Problem J. Apple Family Reunion

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Background

One rainy night, a hooded mare gallops into an empty marketplace and enters a darkened curio shop. She begins to rummage around the shop until a small candle is lit.

"May I help you, traveler?" A stallion, who is the shopkeeper, comes out and asks. Stepping behind the counter, he surmises that something powerful drew the customer to his shop; she points to an amulet inlaid with a crimson gem kept under glass on the shelf behind him.

"Ah, you have a keen eye. The Alicorn Amulet is one of the most mysterious and powerful of all the known magical charms. Uh, ah — I'm afraid this is... far too dangerous."

(The mare tosses a big bag of gold bits onto the counter.)

"Would you like that gift-wrapped?"

...

The next morning, all ponies awake to find themselves changed into permutations!

There are $n!$ ponies in Equestria. The evil magic turned all ponies into $n!$ different permutations of length n.

In order to reduce the panic, Applejack, who changed into permutation A, decides to gather her family as soon as possible. She notices that, for another permutation P, if she can change herself into P with several applications of One-Two-Three-transformation, then P is a member of the Apple family.

The One-Two-Three-transformation goes as follows. Choose 3 adjacent elements A_i, A_{i+1}, A_{i+2}. If A_i is the median of them, you can put it behind A_{i+2}, in other words, transform these three elements into A_{i+1}, A_{i+2}, A_i. If A_{i+2} is the median of them, you can put it in front of A_i, in other words, transform these three elements into A_{i+2}, A_i, A_{i+1}.

In fact, if a permutation could change into another with One-Two-Three-transformations, then they belong to the same family. Let us choose the lexicographically smallest permutation in each family as the representative of that family. Then we can sort all families by the lexicographical order of their representatives, obtaining the list of families $F_1, F_2, \ldots$. For example, the permutation $(1, 2, \ldots, n)$ always belongs to the family F_1.

Each pony should go to the temporary shelter of their family. A permutation $P \in F_i$ implies that pony P should go to shelter i to meet its family.

Please help Applejack find which shelter the Apple family should go to, and how many ponies are in the Apple family.

Input

The first line contains one integer n ($3 \le n \le 10^5$).

The second line contains n integers $A_1, A_2, \ldots, A_n$, denoting the permutation A Applejack changed into.

Output

Output one line containing one integer.

We know there exists a unique i such that $A \in F_i$. **If $i \le 65\,472$, print the number of ponies in the Apple family modulo $998\,244\,353$. Otherwise, just print i modulo $998\,244\,353$ as the answer.**

Example

standard input	*standard output*
3 2 1 3	2

Note

There are 4 families in total:

- $F_1 = \{(1, 2, 3)\}$
- $F_2 = \{(1, 3, 2),\ (2, 1, 3)\}$
- $F_3 = \{(2, 3, 1),\ (3, 1, 2)\}$
- $F_4 = \{(3, 2, 1)\}$

So the Apple family is $F_2 = \{(1, 3, 2),\ (2, 1, 3)\}$, and their representative is $(1, 3, 2)$. So, print the number of ponies in the Apple family, which is 2.