

# 2021 Canadian Computing Olympiad

## Day 1, Problem 1

### Swap Swap Sort

**Time Limit: 3 seconds**

#### Problem Description

There is an array of  $N$  integers, each with a value between 1 and  $K$ . Your friend has an algorithm that can sort this array according to any ordering of the numbers from 1 to  $K$ . The algorithm performs a sequence of **swap** operations, that exchange two **adjacent** elements of the array. The algorithm performs exactly the minimum number of such swaps to sort the array.

The desired ordering of the numbers from 1 to  $K$  is given by a **target permutation**. A target permutation is a sequence of each number from 1 to  $K$  appearing exactly once, in the same order that is desired by the corresponding ordering.

For example, the array [1 4 2 1 2] sorted by the target permutation 4, 1, 2, 3 results in the array [4 1 1 2 2].

You are interested in the number of swaps performed by your friend's algorithm for different target permutations. To explore this, you start with a target permutation of  $1, 2, \dots, K$  and perform  $Q$  operations on it. Each operation swaps two **adjacent** elements of the target permutation. After performing each operation, find the number of swaps your friend's algorithm would make if it was run with the current target permutation. The  $Q$  operations cumulatively change the target permutation, but do not affect the array.

#### Input Specification

The first line contains the three integers  $N$ ,  $K$ , and  $Q$  ( $1 \leq K \leq N \leq 100\,000$ ,  $1 \leq Q \leq 1\,000\,000$ ).

The next line contains  $N$  integers  $a_1, a_2, \dots, a_N$  ( $1 \leq a_i \leq K$ ) specifying the array.

The next  $Q$  lines each contains a single integer  $j$  ( $1 \leq j \leq K - 1$ ), representing the operation of swapping the elements of the target permutation at indices  $j$  and  $j + 1$ .

For 3 of the 25 available marks,  $N, Q \leq 5\,000$ .

For an additional 3 of the 25 available marks,  $Q \leq 100$ .

For an additional 5 of the 25 available marks,  $K \leq 500$ .

#### Output Specification

For each of the  $Q$  operations, output a line containing a single integer; the answer for the current target permutation.

### Sample Input

5 4 3

1 4 2 1 2

3

2

1

### Output for Sample Input

4

2

2

### Explanation of Output for Sample Input

The three target permutations are 1, 2, 4, 3, then 1, 4, 2, 3, then 4, 1, 2, 3. For the final target permutation, your friend's algorithm uses two swaps to sort the array [1 4 2 1 2] to [4 1 1 2 2].

2021 Canadian Computing Olympiad  
Day 1, Problem 2  
**Weird Numeral System**

**Time Limit: 1.5 seconds**

**Problem Description**

Alice enjoys thinking about base- $K$  numeral systems (don't we all?). As you might know, in the standard base- $K$  numeral system, an integer  $n$  can be represented as  $d_{m-1} d_{m-2} \dots d_1 d_0$  where:

- Each digit  $d_i$  is in the set  $\{0, 1, \dots, K - 1\}$ , and
- $d_{m-1}K^{m-1} + d_{m-2}K^{m-2} + \dots + d_1K^1 + d_0K^0 = n$ .

For example, in standard base-3, you would write 15 as 1 2 0, since  $(1) \cdot 3^2 + (2) \cdot 3^1 + (0) \cdot 3^0 = 15$ .

But standard base- $K$  systems are too easy for Alice. Instead, she's thinking about **weird-base- $K$**  systems.

A weird-base- $K$  system is just like the standard base- $K$  system, except that instead of using the digits  $\{0, \dots, K - 1\}$ , you use  $\{a_1, a_2, \dots, a_D\}$  for some value  $D$ . For example, in a weird-base-3 system with  $a = \{-1, 0, 1\}$ , you could write 15 as 1 -1 -1 0, since  $(1) \cdot 3^3 + (-1) \cdot 3^2 + (-1) \cdot 3^1 + (0) \cdot 3^0 = 15$ .

Alice is wondering how to write  $Q$  integers,  $n_1$  through  $n_Q$ , in a weird-base- $K$  system that uses the digits  $a_1$  through  $a_D$ . Please help her out!

**Input Specification**

The first line contains four space-separated integers,  $K$ ,  $Q$ ,  $D$ , and  $M$  ( $2 \leq K \leq 1\,000\,000$ ,  $1 \leq Q \leq 5$ ,  $1 \leq D \leq 5001$ ,  $1 \leq M \leq 2500$ ).

