

## 一、题目概览

中文题目名称	要塞	弹球游戏	线段
英文题目名称	fortress	bounce	c
可执行文件名	fortress	bounce	c
输入文件名	fortress.in	bounce.in	c.in
输出文件名	fortress.out	bounce.out	c.out
时间限制	1s	4s	1s
空间限制	256MB	256MB	256MB
测试点数目	10	捆绑测试	10
测试点分值	10	捆绑测试	10
题目类型	传统	传统	传统
比较方式	全文比较	全文比较	实数比较
是否有部分分	否	否	否

## 二、注意事项：

1. 文件名（程序名和输入输出文件名）必须使用小写。
2. C/C++中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 0。
3. 评测环境为 Windows，使用 lemon 进行评测。
4. 开启 O2 优化，栈空间开大至 256M。

## 要塞 (fortress)

### 【题目描述】

为了防止“对方不想和你说话并向你扔了一只狗”，你修建了一座要塞来防御这些稳定尾翼脱壳穿甲狗。



对方不想和你说话  
并向你扔了一只狗



对方向你扔了一只旋转狗



对方向你扔了一只  
尾翼稳定脱壳穿甲狗

某一个时刻，你发现有  $m$  只狗正向你的要塞飞来，狗从  $1 \sim m$  编号，你需要使用你的要塞炮来拦截这些狗，一共有  $n$  门炮，从  $1 \sim n$  标号，每门炮只能攻击一个目标，由于每门炮的射界和位置不一样，每门炮能攻击到的狗的集合也不一样，每一只狗的耐久度都为 5，每受到一门炮的攻击，狗的耐久度会降低 1。狗会对要塞造成一定的伤害，狗的伤害值与其剩余耐久度有关，关系如下：

耐久度:	5	4	3	2	1	0
伤害值:	3	3	2	2	1	0

如果耐久度变为 0，则这只狗会直接消失，由于每门炮必须攻击一个目标，所以一只狗不会同时被超过 5 门炮攻击，现在你想知道，在所有攻击方案中，最坏情况下所有的狗一共能对你的要塞造成多少伤害(即伤害总和最大)。保证至少存在一种合法的攻击方案。

有多组数据。

### 【输入数据】

第一行，一个整数  $T$ ，表示数据组数

对于每一组数据：

第一行两个整数  $n, m$ ，表示要塞炮的数量和狗的数量。

之后的  $n$  行，每行先是一个整数  $k$ ，表示这一门炮能攻击到  $k$  只狗，然后是  $k$  个整数，表示它能攻击到的狗的编号，保证同一只狗的标号不会在这一行出现两次。

每组数据之后都有一个换行。

### 【输出数据】

一共  $T$  行，每一行一个整数，表示这组数据中最坏情况下的狗对你的要塞造成的总伤害值。

### 【样例输入 1】

```
1
5 5
1 1
1 1
1 1
1 1
1 1
```

**【样例输出 1】**

12

**【样例输入 2】**

1

5 5

1 1

2 1 2

3 1 2 3

3 1 2 3

3 1 2 3

**【样例输出 2】**

14

**【数据范围】**

对于 20%的数据：n,  $m \leq 5$ 。

对于另外 10%的数据： $m \leq 8$ 。

对于另外 10%的数据：每门炮都可以攻击所有的狗。

对于 100%的数据： $n \leq 100$ ,  $m \leq 100$ ,  $T \leq 5$ 。

## 弹球游戏 (bounce)

### 【题目描述】

妹老师引进了一个新的弹球游戏，游戏区域是一个  $n*m$  的网格，格子坐标编号从(1,1)到(n,m)，每个格子里都有一个斜面，斜面有四种摆放方式：



灰色是墙壁白色是空地，球碰到斜面会反弹。

玩家可以选择任何一个格子的中心，向上下左右四个方向之一扔出一颗弹球，弹球会在网格中反复碰撞走出一条路径。弹球不允许与灰色部分两条直角边以及边界碰撞。

例如出现如下碰撞情况玩家直接失败：



可以证明满足条件的路径，最终一定会回到发射点。

同时，初始时刻每相邻两个格子之间都有一面很薄的墙壁，弹球可以穿过并击碎它，击碎会带来收益(可正可负)。  $C_{ij}$  表示击碎  $(i,j)$  和  $(i,j+1)$  之间的墙壁的收益，  $R_{ij}$  表示击碎  $(i,j)$  和  $(i+1,j)$  之间的墙壁的收益。

