提高测试题

（请选手务必仔细阅读本页内容）

一、题目概况

中文题目名称	奇怪的管子	洪水	大都市	耕地
英文题目名称	kra	pow	meg	ork
可执行文件名	kra	pow	meg	ork
输入文件名	kra.in	pow.in	meg.in	ork.in
输出文件名	kra.out	pow.out	meg.out	ork.out
每个测试点时限	1 秒	2 秒	1 秒	3 秒
测试点数目	10	10	10	10
每个测试点分值	10	10	10	10
附加样例文件	无	无	无	无
题目类型	传统	传统	传统	传统

二、提交源代码文件名

对于pascal语言	kra.pas	pow.pas	meg.pas	ork.pas
对于C语言	kra.c	pow.c	meg.c	ork.c
对于 C++语言	kra.cpp	pow.cpp	meg.cpp	ork.cpp

三、编译命令（不包含任何优化开关）

对于 pascal 语言	fpc kra.pas	fpc pow.pas	fpc meg.pas	fpc ork.pas
对于 C 语言	gcc -o kra kra.c -lm	gcc -o pow pow.c -lm	gcc -o meg meg.c -lm	gcc -o ork ork.c -lm
对于C++语言	g++ -o kra kra.cpp -lm	g++ -o pow pow.cpp -lm	g++ -o meg meg.cpp -lm	g++ -o ork ork.cpp -lm

四、允许内存限制

内存上限	256M	256M	256M	256M

五、注意事项

1、 每位选手提交一个以自己编号命名的文件夹，在该文件夹下放4个源程序，然后再在该文件夹下建立四个子目录，名称分别为：kra、pow、meg、ork，再把源代码放入对应的子目录中。

2、 文件夹名、文件名(程序名和输入输出文件名)必须使用英文小写。

3、 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。

4、 统一评测时采用的机器配置为:windows下lemon评测和全国评测系统下评测。

5、 最终测试时，所有编译命令均不打开任何优化开关。

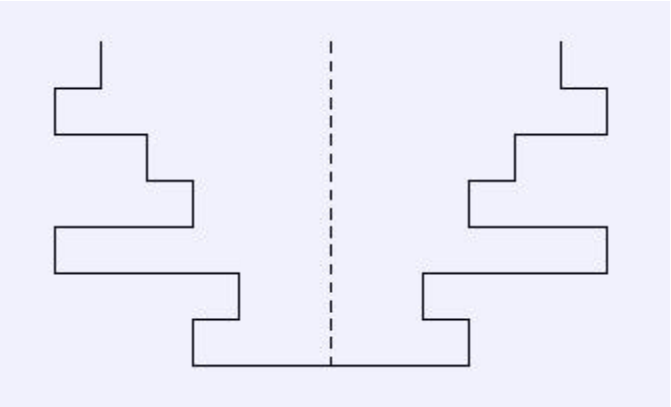
6、 请尽力优化，会收获更多的部分得分。

第一题：奇怪的管子

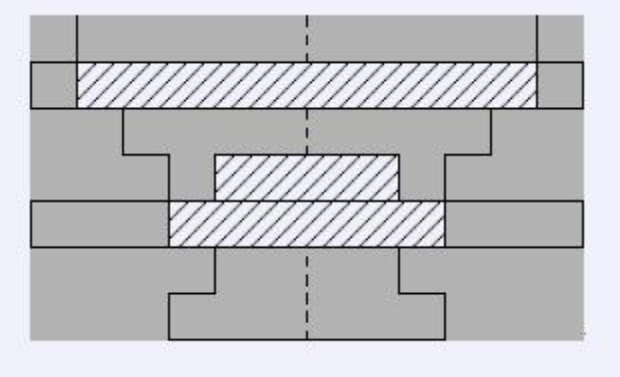
Kra-The Disks

Description

Johnny 在生日时收到了一件特殊的礼物，这件礼物由一个奇形怪状的管子和一些盘子组成. 这个管子是由许多不同直径的圆筒(直径也可以相同) 同轴连接而成. 这个管子的底部是封闭的,顶部是打开的. 下图是由直径为: 5cm, 6cm, 4cm, 3cm, 6cm, 2cm and 3cm 的圆筒组成的管子.



