

1001 小 Q 系列故事——为什么时光不能倒流

Time Limit: 0.1 Seconds Memory Limit: 65536K

我以为我会是最坚强的那一个 我还是高估了自己
我以为你会是最无情的那一个 还是我贬低了自己

就算不能够在一起 我还是为你担心
就算你可能听不清 也代表我的心意

那北极星的眼泪 闪过你曾经的眼角迷离
那玫瑰花的葬礼 埋葬的却是关于你的回忆

如果时光可以倒流 我希望不要和你分离
如果注定分离 我希望不要和你相遇

. ——摘自《小 Q 失恋日记》第 17 卷 520 页

这是码农小 Q 第 58 次失恋了，也是陷得最深的一次。

要知道，小 Q 自从第一次到腾讯公司报到，就被风姿绰约的前台 MM 彻底迷住了，这 1000 多个日日夜夜他无时无刻不在憧憬着他们美好的未来。为了能见到 MM，他每天早到晚归，甘愿加班，连续 3 年被评为优秀员工，并且以全公司最快的速度晋级到四级岗位。就在他终于鼓足勇气准备表白的时候，MM 却满面春风地送来了一包喜糖.....

现在小 Q 专门请了年休假治疗情伤，但情绪总不见好转，每天足不出户，眼睛盯着墙上的钟表，反复念叨：“表白要趁早，时光不倒流，表白要趁早，时光不倒流.....”

假设现在已知当前的时间，让时间倒退回若干，你能计算出钟表显示的时间吗？

Input

输入首先包含一个整数 N，表示有 N 组测试用例。

接下来的 N 行表示 N 个测试用例，每行包括 2 个时间 HH:MM:SS hh:mm:ss

HH:MM:SS 表示当前的时间，hh:mm:ss 表示希望倒退回去的时间。

[Technical Specification]

00<=HH<=11

00<=hh<=99

00<=MM, SS, mm, ss<=59

Output

请计算并输出钟表倒退后显示的时间，要求输出格式为 HH:MM:SS（即时分秒均显示 2 位，不足则补 0），每组数据输出占一行。

Sample Input

2

11:28:32 02:14:21

05:00:00 96:00:01

Sample Output

09:14:11

04:59:59

1002 小明系列故事——女友的考验

Time Limit: 0.2 Seconds Memory Limit: 32768K

终于放寒假了，小明要和女朋友一起去看电影。这天，女朋友想给小明一个考验，在小明正准备出发的时候，女朋友告诉他，她在电影院等他，小明过来的路线必须满足给定的规则：

1、假设小明在的位置是 1 号点，女朋友在的位置是 n 号点，则他们之间有 $n-2$ 个点可以走，小明每次走的时候只能走到比当前所在点编号大的位置；

2、小明来的时候不能按一定的顺序经过某些地方。比如，如果女朋友告诉小明不能经过 $1 \rightarrow 2 \rightarrow 3$ ，那么就要求小明来的时候走过的路径不能包含有 $1 \rightarrow 2 \rightarrow 3$ 这部分，但是 $1 \rightarrow 3$ 或者 $1 \rightarrow 2$ 都是可以的，这样的限制路径可能有多条。

这让小明非常头痛，现在他把问题交给你了。

特别说明，如果 1 2 3 这三个点共线，但是小明是直接从 1 到 3 然后再从 3 继续，那么此种情况是不认为小明经过了 2 这个点的。

现在，小明即想走最短的路尽快见到女朋友，又不想打破女朋友的规定，你能帮助小明解决这个问题吗？

Input

输入包含多组样例，每组样例首先包含两个整数 n 和 m ，其中 n 代表有 n 个点，小明在 1 号点，女朋友在 n 号点， m 代表小明的女朋友有 m 个要求；

接下来 n 行每行输入 2 个整数 x 和 y (x 和 y 均在 `int` 范围)，代表这 n 个点的位置（点的编号从 1 到 n ）；

再接着是 m 个要求，每个要求 2 行，首先一行是一个 k ，表示这个要求和 k 个点有关，然后是顺序给出的 k 个点编号，代表小明不能走 $k_1 \rightarrow k_2 \rightarrow k_3 \cdots \rightarrow k_i$ 这个顺序的路径；

n 和 m 等于 0 的时候输入结束。

[Technical Specification]

$2 \leq n \leq 50$

$1 \leq m \leq 100$

$2 \leq k \leq 5$

Output

对于每个样例，如果存在满足要求的最短路径，请输出这个最短路径，结果保留两位小数；否则，请输出 "Can not be reached!" （引号不用输出）。

Sample Input

3 1

1 1

2 1

3 1

2

1 2

2 1

0 0

1 1

2

1 2

5 3

0 0

5 3

1 2

1 2 2

5 2 1

3

1 2 3

2

4 5

2

1 5

0 0

Sample Output

2.00

Can not be reached!

