

曼哈顿

【问题描述】

P城是M国的著名旅游城市。在市长G先生的治理下，人民安居乐业，城市欣欣向荣。然而，G市长并没有被自己的政绩冲昏头脑，他清醒地意识到城市的治理还存在着一些问题，其中之一就是交通问题。

P城有 m 条横向街道和 n 条纵向街道，横向街道横贯东西，纵向街道纵穿南北，构成了P城整齐的交通网络（如图1所示）。

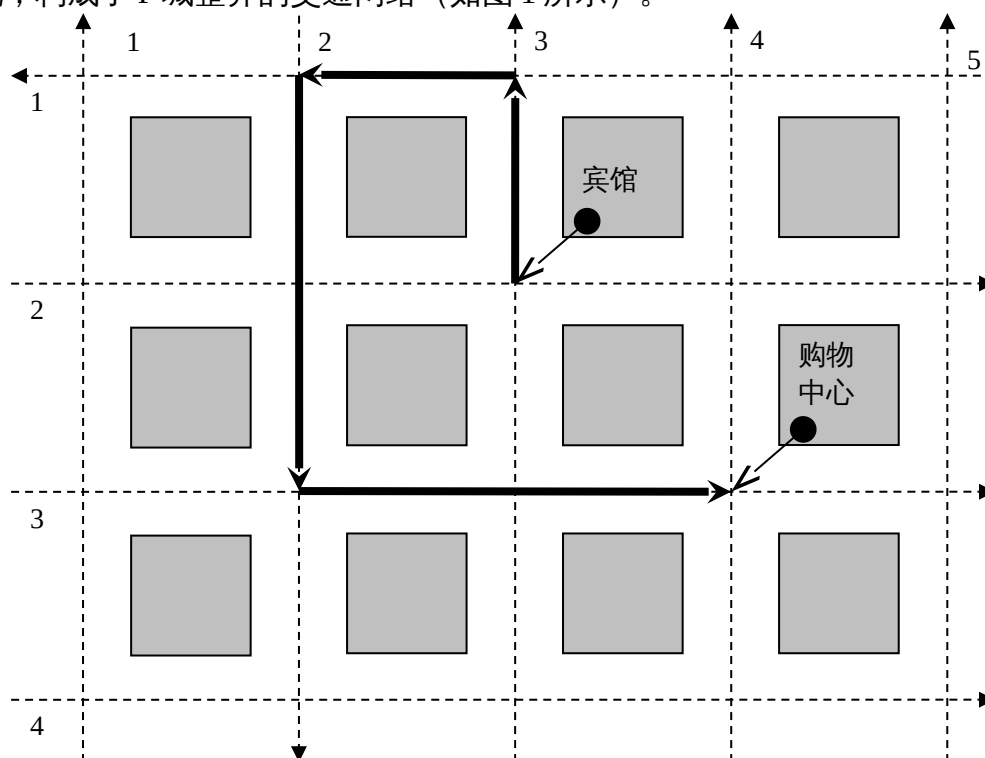


图1

由于街道狭窄，每条街道都只允许单向行驶，单向行驶的方向是事先设定好了的。一条横向街道的行驶方向只能是向东或者向西，一条纵向街道的行驶方向只能是向南或者向北，逆向行驶是绝对禁止的。

这项限制给交通带来了巨大的不便。如图1，很多游人希望从宾馆前往购物中心，但限于街道的行驶方向，他们不得不绕一个大圈才能够到达。

这个问题一直困扰着G市长，每天他都会收到不少游人的来信，抱怨P城不合理的交通设计。但由于街道数目过多，他和他的部下始终不能解决这个问题……

令人高兴的是这个问题不久就可能得以解决。因为最近他们以重金聘请了著名的交通规划大师B先生，请他对P城的交通进行有效合理的改造。

B先生知道不能通过拓宽街道的方法解决问题，因为这样势必影响到街道两旁的旅游景点，这是大家都不希望看到的。于是他准备重新设计街道的行驶

方向（整条街道的行驶方向），使之尽可能满足大家的要求。

B 先生先把 P 城的街道编号，横向街道由北向南编号为 $1, 2, \dots, m$ ，纵向街道由西向东编号为 $1, 2, \dots, n$ 。这样任何一个十字路口的位置都可以用一对正整数来表示，第一个数是该路口所在的横向街道的编号，第二个数是它所在的纵向街道的编号，这对整数被称为该十字路口的坐标。比如图 1 中宾馆所在的十字路口的坐标是 $(2,3)$ 。

