E. Blog Post Rating

It's well-known that blog posts are an important part of Codeforces platform. Every blog post has a global characteristic changing over time — its community rating. A newly created blog post's community rating is 0. Codeforces users may visit the blog post page and rate it, changing its community rating by +1 or -1.

Consider the following model of Codeforces users' behavior. The i-th user has his own estimated blog post rating denoted by an integer a_i. When a user visits a blog post page, he compares his estimated blog post rating to its community rating. If his estimated rating is higher, he rates the blog post with +1 (thus, the blog post's community rating increases by 1). If his estimated rating is lower, he rates the blog post with -1 (decreasing its community rating by 1). If the estimated rating and the community rating are equal, user doesn't rate the blog post at all (in this case we'll say that user rates the blog post for 0). In any case, after this procedure user closes the blog post page and never opens it again.

Consider a newly created blog post with the initial community rating of 0. For each of n Codeforces users, numbered from 1 to n, his estimated blog post rating a_i is known.

For each k from 1 to n, inclusive, the following question is asked. Let users with indices from 1 to k, **in some order**, visit the blog post page, rate the blog post and close the page. Each user opens the blog post only after the previous user closes it. What could be the maximum possible community rating of the blog post after these k visits?

Input

The first line contains a single integer n ($1 \le n \le 5 \cdot 10^5$) — the number of Codeforces users.

The second line contains n integers $a_1, a_2, ..., a_n$ ($-5 \cdot 10^5 \le a_i \le 5 \cdot 10^5$) — estimated blog post ratings for users in order from 1 to n.

Output

For each k from 1 to n, output a single integer equal to the maximum possible community rating of the blog post after users with indices from 1 to k, in some order, visit the blog post page, rate the blog post, and close the page.

Examples

input

```
4
2 0 2 2
```

output

```
1
1
2
2
```

input

```
7
2 -3 -2 5 0 -3 1
```

output

```
1
0
-1
0
1
```

```
1
2
```