注意事项

每个题目都有一个附件包，你可以从CMS系统和你的机器上获得这个附件包。

对于提交答案型(Output-only)的题目:

- 附件包里包含输入测试数据和测试数据样例。每一组测试数据都是一个独立的子任务。
- 你可以将多个输出文件打包成一个zip文件进行提交。为达到此目的，你的输出文件应该命名为`??.out`, 其中`??`是测试数据的编号(例如, `03.out`)。你可以使用下面的命令把多个输出文件压缩成一个zip文件: `zip output.zip *.out`。
- 对于提交答案型的题目，你最多可以提交100次。在每次提交中，你可以为测试数据的任一个子集提交输出文件。

对于其他类型的题目:

- 附件包里包含评分程序样例、实现样例、测试数据样例、以及编译脚本。
- 你只能提交一个文件，并且最多可以提交50次。
- 你提交的所有程序都不能从标准输入中读入，也不能打印到标准输出或与任何其他文件进行交互。但是，你的程序可以输出到标准错误流。
- 在题目描述的上方给出了你应当提交的文件名。它应当实现在任务描述部分中所述的过程（函数段），并遵照实现样例中列出的参数和返回值等。
- 你可以根据需要自行实现其它的过程（函数段）。
- 当使用评分程序样例测试你的程序时，你的输入应当与任务描述中规定的格式和限制相匹配，否则，有可能发生未予列出的行为。

约定

题目描述中给出了基于通用数据类型名`bool`, `integer`, `int64`, 以及 `int[]`(数组)的参数和返回值。

对于所支持的每一种编程语言，下表中列出了评分程序所使用的相应数据类型或实现:

语言	**bool**	**integer**	**int64**	**int[]**	数组 **a** 的长度
C++	`bool`	`int`	`long long`	`std::vector<int>`	`a.size()`
Pascal	`boolean`	`longint`	`int64`	`longint`型的数组	`Length(a)`
Java	`boolean`	`int`	`long`	`int[]`	`a.length`

限制

题目	时间限制	内存限制
nowruz	提交答案型	提交答案型
wiring	1 sec	256 MB
train	2 sec	256 MB

Nowruz诺鲁孜节

再过几天就是诺鲁孜节了(波斯人的新年)，爷爷邀请他的全家人到他的花园来聚会。在众多的宾客中有k个小孩。为了让这些孩子们在聚会中更开心，爷爷打算让他们玩一个捉迷藏的游戏。

整个花园可以看成一个有$m \times n$个方格的网格。其中有一些（或许没有）方格被岩石堵住了，而剩下的方格就称为*空格*。如果两个格子共享同一条边，我们就称这两个格子是邻*居*。因此，每一个方格最多有4个邻居：两个水平方向的和两个垂直方向的。爷爷想把花园变成一个迷宫。为达此目的，他会在花园中的一些*空格*上种植灌木来堵住它们。而这些被灌木丛堵住的方格就不再是*空格*了。

一个迷宫必须具有下面所述的性质。在迷宫中的任意一对空格a和b之间都只会恰有唯一的一条简单路径相连。而这条由a到b的简单路径就是一个从空格a开始并以空格b结束的空格序列，序列中所有的方格必须是不同的，而且每两个相连的方格都是邻居。

一个小孩能够躲藏的方格当且仅当这个方格是*空格*，而且它恰有唯一一个邻居是空格。同一个空格内只能躲藏一个小孩。

题目会给出整个花园的地图作为输入文件。你的任务就是帮助爷爷构造一个能够躲藏尽量多小孩的迷宫。

实现细节

这是一个提交答案类型的题目，而且它有部份分。题目会给出10个描述爷爷花园的输入文件。对于每个输入文件你应该提交一个含有迷宫的地图作为输出文件。我们会根据你提交的每个输出文件中的迷宫能够躲藏的小孩数目来给出你的分数。

这道题目你不需要提交任何源代码。

输入格式

每个输入文件都描述了一个表示整个花园的网格，我们也会给出爷爷邀请的小孩数目 k。格式如下：

- 第1行： 1: m n k
- 第 $1+i$ ($1 \leq i \leq m$)行： 网格中的第 i 行，它是一个长度为 n的字符串， 包含以下的字符(中间没有空格):
  - '.'：一个空格，
  - '#': 一块岩石。

