



---

# R E P O R T

## C프로그래밍2 과제6

과목명	C 프로그래밍 II
분반	2 분반
교수	정 구 철
학번	2020136129
이름	최 수 연
제출일	2020년 10월 24일 토요일

구조체 - 6th

1. point 구조체를 만들고 (2, 3)으로 초기화시키고 출력하시오

☐ 구조체에 들어가는 변수는 int x, int y

```
#include <stdio.h>

struct point {
    int x;
    int y;
};

int main(void)
{
    struct point s = { 2, 3 };
    printf("%d, %d\n", s.x, s.y);
    return 0;
}
```

2, 3

2. 1번에서 만든 point구조체를 이용해서 사각형을 나타내는 rect 구조체를 만드시오. 좌표는 (2,3), (20,30) 그리고 rect구조체의 내용물을 출력하시오

□ point p1,p2를 이용해 구조체를 만들면됨

```
#include <stdio.h>

struct point {
    int x;
    int y;
};

struct rect {
    struct point p1;
    struct point p2;
};

int main(void)
{
    struct rect r;
    r.p1.x = 2;
    r.p1.y = 3;
    r.p2.x = 20;
    r.p2.y = 30;
    printf("p1 = (%d, %d)\np2 = (%d, %d)\n", r.p1.x, r.p1.y, r.p2.x, r.p2.y);
    return 0;
}
```

```
p1 = (2, 3)
p2 = (20, 30)
```

3. 2에서 만든 사각형의 둘레를 구하시오

```
#include <stdio.h>

struct point {
    int x;
    int y;
};

struct rect {
    struct point p1;
    struct point p2;
};

int main(void)
{
    struct rect r;
    int w, h, peri;
    r.p1.x = 2;
    r.p1.y = 3;
    r.p2.x = 20;
    r.p2.y = 30;
    w = r.p2.x - r.p1.x;
    h = r.p2.y - r.p1.y;
    peri = 2 * w + 2 * h;
    printf("둘레는 %d입니다.\n", peri);
    return 0;
}
```

둘레는 90입니다.

4. 2에서 만든 사각형의 넓이를 구하시오

```
#include <stdio.h>

struct point {
    int x;
    int y;
};

struct rect {
    struct point p1;
    struct point p2;
};

int main(void)
{
    struct rect r;
    int w, h, area;
    r.p1.x = 2;
    r.p1.y = 3;
    r.p2.x = 20;
    r.p2.y = 30;
    w = r.p2.x - r.p1.x;
    h = r.p2.y - r.p1.y;
    area = w * h;
    printf("넓이는 %d입니다.\n", area);
    return 0;
}
```

넓이는 486입니다.

5. 2에서 만든 사각형을 A라고 하자. (2,3), (20,31)의 좌표를 가지는 사각형 B를 만들고 A와 B가 같은 사각형인지 확인하시오

```
#include <stdio.h>

struct point {
    int x;
    int y;
};
struct rectA {
    struct point p1;
    struct point p2;
};
struct rectB {
    struct point p1;
    struct point p2;
};
int main(void)
{
    struct rectA rA;
    struct rectB rB;
    rA.p1.x = 2;
    rA.p1.y = 3;
    rA.p2.x = 20;
    rA.p2.y = 30;
    rB.p1.x = 2;
    rB.p1.y = 3;
    rB.p2.x = 20;
    rB.p2.y = 31;
    if ((rA.p1.x == rB.p1.x) && (rA.p1.y == rB.p1.y) && (rA.p2.x == rB.p2.x)
    && (rA.p2.y == rB.p2.y))
        printf("두 사각형은 같은 사각형입니다.\n");
    else
        printf("두 사각형은 같은 사각형이 아닙니다.\n");
    return 0;
}
```

두 사각형은 같은 사각형이 아닙니다.

6. point 구조체 배열을 만들어서 2번에서 만든 사각형 A 안의(테두리 포함) 모든 좌표(int라 서 유한함)를 담고 출력하시오.

