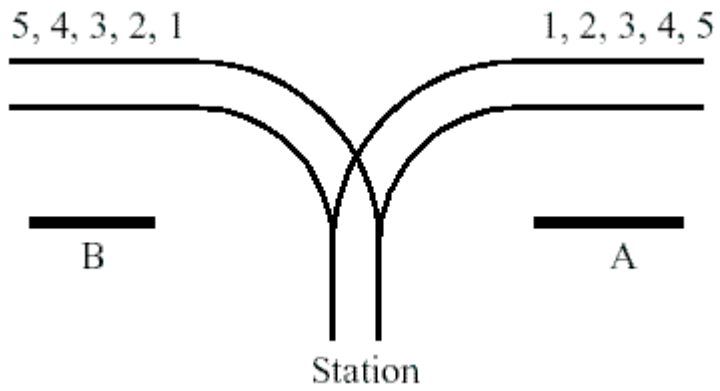
题目 1　　Rails

输入文件：rails.in
输出文件：rails.out
提交文件：rails.exe, 以及 rails.pas 或 rails.c 或 rails.cpp

某城市有一个火车站，其中的铁路如下图所示：



每辆火车都从 A 方向驶入车站，再从 B 方向驶出车站，同时它的车厢可以进行某种形式的重新组合。假设从 A 方向驶来的火车有 N 节车厢($N \le 1000$)，分别按顺序编号为 1, 2, …, N。负责车厢调度的工作人员需要知道能否使它以 $a_1, a_2, \ldots, a_N$ 的顺序从 B 方向驶出。请你给他写一个程序，用来判断能否得到指定的车厢顺序。假定在进入车站之前每节车厢之间都是不连着的，并且它们可以自行移动，直到处在 B 方向的铁轨上。另外假定车站里可以停放任意多节的车厢。但是一旦当一节车厢进入车站，它就不能再回到 A 方向的铁轨上了，并且一旦当它进入 B 方向的铁轨后，它就不能再回到车站。

输入

输入文件包含很多段，每一段是很多行。除了最后一段外，每一段都定义了一辆火车以及很多所需要的重组顺序。每一段的第一行是上面所说的整数 N，接下来的每一行都是 1, 2, …, N 的一个置换，每段的最后一行是数字 0。

最后一段只包含数字 0。

输出

输出文件中的每一行都和输入文件中的一个描述置换的行相对应，并且用 Yes 表示可以把它们编排成所需的顺序，否则用 No 表示。另外，用一个空行表示输入文件的对应段的结束。输入文件中最后的空段不需要在输出文件中有内容相对应。

例

输入文件：

```
5
1 2 3 4 5
5 4 1 2 3
0
6
6 5 4 3 2 1
0
0
```

输出文件：

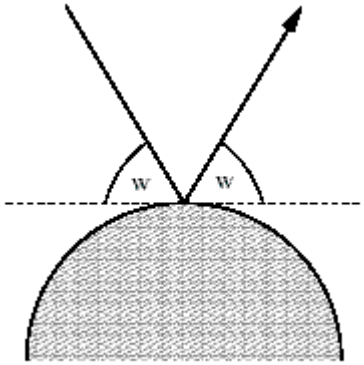
```
Yes
No

Yes
```

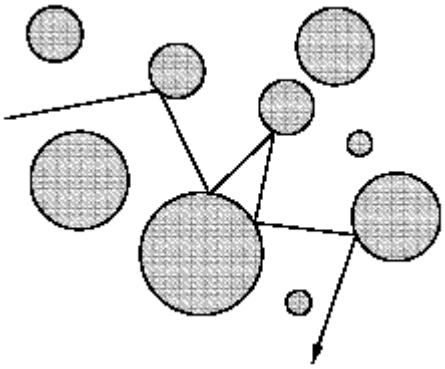
题目 2　　Reflections

输入文件：reflect.in
输出文件：reflect.out
提交文件：reflect.exe, 以及 reflect.pas 或 reflect.c 或 reflect.cpp