输出格式

- 第 i ($1 \leq i \leq m$)行: 迷宫中的第 i 行 (种植了灌木的花园). 它是一个长度为 n的字符串， 包含以下字母 (中间没有空格):
  - '.': 一个空格,
  - '#': 一块岩石,
  - 'X': 一个灌木. (注意字母 X 必须为大写字母)

限制条件

- $1 \leq m, n \leq 1024$

评分

一个 *有效* 的输出文件必须符合下列所有的条件：

- 除了把输入文件中的任意多个字母'.' 修改成字母 'X' (即被灌木堵塞)外，输出的地图必须和输入地图完全一样。
- 输出的地图必须符合在上文中提及的迷宫的所有性质。

对于某一个测试数据，如果你的输出不是*有效*的，你的这个测试数据的得分将会是 0。反之，你的得分是$\min(10, 10 \cdot l/k)$，向下取值至小数后二位，这里的l是指你输出的迷宫中能够最多藏着的小孩数目，而k则表示在输入文件中题目要求你躲藏的小孩数目。对于一个测试数据，你能够得到10分，当且仅当你的输出是一个能够躲藏 k个或更多个小孩的迷宫。

对于每组测试数据都存在一个能得到10分的答案。

请注意如果你的答案是有效的，但根据上述公式你的得分仍然是0分，则在评分系统中，显示的结果将会是'Wrong Answer'。

例子

考虑以下输入：

```
4 5 5
....#
.#..#
...#.
....#
```

以下是其中一个有效的输出：

```
.X.X#
.#..#
...#X
XX..#
```

因为$l = 4$个小孩能够藏在这个迷宫中，所以这个解答能够得到$10 \cdot 4/5 = 8$分。小孩能够躲藏的方格如下以O所示。

```
OXOX#
.#.O#
...#X
XX.O#
```

以下的三个输出都不是有效的输出：

```
.XXX#          ...X#          XXXX#
.#XX#          .#.X#          X#XX#
...#.          ...#X          ..X#X
XX..#          XXXX#          ..XX#
```

在最左边的输出中，左上角的空格和最右列（位于右下方）的空格之间并没有一条简单路径。

在其他的两个输出中，对于任意两个空格之间都恰有两条简单路径相连。

Wiring接线

Maryam　是一位电机工程师。她正在为一座通讯塔设计接线方案。在这个塔上有一些分布在不同高度的连接点。一条电线可以用来将任何两个连接点连接起来。每一个连接点都可以接上任意数目的电线。而连接点共有两种: 分别为红色连接点及蓝色连接点。

为了表述方便起见，通讯塔会被视为一条直线，而那些红色及蓝色连接点会被视为在这条直线上的一些非负整数坐标。一条电线的长度是该电线所连接的两个连接点间的距离。

你要做的是帮 Maryam 找出一个接线的方案，使得满足以下条件:

1. 每个连接点上最少有一条电线连接到一个不同颜色的连接点上
2. 所用的电线的总长度为最短。

实现细节

你需要实现以下的子程序:

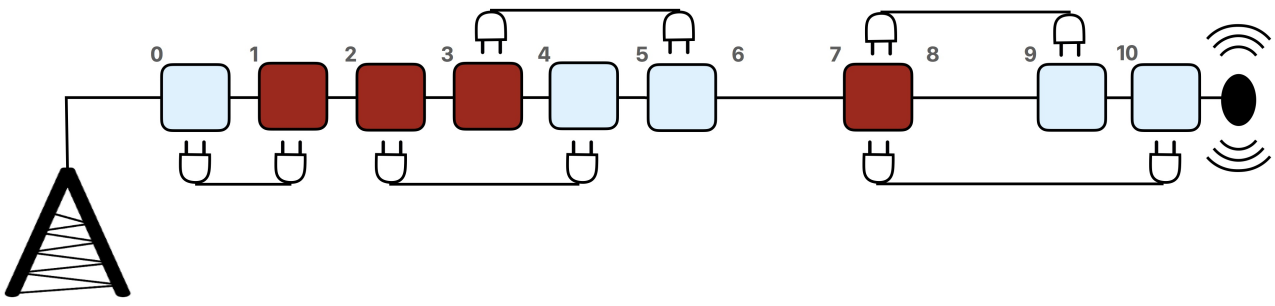
```
int64 min_total_length(int[] r, int[] b)
```

- r: 一个长度为 n 的数组，其内以升序排列着所有红色连接点的位置。
- b: 一个长度为 m 的数组，其内以升序排列着所有蓝色连接点的位置。
- 这个子程序需返回在所有可能的连接方案中，最短电线总长度的那个方案的电线总长度作为其返回值。
- 请注意这个子程序的返回值的类型为 `int64`。

