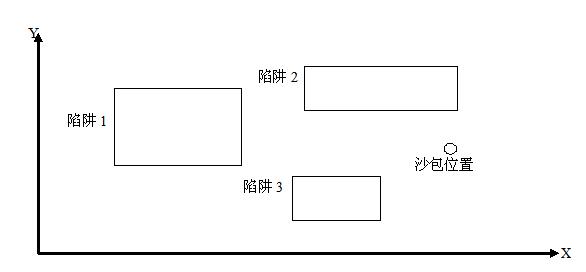
## 2555 人人都能参加第30届校田径运动会了

### 一、题目

#### 问题描述

杭州师范大学第29届田径运动会圆满的闭幕了，本届运动会是我校规模最大，参赛人数最多的一次运动会。在两天半时间里，由学生、教工组成的61支代表队共2664名运动员参加了比赛。比赛期间，运动健儿赛出了风格、赛出了水平，共有9人次打破6项校纪录。 我们寝室的4名同学是我班最卖力的啦啦队员，每天都在看台上为班级里的运动员们加油助威，为我班获得精神文明奖立下了汗马功劳。可是遗憾的是，与我校的其他近2万名同学一样，我们自己不能上场表演 ：( 于是，我们4名同学为下一届校运会发明了一种人人都能参加的比赛项目： 在地面上有N 个大小不等的长方形陷阱，每个陷阱的周长各不相同，每个参赛者都有一个沙包，闭上眼睛把它扔向地面，如果沙包掉到了某个陷阱里，那么这个参赛者根据这个陷阱的周长长度（如50米），绕跑道跑陷阱的周长长度（如50米），如果沙包没有掉到任何一个陷阱里，那么恭喜你，你跑0米。 有m<20000个同学参加了比赛，为了给跑步跑得最多的三位同学(冠军、亚军、季军)颁发安慰奖，必须给这m个同学的跑的长度按从多到少排序。 如下图一样的坐标系与长方形，这些长方形（陷阱）的四条边都与X轴或Y轴平行，它们之间互不相交，它们的左上角顶点的坐标与右下角顶点的坐标已知，给定一个你扔出去的沙包（看作是一个点）的坐标，可以得到你要跑的距离。（注意，这里的坐标值都不超过10000） 

#### 输入数据

第一行是两个正整数m<20000，n<100，它表示有m 个同学参加了扔沙包比赛，有n个陷阱。 接下去m行是m个同学扔出去的沙包的坐标，每一行都是两个正整数。 接下去的n行是陷阱的坐标，每行有4个正整数，它们从左到右分别是：陷阱左下角顶点的横坐标的值、陷阱左下角顶点的纵坐标的值，陷阱右上角顶点的横坐标的值、陷阱右上角顶点的纵坐标的值。

#### 输出数据

m个同学按跑的距离的多少，从多到少输出，一个数字一行。

#### 输入样例

5 3  
15 27  
32 93  
22 3  
98 4  
65 23  
22 65 100 76  
2 5 7 9  
54 6 94 24

#### 输出样例

116  
0  
0  
0  
0

#### 题目来源

HDU 2555 http://acm.hdu.edu.cn/showproblem.php?pid=2555

### 二、题解

#### 解题思路

题目大意是给定一个多个坐标，再划定多个矩形范围，确定坐标在哪个范围内并根据范围的周长排序。

使用两个二维数组存储沙包坐标和矩形范围两个顶点的坐标，在输入范围顶点的值时计算各沙包坐标是否落在范围内，并用一个数组存储该范围周长（若不落在任何范围里，则其对应的周长为0）。再将存储周长的数组排序，输出排序后的各个数组元素值。

#### 参考程序

#include<stdio.h>  
int main()  
{  
 int n,m,i,j,temp;  
 int size1[20000][2],size2[100][4],size[20000];  
 scanf("%d%d",&n,&m);  
 for(i=0;i<n;i++)  
 scanf("%d%d",&size1[i][0],&size1[i][1]);  
 for(i=0;i<m;i++)  
 for(j=0;j<4;j++)  
 scanf("%d",&size2[i][j]);  
 for(i=0;i<n;i++){  
 for(j=0,temp=1;j<m;j++){  
 if(size1[i][0]>=size2[j][0]&&size1[i][0]<=size2[j][2]&&size1[i][1]>=size2[j][1]&&size1[i][1]<=size2[j][3]){  
 size[i]=(size2[j][2]-size2[j][0]+size2[j][3]-size2[j][1])\*2;  
 temp=0;  
 break;  
 }  
 }  
 if(temp) size[i]=0;  
 }  
 for(i=0;i<n-1;i++)  
 for(j=0;j<n-i;j++)  
 {  
 if(size[j]<size[j+1])  
 {  
 temp=size[j];  
 size[j]=size[j+1];  
 size[j+1]=temp;  
 }  
 }  
 for(i=0;i<n;i++) printf("%d\n",size[i]);  
 return 0;  
}

#### 复杂度分析

无

#### 编程技巧

无