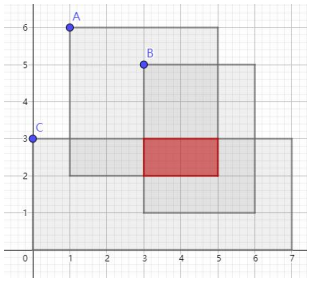
矩形相交的面积

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

示例 1

输入：

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

输出：

```
2
```

说明：

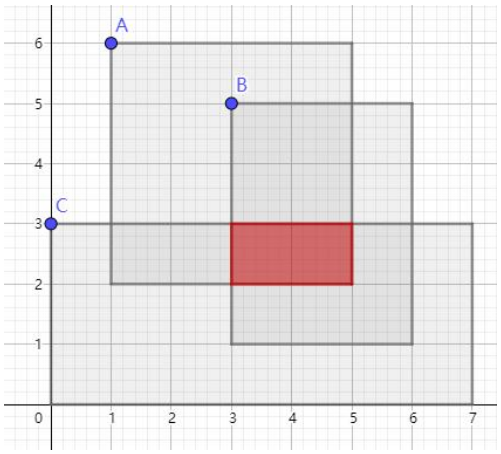
给定 3 个矩形 A，B，C

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 marray {

    int x, y;

    int l, h;

};


bool solve(marray& a, marray& b) {

    int t_x, t_y, t_l, t_h;

    if (a.x > b.x && b.x + b.l > a.x) {

        t_x = a.x;

        if (b.x + b.l < a.x + a.l)

            t_l = b.x + b.l - a.x;
```

```
else
    t_l = a.l;
if (a.y > b.y && a.y - a.h < b.y) {
    t_y = b.y;
    if (a.y - a.h > b.y - b.h)
        t_h = b.y - (a.y - a.h);
    else
        t_h = b.h;
    a.x = t_x, a.y = t_y, a.l = t_l, a.h = t_h;
    return true;
} else if (a.y < b.y && b.y - b.h < a.y) {
    t_y = a.y;
    if (b.y - b.h > a.y - a.h)
        t_h = a.y - (b.y - b.h);
    else
        t_h = a.h;
    a.x = t_x, a.y = t_y, a.l = t_l, a.h = t_h;
    return true;
} else if (a.y == b.y) {
    t_y = a.y;
    t_h = a.h < b.h ? a.h : b.h;
    a.x = t_x, a.y = t_y, a.l = t_l, a.h = t_h;
```

```
            return true;
        }
    } else if (a.x < b.x && a.x + a.l > b.x) {
        t_x = b.x;
        if (a.x + a.l < b.x + b.l)
            t_l = a.x + a.l - b.x;
        else
            t_l = b.l;
        if (b.y > a.y && b.y - b.h < a.y) {
            t_y = a.y;
            if (b.y - b.h > a.y - a.h)
                t_h = a.y - (b.y - b.h);
            else
                t_h = a.h;
            a.x = t_x, a.y = t_y, a.l = t_l, a.h = t_h;
            return true;
        } else if (b.y < a.y && a.y - a.h < b.y) {
            t_y = b.y;
            if (a.y - a.h > b.y - b.h)
                t_h = b.y - (a.y - a.h);
            else
                t_h = b.h;
```

```
            a.x = t_x, a.y = t_y, a.l = t_l, a.h = t_h;
            return true;
        } else if (a.y == b.y) {
            t_y = a.y;
            t_h = a.h < b.h ? a.h : b.h;
            a.x = t_x, a.y = t_y, a.l = t_l, a.h = t_h;
            return true;
        }
    } else if (a.x == b.x) {
        t_x = a.x;
        t_l = a.l < b.l ? a.l : b.l;
        if (a.y > b.y && a.y - a.h < b.y) {
            t_y = b.y;
            if (a.y - a.h > b.y - b.h)
                t_h = b.y - (a.y - a.h);
            else
                t_h = b.h;
            a.x = t_x, a.y = t_y, a.l = t_l, a.h = t_h;
            return true;
        } else if (a.y < b.y && b.y - b.h < a.y) {
            t_y = a.y;
            if (b.y - b.h > a.y - a.h)
```

```
                t_h = a.y - (b.y - b.h);
            else
                t_h = a.h;
            a.x = t_x, a.y = t_y, a.l = t_l, a.h = t_h;
            return true;
        } else if (a.y == b.y) {
            t_y = a.y;
            t_h = a.h < b.h ? a.h : b.h;
            a.x = t_x, a.y = t_y, a.l = t_l, a.h = t_h;
            return true;
        }
    }
    return false;
}

int main() {
    marray a, b, c;
    cin >> a.x >> a.y >> a.l >> a.h;
    cin >> b.x >> b.y >> b.l >> b.h;
    cin >> c.x >> c.y >> c.l >> c.h;
    if (solve(a, b)) {
        if (solve(a, c)) {
```

```
            cout << a.l* a.h << endl;
        } else {
            cout << '0' << endl;
        }
    } else {
        cout << '0' << endl;
    }
    return 0;
}
```