CODECON

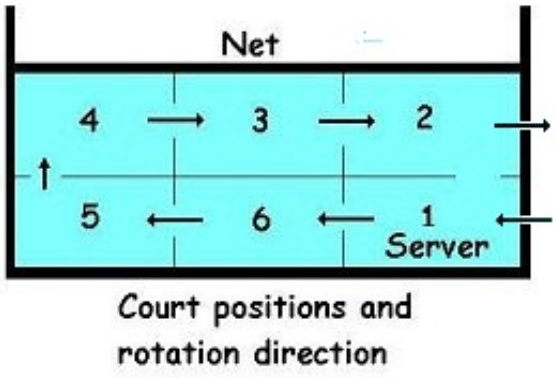
Memory Limit: 1024 MB
Time Limit: 5 s

Volleyball Rotations (100 points)

Introduction

You are the Captain of the **Bloomberg Volleyball Team**. In order to make an effective lineup you want to know who will be on the court after a certain number of **rotations** as well as **where on the court they will be**.

In volleyball, 6 players are on the court at a time and they **rotate clockwise one position** every time their team gains the right to serve. A volleyball court is referenced with numbers 1 - 6. Starting from the lower right corner is position 1 going all the way around the court in a counter-clockwise direction to position 6 at the lower middle.



Court positions and rotation direction

You have **will have anywhere between 7 and 26 people on your team**. Because your team is > 6, there will be players waiting off court on the **sidelines** while six play. Every **rotation**, one of these **sideline** players will come into the court at position 1 (server position), while one player will leave the court at position 2 and join the **sideline at the back of the line**.

The **setter** will **always** be in the **starting lineup** and will not be rotated off the court. This means that when the **setter** is at position 2, when it comes time for a rotation instead of leaving the court and joining the sideline, they will instead go directly to position 1.

Input Specifications

Your program will take from **STDIN:**

- The first line will be the number of rotations that the team will undergo $1 \leq N \leq 200$
- The second line will be the name of the **setter**. Names will only contain alphabetical characters.
- The third line will be the starting lineup. The setter will always be in the starting lineup. Each player name will be a **unique string**. The names are **space delimited**. There will always be **6** people in the starting lineup.
- The fourth line will contain the players on the sidelines. The first token will be the number of players on the sideline $1 \leq S \leq 20$, followed by that many space-delimited unique player names.The first name will be the first to rotate in.

Output Specifications

Based on the input, print out the **lineup** after the **N number of rotations** ordering them based on which **position on the court** they will end up.

Sample Input/Output

Input

```
3
A
A B C D E F
4 G H I J
```

Output

```
I E F A G H
```

Explanation

The starting lineup was:
######<-Net
D C B
E F A
[G H I J]<- Sideline Players

Notice how A is in position 1, B position 2 etc. The input order will correspond to where they are on the court, the first player is in position 1, second position 2 etc.
After 3 rotations, the lineup looks like so:
######<-Net
A F E
G H I
[J B C D]<- Sideline Players

Therefore the answer, starting with player at position 1, is I E F A G H

Input

```
6
A
A B C D E F
4 G H I J
```

Output

```
A G H I J B
```

Explanation

This demonstrates what happens to A (the setter). Since they do not come off the court they rotated back to position 1 on the sixth rotation.

The starting lineup was:
######<-Net
D C B
E F A
[G H I J]<- Sideline Players

After 5 rotations, the lineup looks like so:
######<-Net
H G A
I J B
[C D E F]<- Sideline Players

The last rotation takes A back to position 1 and none of the Sideline Players get to come in. This results in the lineup:

######<-Net
I H G
J B A
[C D E F]<- Sideline Players

Therefore the answer, starting with player at position 1, is A G H I J B