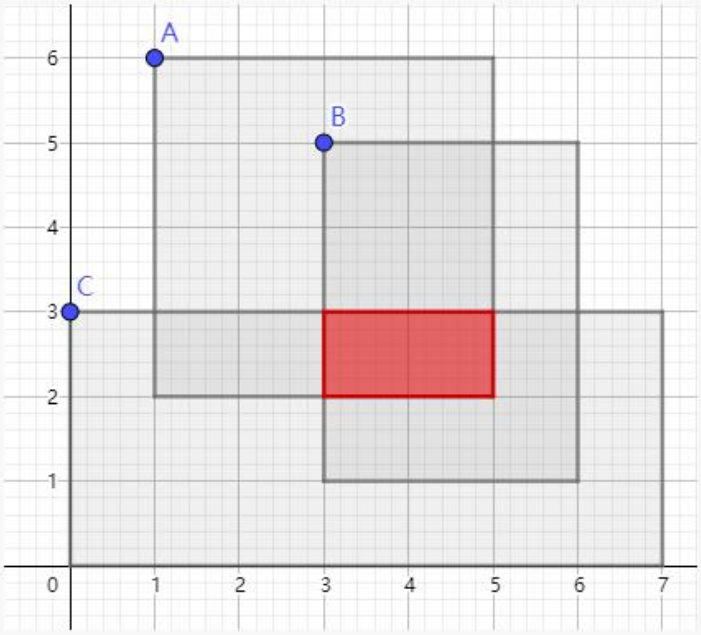
题目描述：在坐标系中，给定3个矩形，求相交区域的面积。




输入描述：3行输入分别为3个矩形的位置，分别代表
'左上角x坐标'，'左上角y坐标'，'矩形宽'，'矩形高'
-1000 <= x,y < 1000

输出描述：输出3个矩形相交的面积，不相交的输出0

补充说明：

示例 1

输入：

```
1 6 4 4
3 5 3 4
0 3 7 3
```

输出：

```
2
```

说明：

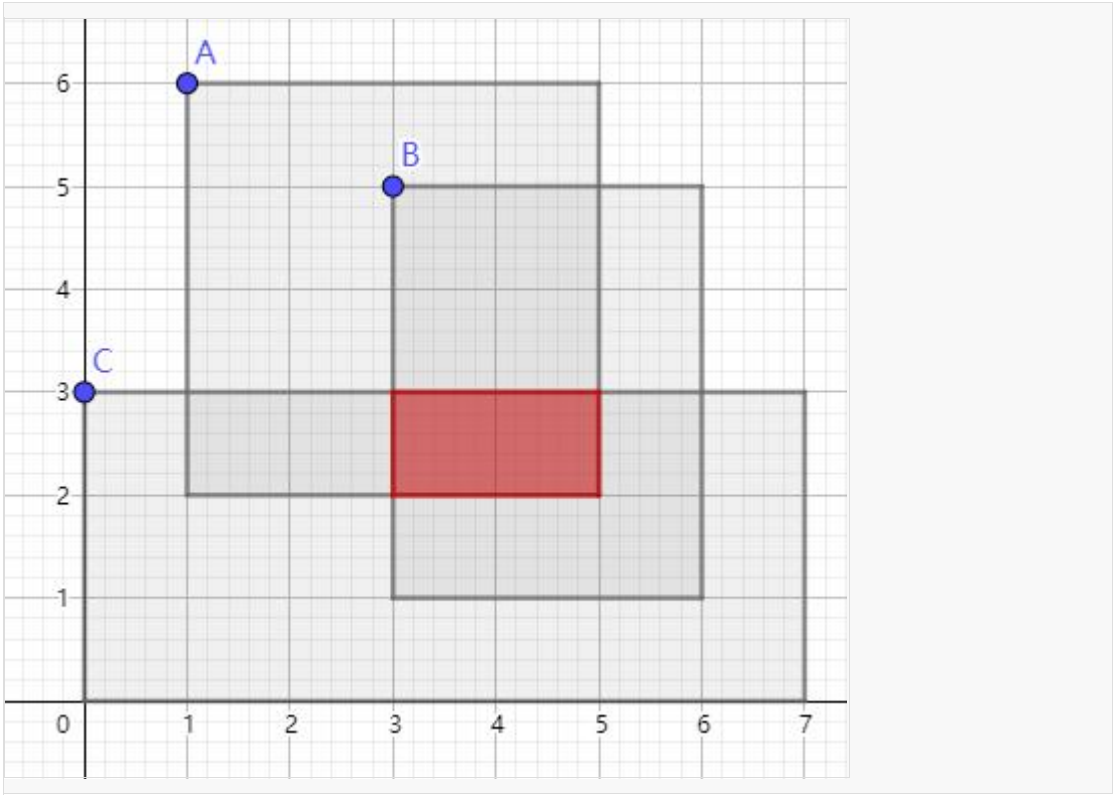
给定 3 个矩形 A，B，C

A：左上角坐标(1, 6)，宽 4，高 4

B：左上角坐标(3, 5)，宽 3，高 4

C：左上角坐标(0, 3)，宽 7，高 3

3 个矩形的相交面积为 2，如图所示




```
#include <iostream>
using namespace std;

struct Rect {
	int x;
	int y;
	int w;
	int h;
};
Rect intersect(Rect a, Rect b);
bool isValidRect(Rect a);
int main() {
	Rect a = {0};
	Rect b = {0};
	Rect c = {0};
	cin >> a.x >> a.y >> a.w >> a.h;
	a.y = a.y - a.h;
	cin >> b.x >> b.y >> b.w >> b.h;
	b.y =    b.y - b.h;
	cin >> c.x >> c.y >> c.w >> c.h;
	c.y = c.y - c.h;

	Rect result = intersect(a, b);
//	std::cout << result.x << " "<<result.y << " "<< result.w << " " << result.h << std::endl;
	if (isValidRect(result)) {
```

```
            result = intersect(result, c);
        }

        if (isValidRect(result)) {
            std::cout << result.w* result.h << std::endl;
        } else {
            std::cout << 0 << std::endl;
        }

        return 0;
}
bool isValidRect(Rect a) {
        return a.w > 0 && a.h > 0;
}
Rect intersect(Rect a, Rect b) {
        Rect ab = {0};
        ab.x = max(a.x, b.x);
        ab.y = max(a.y, b.y);

        int peer_x = min(a.x + a.w, b.x + b.w);
        int peer_y = min(a.y + a.h, b.y + b.h);
        ab.w = peer_x - ab.x;
        ab.h = peer_y - ab.y;
        return ab;
}
// 64 位输出请用 printf("%lld")
```