Selection Sort Count

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Nikuniku learns the Selection Sort today. To better understand this algorithm, she implements it again and computes the number of swaps for each round, as the following pseudocode shows:

Algorithm 1 Selection Sort

```
1: function SORT(Permutation P)
2:     for i in [1, n − 1] do
3:         c_i ← 0
4:         for j in [i + 1, n] do
5:             if P_i > P_j then
6:                 swap(P_i, P_j)
7:                 c_i ← c_i + 1
```

Nikuniku wants to know how many different kinds of permutations P of length n there are, so that after running the algorithm, the number of swaps of each round (denoted as the counter c in the pseudocode) matches a given $t_1, \ldots, t_{n-1}$. You only have to output the answer modulo 998244353. A permutation of length n consists of $1, \ldots, n$ exactly once.

Input

The first line has one integer n ($2 \leq n \leq 2 \cdot 10^5$).

In the following line, there are $n-1$ integers $t_1 \ldots t_{n-1}$. It is guaranteed that there exists at least one permutation satisfying the requirement.

Output

Output the answer in one line, modulo 998244353.

Examples

standard input	standard output
3 1 1	2
6 0 1 1 1 0	4
6 0 1 2 1 1	6