每个圆筒的高度都是相等的, 玩具中所带的盘子也是一些高度和它们相同的圆筒,直径有大有小.
Johnny 发明了一种游戏,把盘子从管子顶部一个接一个的扔下去,他想知道最后这些盘子落在了哪,假设盘子落下过程中圆心和管子的轴一直保持一致,比如说我们丢下去三个盘子: 3cm, 2cm and 5cm, 下图展示了最终它们的停止位置:



如图可以知道,盘子掉下去以后,要么被某个圆筒卡住,要么就是因为掉在了以前的一个盘子上而停住.
Johnny 想知道他最后扔下去的那个盘子掉在了哪个位置,你来帮他把.

Input

第一行两个整数 n 和 m (1<= n, m<= 300 000) 表示水管包含的圆筒数以及盘子总数.
第二行给出 n 个整数 r1, r2,...,rn (1 <=ri<= 1 000 000 000 for 1<= i<= n) 表示水管从上到下所有圆筒的直径. 第三行给出 m 个整数 k1, k2,..., km (1<= kj<= 1 000 000 000 for 1<= j<= m) 分别表示 Johnny 依次扔下去的盘子直径.

Output

一个整数输出最后一个盘子掉在了哪一层,如果盘子不能扔进水管,那么打印 0.

Sample Input

7 3
5 6 4 3 6 2 3
3 2 5

Sample Output

2

第二题：洪水

pow

Description

AKD 市处在一个四面环山的谷地里。最近一场大暴雨引发了洪水，AKD 市全被水淹没了。Blue Mary，AKD 市的市长，召集了他的所有顾问（包括你）参加一个紧急会议。经过细致的商议之后，会议决定，调集若干巨型抽水机，将它们放在某些被水淹的区域，而后抽干洪水。

你手头有一张 AKD 市的地图。这张地图是边长为 m*n 的矩形，被划分为 m*n 个 1*1 的小正方形。对于每个小正方形，地图上已经标注了它的海拔高度以及它是否是 AKD 市的一个组成部分。地图上的所有部分都被水淹没了。并且，由于这张地图描绘的地面周围都被高山所环绕，洪水不可能自动向外排出。显然，我们没有必要抽干那些非 AKD 市的区域。

每个巨型抽水机可以被放在任何一个 1*1 正方形上。这些巨型抽水机将持续地抽水直到这个正方形区域里的水被彻底抽干为止。当然，由连通器原理，所有能向这个格子溢水的格子要么被抽干，要么水位被降低。每个格子能够向相邻的格子溢水，“相邻的”是指（在同一高度水平面上的射影）有公共边。

Input

第一行是两个数 m,n(1<=m，n<=1000).
以下 m 行，每行 n 个数，其绝对值表示相应格子的海拔高度；若该数为正，表示他是 AKD 市的一个区域；否则就不是。
请大家注意:所有格子的海拔高度其绝对值不超过 1000,且可以为零.

Output

只有一行，包含一个整数，表示至少需要放置的巨型抽水机数目。

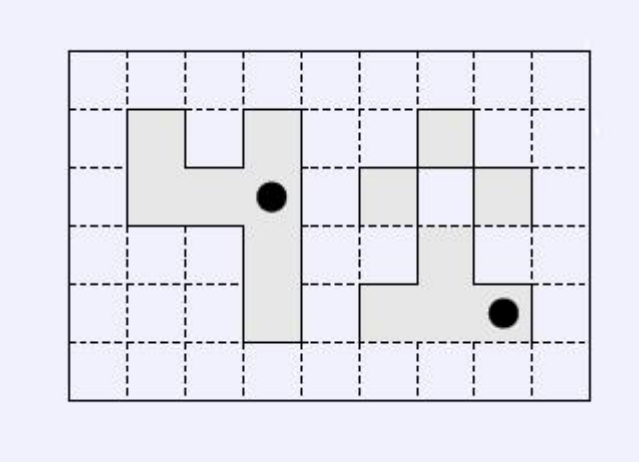
Sample Input

```
6 9
-2 -2 -1 -1 -2 -2 -2 -12 -3
-2 1 -1 2 -8 -12 2 -12 -12
-5 3 1 1 -12 4 -6 2 -2
-5 -2 -2 2 -12 -3 4 -3 -1
-5 -6 -2 2 -12 5 6 2 -1
-4 -8 -8 -10 -12 -8 -6 -6 -4
```

