Cardsharper

Zenek is a well known (at least in Byteotia) card-sharper. He spent most of his best years practicing one card shuffle with his deck of **n** cards, which for simplicity we will call 1, 2, ..., **n**. Unfortunately, it turns out that knowing this one card shuffle **a** is not enough to earn a good living. To become rich and famous Zenek needs to know **k** shuffles $\mathbf{c_1}$, ..., $\mathbf{c_k}$. As he doesn't have enough time to learn all of them, he decided to learn only one shuffle **b** so that using both **a** and **b** he will be able to perform as many of $\mathbf{c_i}$ as it is possible.

Each shuffle is described by **n** numbers $\mathbf{t_1}$, $\mathbf{t_2}$, ..., $\mathbf{t_n}$. Such description means that after performing shuffle, card that was originally at position **i** will be at position $\mathbf{t_i}$.

Task

Find shuffle **b** maximizing number of shuffles that can be performed.

Input

First line contains **n** ($2 \le \mathbf{n} \le 52$). Second line contains **n** numbers $\mathbf{a_1}$, $\mathbf{a_2}$, ..., $\mathbf{a_n}$ describing shuffle that Zenek already knows. Third line contains **k** ($2 \le \mathbf{k} \le 6$). **i**-th of the next **k** lines contains description of $\mathbf{c_i}$.

Output

First line contains description of the shuffle **b** that Zenek should learn. **i**-th of the next **k** lines contains:

- -1 when it is not possible to perform $\mathbf{c_i}$ using only **a** and **b**
- **m**, $\mathbf{r_1}$, $\mathbf{r_2}$, ..., $\mathbf{r_m}$ ($0 \le \mathbf{m} \le 500000$, $0 \le \mathbf{r_i} \le 10^6$) meaning that applying **a** $\mathbf{r_1}$ times, then **b** $\mathbf{r_2}$ times, then **a** $\mathbf{r_3}$ times and so on is the same as applying shuffle $\mathbf{c_i}$ once.

Examples

Input

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

Output

```
2 1 3 4 5
3 4 1 1
0
9 1 1 3 1 4 1 1 1 1
```

Input

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

Output

1 3 2 4 5
2 0 1
-1
-1