□ 배열의 크기는 자유

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

```
struct point {  
    int x;  
    int y;  
};
```

```
struct rect {  
    struct point p1;  
    struct point p2;  
    struct point p3[500];  
};
```

```
int main(void)
```

```
{
```

```
    int a = 0;
```

```
    struct rect r;
```

```
    r.p1.x = 2;
```

```
    r.p1.y = 3;
```

```
    r.p2.x = 20;
```

```
    r.p2.y = 30;
```

```
    for (int i = r.p1.x; i <= r.p2.x; i++)
```

```
    {
```

```
        for (int j = r.p1.y; j <= r.p2.y; j++, a++)
```

```
        {
```

```
            r.p3[a].x = i;
```

```
            r.p3[a].y = j;
```

```
            printf("(%d, %d)", r.p3[a].x, r.p3[a].y);
```

```
        }
```

```
    }
```

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

```
    return 0;
```

```
}
```

```
(2, 3)(2, 4)(2, 5)(2, 6)(2, 7)(2, 8)(2, 9)(2, 10)(2, 11)(2, 12)(2, 13)(2, 14)(2, 15)(2, 16)(2, 17)(2, 18)(2, 19)(2, 20)(2, 21)(2, 22)(2, 23)(2, 24)(2, 25)(2, 26)(2, 27)(2, 28)(2, 29)(2, 30)(3, 3)(3, 4)(3, 5)(3, 6)(3, 7)(3, 8)(3, 9)(3, 10)(3, 11)(3, 12)(3, 13)(3, 14)(3, 15)(3, 16)(3, 17)(3, 18)(3, 19)(3, 20)(3, 21)(3, 22)(3, 23)(3, 24)(3, 25)(3, 26)(3, 27)(3, 28)(3, 29)(3, 30)(4, 4)(4, 5)(4, 6)(4, 7)(4, 8)(4, 9)(4, 10)(4, 11)(4, 12)(4, 13)(4, 14)(4, 15)(4, 16)(4, 17)(4, 18)(4, 19)(4, 20)(4, 21)(4, 22)(4, 23)(4, 24)(4, 25)(4, 26)(4, 27)(4, 28)(4, 29)(4, 30)(5, 5)(5, 6)(5, 7)(5, 8)(5, 9)(5, 10)(5, 11)(5, 12)(5, 13)(5, 14)(5, 15)(5, 16)(5, 17)(5, 18)(5, 19)(5, 20)(5, 21)(5, 22)(5, 23)(5, 24)(5, 25)(5, 26)(5, 27)(5, 28)(5, 29)(5, 30)(6, 6)(6, 7)(6, 8)(6, 9)(6, 10)(6, 11)(6, 12)(6, 13)(6, 14)(6, 15)(6, 16)(6, 17)(6, 18)(6, 19)(6, 20)(6, 21)(6, 22)(6, 23)(6, 24)(6, 25)(6, 26)(6, 27)(6, 28)(6, 29)(6, 30)(7, 7)(7, 8)(7, 9)(7, 10)(7, 11)(7, 12)(7, 13)(7, 14)(7, 15)(7, 16)(7, 17)(7, 18)(7, 19)(7, 20)(7, 21)(7, 22)(7, 23)(7, 24)(7, 25)(7, 26)(7, 27)(7, 28)(7, 29)(7, 30)(8, 8)(8, 9)(8, 10)(8, 11)(8, 12)(8, 13)(8, 14)(8, 15)(8, 16)(8, 17)(8, 18)(8, 19)(8, 20)(8, 21)(8, 22)(8, 23)(8, 24)(8, 25)(8, 26)(8, 27)(8, 28)(8, 29)(8, 30)(9, 9)(9, 10)(9, 11)(9, 12)(9, 13)(9, 14)(9, 15)(9, 16)(9, 17)(9, 18)(9, 19)(9, 20)(9, 21)(9, 22)(9, 23)(9, 24)(9, 25)(9, 26)(9, 27)(9, 28)(9, 29)(9, 30)(10, 10)(10, 11)(10, 12)(10, 13)(10, 14)(10, 15)(10, 16)(10, 17)(10, 18)(10, 19)(10, 20)(10, 21)(10, 22)(10, 23)(10, 24)(10, 25)(10, 26)(10, 27)(10, 28)(10, 29)(10, 30)(11, 11)(11, 12)(11, 13)(11, 14)(11, 15)(11, 16)(11, 17)(11, 18)(11, 19)(11, 20)(11, 21)(11, 22)(11, 23)(11, 24)(11, 25)(11, 26)(11, 27)(11, 28)(11, 29)(11, 30)(12, 12)(12, 13)(12, 14)(12, 15)(12, 16)(12, 17)(12, 18)(12, 19)(12, 20)(12, 21)(12, 22)(12, 23)(12, 24)(12, 25)(12, 26)(12, 27)(12, 28)(12, 29)(12, 30)(13, 13)(13, 14)(13, 15)(13, 16)(13, 17)(13, 18)(13, 19)(13, 20)(13, 21)(13, 22)(13, 23)(13, 24)(13, 25)(13, 26)(13, 27)(13, 28)(13, 29)(13, 30)(14, 14)(14, 15)(14, 16)(14, 17)(14, 18)(14, 19)(14, 20)(14, 21)(14, 22)(14, 23)(14, 24)(14, 25)(14, 26)(14, 27)(14, 28)(14, 29)(14, 30)(15, 15)(15, 16)(15, 17)(15, 18)(15, 19)(15, 20)(15, 21)(15, 22)(15, 23)(15, 24)(15, 25)(15, 26)(15, 27)(15, 28)(15, 29)(15, 30)(16, 16)(16, 17)(16, 18)(16, 19)(16, 20)(16, 21)(16, 22)(16, 23)(16, 24)(16, 25)(16, 26)(16, 27)(16, 28)(16, 29)(16, 30)(17, 17)(17, 18)(17, 19)(17, 20)(17, 21)(17, 22)(17, 23)(17, 24)(17, 25)(17, 26)(17, 27)(17, 28)(17, 29)(17, 30)(18, 18)(18, 19)(18, 20)(18, 21)(18, 22)(18, 23)(18, 24)(18, 25)(18, 26)(18, 27)(18, 28)(18, 29)(18, 30)(19, 19)(19, 20)(19, 21)(19, 22)(19, 23)(19, 24)(19, 25)(19, 26)(19, 27)(19, 28)(19, 29)(19, 30)(20, 20)(20, 21)(20, 22)(20, 23)(20, 24)(20, 25)(20, 26)(20, 27)(20, 28)(20, 29)(20, 30)
```