21.65

1003 吉哥系列故事——完美队形 I

Time Limit: 1.0 Seconds Memory Limit: 65536K

吉哥这几天对队形比较感兴趣。

有一天，有 n 个人按顺序站在他的面前，他们的身高分别是 $h[1], h[2] \dots h[n]$ ，吉哥希望从中挑出一些人，让这些人形成一个新的队形，新的队形若满足以下三点要求，则称之为完美队形：

- 1、挑出的人保持他们在原队形的相对顺序不变；
- 2、左右对称，假设有 m 个人形成新的队形，则第 1 个人和第 m 个人身高相同，第 2 个人和第 $m-1$ 个人身高相同，依此类推，当然，如果 m 是奇数，中间那个人可以任意；
- 3、从左到中间那个人，身高需保证递增，如果用 H 表示新队形的高度，则 $H[1] < H[2] < H[3] \dots < H[mid]$ 。

现在吉哥想知道：最多能选出多少人组成完美队形？

Input

第一行输入 T ，表示总共有 T 组数据($T \leq 20$)；

每组数据先输入原先队形的人数 $n(1 \leq n \leq 200)$ ，接下来一行输入 n 个整数，表示按顺序从左到右原先队形位置站的人的身高 ($50 \leq h \leq 250$ ，不排除特别矮小和高大的)。

Output

请输出能组成完美队形的最多人数，每组数据输出占一行。

Sample Input

```
2
3
1 2 1
4
1 2 2 1
```

Sample Output

```
3
4
```

1004 吉哥系列故事——完美队形 II

Time Limit: 1.0 Seconds Memory Limit: 65536K

吉哥又想出了一个新的完美队形游戏！

假设有 n 个人按顺序站在他的面前，他们的身高分别是 $h[1], h[2] \dots h[n]$ ，吉哥希望从中挑出一些人，让这些人形成一个新的队形，新的队形若满足以下三点要求，则就是新的完美队形：

- 1、挑出的人保持原队形的相对顺序不变，且必须都是在原队形中连续的；
- 2、左右对称，假设有 m 个人形成新的队形，则第 1 个人和第 m 个人身高相同，第 2 个人和第 $m-1$ 个人身高相同，依此类推，当然如果 m 是奇数，中间那个人可以任意；
- 3、从左到中间那个人，身高需保证不下降，如果用 H 表示新队形的高度，则 $H[1] \leq H[2] \leq H[3] \dots \leq H[mid]$ 。

现在吉哥想知道：最多能选出多少人组成新的完美队形呢？

Input

输入数据第一行包含一个整数 T ，表示总共有 T 组测试数据($T \leq 20$)；

每组数据首先是一个整数 n ($1 \leq n \leq 100000$)，表示原先队形的人数，接下来一行输入 n 个整数，表示原队形从左到右站的人的身高 ($50 \leq h \leq 250$ ，不排除特别矮小和高大的)。

Output

请输出能组成完美队形的最多人数，每组输出占一行。

Sample Input

```
2
3
1 2 1
4
1 2 2 1
```

Sample Output

```
3
4
```

1005 湫湫系列故事——设计风景线

Time Limit: 3 Seconds Memory Limit: 65536K

随着杭州西湖的知名度的进一步提升，园林规划专家湫湫希望设计出一条新的经典观光线路，根据老板马小腾的指示，新的风景线最好能建成环形，如果没有条件建成环形，那就建的越长越好。

现在已经勘探确定了 n 个位置可以用来建设，在它们之间也勘探确定了 m 条可以设计的路线以及他们的长度。请问是否能够建成环形的风景线？如果不能，风景线最长能够达到多少？

其中，可以兴建的路线均是双向的，他们之间的长度均大于 0。

Input

测试数据有多组，每组测试数据的第一行有两个数字 n, m ，其含义参见题目描述；接下去 m 行，每行 3 个数字 $u \ v \ w$ ，分别代表这条线路的起点，终点和长度。

[Technical Specification]

1. $n \leq 100000$
2. $m \leq 1000000$
3. $1 \leq u, v \leq n$
4. $w \leq 1000$

Output

对于每组测试数据，如果能够建成环形（并不需要连接上去全部的风景点），那么输出 YES，否则输出最长的长度，每组数据输出一行。

Sample Input

```
3 3
1 2 1
2 3 1
3 1 1
```

Sample Output

YES