Light up!

If you want to keep up with the Joneses, you can't start early enough putting up christmas decorations. Unfortunately only few things are more detrimental to

(a) your overall mental health

(b) any form of interhuman relations

than getting those damned christmas lights to work.

Building on experience from previous years, Walter Hoppenstedt is worried about (b) and hence has already precautionally complimented the rest of the family out of the house to do their christmas shopping. However, this means that he now has to do all the work on his own—and hence (a) is in acute danger.

Summoning up all of his acrobatic talent Walter has already strung the n ($1 \leq n \leq 1\,000$) christmas lights in a symmetric and aesthetically pleasing fashion around the fir tree in their garden and has found an unobtrusive way to connect them to an electrical outlet in the basement. However, after plugging in he notices that some lights haven't lit up.

For each light, there are two possible reasons for this: either the bulb is not working or the socket isn't. Walter now wants to put the bulbs into the sockets in such a way that the maximal number of lights will shine. However, with no ammeter at hand, the only way to get information about the christmas lights is to pull some bulbs out of their sockets and then put them into other sockets.

However, since this means that he first has to unplug the christmas lights (nobody wants to get electrocuted, after all) and then plug them in again (so he has to walk down into the basement twice, which takes quite some time), we may assume that the time for one such attempt is independent of the number of sockets changed. However, Walter has only time for 1 500 attempts remaining until the rest of his family will come back home. To prevent the potential disgrace of needing their help, write a program that tells him which bulbs to put into which sockets if he wants the maximal amount of bulbs to light up.

Remark

Fortunately the christmas lights in question are a rather modern product—meaning that a single broken bulb or socket won't cause all the other lights to remain dark, too. In other words: any given light will shine precisely if neither its bulb nor socket are broken.

Implementation

This is an interactive task. Initially, you are given the number n of lights on standard input. Afterwards you can issue the following commands by writing the respective lines to standard output. You can read the answer to your query (if applicable) from standard input directly after this.

The available commands are as follows:

- **Q** x_1 … x_n, where the x_i are pairwise distinct integers between 1 and n, asks Walter to put the x_i^{th} bulb into the i^{th} socket ($i = 1, \ldots, n$). You will get a line consisting of n integers as answer: here the i^{th} is integer is 1 if the respective light is shining, otherwise 0.
- **A** x_1 … x_n (with the same constraints as before) is to be issued precisely once—your program will automatically be terminated afterwards. This means that you are convinced that the given assignment of bulbs to sockets is optimal.

Your program is only allowed to issue 1 500 queries of the first type. If your output does not satisfy these conditions, you will get “Wrong Answer” for the respective testcase.

Important notice

In order to ensure that communication between the two programs works, you have to flush standard output after each of your program's outputs. When using *cout* it suffices to append ≪ *flush* to each output, e.g.

cout ≪ "Q 1 2 3 4\n" ≪ *flush*;

or to always use *endl* for line breaks. When using *printf* you have to call *fflush*(*stdout*) after each output.

Constraints

We always have $1 \le n \le 1\,000$.

Subtask 1 (20 points). $n \le 5$

Subtask 2 (20 points). $n \le 30$

Subtask 3 (40 points). $n \le 400$

Subtask 4 (20 points). No further constraints.

Sample case

The public testcase describes $N = 4$ christmas lights with precisely the first two bulbs and the last two sockets working.

Your program	Grader	Explanation
	`4`	the number of lights
`Q 1 2 3 4`		
	`0 0 0 0`	no light is shining
`Q 4 3 2 1`		
	`0 0 1 1`	the last two lights are shining
`A 4 3 2 1`		your program wants to output the answer your solution is correct and gets accepted

Limits

Time: 1.5 s
Memory: 256 MiB

The above time limit also contains the runtime of our grader; however, it is guaranteed that our program needs at most half of the above time to answer the 1 500 queries allowed.

Feedback

restricted feedback is given for this task, i.e. the score shown equals your real score on this submission. However, for each testcase group you are only shown the verdict for the first testcase with minimal score in that group. (The order of the testcases in each of the groups is fixed.)