(문제 7~10). 과제 4의 문제 7~10은 구조체를 사용하면 좀 더 맛있는 코드 작성이 가능합니다. 왜 구조체를 사용해야 하는지 느껴보시길 바랍니다.

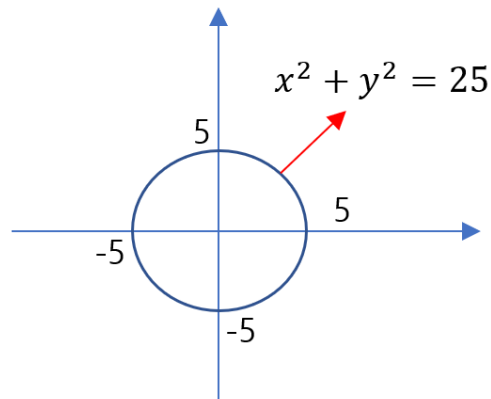


그림1. 반지름이 5이고, 중심 좌표가 (0,0)인 원

그림1과 같이 반지름이 5이고, 중심 좌표가 (0,0)인 원이 있다고 하자.

x 좌표 하나를 넣었을 때, y좌표를 출력하는 함수 `get_single_y`의 원형은 다음과 같이 정의된다.

- `int get_single_y(double x, struct point *y)`

x좌표들을 넣었을 때(배열을 넣었을 때), y 값들을 출력하는 함수 `get_multi_y`의 원형은 다음과 같이 정의된다.

- `int get_multi_y(double *x, struct point *y)`

7. 양수와 음수값을 동시에 받을 구조체 `point`를 정의하시오. 그리고 (1,-1)을 넣고 출력해보시오

```
#include <stdio.h>

struct point
{
    double a, b;
};

int main(void)
{
    struct point p1;
    printf("양수, 음수 입력: ");
    scanf_s("%lf %lf", &p1.a, &p1.b);
    printf("(%lf, %lf)\n", p1.a, p1.b);
    return 0;
}
```

```
양수, 음수 입력: 1 -1
(1.000000, -1.000000)
```



8. get\_single\_y를 구현하시오. 그리고 x는 7을 집어넣었을 때 결과값을 출력하시오.