例子

```
min_total_length([1, 2, 3, 7], [0, 4, 5, 9, 10])
```

以下的图表述了样例中的数据。



- 图中以水平的方式表示出相关的通讯塔。
- 因题目打印是黑白色的，所以红色接点以较深色来表示，而蓝色接点则以较浅色来表示。
- 图中有 4 个红色的连接点，其位置分别为 $1, 2, 3$ 及 7。
- 图中有 5 个蓝色的连接点，其位置分别为 $0, 4, 5, 9$ 及 10。
- 该例的最优解的电线总长度为 $1 + 2 + 2 + 2 + 3 = 10$，所以子程序的返回值为 10。
- 请注意共有两条电线连接在位置为7的连接点上。

限制条件

- $1 \leq n, m \leq 100\,000$,
- $0 \leq r[i] \leq 10^9$ (对于所有 $0 \leq i \leq n - 1$),
- $0 \leq b[i] \leq 10^9$ (对于所有 $0 \leq i \leq m - 1$),
- 数组 r 及数组 b 都已经按升序排好序。
- 在数组 r 及 b 内的所有 $n + m$ 个值均是不同的。

子任务

1. (7 分) $n, m \leq 200$。
2. (13 分) 所有红色接点的位置坐标小于任何蓝色接点的位置坐标。
3. (10 分) 在每 7 个连续（接续）的连接点内必有最少一个红色接点及一个蓝色接点。
4. (25 分) 所有接点在 $[1, n + m]$ 范围内有不同的位置坐标。
5. (45 分) 没有任何附加的限制。

评分程序样例

评分程序样例将会读入以下格式的数据:

- 第 1 行: n m
- 第 2 行: $r[0]$ $r[1]$ $\ldots$ $r[n-1]$
- 第 3 行: $b[0]$ $b[1]$ $\ldots$ $b[m-1]$

评分程序样例将会输出单独一行数据，上面含有 `min_total_length` 的返回值。

Toy Train玩具火车

Arezou和她的兄弟Borzou是双胞胎。他们收到的生日礼物是一套好玩的玩具火车。他们用它建了一个有n个车站和m段*单向轨道*的铁路系统。这些车站的编号是从0到$n-1$。每段轨道都始于某一车站，然后终于同一车站或其他车站。每个车站至少会有一段轨道以它为起点。

其中有些车站是*充电车站*。无论何时，如果火车抵达某个充电车站，它都会被充到满电。满电火车拥有足够的动力连续地驶过n段轨道，但是如果不再充电的话，在即将进入第$n+1$段轨道时它就会因电已用光而停车。

每个车站都有一个轨道开关，可以扳向任一以该车站为起点的轨道。火车从某个车站驶出时，驶向的正是该车站的开关所扳向的轨道。

这对双胞胎打算用他们的火车玩个游戏。他们已经分完了所有的车站：每个车站要么归Arezou，要么归Borzou。游戏里面只有一列火车。游戏开始时，这列火车停在车站s，并且充满了电。为启动游戏，车站s的拥有者把车站s的开关扳向某个以s为起点的轨道。随后他们启动火车，火车也就开始沿着轨道行驶。

无论何时，在火车首次进入某一车站时，该车站的拥有者都要扳定车站开关。开关一旦扳定，它就会保持状态不变直到游戏结束。因此，火车如果开到了某个曾经进过的车站，就会沿着与之前相同的轨道开出该车站。

由于车站数量是有限的，火车的行驶最终都会落入某个*环路*。环路是指一系列*不同*的车站$c[0], c[1], \cdots, c[k-1]$，其中火车在离开车站$c[i]$ $(0 \leq i < k-1)$后驶上连向车站$c[i+1]$的轨道，在离开车站$c[k-1]$后驶上连向车站$c[0]$的轨道。一个环路可能只包括一个车站(此时$k=1$)，即火车从车站$c[0]$驶出后又驶上了连向$c[0]$的轨道。

如果火车能够连续行驶跑个没完，Arezou就赢了。否则火车最后会把电用光而停车，这样Borzou就赢了。换句话说，如果在车站$c[0], c[1], \cdots, c[k-1]$中至少有一个充电车站，且使得火车能够不断地充电而沿着环路跑个没完，Arezou赢。否则，它就会最终把电用光（有可能是在沿着环路跑好几圈后），Borzou赢。

