



题目名称	驾车旅行	信号塔	蹦极	智慧珠游戏
程序文件名	drive	tower	precision	ball
输入文件名	drive.in	tower.in	precision.in	ball.in
输出文件名	drive.out	tower.out	precision.out	ball.out
每个测试点时限	1 秒	1 秒	2 秒	2 秒
内存限制	128 MB	128 MB	128 MB	128 MB
测试点数目	10	10	10	10
每个测试点分值	10	10	10	10
是否有部分分	无	无	无	无
评测方式	Normal	Normal	Special Judge	Normal

提交源程序需加后缀

对于 Pascal 语言	drive.pas	tower.pas	precision.pas	ball.pas
对于 C 语言	drive.c	tower.c	precision.c	ball.c
对于 C++ 语言	drive.cpp	tower.cpp	precision.cpp	ball.cpp

评测环境:

Intel(R) Core(TM) i3-370M CPU @2.40GHz 2.39GHz, 2.00GB RAM

Cena 0.8.2 @ Windows 8 Release Preview x64

C++选手注意 Windows 7/8/Vista 可以使用 %lld 输入输出 64 位整数。

最终评测时，所有编译命令将打开-O2 优化开关。

Nescafe 25

驾车旅行 (drive.pas/c/cpp)

题目描述

Freda 和 Rainbow 一起驾车出游，这天，他们来到了 Poetic Island 市。

Poetic Island 的街道纵横交错，南北方向和东西方向的双向街道各有 n 条，相邻街道的距离均为 L 英里。同时，每条街道有它自己的最高速度限制。

当然，这 $2*n$ 条街道构成了 $n*n$ 个十字路口，其中西南角交叉口的坐标为 $(1,1)$ ，东北角交叉口的坐标为 (n,n) 。在所有交叉口均可任意改变行驶方向，但是只能在交叉口处改变速度。

现在，Rainbow 和 Freda 要从交叉口 (sx, sy) 开车行驶到 (tx, ty) 。他们只想沿着路程最短的线路行驶，并且行驶时间必须在给定的闭区间 $[t1, t2]$ 分钟内，车速必须是 5 的正整数倍。若车速为 v 千米/小时，则每升汽油能行驶的路程为 $80 - 0.03v^2$ 千米。

Rainbow 想知道最早到达终点的方案是什么，Freda 想知道最省油的方案是什么。但是 Rainbow 和 Freda 简直弱爆了有木有，请你帮他们计算一下吧……

输入格式

输入第一行为两个整数 n, L 。

第二行包含 n 个正整数，从南到北描述 n 条东西走向的街道的速度限制。

第三行包含 n 个正整数，从西到东描述 n 条南北走向的街道的速度限制。

第四行包含六个正整数 $sx, sy, tx, ty, t1, t2$ 。

输出格式

若有解，输出两行，分别描述最早到达的方案（若有多种方案，选择其中最省油的）和最省油的方案（如果有多种方案，选择其中最早到达的）。每种方案用两个数表示，第一个数表示用时多少分钟（向上取整）；第二个数表示耗油量多少升（四舍五入保留两位小数）。

若无解，仅输出一行：No。

样例输入

```
6 20
30 40 50 50 50 50
50 50 50 50 50 40
1 1 6 6 300 320
```

样例输出

```
300 6.25
318 5.60
```

Nescafe 25

数据范围与约定

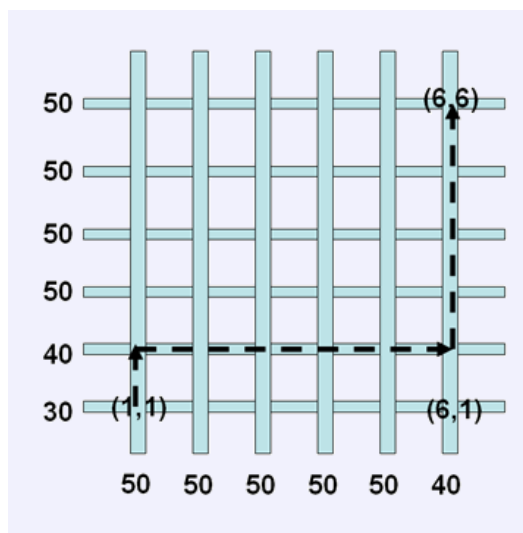
对于 20% 的数据， $n \leq 4$;

对于 50% 的数据， $n \leq 8$;

