时间限制均为 1 S，内存限制均为 128 M。

问题1：奶牛拍照（leftout）

Farmer John 正在尝试给他的牛群拍照。根据以往的经验，他知道这一工作往往结果不怎么样。

这一次， Farmer John 购买了一台昂贵的无人机，想要拍一张航拍照。为了使照片尽可能好看，他想让他的奶牛们在拍照时都朝向同一个方向。奶牛们现在在一块有围栏的草地上排列成 $N \times N$ （ $2 \leq N \leq 1000$ ）的方阵，例如：

```
RLR
RRL
LLR
```

这里，字符 `R` 表示一头朝右的奶牛，字符 `L` 表示一头朝左的奶牛。由于奶牛们都挤在一起， Farmer John 没办法走到某一头奶牛面前让她调转方向。他能做的只有对着某一行或某一列的奶牛喊叫让她们调转方向，使得被叫到的这一行或列内的所有 `L` 变为 `R` ， `R` 变为 `L` 。 Farmer John 可以对任意多的行或列发号施令，也可以对同一行或列多次发令。

就如同 Farmer John 想象的，他发现他不可能让他的奶牛们都朝向同一个方向，他最多能让除了一头之外的所有奶牛都朝向相同的方向，请找出这头奶牛。

输入格式（文件名：leftout.in）

输入的第一行包含 N 。以下 N 行描述了奶牛方阵的第 $1 \ldots N$ 行，每行包含一个长度为 N 的字符串。

输出格式（文件名：leftout.out）

输出一头奶牛的行列坐标，满足如果这头奶牛被调转方向， Farmer John 就可以使他的所有奶牛都朝向同一个方向。如果不存在这样的奶牛，输出 `-1` 。如果存在多头这样的奶牛，输出其中行坐标最小的，如果多头这样的奶牛具有相同的行坐标，输出其中列坐标最小的。

输入样例

```
3
RLR
RRL
LLR
```

输出样例

```
1 1
```

样例解释

在这个例子中，位于第 1 行第 1 列（左上角）的奶牛是那头令人讨厌的奶牛，因为 Farmer John 可以喊叫第 2 行和第 3 列来让所有奶牛都面向左侧，只有这一头奶牛面向右侧。

问题2：假期安排（vacation）

奶牛航空公司计划在奶牛栖息的 N 个牧场（ $1 \leq N \leq 200$ ）上建立航线。正如所有航空公司都会做的那样，有 K 个牧场（ $1 \leq K \leq 100, K \leq N$ ）被选为航司枢纽。牧场标号为 $1 \ldots N$，被选为航司枢纽的牧场标号为 $1 \ldots K$ 。

目前有 M （ $1 \leq M \leq 10000$ ）条单程航班连接这些牧场。航班 i 从牧场 u_i 飞往 v_i，搭乘这趟航班花费 d_i 美元（ $1 \leq d_i \leq 10^6$ ）。

奶牛航空公司最近知道奶牛们有 Q （ $1 \leq Q \leq 10000$ ）次单程旅行。第 i 次旅行是从牧场 a_i 到 b_i 。为了能够从 a_i 飞往 b_i，这次旅行可能需要搭乘多趟航班（甚至有可能多次飞到同一个牧场），但必须途径至少一个航司枢纽（航司枢纽可能是这次旅行的起点或终点，也可能不是）。这样的限制可能使得没有合法的从 a_i 飞往 b_i 的路线。对于所有拥有合法线路的旅行，帮助奶牛航空公司求出这次旅行最少需要花费多少美元。

输入格式（文件名：vacation.in）

第 1 行：四个整数 N 、 M 、 K 、 Q 。

第 $2 \ldots M+1$ 行：第 $i+1$ 行包含第 i 趟航班的 u_i 、 v_i 、 d_i 。

第 $M+2 \ldots M+Q+1$ 行：第 $M+i+1$ 行包含第 i 次旅行的 a_i 、 b_i 。

输出格式（文件名：vacation.out）

第 1 行：Q 次旅行中拥有合法路线的旅行的数量。

第 2 行：对于所有拥有合法路线的旅行，它们的最小总花费。

输入样例

```
3 3 1 3
3 1 10
1 3 10
1 2 7
3 2
2 3
1 2
```

输出样例

```
2
24
```

样例解释

有 3 个牧场（标号为 $1\ldots3$），牧场 1 为航司枢纽。从牧场 3 到 1 有一趟花费为 10 美元的航班，以此类推。我们需要关注 $3\to2$、$2\to3$ 和 $1\to2$ 这三次旅行。

$3\to2$ 的旅行只有一条可行路线，花费为 $10+7$。$2\to3$ 的旅行没有合法路线，因为没有从牧场 2 起飞的航班。$1\to2$ 的旅行只有一条可行路线，花费为 7。

问题3：爬山（climb）

Farmer John 发现，他的奶牛们进行日常锻炼后产奶量会更高。因此他决定让他的 N 头奶牛（ $1\le N\le25000$ ）来回爬山。

奶牛 i 上山需要 U_i 分钟，下山需要 D_i 分钟。奶牛们是驯养过的动物，每头奶牛都需要一个人帮助她们爬山，但由于财政紧张，一共只有两个人能帮助奶牛们： Farmer John 和他的侄子 Farmer Don 。 FJ 计划指导奶牛们上山， FD 指导奶牛们下山。由于每头奶牛都需要指导，并且在上山或下山的过程中只有一个人能帮助她们，因此在任何时刻都只能有一头奶牛上山（ FJ 提供帮助）和一头奶牛下山（ FD 提供指导）。山顶上可能积累有一群奶牛等待 FD 帮助她们下山。奶牛们下山的顺序可以和上山的顺序不同。

请求出所有 N 头奶牛完成一次爬山活动所需的最少总时间。

输入格式（文件名：climb.in）

第 1 行：奶牛数量 N 。

第 $2\ldots N+1$ 行：第 $i+1$ 行包含两个整数 U_i 、 D_i （ $1\le U_i,D_i\le50000$ ）。

输出格式（文件名：climb.out）

第 1 行：所有 N 头奶牛完成一次爬山活动所需的最少总时间。

输入样例

```
3
6 4
8 1
2 3
```

输出样例

```
17
```

样例解释

上山和下山的顺序都为奶牛 $3,1,2$ ，此时总时间为 17 。