Problem B. Queue Sorting

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Miss Burger has learned a lot of data structures recently. Her favorite one is the queue. Now, she wants to use two queues to implement a sorting algorithm.

To do so, she creates two queues A and B in the beginning. They are all initialized to be empty. For a queue, you can only conduct two kinds of operations:

- Insert: inserting an element to the back of the queue.
- Eject: acquire and remove the first element in the front.

The integers we will sort only include 1 to n. For each i ($1 \le i \le n$), we let a_i denote the amount of i in total. Therefore, $\sum_{i=1}^{n} a_i$ integers are to be sorted in total, denoted as m.

Miss Burger first rearranges all m integers into a sequence b in arbitrary order. We say b is legal if and only if she can sort all integers in non-decreasing order in the following way:

- For each integer in b in order, she inserts it into one of the two queues.
- After all integers are inserted, she begins to eject them out one by one in arbitrary order. Here, each time, she can choose to eject an integer from one of the non-empty queues and concatenate it into the resulting sequence.

Now, Miss Burger wants to know how many different sequences $\{b_i\}_{i=1\ldots m}$ can be sorted by her two queues. Two sequences $\{b_i\}_{i=1\ldots m}$ are considered different if there exists a position i where their values b_i of these two sequences are different. As the answer may be large, your output should be modulo 998 244 353.

Input

The first line contains an integer n ($1 \le n \le 500$), denoting the types of numbers.

The second line contains n integers, $a_1, \ldots, a_n$ ($0 \le a_i \le n$), where a_i denotes the amount of number i.

It is guaranteed that $1 \le \sum_{i=1}^{n} a_i \le 500$.

Output

Output a single line containing only one integer, denoting the number of legal sequences.

As the answer may be large, your output should be modulo 998 244 353.

Example

standard input	standard output
4 1 1 1 1	14