下面是一个圆柱的截面图，当一束光线设向该柱时，该光线会以和入射角相同的反射角反射出去。



而下图是一束光线在若干各柱子之间连续反射的情况：



你的任务是写一个程序，计算出一个给定的入射光线在给定的若干各柱子之间的反射情况。

输入

输入文件包含很多段，每一段是很多行。除了最后一段外，每一段都定义了若干个柱子的位置和大小以及一束入射光线的起点和方向。每一段的第一行是表示柱子个数的整数 n（$n \le 25$），接下来有 n 行，每行都是三个整数 x_i, y_i, r_i ($r_i \ge 0$)，前面两个表示第 i 个柱子的中心坐标，最后一个表示第 i 个柱子的半径。每段的最后一行是四个整数 x, y, d_x, d_y，其中(x,

y)表示光线的起点，(d_x, d_y)表示光线的方向矢量，d_x 和 d_y 中至少有一个不为 0。

所有的柱面都是分离不相连的，光线不会从柱子里面发出，也永远不会和柱子相切。

最后一段只包含数字 0 表示结束。不用处理这个段。

输出

对每个描述场景的段首先输出场景的编号，接着输出光线首先击中的十个柱子的编号（柱子的编号就是它们在输入文件中的顺序号）。

如果一束光线至多击中了十个柱子（然后它就射向远方），那么在最后一个它击中的柱子编号后面输出 inf。如果它击中了多于十个柱子，那么在第十个柱子的编后后面输出三个点(…)。输入文件中最后的空段不需要在输出文件中有内容相对应。

例

输入文件：

```
3
3 3 2
7 7 1
8 1 1
3 8 1 -4
2
0 0 1
5 0 2
2 0 1 0
0
```

输出文件：

```
Scene 1
1 2 1 3 inf
Scene 2
2 1 2 1 2 1 2 1 2 1 ...
```

题目 3　　Robot

输入文件：robot.in
输出文件：robot.out
提交文件：robot.exe, 以及 robot.pas 或 robot.c 或 robot.cpp

某机器人研究所正在研制一种机器人，目的是让它能以最少的时间来从一个地方移动到另一个地方。该机器人只能沿直线（轨道）移动，并且所有的轨道构成了矩形的方格。相邻的轨道之间间隔 1 米，整个场地为 N×M 米并且全部被方格覆盖。场地边缘离最近的轨道的距离也是 1 米。机器人是直径 1.6 米的圆形，轨道必须经过它的圆心。机器人只能面向东南西北四个方向之一，并且轨道都是南北向和东西向的。机器人只能向它面对的方向前进，并且它只可以在轨道交叉的地方改变它面对的方向，而且初始时它是朝向一个轨道的方向的。场地中的每个障碍物都是 1 米×1 米的占地面积，每个障碍物都处在一个由轨道围成的 1 米×1 米的方格中。机器人的动作由两种命令控制，分别是 GO 和 TURN。

GO 命令有一个整数参数 $n \in \{1, 2, 3\}$，接到该命令后机器人向它面向的方向前进 n 米。

TURN 命令有一个参数，该参数或者为 left，或者为 right。接到该命令后它会按照参数所指定的方向左转或右转 90 度。

每个命令的执行耗时 1 秒。

请你帮研究员们写一个程序，要求该程序能够计算出机器人从一个地方移动到另一个地方所需的最小时间。

输入

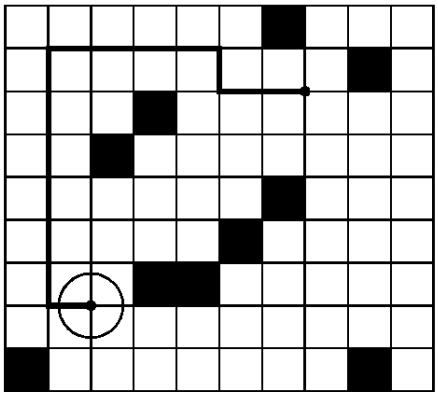
输入文件包含很多段，每一段是很多行。除了最后一段外，每一段都定义了一个场地以及机器人的初始位置和方向。每一段的第一行是用单个空格分开的两个整数 M≤50 和 N≤50，接下来有 M 行，每行有 N 个用单个空格分开的 0 或 1，分别表示空地和障碍物（轨道在空地之间）。每段的最后一行是四个正整数 B_1, B_2, E_1, E_2，每个整数后面有一个空格，接着用一个单词表示机器人初始的方向。B_1, B_2 表示机器人的初始位置相对西北角的坐标，E_1, E_2 表示机器人的目标位置相对西北角的坐标，对机器人到达目标位置时的方向不作要求。我们用（行，列）型的坐标，也就是说左上角（西北角）的坐标为 0, 0，右下角（东南角）的坐标为 M, N。方向用 north, west, south 和 east 四个单词之一给出。