除此之外，妹老师还设置了  $K$  个特殊格子，它们的中心必须被经过至少一次，否则游戏算作失败。

假老师可以任意安排每个格子里斜面的摆放方式，并且可以在任意一些格子中心扔出任意多个弹球，他想知道他能获得的最大收益是多少(为了使游戏不失败，收益可能为负)。

有多组数据

### 【输入数据】

第一行一个整数  $T$  表示数据组数。

对于每组数据：

第一行两个整数  $n,m$  表示网格边长，接下来  $n$  行每行  $m-1$  个整数表示  $C_{ij}$ ，接下来  $n-1$  行每行  $m$  个整数表示  $R_{ij}$ 。

接下来一行一个整数  $K$  表示特殊格子数量，接下来  $K$  行每行两个整数  $X_i,Y_i$  表示特殊格子坐标。

### 【输出数据】

每组数据输出一行一个整数表示最大收益，若游戏必然失败输出 Impossible。

### 【样例输入】

```
2
4 4
0 0 -1
```

0 1 0  
0 -1 -1  
0 1 0  
1 0 1 0  
-1 -1 0 0  
1 1 -1 -1  
1  
3 3  
2 3  
0 0  
0 0  
0 0 0  
2  
1 1  
2 3

**【样例输出】**

2  
Impossible

**【数据范围】**

subtask1[9pts]:  $nm \leq 10$ 。

subtask2[11pts]:  $nm \leq 20$ 。

subtask3[20pts]:  $R_{ij} = C_{ij} = 0$ 。

subtask4[22pts]:  $K = 0$ 。

subtask5[15pts]:  $n, m \leq 10$ 。

subtask6[23pts]: 无特殊条件。

对于全部数据,  $1 \leq T \leq 50$ ,  $1 \leq n, m \leq 30$ ,  $|C_{ij}|, |R_{ij}| \leq 500$ ,  $0 \leq K \leq 100$ ,  $1 \leq X_i \leq n$ ,  $1 \leq Y_i \leq m$ 。

## 线段 (c)

### 【题目描述】

给出若干条线段，用 $(x1,y1)$ ,  $(x2,y2)$ 表示其两端点坐标，现在要求支持两种操作：

0 x1 y1 x2 y2: 表示加入一条新的线段， $(x1,y1)-(x2,y2)$

1 x0: 询问所有线段中，x 坐标在  $x_0$  处的最高点的 y 坐标是什么，如果对应位置没有线段，则输出 0。

### 【输入数据】

第一行两个正整数 N,M 为初始的线段个数和操作个数

接下来 N 行，每行四个整数，表示一条线段

接下来 M 行，每行为一个操作 “0 x1 y1 x2 y2” 或 “1 x0”。

### 【输出数据】

对于每一个询问操作，输出一行一个实数，当你的答案与 std 答案绝对误差不超过  $1e-2$  时视为正确。

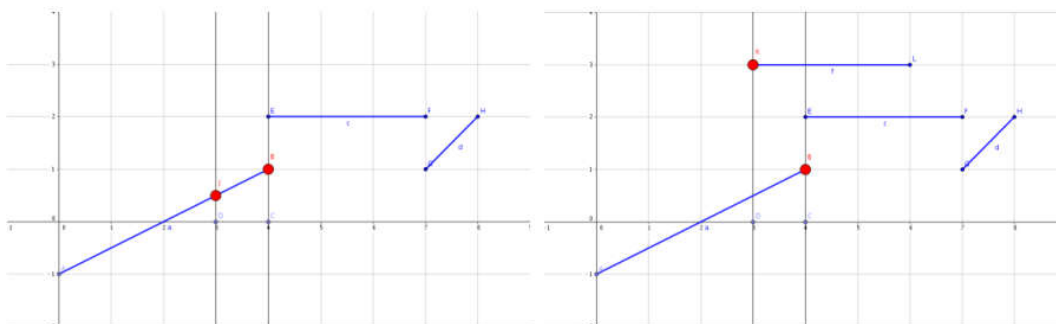
### 【样例输入】

```
3 4
0 -1 4 1
4 2 7 2
7 1 8 2
1 4
1 3
0 3 3 6 3
1 3
```

### 【样例输出】

```
2.000000
0.500000
3.000000
```

### 【样例示图】



### 【数据范围】

对于 10% 的数据:  $n, m \leq 1000$ 。

另外 20% 的数据: 所有的 1 操作都在 0 操作之后。

另外 20% 的数据: 所有线段的两端的 x 坐标都包含所有的询问的 x 坐标，你可以将每条线段当做直线处理。

对于 100% 的数据:  $n \leq 50000, m \leq 150000$ ,  $x1, x2, y1, y2, x0$  均为整数，

$0 < x0 \leq 10^5, |x1|, |x2|, |y1|, |y2| \leq 10^6$ 。