

Art Collections (art)

While your days as an art thief are long past, this does not mean that you lost interest in contemporary art. Unfortunately, you've been pretty busy lately with BOI preparations. That's why you have lost track of how the N hottest contemporary art collections (conveniently numbered from 1 to N) rank according to value. Since simply asking someone would be quite embarrassing, you will have to resort to different means: *anonymous online rankings*.

That is, you will repeatedly do the following: You first guess a ranking of the N art collections (based on their value, most expensive first), then publish this ranking on some website, and finally wait for the collection owners' complaints in the comments section. As you don't want to read each individual comment, you will only keep track of the total number of complaints you receive. Fortunately, the owners' behaviour is very reliable: Each of them will complain exactly once for each collection that ranks higher than their own in your guessed ranking although it doesn't in the true ranking, but none will complain about collections you erroneously guessed to rank lower than theirs. You can assume that the values of all collections are distinct.



"Readers also liked: 13 SHOCKING applications of Dijkstra's algorithm computer scientists don't want you to know about!"

However, as publishing a ranking puts your anonymity at risk,* you only want to publish up to 4 000 guessed rankings before finding the correct ranking of the collections. Write a program that helps you to decide which rankings to publish!

Communication

This is a communication task. You must implement the function **void solve(int N)** where N is as described above. For each testcase, this function is called exactly once. Inside *solve*, you can use the following other functions provided by the grader:

- **int publish(std::vector<int> R)** publishes a ranking R of the collections on the website. R must be a permutation of the numbers 1 to N with the collections you guess to be more expensive first. The function returns the number of complaints you receive after publishing this ranking. You can call this function at most 4 000 times per testcase.
- **void answer(std::vector<int> R)** states that you have found the correct ranking R of the collections, in the same format as for *publish*. You must call *answer* exactly once; your program will be automatically terminated afterwards.

If any of your function calls does not satisfy the above constraints, your program will be immediately terminated and judged as **Not correct** for the respective testcase. You must not write to standard output or read from standard input; otherwise you may receive the verdict **Security violation!**

You must include the file `art.h` in your source code. To test your program locally, you can link it with `sample_grader.cpp`, which can be found in the attachment for this task in CMS (see below for a description of the sample grader, and see `sample_grader.cpp` for instructions on how to run it with your program). The attachment also contains a sample implementation as `art_sample.cpp`.

* Definitely because of your distinctive writing style and not because you have a tendency to accidentally sign them with your name.



Constraints

We always have $2 \leq N \leq 4\,000$.

Subtask 1 (5 points). $N \leq 6$

Subtask 2 (15 points). $N \leq 40$

Subtask 3 (15 points). $N \leq 250$

Subtask 4 (15 points). $N \leq 444$

Subtask 5 (20 points). $N \leq 2\,000$

Subtask 6 (30 points). No further constraints.

Example Interaction

Consider a testcase with $N = 3$ where collection 1 is the most expensive, followed by 3, and then collection 2 being the least expensive. First, the grader calls your function *solve* as *solve(3)*. Then, a possible interaction between your program and the grader could look as follows:

Your program	Return value	Explanation
<i>publish</i> ({1, 2, 3})	1	you get a single complaint from the owner of collection 3
<i>publish</i> ({2, 3, 1})	3	you get two complaints from the owner of collection 1 and one from the owner of collection 3
<i>answer</i> ({1, 3, 2})	—	you are convinced that you have found the correct ranking your solution is correct and is accepted

Grader

The sample grader first expects on standard input two lines. The first line should contain the integer N . The second line should contain a list of N space-separated integers, the correct ranking of the collections in the same format as for *publish* and *answer*. Then, the grader calls *solve(N)* and writes to standard output a protocol of all grader functions called by your program. Upon termination, it writes one of the following messages to standard output:

Invalid input. The input to the grader via standard input was not of the above format.

Invalid published ranking. You called *publish* with invalid parameters.

Too many published rankings. You called *publish* more than 4 000 times.

No answer. The function *solve* terminated without calling *answer*.

Wrong answer. You called *answer* with an incorrect ranking.

Correct: p published ranking(s). You called *answer* with the correct ranking and there were p calls to *publish*.

In contrast, the grader actually used to judge your program will only output **Not correct** (for any of the above errors) or **Correct**. Both the sample grader and the grader used to judge your program will terminate your program automatically whenever one of the above errors occurs or after your program calls *answer*.



BOI 2022
Lübeck, Germany
April 28 – May 3, 2022

Day 1
Task: **art**
Language: **en**

Limits

Time: 3 s

Memory: 512 MiB