

## 컴퓨터정보공학부 2021202058 송채영

문제 1 (소요시간: 10분)

\*주석으로 라인 별 분석을 대신함

소스코드

```
#include <stdio.h>

int* sum(int a, int b); // int형 변수의 주소를 반환하는 함수 선언

int main(void)
{
    int* resp; // 반환값을 저장할 포인터 resp
    resp = sum(10, 20); // resp에 반환된 주소 저장
    printf("두 정수의 합 : %d\n", *resp); // resp가 가리키는 변수의 값 출력
    return 0;
}

int* sum(int a, int b) // int형 변수의 주소를 반환하는 함수
{
    static int res; // 정적 지역 변수 res 선언
    res = a + b; // a + b의 합을 res에 저장
    return &res; // 정적 지역 변수의 주소값 반환
}
```

예상 결과

두 정수의 합 : 30

결과화면

예상결과와 동일하다.

```
Microsoft Visual Studio 디버그 콘솔
두 정수의 합 : 30
C:\Users\82104\source\repos\Project3\x64\Debug\Project3.exe(프로세스 15632개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

## 문제 2 (소요시간: 15분)

### 소스코드

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

```
int main()
```

```
{
```

```
    int a[6] = {5, 10, 15, 20, 25, 30}; // 배열 선언과 동시에 초기화
```

```
    int* p = a;
```

```
    p = a;
```

```
    for (int i = 0; i < 5; i++)
```

```
    {
```

```
        printf(" %d ", *p++); // p가 가르키는 값을 a에 대입후 p 증가 (주소 증가)
```

```
    }
```

```
    printf("\n");
```

```
    p = a;
```

```
    for (int i = 0; i < 5; i++)
```

```
    {
```

```
        printf(" %d ", (* p)++); // p가 가르키는 값을 a에 대입후 p가 가르키는 값 i를  
        증가 (값 증가)
```

```
    }
```

```
    printf("\n");
```

```
    p = a;
```

```
    for (int i = 0; i < 5; i++)
```

```
    {
```

```
        printf(" %d ", *++p); // p를 증가시킨 후에 p가 가르키는 값을 a에 대입 (주소 증  
        가)
```

```
    }
```

```
    printf("\n");
```

```

    p = a;

    for (int i = 0; i < 5; i++)
    {
        printf(" %d ", ++*p); // p가 가르키는 값을 가져오고 그 값을 증가해 a에 대입
(값 증가)
    }

    printf("Wn");

    return 0;
}

```

### 결과화면

```

5 10 15 20 25
5 6 7 8 9
10 15 20 25 30
11 12 13 14 15

C:\Users\#82104\source\repos\Project3\#x64\Debug\Project3.exe(프로세스 21676개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...

```

### 고찰

++\*p에서 마지막 값이 쓰레기 값이 나오므로 실행결과가 제대로 됐는지 확인하기 위해 배열을 `int a[6] = {5, 10, 15, 20, 25, 30};` 다음과 같이 선언 후 초기화 해주었다. 또한 ++\*p의 값이 11부터 시작되는 이유는 두 번째 (\*p)++에서 `a[6] = {10, 15, 20, 25, 30, 35}`로 바뀌었기 때문이다.

### 문제 3 (소요시간: 60분 이상)

#### 소스코드

```

#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>
#include <string.h>

void format_display(char money[]);

int main()
{
    char money[100]; // 문자열 저장

    printf("단위를 뺀 금액 입력하고 Enter > ");

    scanf("%s", money);
}

```

```

        format_display(money);

        return 0;
    }

void format_display(char money[]) // 문자열로 숫자를 입력 받아 천 단위 단위 표시 출력 함수
{
    int i, len;

    const char* str = money; // 길이를 구할 문자열

    len = strlen(str);

    for (i = 0; i < strlen(str); i++) // 3자리 마다 콤마 출력
    {
        printf("%c", money[i]);

        len--; // 3자리 단위로 출력하기 위한 논리

        if (len % 3 == 0 && len != 0) // 콤마 추가
        {
            printf(",");
        }
    }

    return 0;
}

```

## 결과화면



```

Microsoft Visual Studio 디버그 콘솔
단위를 뺀 금액 입력하고 Enter > 234345456
234,345,456
C:\Users\82104\source\repos\Project3\Debug\Project3.exe(프로세스 24816개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...

```

## 고찰

문자열을 숫자로 입력 받아 천 단위 단위로 출력하는 함수를 짜는데 오래 걸렸다. 처음 문자열로 입력 받지 않고 코드를 짜고, 3자리 단위로 출력하는 부분에서 어떻게 구현해야 할지 생각하다 보니 시간이 더욱 오래 걸렸던 것 같다. 단위를 뒤에서부터 출력하는 것을 len 변수를 통해 해결하였다.