변수에 포인터를 하라.

```
#include <stdio.h>  
void get_sum_diff(int x, int y, int* p_sum, int* p_diff);  
  
int main(void)  
{  
    int sum = 0, diff = 0;  
    get_sum_diff(100, 200, &sum, &diff);  
    printf("원소들의 합=%d\n", sum);  
    printf("원소들의 차=%d\n", diff);  
    return 0;  
}  
  
void get_sum_diff(int x, int y, int* p_sum, int* p_diff)  
{  
    *p_sum = x + y;  
    *p_diff = x - y;  
}
```

제를 참고



```
#include <stdio.h>

void exchange(int *, int *);

int main()
{
    int cheoli=10, metel=20;
    exchange(&cheoli, &metel);
    return 0;
}

void exchange(int *cp, int *mp)
{
    int temp;
    temp=*cp;
    *cp=*mp;
    *mp=temp;
}
```



<11>

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

```
void exchange(int *, int *);
```

```
int main()
```

```
{
```

```
    int cheoli=10, metel=20;
```

```
    exchange(&cheoli, &metel);
```

```
    return 0;
```

```
}
```

```
void exchange(int *cp, int *mp)
```

```
{
```

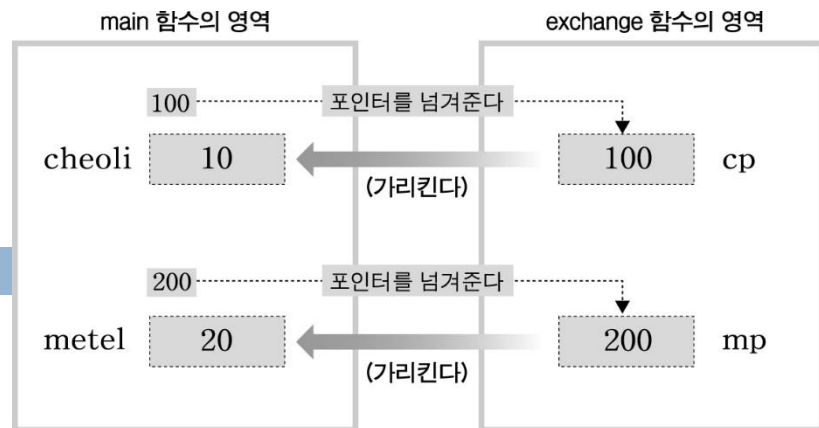
```
    int temp;
```

```
    temp=*cp;
```

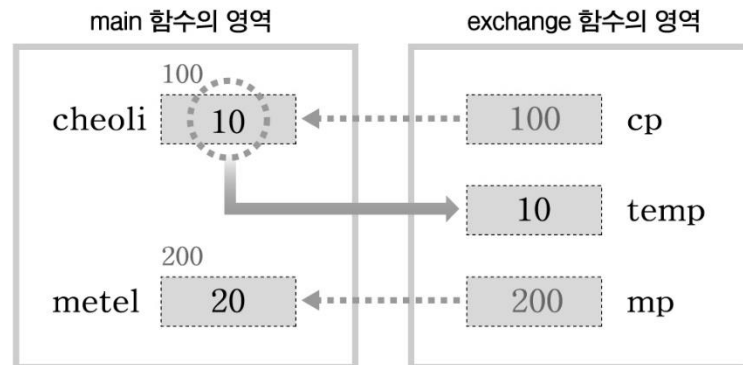
```
    *cp=*mp;
```

```
    *mp=temp;
```

```
}
```



```
temp = *cp;
```



```
*cp = *mp;
```

```
*mp = temp;
```

