

DMOPC '17 Contest 5 P6 - Bridges

Time limit: 2.5s Memory limit: 256M

You are the leader of an island nation. Your nation consists of N islands conveniently labelled $1, 2, \dots, N$. Each island has v_i inhabitants. Currently, the only way of travelling between any two islands is by sea. You have created plans to build $N - 1$ bridges labelled $1, 2, \dots, N - 1$ to connect your nation. Specifically, bridge i will connect islands i and $i + 1$ for all $i = 1, 2, \dots, N - 1$. Now all that's left is to build these $N - 1$ bridges and the people of your nation will rejoice!

The bridges will be built one at a time. Once each bridge is built, an opening ceremony will be held for that bridge. The bridge will be allowed for public use once the opening ceremony ends. At the opening ceremony for bridge i , all inhabitants who can currently reach island i by land (including already-built bridges) will gather on one side of the bridge and all inhabitants who can currently reach island $i + 1$ by land (including already-built bridges) will gather on the other. This ceremony will generate a unity value u_i which is the product of the number of inhabitants on each side.

This process doesn't seem very interesting, so you decide to play a game with one of your advisors to make things more fun. Your advisor has chosen an order to build the $N - 1$ bridges. They have determined the unity values u_1, u_2, \dots, u_{N-1} which will result from this order. Your task is to find an order to build the bridges which obtains the same array of unity values.

However, your advisor believes that simply giving you their array of unity values will make it too easy. Instead, you are allowed to ask them up to Q questions. You will give your advisor an order to build the bridges - a permutation of $1, 2, \dots, N - 1$. Say that x_i is the unity value of bridge i from the advisor's construction order and y_i is the unity value of bridge i from your construction order. Your advisor will only tell you $|x_1 - y_1| + |x_2 - y_2| + \dots + |x_{N-1} - y_{N-1}|$. **Note that x_i and y_i are not the unity values of the i^{th} bridge in the construction order. They are the unity values of bridge i , once it has been constructed.**

Constraints

$1 \leq v_i \leq 100$ for all subtasks.

Subtask 1 [30%]

$2 \leq N \leq 20$
 $Q = 2N^2$

Subtask 2 [70%]

$2 \leq N \leq 400$
 $Q = 2N$

Input Specification

The first line will contain two space-separated integers N and Q .

The next line will contain N space-separated integers. The i^{th} of these will be v_i , the number of inhabitants on island i .

Interaction

This is an interactive problem. After reading the initial two lines of input, your program can make queries.

Each query must be a single line containing $N - 1$ space-separated integers. These integers must be a permutation of $1, 2, \dots, N - 1$ indicating the order which the bridges should be built. The first integer will indicate the index of the first bridge built, and so on.

After printing and flushing, a new line with a single integer will appear as input. It will be the value $|x_1 - y_1| + \dots + |x_{N-1} - y_{N-1}|$ as described in the problem statement.

You cannot make more than Q queries. Also, you do not need to use all Q queries.

Your program will be deemed correct if the value $|x_1 - y_1| + \dots + |x_{N-1} - y_{N-1}|$ is ever 0 for your queries. The judge will halt interaction once it has printed 0.

Note: To flush, you can use `fflush(stdout);` in C++, or `System.out.flush();` in Java, or `import sys; sys.stdout.flush()` in Python 2/3. For other languages, search in its documentation.

Sample Interaction

`>>>` denotes your output. You should not print this out in your actual program.

```
5 10
1 2 3 2 1
>>> 1 2 3 4
12
>>> 1 2 4 3
24
>>> 1 3 2 4
0
```

Explanation for Sample

The advisor's order is `3 1 2 4`. Then $u_3 = 3 \cdot 2 = 6$, $u_1 = 1 \cdot 2 = 2$, $u_2 = (1 + 2) \cdot (3 + 2) = 15$, $u_4 = (1 + 2 + 3 + 2) \cdot 1 = 8$.

The first query gives the following unity values:

$u_1 = 1 \cdot 2 = 2$, $u_2 = (1 + 2) \cdot 3 = 9$, $u_3 = (1 + 2 + 3) \cdot 2 = 12$, $u_4 = (1 + 2 + 3 + 2) \cdot 1 = 8$.

The second query gives the following unity values: $u_1 = 1 \cdot 2 = 2$, $u_2 = (1 + 2) \cdot 3 = 9$, $u_4 = 2 \cdot 1 = 2$, $u_3 = (1 + 2 + 3) \cdot (2 + 1) = 18$.

The third query gives the following unity values: $u_1 = 1 \cdot 2 = 2$, $u_3 = 3 \cdot 2 = 6$, $u_2 = (1 + 2) \cdot (3 + 2) = 15$, $u_4 = (1 + 2 + 3 + 2) \cdot 1 = 8$.