最后一段只包含数字 M = 0 和 N = 0。

输出

输出文件中的每一行都和输入文件中的一个段相对应，用一个整数给出机器人从初始位置面朝初始方向经过若干命令移动到目标位置所需的最小时间，如果机器人不可能移动到目标位置则输出-1。输入文件中最后的空段不需要在输出文件中有内容相对应。

例



输入文件：

```
9 10
0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 1 0
0 0 0 1 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 1 0 0 0 0
0 0 0 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 1 0
7 2 2 7 south
0 0
```

输出文件：

12

题目 4　　Coins

输入文件：coins.in
输出文件：coins.out
提交文件：coins.exe, 以及 coins.pas 或 coins.c 或 coins.cpp

有十二个硬币，其中有一个是假的，但是从外表分辨不出来。假币和真币唯一的区别是重量不同，但不知到底是重还是轻。有一个精确的天平，你的任务是使用它称三次就找出假币。例如，如果在天平的两边各放一枚硬币时天平平衡，可知这两枚硬币都是真的。而如果接着把它们中间的一枚和第三枚硬币分别放在天平的两边时天平不平衡，就可以知道第三枚硬币是假的，并且可以根据天平倾斜的方向判断出假币比真币轻还是重。

输入

输入文件的第一行是整数 n(n>0)表示后面的测试数据的组数。每组测试数据包含三行输入，每行代表一次测量。十二枚硬币分别编号为 A 到 L。有两个串分别表示天平左右两边所放的硬币，第三个串表示一次称重的结果，由 up, down 和 even 三个单词之一给出，分别表示天平的右端升起，降下还是保持水平。

输出

对于每组测试数据，输出假币的编号以及它比真币轻(light)还是重(heavy)，我们保证结果可以唯一确定。

例

输入文件：

1
ABCD EFGH even
ABCI EFJK up
ABIJ EFGH even

输出文件：

K is the counterfeit coin and it is light.

题目 5　　FastFood

输入文件：fastfood.in
输出文件：fastfood.out
提交文件：fastfood.exe, 以及 fastfood.pas 或 fastfood.c 或 fastfood.cpp

在一条公路旁有很多快餐连锁店，还需要在公路旁修一些仓库，用来给这些餐馆补给物资。很自然的，我们希望每个餐馆和补给它的仓库的距离总和最小，你的任务是写一个程序给出该最小值以及各个仓库的位置和分配。

更确切的说，我们会给你 n 个整数 $d_1 < d_2 < \ldots < d_n$，分别表示 n 个餐馆的位置坐标（沿着公路，假设公路为直线）。然后还会给你一个整数 k ($k \le n$)表示仓库的数目。

这 k 个仓库会修在 k 个不同的餐馆处，每个餐馆分配离它最近的仓库。总的距离和定义如下：

$$\sum_{i=1}^{n} | d_i - (\text{给餐馆} i \text{供给的仓库的位置}) |$$

写一个程序计算出仓库的位置，使得上述距离总和最小。

输入

输入文件包含很多段，每一段是很多行。除了最后一段外，每一段都定义了一些餐馆的位置以及仓库的个数。每一段的第一行是两个整数 n 和 k，分别满足 $1 \le n \le 200, 1 \le k \le 30, k \le n$。接下来是 n 行，每行一个整数，顺序的表示餐馆的位置。

最后一段只包含 n=k=0，这段不用处理。

输出

对于每个段，首先输出号码，接着以如下方式输出仓库的位置：对于每个仓库用一行输出它的位置和它供给的餐馆编号。如果有多于一个的最优解，只需输出其中任何一个。最后用一行输出总的距离和。

每个段后输出一个空行。

例

输入文件：

```
6 3
5
6
12
19
20
27
0 0
```

输出文件：

```
Chain 1
Depot 1 at restaurant 2 serves restaurants 1 to 3
Depot 2 at restaurant 4 serves restaurants 4 to 5
Depot 3 at restaurant 6 serves restaurant 6
Total distance sum = 8
```