Point on rectangle

The picture below shows class **Point** which represents a point on 2 dimensional plane.

Rect represents a rectangle which contains a bottom-left point and top-right point of the rectangle (which sides are parallel with X-axis and Y-axis)

Your task is to implement method **area** and **contains** from class **Rect**. **area** is a method which returns area of the rectangle and **contains** is a method which test whether a point is in the rectangle or not. If it is, return **True**. If it is not, return **False**.

A point on the rectangle's side is "in" the rectangle.

```
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y
    def __str__(self):
        return "("+str(self.x)+","+str(self.y)+")"

class Rect:
    def __init__(self, p1, p2):
        self.lowerleft = p1
        self.upperright  = p2

    def area(self):
        ???

    def contains(self, p):
        ???

x1,y1,x2,y2 = [int(e) for e in input().split()]
lowerleft = Point(x1,y1)
upperright = Point(x2,y2)
rect = Rect(lowerleft, upperright)
print(rect.area())
m = int(input())
for i in range(m):
    x,y = [int(e) for e in input().split()]
    p = Point(x,y)
    print(rect.contains(p))
```

Grey area on the program is used to read 2 points from the rectangle then returns an area of it, after that, check if some input points are in the rectangle. (you don't have to change anything on the grey area. Please implement only in method **area** and **contains**)

Input

The first line contains 4 integer which represent (x,y) position of a bottom-left point and top-right point of the rectangle.

The next line is an integer m which is a number of following commands.

For the next m lines, each line contains 2 integer which is a position x, y of a point which will be tested whether it is in the rectangle.

Output

The first line is an area of the rectangle.

Next m lines are result of the test.

Example

Input (from keyboard)	Output (on screen)
2 2 10 10 4 0 0 2 4 3 5 10 1	64 False True True False