现在给你一个这样的铁路系统。Arezou和Borzou将会玩n轮游戏。其中在第s轮游戏中（$0 \leq s \leq n-1$），火车最初停在车站s上。你的任务是，对每一轮游戏，判断是否无论Borzou怎么玩，Arezou都必胜。

实现细节

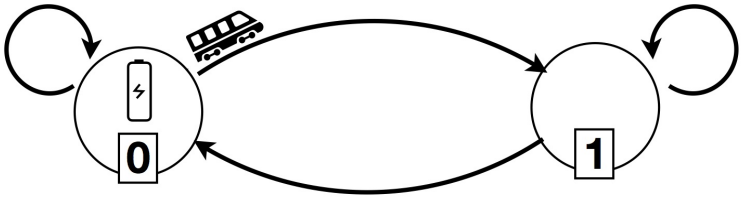
你需要实现下面的函数：

```
int[] who_wins(int[] a, int[] r, int[] u, int[] v)
```

- a: 长度为n的数组。如果Arezou拥有车站i，则$a[i] = 1$；否则Borzou拥有车站i，且$a[i] = 0$。
- r: 长度为n的数组。如果车站i是充电车站，则$r[i] = 1$。否则$r[i] = 0$。
- u和v: 长度为m的数组。对于所有$0 \le i \le m - 1$，存在某一单向轨道，其起点为$u[i]$，终点为$v[i]$.
- 该函数需要返回一个长度为n的数组w。对于每个$0 \le i \le n - 1$，如果在火车最初停在车站i的游戏中，不管Borzou怎么玩，Arezou都能赢，则$w[i]$的值应为1。否则$w[i]$的值应为0。

例子

```
who_wins([0, 1], [1, 0], [0, 0, 1, 1], [0, 1, 0, 1])
```



- 这里有2个车站。Borzou拥有充电车站0。Arezou拥有车站1，但是它不是充电车站。
- 这里有4段轨道$(0, 0), (0, 1), (1, 0)$和$(1, 1)$，其中(i, j)表示一个以车站i为起点、车站j为终点的单向轨道。
- 考虑火车最初停在车站0的游戏。如果Borzou将车站0的开关扳向轨道$(0, 0)$，那么火车就会沿着这个环形轨道绕个没完（注意，车站0是一个充电车站）。在这种情况下，Arezou赢。否则，如果Borzou把车站0的开关扳向轨道$(0, 1)$，Arezou可以把车站1的开关扳向轨道$(1, 0)$。这样的话，火车将会在两个车站之间绕个不停。Arezou还是会赢，因为车站0是充电车站，火车将跑个没完。因此，无论Borzou怎么玩，Arezou都会赢。
- 根据类似的逻辑，在火车最初停在车站1的游戏中，无论Borzou怎么玩，Arezou也都会赢。因此，函数应当返回$[1, 1]$。

数据范围和限制

- $1 \le n \le 5000$。
- $n \le m \le 20\,000$。
- 至少会有一个充电车站。
- 每个车站至少会有一段轨道以它为起点。
- 可能会有某个轨道的起点和终点是相同的（即$u[i] = v[i]$）。
- 所有轨道两两不同。也就是说，不存在这样的两个下标i和j（$0 \le i < j \le m - 1$），使得$u[i] = u[j]$且$v[i] = v[j]$。
- 对于所有$0 \le i \le m - 1$，都有$0 \le u[i], v[i] \le n - 1$。

子任务

1. (5分) 对于所有$0 \leq i \leq m-1$，都有$v[i] = u[i]$或者$v[i] = u[i] + 1$。
2. (10分) $n \leq 15$。
3. (11分) Arezou拥有所有车站。
4. (11分) Borzou拥有所有车站。
5. (12分) 充电车站的数量为1。
6. (51分) 无任何限制。

评分程序样例

评分程序样例会按照下述格式来读取输入数据：

- 第1行： n m
- 第2行： $a[0]$ $a[1]$ $\ldots$ $a[n-1]$
- 第3行： $r[0]$ $r[1]$ $\ldots$ $r[n-1]$
- 第$4+i$行(对于所有$0 \leq i \leq m-1$)： $u[i]$ $v[i]$

评分程序样例会按照下述格式打印出`who_wins`的返回结果：

- 第1行： $w[0]$ $w[1]$ $\ldots$ $w[n-1]$