



Task Permutation

 0.69 sec.  256 MB

For some of you this may be your first encode-decode task. We are sure that you will like it 😊

You are given a permutation* of the numbers from 1 to N. You aim to put as many -1s in it as possible. The detailed description is below.

*Permutation - any sequence made up of N elements in which every number from 1 to N occurs exactly once.



Implementation details

Your task is to implement two functions:

```
std::vector<int> encode(std::vector<int> a);
```

It will be called once by the jury and must encode the permutation a . You encode it by replacing numbers of the permutation with -1 . Once you have decided which numbers to replace, you return the changed permutation.

```
std::vector<int> decode(std::vector<int> a);
```

This function will be called once by the jury and must decode the permutation which you encoded in the previous function (that is, as an argument you receive your permutation with the -1s that you put in it). You have to decode it by replacing every -1 with the number which is the smallest one unused so far. The goal is that after these changes the restored permutation is the same as the one that was given to encode.

Your code **cannot** contain global memory! This means that the two functions cannot communicate and store data which the other one can access.



Constraints

- $1 \leq N \leq 10^5$



Scoring

Let the number of -1s that you have placed in encode be g , and the optimal number be x . If the permutation returned by decode is different from the original one, you get 0 points on the respective test. Otherwise, the points are computed by the following formula:

$$points = \min(1, \frac{g}{x})$$



Subtasks

Subtask	Points	Additional constraints
1	9	$a_i = i + 1, 0 \leq i \leq N - 1$
2	11	$1 \leq N \leq 16$ or 17
3	13	In the sequence with optimally placed -1s, it occurs only once.
4	67	There are no constraints.



Example

Jury	Contestant
$encode(4, 1, 2, 3, 5)$	$4, -1, 2, 3, 5$
$decode(4, -1, 2, 3, 5)$	$4, 1, 2, 3, 5$



Sample grader

The first line contains the size of the sequence N , and the next line – the sequence itself. The grader prints whether the sequence matches on the two calls and how many -1s have been placed.