The second line contains  $D$  distinct integers,  $a_1$  through  $a_D$  ( $-M \leq a_i \leq M$ ).

Finally, the  $i$ -th of the next  $Q$  lines contains  $n_i$  ( $-10^{18} \leq n_i \leq 10^{18}$ ).

For 8 of the 25 available marks,  $M = K - 1 \leq 400$ ,  $K = D \leq 801$ .

**Output Specification**

Output  $Q$  lines, the  $i$ -th of which is a weird-base- $K$  representation of  $n_i$ . If multiple representations are possible, any will be accepted. The digits of the representation should be separated by spaces. Note that 0 must be represented by a non-empty set of digits.

If there is no possible representation, output **IMPOSSIBLE**.

**Sample Input 1**

3 3 3 1

-1 0 1

15

8

-5

**Output for Sample Input 1**

1 -1 -1 0

1 0 -1

-1 1 1

**Explanation of Output for Sample Input 1**

We have:

$$(1) \cdot 3^3 + (-1) \cdot 3^2 + (-1) \cdot 3^1 + (0) \cdot 3^0 = 15,$$

$$(1) \cdot 3^2 + (0) \cdot 3^1 + (-1) \cdot 3^0 = 8, \text{ and}$$

$$(-1) \cdot 3^2 + (1) \cdot 3^1 + (1) \cdot 3^0 = -5.$$

**Sample Input 2**

10 1 3 2

0 2 -2

17

**Output for Sample Input 2**

IMPOSSIBLE

2021 Canadian Computing Olympiad  
Day 1, Problem 3  
**Through Another Maze Darkly**

**Time Limit: 8 seconds**

**Problem Description**

There is a maze that is formed by connecting  $N$  rooms via  $N - 1$  corridors. The rooms are numbered 1 to  $N$  and each room has the shape of a circle. The corridors have the following constraints:

- Each corridor forms a connection between two distinct rooms.
- Every pair of rooms is connected by exactly one path of connected corridors.

One difficulty in navigating through this maze is that the lights are all out, so you cannot see where you are. To help with navigation, each room contains a laser pointer that initially points to a corridor. Consider the following strategy:

- Rotate the room's laser pointer clockwise until it points to another corridor.
- Follow the room's laser pointer and traverse the corridor.
- Repeat the previous two steps indefinitely.

You created  $Q$  queries to investigate this strategy. For each query, you are given an integer  $K$  and you start at room 1. Determine the final room after traversing exactly  $K$  corridors. All laser pointers will reset to their original orientation after each query.

**Input Specification**

The first line contains the integers  $N$  and  $Q$  ( $2 \leq N \leq 800\,000$ ,  $1 \leq Q \leq 800\,000$ ).

The next  $N$  lines describe the layout of the maze, with the  $i^{\text{th}}$  of these lines describing room  $i$ . Specifically, it contains an integer  $k$ , the number of corridors connecting to room  $i$ , followed by  $k$  integers,  $c_1 c_2 \dots c_k$ , indicating the rooms that these  $k$  corridors lead to, in clockwise order. Lastly, room  $i$ 's laser pointer initially points to the corridor leading to room  $c_1$ .

The final  $Q$  lines describe a query, with each line containing an integer  $K$  ( $1 \leq K \leq 10^{15}$ ).

For 4 of the 25 available marks, the  $i^{\text{th}}$  corridor forms a connection between room  $i$  and room  $i + 1$ .

For an additional 4 of the 25 available marks,  $N \leq 2000$  and  $Q \leq 2000$ .

For an additional 12 of the 25 available marks,  $Q = 1$ .

### Output Specification

Output  $Q$  lines answering the queries in order.

### Sample Input

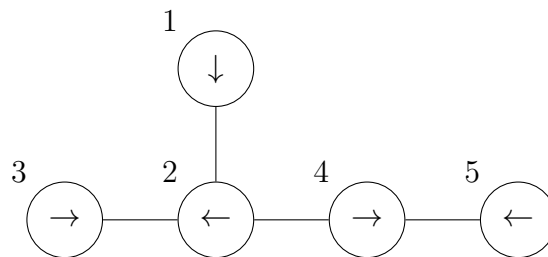
```
5 6
1 2
3 3 1 4
1 2
2 5 2
1 4
1
2
3
4
5
6
```

### Output for Sample Input

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

### Explanation of Output for Sample Input

This is the initial state of the maze.



The strategy will visit these rooms in order:

1, 2, 1, 2, 4, 2, 3, ...