对于 100% 的数据， $1 \leq n \leq 10$, $1 \leq L \leq 20$, $0 \leq t_1 \leq t_2 \leq 1000$, 各条路速度限制不超过 50。

样例说明

在满足题目限制条件的前提下，样例的最快路线为以 40 千米/小时为速度匀速前进，路程为 200 千米，因此时间为 5 小时，每升汽油可以行驶 $80 - 0.03 \times 40 \times 40 = 32$ 千米，因此耗油量为 $200 / 32 = 6.25$ 升。最省油路线是先以 40 千米/小时行驶 120 千米，然后以 35 千米/小时行驶 80 千米，耗油量为 $120 / 32 + 80 / (80 - 0.03 \times 35 \times 35) = 5.60$ 升。下图的路线可以同时满足两种方案（其中第二种方案需要在 (6,2) 处改变速度）。



Nescafe 25

信号塔 (tower.pas/c/cpp)

输入格式

Rainbow 和 Freda 穿过了纵横交错的道路，来到了 Poetic Island 市旁边的一座山脚下。这里有排成一行的 n 座信号塔，其中从左到右第 i 座的高度为 $h[i]$ 。Rainbow 和 Freda 将要在该市盖房子定居了，但是他们认为这 n 座高低不一的信号塔很是难看。于是他们打算把这 n 座信号塔改造成从左到右高度单调不减的。

Rainbow 和 Freda 可是具有超能力的哦~~~ 他们每次可以选择一座塔，把它合并到两侧相邻的一座塔上，合并后的高度为两座塔的高度之和。每次合并完成后，被合并的两座塔会消失；合并成的塔就看做一座新塔，不能再分开。因此每次合并后塔的总数会减少 1。他们想知道，完成他们的目标至少需要多少次合并呢？Rainbow 和 Freda 还是弱爆了有木有！于是请你帮忙计算一下……

输出格式

第一行一个整数 n 。

第二行 n 个整数，第 i 个整数表示 $h[i]$ 。

样例输入

```
8
1 9 9 4 1 2 2 9
```

样例输出

```
3
```

数据范围与约定

对于 50% 的数据， $0 < n \leq 5000$ 。

对于 100% 的数据， $0 < n \leq 200000$ ， $0 < h[i] \leq 2147483647$ ， h 均为随机生成。

Nescafe 25

蹦极 (precision.pas/c/cpp)

题目描述

Alice 制作了 n 个人偶，编号为 1 到 n ，并决定用这些人偶来表演蹦极，她又制作了一个蹦极台，这个蹦极台有 k 个不同的高度可供选择，分别是 $h, 2h, \dots, kh$ ，注意，每个高度上只能站一个人偶，人偶都是独一无二的，她们也无法同时在两个高度表演。

Alice 制作的这 n 个人偶并不完全相同，至少他们的强度 w_i 是不一定相同的，如果有一个强度较低的人偶发现另一个强度较高的人偶在更低的地方表演蹦极，她就会非常不满，拒绝蹦极。（是的，自律人形 :P）此外，这些人偶都是会飞行的，因此她们要求完全由自己爬上蹦极位置，不需要 Alice 的帮助。一开始，所有的人偶都在一起，如果一个人偶的飞行速度是 v_i ，她要爬上高度为 H 的地方蹦极需要 H/v_i 秒，所有人偶会同时开始向目标飞行。

为了让 Mukyu 观看时人偶们就位所用的时间最少，Alice 设计了一种方案来安排 n 个人偶中的其中 k 个表演蹦极。经历过上次梭哈题目后，Alice 现在对现代科技非常不屑，她认为现代科技无法完美的完成这一任务。显然，你的任务就是证明 Alice 的话是错误的。

输入格式

第一行包含一个正整数 T ，表示有 T 组测试数据。

每组测试数据的第一行包含三个整数 n, k, h ，表示有 n 个人偶，蹦极台有 k 个高度，分别为 $h, 2h, \dots, kh$ 。

接下来的一行中有 n 个整数，代表 w_i ，即每个人偶的强度。

接下来的一行中有 n 个整数，代表 v_i ，即每个人偶的飞行速度。

输出格式

对于每组测试数据，输出 k 个人偶的编号，表示使这 k 个人偶在 $h, 2h, \dots, kh$ 上表演可以使所需的准备时间最短。如果有多种方案，你可以输出任意一种。

