Problem D. Classical DP Problem

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Let us consider a grid of squares with n rows and n columns. Arbok has cut out some part of the grid so that, for each $i = 1, 2, \ldots, n$, only the leftmost a_i squares are remaining in the i-th row from the top. The values of a_i satisfy $a_1 \le a_2 \le \ldots \le a_n$: that is, the grid looks like a Young diagram. Now, Arbok wants to place rooks into some of the remaining squares of the grid.

A rook is a chess piece that occupies one square and can move horizontally or vertically, through any number of unoccupied squares.

Let's say that a square is covered if it either contains a rook, or a rook can be moved to this square in one move.

Find r, the smallest number of rooks Arbok needs to place into some of the remaining squares so that every remaining square is covered. Also find w, the number of ways to put r rooks to satisfy the same condition, modulo $998\,244\,353$.

Input

The first line contains a single integer n, denoting the size of the grid ($1 \le n \le 5000$).

The second line contains n integers $a_1, a_2, \ldots, a_n$, denoting the widths of the rows left by Arbok ($1 \le a_1 \le a_2 \le \ldots \le a_n \le n$).

Output

Print two integers r and w, denoting the smallest number of rooks Arbok needs to place so that every remaining square is covered, and the number of ways to put r rooks to achieve the same, modulo $998\,244\,353$.

Example

standard input	*standard output*
3 1 2 3	2 6

Note

In the first example test, one rook is not enough to cover every square, but two rooks are enough, and there are six ways to place two rooks to cover every square (R denotes a rook, * denotes an empty square):

```
R       *       *       *       *       *
**      R*      R*      R*      *R      **
*R*     R**     *R*     **R     R**     RR*
```