

## C. Eleventh Birthday

time limit per test: 2 seconds  
memory limit per test: 512 megabytes  
input: standard input  
output: standard output

It is Borya's eleventh birthday, and he has got a great present:  $n$  cards with numbers. The  $i$ -th card has the number  $a_i$  written on it. Borya wants to put his cards in a row to get one greater number. For example, if Borya has cards with numbers 1, 31, and 12, and he puts them in a row in this order, he would get a number 13112.

He is only 11, but he already knows that there are  $n!$  ways to put his cards in a row. But today is a special day, so he is only interested in such ways that the resulting big number is divisible by eleven. So, the way from the previous paragraph is good, because  $13112 = 1192 \times 11$ , but if he puts the cards in the following order: 31, 1, 12, he would get a number 31112, it is not divisible by 11, so this way is not good for Borya. Help Borya to find out how many good ways to put the cards are there.

Borya considers all cards different, even if some of them contain the same number. For example, if Borya has two cards with 1 on it, there are two good ways.

Help Borya, find the number of good ways to put the cards. This number can be large, so output it modulo 998244353.

### Input

Input data contains multiple test cases. The first line of the input data contains an integer  $t$  — the number of test cases ( $1 \leq t \leq 100$ ). The descriptions of test cases follow.

Each test is described by two lines.

The first line contains an integer  $n$  ( $1 \leq n \leq 2000$ ) — the number of cards in Borya's present.

The second line contains  $n$  integers  $a_i$  ( $1 \leq a_i \leq 10^9$ ) — numbers written on the cards.

It is guaranteed that the total number of cards in all tests of one input data doesn't exceed 2000.

### Output

For each test case output one line: the number of ways to put the cards to the table so that the resulting big number was divisible by 11, print the number modulo 998244353.

### Example

input
4 2 1 1 3 1 31 12 3 12345 67 84 9 1 2 3 4 5 6 7 8 9
output
2 2 2 31680