UVa Online Judge

12402 Parallel Missions

Ray has always been a fan of **Lineage II**, the (previously) most popular MMORPG (Massively Multiplayer Online Role-Playing Game) in Korea and Hong Kong. If you have ever tried the game, you'll know that the procedures of the missions are harsh and time consuming. Most of the time is spent on travelling around the world, to kill a definite amount of a specific kind of monster, or to talk to some NPCs (Non-Player Characters). You may of course avoid the optional missions, but some missions are inevitable.

In order to have stronger combat skills or special abilities (e.g. High grade weapon/armor creation), you need to upgrade to advanced jobs (e.g. from Priest to Bishop) through finishing several extremely tiring missions. One way to minimise the duration is to solve several missions simutaneously.

Here's an example:

```
Mission 1 asks player to travel in the following order: 2 -> 3 -> 5 -> 1 -> 2
Mission 2 asks player to travel in the following order: 5 -> 2 -> 3 -> 4 -> 5
Mission 3 asks player to travel in the following order: 4 -> 2 -> 4 -> 1 -> 4
```

Suppose the player is currently in city 4 and assume that the time needed to travel between the cities are the same (say, **1 time unit**). The following route is optimal, taking only 8 time units:

```
4 -> 5 -> 2 -> 3 -> 4 -> 5 -> 1 -> 2 -> 4
```

It's not always easy to pick an optimal path. You are going to **write a program** to help.

Input

Input contains several test cases. The first line of each case gives a single integer which indicates the number of missions to be solved simutaneously. The first number of the next lines indicate the number of procedures of the missions. The integers following are the cities needed to be travelled (in order). The last line of the case consists of a single integer which locates the starting position of the player.

To make this problem easier we impose the following small input ranges:

- There are no more 10 cities in the world, and at most 10 missions.
- The number of procedures of each mission does not exceed 6.
- For an optimal path, travelling time required is always less than 20.

Output

For each test case, output a path such that the travelling time is minimized.

Sample Input

```
3
5 2 3 5 1 2
5 5 2 3 4 5
5 4 2 4 1 4
4
```

Sample Output

```
4 5 2 3 4 5 1 2 4
```