经过长期调查，他整理出了游人们提得相对集中的一些要求。每条要求都可以写成如下的形式：从一个十字路口到另一个十字路口的最短路径的长度必须等于它们之间的曼哈顿距离。所谓曼哈顿距离是指两个十字路口在东西方向上的距离加上在南北方向上的距离，坐标分别为 (x_1, y_1) 和 (x_2, y_2) 的两个十字路口之间的曼哈顿距离为 $|x_1 - x_2| + |y_1 - y_2|$ 。

好了，B 先生已经知道了 P 城目前所有街道的行驶方向和游人们提得相对集中的要求，他能不能重新设计街道的行驶方向，使之满足所有要求呢？

另外，改变每条街道的行驶方向都有一定的工作量，工作量的大小因道路而异。B 先生不仅想找到一个可行的改造计划，而且还希望这个计划的总工作量尽可能小。你能帮帮他吗？

【输入文件】

输入文件的第一行有两个正整数 m 和 n ，分别表示横向街道和纵向街道的数目。

第二行是一个长度为 m 的字符串，由北向南列出了 m 条横向街道改造前的行驶方向。E 表示向东，W 表示向西。

第三行是一个长度为 n 的字符串，由西向东列出了 n 条纵向街道改造前的行驶方向。S 表示向南，N 表示向北。

第四行有 m 个非负整数，由北向南列出了改变每条横向街道的行驶方向的工作量。

第五行有 n 个非负整数，由西向东列出了改变每条纵向街道的行驶方向的工作量。

第六行是一个正整数 k ，表示游人们提得相对集中的要求的数目。

接下来的 k 行，每行有四个正整数 x_1, y_1, x_2, y_2 ，表示一条要求。这条要求的内容是希望从坐标为 (x_1, y_1) 的十字路口到坐标为 (x_2, y_2) 的十字路口的最短路径的长度等于这两个路口之间的曼哈顿距离。

【输出文件】

第一行是一个字符串，“possible”或者“impossible”（引号不输出）。输出“possible”表示可以通过改变街道的行驶方向满足输入数据中的所有要求，输出“impossible”表示无论怎么设计都不可能满足输入数据中的所有要求。

如果在第一行输出的是“possible”的话，在第二行输出一个整数，表示最小的总工作量，在第三行输出一个长度为 m 的字符串，由北向南列出改造后的 m 条横向街道的行驶方向，E 表示向东，W 表示向西，在第四行输出一个长度为 n 的字符串，由西向东列出改造后的 n 条纵向街道的行驶方向，S 表示向南，N 表示向北。



【约定】

- $m \leq 10$
- $n \leq 100$
- $k \leq 100$
- 改变一条街道的行驶方向的工作量不超过 10000

【样例输入】

```
2 3
WE
NNS
3 9
1 4 2
2
1 3 2 1
2 3 2 2
```

【样例输出】

```
possible
9
ww
NNS
```

【评分方法】

- 如果你的输出文件的第一行是“impossible”，
 - ✓ 如果确实无解，则该测试点满分。
 - ✓ 如果实际有解，则该测试点 0 分。
- 如果你的输出文件的第一行是“possible”，
 - ✓ 如果你的程序输出的方案不可行，则该测试点 0 分。
 - ✓ 如果你的程序输出的总工作量与实际总工作量不一致，则该测试点 0 分。
 - ✓ 如果你的程序输出的方案可行，但总工作量不是最小的，则该测试点 4 分。
 - ✓ 如果你的程序输出的方案可行，且总工作量最小，则该测试点满分。