- ☐ x의 값이 반지름의 범위를 넘었을 때, -1을 반환하여야 함.
- ☐ main에서 함수가 -1을 반환했을 때 "input range is out"이라고 출력되어야함(함수 내에서 출력하는 것 아님)

```
#include <stdio.h>
#include <math.h>
int get_single_y(double x, struct point* y);

struct point
{
    double a, b;
};

int main(void)
{
    double x;
    struct point p1;
    struct point* p2 = &p1;
    printf("x 좌표 입력: ");
    scanf_s("%lf", &x);
    if (get_single_y(x, p2) == -1)
        printf("input range is out\n");
    return 0;
}

int get_single_y(double x, struct point* y)
{
    if (x < -5 || x > 5)
        return -1;
    else
    {
        y->a = -sqrt(25 - x * x);
        y->b = sqrt(25 - x * x);
        printf("(%lf, %lf) or (%lf, %lf)\n", x, y->a, x, y->b);
    }
    return 0;
}
```

```
x 좌표 입력: 7
input range is out
```

9. 8에서 만든 get\_single\_y를 가지고 x에 3을 집어넣었을 때 결과값을 출력하시오.

```
#include <stdio.h>
#include <math.h>
int get_single_y(double x, struct point* y);

struct point
{
    double a, b;
};
int main(void)
{
    double x;
    struct point p1;
    struct point* p2 = &p1;
    printf("x 좌표 입력: ");
    scanf_s("%lf", &x);
    if (get_single_y(x, p2) == -1)
        printf("input range is out\n");
    return 0;
}
int get_single_y(double x, struct point* y)
{
    if (x < -5 || x > 5)
        return -1;
    else
    {
        y->a = -sqrt(25 - x * x);
        y->b = sqrt(25 - x * x);
        printf("(%lf, %lf) or (%lf, %lf)\n", x, y->a, x, y->b);
    }
    return 0;
}
```

```
x 좌표 입력: 3
(3.000000, -4.000000) or (3.000000, 4.000000)
```

10. get\_multi\_y(double \*x, struct point \*y)를 구현하고 결과값을 출력하시오.

- ☐ x의 값이 하나라도 반지름의 범위를 넘으면, -1을 반환하여야 함.
- ☐ 배열 x,y1,y2의 SIZE는 5로 통일
- ☐ x[SIZE] = {1, 1.5, 2, 2.5, 3}로 정의
- ☐ Hint. get\_single\_y는 y->y1 = 1 이런 형태로 값을 넣어야 했지만, y가 구조체 배열인 경우는 조금 다르게 접근할 필요가 있음. 예전에 배운 포인터를 잘 생각해서 어떻게 해야할지 생각하시오

```
#include <stdio.h>
#include <math.h>
#define SIZE 5
int get_multi_y(double* x, struct point* y);
struct point
{
    double a, b;
};
int main(void)
{
    double x[SIZE] = { 1, 1.5, 2, 2.5, 3 };
    struct point y1[SIZE];
    if (get_multi_y(x, y1) == -1)
        printf("input range is out\n");
    return 0;
}
int get_multi_y(double* x, struct point* y)
{
    for (int i = 0; i < 5; i++)
    {
        if (x[i] < -5 || x[i] > 5)
            return -1;
        else
        {
            y[i].a = -sqrt(25 - x[i] * x[i]);
            y[i].b = sqrt(25 - x[i] * x[i]);
            printf("(%.1lf, %.1lf) or (%.1lf, %.1lf)\n", x[i], y[i].a, x[i], y[i].b);
        }
    }
    return 0;
}
```

```
(1.0, -4.898979) or (1.0, 4.898979)
(1.5, -4.769696) or (1.5, 4.769696)
(2.0, -4.582576) or (2.0, 4.582576)
(2.5, -4.330127) or (2.5, 4.330127)
(3.0, -4.000000) or (3.0, 4.000000)
```