Fast Bogosort

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

Have you heard of Bogosort? Bogosort is a very interesting random sorting algorithm, which can sort an array of n elements in the best-case time complexity of $O(n)$. The algorithm can be described as follows:

- Uniformly shuffle the entire array.
- Check if the array is sorted.

Yes, after reading the algorithm above, you will realize that the expected time complexity of Bogosort is actually $O(n \cdot n!)$!

Bogosort is so fascinating, but it runs so slowly, which is quite unfortunate. Therefore, let's help improve it by optimizing it using the following algorithm, Fast Bogosort, to sort a permutation of numbers from 1 to n:

- If the array is already sorted, stop.
- Otherwise, divide the array into as many segments as possible, such that each segment corresponds to the interval $[l_i, r_i]$ and contains exactly the permutation of numbers $[l_i, r_i]$. Note that the segments must not overlap and their union must be the entire range $[1, n]$.
- For each segment $[l, r]$ that has been divided, **if** $l < r$, call `shuffle(l, r)` to uniformly shuffle the numbers within that segment.

If we only care about the number of calls to `shuffle(l, r)`, it is clear that the original Bogosort has an expected number of calls of $n!$. Now, given a permutation of numbers from 1 to n, please calculate the expected number of times Fast Bogosort will call `shuffle(l, r)`.

Input

The first line of the input contains one integer n ($1 \le n \le 10^5$), indicating the length of the array.

The second line contains n integers $a_1, a_2, \ldots, a_n$, indicating the content of the array. It is guaranteed that $a_1, a_2, \ldots, a_n$ form a permutation of $1 \sim n$.

Output

Output the expected number of times Fast Bogosort will call `shuffle(l, r)`. It can be proved that the expected number is always a rational number. Additionally, under the constraints of this problem, it can also be proved that when that value is represented as an irreducible fraction $\frac{P}{Q}$, we have $Q \not\equiv 0 \pmod{998244353}$. Thus, there is a unique integer R such that $R \times Q \equiv P \pmod{998244353}, 0 \le R < 998244353$. Report this R.

Examples

standard input	standard output
5 2 1 5 3 4	332748123
10 4 2 3 1 6 5 9 7 10 8	453747445