样例输入

```
5 3 2
1 2 3 2 1
1 2 1 2 10
```

样例输出

```
5 2 4
```

样例说明

编号为 5 的人偶站在高度 2 ，她飞到目标需要 $2/10 = 0.2$ 秒。

编号为 2 的人偶站在高度 4 ，她飞到目标需要 $4/2 = 2$ 秒。

Nescafe 25

编号为 3 的人偶站在高度 6，她飞到目标需要 $6/2 = 3$ 秒。

因此，最短的准备时间为 3 秒，可以证明，没有其他方案所需的世界比这种更短。

数据范围与约定

对于 20% 的数据，保证 $N \leq 300$ 。

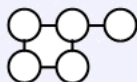
对于 40% 的数据，保证 $N \leq 5000$ 。

对于 100% 的数据，保证 $N \leq 100000$ ， $T \leq 10$ 。

提示

C++ 选手如果使用 `cin` 读入数据很可能因此超时，推荐使用 `scanf/printf`。

学好英语很重要。



Nescafe 25

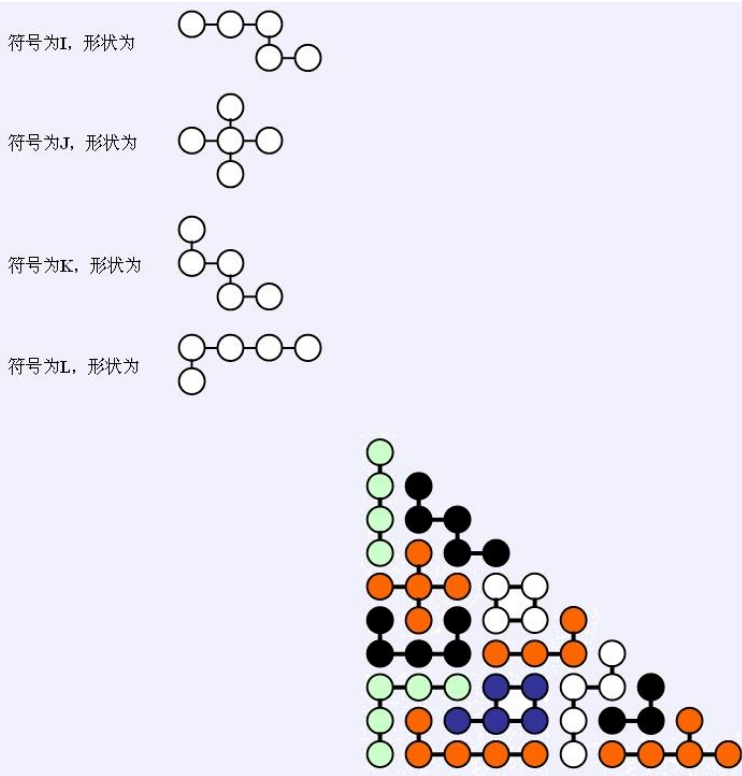


图2示出了一种拼盘方案。为便于描述可将图2抽象为图3，就可以用一个数据为字符的二维数组来表示了。

```
B
B K
B K K
B J K K
J J J D D
G J G D D C
G G G C C C I
E E E H H I I A
E L H H H I A A F
E L L L L I F F F F
```

图3

对于由珠子构成的零件，可以放到盘件的任一位置，条件是能有地方放，且尺寸合适，所有的零件都允许旋转(0°、90°、180°、270°)和翻转(水平、竖直)。

现给出一个盘件的初始布局，求一种可行的智慧珠摆放方案，使所有的零件都能放进盘件中。

输入格式

文件中包含初始的盘件描述，一共有 10 行，第 i 行有 i 个字符。如果第 i 行的第 j 个字符是字母”A”至”L”中的一个，则表示第 i 行第 j 列的格子上已经放了零件，零件的编号为对应的字母。如果第 i 行的第 j 个字符是”.”，则表示第 i 行第 j 列的格子上没有放零件。 输入保证预放的零件已摆放在盘件中。

输出格式

如果能找到解，向输出文件打印 10 行，为放完全部 12 个零件后的布局。其中，第 i 行应包含 i 个字符，第 i 行的第 j 个字符表示第 i 行第 j 列的格子上放的是哪个零件。 如果无解，输出单独的一个字符串’No solution’(不要引号，请注意大小写)。 所有的数据保证最多只有一组解。

Nescafe 25

样例输入

.
..
...
....
.....
.....C
...CCC.
EEHH..
E.HHH....
E.....

样例输出

B
BK
BKK
BJKK
JJDD
GJGDDC
GGGCCI
EEHHIIA
ELHHHIAAF
ELLLLIFFF