Sample Output

```
2
```

Hint



第三题：大都市

meg

Description

在经济全球化浪潮的影响下,习惯于漫步在清晨的乡间小路的邮递员 Blue Mary 也开始骑着摩托车传递邮件了。不过，她经常回忆起以前在乡间漫步的情景。
昔日，乡下有依次编号为 1..n 的 n 个小村庄，某些村庄之间有一些双向的土路。从每个村庄都恰好有一条路径到达村庄 1（即比特堡）。并且，对于每个村庄，它到比特堡的路径恰好只经过编号比它的编号小的村庄。另外，对于所有道路而言，它们都不在除村庄以外的其他地点相遇。在这个未开化的地方，从来没有过高架桥和地下铁道。
随着时间的推移，越来越多的土路被改造成了公路。至今，Blue Mary 还清晰地记得最后一条土路被改造为公路的情景。现在，这里已经没有土路了——所有的路都成为了公路，而昔日的村庄已经变成了一个大都市。
Blue Mary 想起了在改造期间她送信的经历。她从比特堡出发，需要去某个村庄，并且在两次送信经历的间隔期间,有某些土路被改造成了公路.现在 Blue Mary 需要你的帮助：计算出每次送信她需要走过的土路数目。（对于公路，她可以骑摩托车；而对于土路，她就只好推车了。）

Input

第一行是一个数 n.
以下 n-1 行，每行两个整数 a，b（1<=a<b<=n），表示改造开始之前有一条土路连接着村庄 a 和村庄 b.
以下一行包含一个整数 m，表示 Blue Mary 曾经在改造期间送过 m 次信。
以下 n+m-1 行，每行有两种格式的若干信息，表示按时间先后发生过的 n+m-1 次事件:
若这行为 A a b(1<=a<b<=n),则表示城市 a,b 之间的土路被改成公路，保证不会重复；
若这行为 W a, 则表示 Blue Mary 曾经从比特堡送信到村庄 a。
输入保证：

对于 70%数据：1<=N,M<=50

对于所有数据：1<=N,M<=100000;

Output

有 m 行，每行包含一个整数，表示对应的某次送信时经过的土路数目。

Sample Input

```
5
1 2
1 3
1 4
4 5
4
W 5
A 1 4
W 5
A 4 5
W 5
W 2
A 1 2
A 1 3
```

Sample Output

```
2
1
0
1
```

第四题：奇怪的管子

Ork-Ploughing

Description

Byteasar 想耕种他那块矩形的田，他每次能耕种矩形的一边（上下左右都行），在他每次耕完后，剩下的田也一定是矩形，每块小区域边长为 1，耕地的长宽分别为 m 和 n，不幸的是 Byteasar 只有一匹老弱的马，从马开始耕地开始，只有当它耕完了一边才会停下休息。但有些地会非常难耕以至于马会非常的累，因此 Byteasar 需要特别小心。**当耕完了一边之后，马可以停下来休息恢复体力**。每块地耕种的难度不一，但是 Byteasar 都非常清楚。我们将地分成 m x n 块单位矩形——我们用坐标（I,j）来定义它们。每块地都有一个整数 T[I,J],来定义（I,j）的耕种难度。所以每次马耕一边地时的难度就是所有它耕种的地的难度总和，对于这匹虚弱的马而言，这个值不能超过他的体力值。Byteasar 想知道在马不死掉的情况下最少需要耕多少次才能把地耕完。

Input

第一行三个整数，K,M,N。1<=k<=200 000 000,1<=m,n<=2000.其中 K 表示马的体力值。
接下来 N 行每行 M 个整数表示难度值。（0<=难度值<=10 000）

Output

一个整数表示最少的耕种次数（保证马能耕完）。

Sample Input

12 6 4
6 0 4 8 0 5
0 4 5 4 6 0
0 5 6 5 6 0
5 4 0 0 5 4

Sample Output

8