Task

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 题目名称 | 高楼 | 朋友 | 纸带 | 猜数 |
| 可执行文件名 | mansion | friend | ribbon | guess |
| 输入文件名 | mansion.in | friend.in | ribbon.in | guess.in |
| 输出文件名 | mansion.out | friend.out | ribbon.out | guess.out |
| 每个测试点时限 | 2秒 | 5秒 | 2秒 | 1秒 |
| 内存限制 | 1024MB | 1024MB | 1024MB | 1024MB |
| 测试点数目 | 20 | 20 | 20 | 20 |
| 每个测试点分值 | 5 | 5 | 5 | 5 |
| 是否有部分分 | 无 | 无 | 无 | 无 |
| 题目类型 | 传统 | 传统 | 传统 | 传统 |

提交源程序须加后缀：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pascal语言 | pas | pas | pas | pas |
| C 语言 | c | c | c | c |
| C++ 语言 | cpp | cpp | cpp | cpp |

**注意：最终测试时，所有编译命令均不打开任何优化开关。**

**高楼**

**问题描述**

一条街上有N座高楼，它们的高度分别是从1到N。已知从街的左边可以看到L栋高楼，从右边可以看到R栋。问有多少种高楼的排列方案满足这个条件。

**输入格式**

输入只有一行，三个整数，N，L，R，分别表示高楼总数，从左右分别能够看到的高楼数。

**输出格式**

输出只有一行，一个整数，表示方案数模109+7。

**样例输入**

5 3 2

**样例输出**

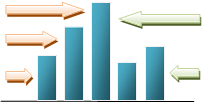
18

**数据规模**

对于20%的数据满足，N ≤ 10。

对于70%的数据满足，N ≤ 300。

对于100%的数据满足，1 ≤ N ≤ 5000；1 ≤ L，R ≤ N。

**样例解释**

一种满足样例要求的布局方案如右图。



**朋友**

**问题描述**

有三个人，他们是好朋友，分别叫做X，Y，Z。有一天他们想要聚在一起玩，想在其中一人的位置集合。

他们生活的城市很奇怪，从地图上看是一个N×M的矩形方阵，每个格子代表一个街区。这里的居民出门不走路，也不坐交通工具，而是依靠一种弹射器来移动。每个街区都装有弹射器，而使用弹射器是需要支付一定费用，而且每个弹射器都有自己的弹射能力。

如果第i行第j列的弹射器有Ai，j的弹射能力和Bi，j的费用,那么任何一个居民在(i，j)处支付Bi，j的费用就可以到达曼哈顿距离不超过Ai，j的任意位置。如下图所示，从红色街区交费以后可以跳到周围的任意蓝色街区。



现在已知X，Y，Z的坐标，求在哪里集合大家需要花的费用总和最低。

**输入格式**

第一行，两个整数，N和M，分别表示行数和列数。

接下来是2个N×M的矩阵，表示弹射能力Ai，j和费用Bi，j。

最后三行，每行两个整数，分别表示X，Y，Z所在的行号和列号。

**输出格式**

第一行，一个字符，X、Y或者Z，表示最优集合地点。如果存在多个最优集合地点，输出字典序较小的。

第二行，一个整数，表示最小费用。

如果无法集合，只输出一行“NO”（不含引号）。

**样例输入**

4 4

0 0 0 0

1 2 2 0

0 2 2 1

0 0 0 0

5 5 5 5

5 5 5 5

5 5 5 5

5 5 5 5

2 1

3 4

2 2

**样例输出**

Z

15

**数据规模**

对于20%的数据满足，N，M ≤ 10；Bi，j ≤ 20。

对于50%的数据满足，N，M ≤ 100；Bi，j ≤ 20。

对于100%的数据满足，1 ≤ N，M ≤ 200；0 ≤ Bi，j ≤ 109；  
0 ≤ Ai，j ≤ 1000。

**纸带**

**问题描述**

有一个无限长纸带，上面被划分成若干个格子。现在需要对其进行N次操作，第i次操作为在[Li，Ri]上擦除曾经写上的数字（假如有的话），并且写上数字i。询问最终可以看到多少种数字。

**输入格式**

第一行，一个整数N，表示操作数。

接下来M行，每行两个整数，Li和Ri，表示第i次操作的左端点和右端点。

**输出格式**

输出只有一行，一个整数，表示最终看到的数字种数。

**样例输入**

4

0 5

3 8

5 6

4 7

**样例输出**

3

**数据规模**

对于20%的数据满足，N ≤ 100；Ri ≤ 10000。

对于40%的数据满足，N ≤ 1000。

对于70%的数据满足，N ≤ 105。

对于100%的数据满足，1 ≤ N ≤ 106；0 ≤ Li ＜ Ri ≤ 109。

**样例说明**

如图所示。



可以看到的3种不同的数字（这里将数字由颜色代替），第3次操作被第4次操作所覆盖。

**猜数**

**问题描述**

有一个被猜数X，是1到N的范围内的整数，你每次可以给出一个整数Y。你会在你问C个问题之后得到你这个问题的回答，即X与Y的大小关系。如果你得到了K次X＜Y的回答，游戏就结束，你要避免这种情况的发生。询问在最坏情况下，你最少要猜多少次才能得到X=Y的答案。

**输入格式**

第一行，一个整数，数据组数T。

接下来共有T行，每行三个整数，N，K，C，如问题描述所示。

**输出格式**

输出共有T行，每行一个整数，表示第T组数据最坏情况下的最少猜测次数。

**样例输入**

1

6 2 1

**样例输出**

5

**样例解释**

一种可能的猜数过程如下：



猜测：3

应答：（无应答）

猜测：5

应答：X＞3

猜测：6

应答：X＜5

猜测：4

应答：X＜6

猜测：4

应答：X=4

**数据规模**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 数据点 | N | K | C | 数据点 | N | K | C |
| 1 | N≤100 | K=N | C=0 | 11 | N≤100 | K=N | C=1 |
| 2 | N≤1000 | 12 | N≤1000 |
| 3 | N≤106 | 13 | N≤106 |
| 4 | ­N≤109 | 14 | ­N≤109 |
| 5 | N≤1018 | 15 | N≤1018 |
| 6 | N≤100 | K≤100 | C=0 | 16 | N≤100 | K≤100 | C=1 |
| 7 | N≤1000 | 17 | N≤1000 |
| 8 | N≤104 | 18 | N≤104 |
| 9 | ­N≤105 | 19 | ­N≤105 |
| 10 | 20 |

对于100%的数据满足，1 ≤ T ≤ 100；N ≥ 1；K ≥ 1；C ≥ 0。