Increasing Split

Time limit: 1.5 seconds
Memory limit: 1024 MiB

Boris and Ihor have found a sequence a of N positive integers, $a_0, a_1, \ldots, a_{N-1}$, and want to split it between themselves. Unable to agree on the split, they have asked you to act as their arbiter.

Before making any decisions, you are shown the entire sequence. You then process the elements of a from left to right – first a_0, then a_1, and so on up to a_{N-1}. As each element is processed, you must give it to exactly one of Boris and Ihor.

Both of them insist that the elements they receive form a strictly increasing sequence in the order they are received. That is, every element you give to a person must be strictly greater than the previous element you gave to that same person. The first element a person receives may have any value, and it is possible that one of them may receive no elements at all.

Boris and Ihor also cannot agree on how many elements each of them should receive. Therefore, for every integer K from 0 to N, you should determine whether it is possible to split the elements so that Boris receives exactly K elements and the two resulting sequences are strictly increasing. You should consider each value of K as an independent question, which means that all $N + 1$ split processes are fully independent from each other.

For example, let $a = [3, 1, 4, 5, 5]$ and consider the following cases:

- for $K = 3$: you may give $a_0 = 3$ to Boris, $a_1 = 1$ to Ihor, $a_2 = 4$ to Boris, $a_3 = 5$ to Boris, and $a_4 = 5$ to Ihor. In the end, Boris receives 3, 4 and 5 in that order, and Ihor receives 1 and 5, in that order. Both resulting sequences are strictly increasing and Boris has received 3 elements, so $K = 3$ is possible.
- for $K = 0$: Boris receives nothing and Ihor receives everything. Ihor's sequence begins with $3, 1$, which is not strictly increasing, so $K = 0$ is impossible.

For this example, the only possible valid values of K are 2 and 3.

Implementation details

The function you must implement is:

```
std::vector<bool> increasing_split(std::vector<int> a)
```

- a: the sequence of numbers of size N;

This function must return a boolean array of size exactly $N + 1$. The element at index i must be `true` if the elements can be split so that Boris receives exactly i of them and the two resulting sequences are strictly increasing, and `false` otherwise. This function is called exactly once per test.

Constraints

- $2 \leq N \leq 4 \cdot 10^5$;
- $1 \leq a_i \leq 10^9$ for each $0 \leq i < N$.

Examples

Sample grader input	Sample grader output
5 3 1 4 5 5	001100
4 1 2 3 4	11111

Explanation of example 1

We have $a = [3, 1, 4, 5, 5]$, the same sequence used in the statement above. For each value of K:

- $K = 0$: as explained in the statement, no valid split exists, so the answer is 0;
- $K = 1$: it can similarly be shown that no valid split exists in which Boris receives exactly one element, so the answer is 0;
- $K = 2$: you may give $a_1 = 1$ and $a_4 = 5$ to Boris, and $a_0 = 3$, $a_2 = 4$ and $a_3 = 5$ to Ihor. Boris receives 1 and 5 in that order, Ihor receives 3, 4 and 5 in that order; both sequences are strictly increasing, so the answer is 1;
- $K = 3$: as explained in the statement, a valid split exists, so the answer is 1;
- $K = 4$ and $K = 5$: no valid split exists, so the answer is 0.

Explanation of example 2

We have $a = [1, 2, 3, 4]$, which is already strictly increasing. Therefore, regardless of how the elements are split between Boris and Ihor, each of them receives a strictly increasing sequence. Hence every value of K from 0 to 4 is possible.

Subtasks

Subtask	Points	N	Additional constraints
0	0	–	The examples.
1	10	$N \leq 18$	–
2	5	$N \leq 4 \cdot 10^5$	$a_i \leq a_{i+1}$ for each $0 \leq i < N-1$.
3	5	$N \leq 4 \cdot 10^5$	$a_0 \geq \max(a_1, a_2, \ldots, a_{N-1})$.
4	16	$N \leq 4 \cdot 10^5$	a is a permutation of all integers from 1 to N. For every index $0 \leq i < N-1$ such that $a_i < a_{i+1}$, the prefix $a_0, a_1, \ldots, a_i$ is a permutation of all integers from 1 to $i+1$.
5	21	$N \leq 5\,000$	a is a permutation of all integers from 1 to N. For every index $0 \leq i < N-1$ such that $\max(a_0, a_1, \ldots, a_i) < a_{i+1}$, the prefix $a_0, a_1, \ldots, a_i$ is a permutation of all integers from 1 to $i+1$.
6	17	$N \leq 4 \cdot 10^5$	a is a permutation of all integers from 1 to N. For every index $0 \leq i < N-1$ such that $\max(a_0, a_1, \ldots, a_i) < a_{i+1}$, the prefix $a_0, a_1, \ldots, a_i$ is a permutation of all integers from 1 to $i+1$.
7	16	$N \leq 5\,000$	–
8	10	$N \leq 4 \cdot 10^5$	–

Sample grader

The input format is the following:

- line 1: one integer – the value of N;
- line 2: N integers $a_0, a_1, \ldots, a_{N-1}$ – the elements of the sequence.

If the returned array does not have size $N+1$, the sample grader will output the verdict `WA: Returned array does not have size N+1`. Otherwise, the output format is the following:

- line 1: a binary string of length $N+1$. Its character at index i is `1` if $K = i$ is possible, and `0` otherwise.