

[구조체 실습1] student 구조체

※ student 구조체를 이용한 다음의 프로그램을 실행 예를 참고하여 완성하십시오.

```
#include <stdio.h>
#define MAX_STD 5
```

```
struct student {
    char name[20];
    int korean, english, math;
    double average;
};
```

```
int main(void)
{
    struct student std[MAX_STD];
    int i;
    double total_average = 0;
```

```
    printf("%d 명의 학생 정보를 입력하세요.\n", MAX_STD);
```

// 학생 정보 입력 및 평균 계산

```
    total_average /= MAX_STD;
```

// 전체 평균 계산

```
    printf("\n이름      국어 영어 수학 평균\n");
```

// 학생 정보 출력(실행 예 참고)

```
    printf("전체 평균 : %.2f\n", total_average);
```

```
    return 0;
```

```
}
```

```
C:\WINDOWS\system32\cmd.exe
국어, 영어, 수학 점수 : 90 28 71
이름 : 고길동
국어, 영어, 수학 점수 : 30 100 65
이름 : 돌리
국어, 영어, 수학 점수 : 88 71 39
이름 : 도우너
국어, 영어, 수학 점수 : 93 33 56
이름 : 또치
국어, 영어, 수학 점수 : 100 71 83

이름      국어 영어 수학 평균
고길동    90  28  71  63.00
고길동    30 100  65  65.00
돌리      88  71  39  66.00
도우너    93  33  56  60.67
또치     100  71  83  84.67
전체 평균 : 67.87
계속하려면 아무 키나 누르십시오 . . .
```

[구조체 실습2] 구조체를 다루는 함수

※ 다음의 요구사항에 맞게 구조체를 다루는 함수를 정의하여 프로그램을 완성하시오(단, 제시된 함수의 원형은 변경할 수 없다).

-
- 1) void input_student(struct student *s);
- student 구조체 변수의 주소를 전달받아 이름, 국, 영, 수 점수를 입력받고, 평균을 계산해주는 함수
 - 2) void output_student(struct student s);
- student 구조체 변수를 전달받아 이름, 국, 영, 수, 평균을 출력하는 함수
 - 3) double Compute_class_avg(struct student p[], int size);
- student 구조체 배열과 배열의 크기를 전달받아 전체 평균을 계산하여 반환해주는 함수
 - 4) int Find_First(struct student p[], int size);
- student 구조체 배열과 배열의 크기를 전달받아 1등 학생의 인덱스를 반환해주는 함수
-

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

```
#define MAX_STD 5
```

```
struct student {
    char name[20];
    int korean, english, math;
    double average;
};

void input_student(struct student *s);
void output_student(struct student s);
double compute_class_avg(struct student p[], int size);
int Find_First(struct student p[], int size);
```

```
int main(void)
```

```
{
    struct student std[MAX_STD];
    int i, first;
    double total_average = 0;
```

```
    printf("%d 명의 학생 정보를 입력하세요.\n", MAX_STD);
```

```
    for( i = 0 ; i < MAX_STD ; i++ )
```

```
         // input_student() 함수 호출문
```

```
    printf("\n이름      국어 영어 수학   평균\n");
```

```
    for( i = 0 ; i < MAX_STD ; i++ )
```

```
         // output_student() 함수 호출문
```

```
    // compute_class_avg() 함수 호출문
```

```
    
```

```
    printf("\n전체 평균 : %.2f\n", total_average);
```

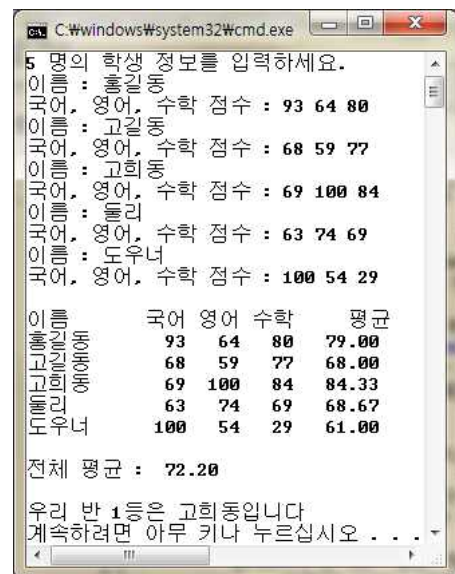
```
    // Find_First() 함수 호출문
```

```
    
```

```
    printf("\n우리 반 1등은 %s입니다\n", std[first].name);
```

```
    return 0;
```

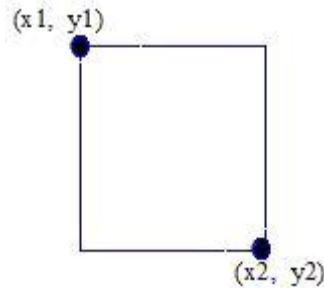
```
}
```



[구조체 실습3]: Rectangle 구조체1

사각형에 대한 구조체 **Rectangle**은 다음과 같이 사각형의 대각끝점으로 정의된다. 다음의 요구사항에 맞도록 프로그램을 작성해봅시다 (여러 번 누누히 강조하지만, 요구사항을 정확히 이해하시고 요구사항에 맞는 코드를 작성하셔야 합니다).

```
typedef struct point{
    int x, y;
} point_t;
typedef struct rectangle{
    point_t p1;
    point_t p2;
} rect_t;
```



* 프로그램 요구 사항 *

=====

- main() 함수

- 1) **rectangle** 구조체 변수를 선언한다.
- 2) 사용자로부터 직사각형의 두 점 좌표를 입력받는다.
- 3) **computeArea()** 함수를 호출하여 사각형의 면적을 계산하여 출력한다.
- 4) **isSquare()** 함수를 호출하여 정사각형 여부를 검사한 결과를 출력한다.

- double ComputeArea(rect_t r);

- 1) **computeArea()** 함수는 직사각형 구조체 변수를 매개변수로 전달받아, 직사각형의 면적을 계산하여 반환하는 함수이다(함수의 원형 변경 불가)

- int isSquare(const rect_t *p)

- 1) **isSquare()** 함수는 사각형 구조체를 하나 전달받아, 정사각형인지를 검사하여 정사각형이면 1을, 직사각형이면 0을 반환하는 함수이다(함수의 원형 변경 불가)

=====

다음은 실행 예입니다.

```
C:\WINDOWS\system32\cmd.exe
사각형의 대각 점 좌표 입력
첫번째 점 좌표: 1 6
두번째 점 좌표: 4 10

사각형의 면적: 12.00

입력하신 사각형은 직사각형입니다.
계속하려면 아무 키나 누르십시오 . . .
```

```
C:\WINDOWS\system32\cmd.exe
사각형의 대각 점 좌표 입력
첫번째 점 좌표: -2 5
두번째 점 좌표: 4 -1

사각형의 면적: 36.00

입력하신 사각형은 정사각형입니다.
계속하려면 아무 키나 누르십시오 . . .
```