Beggar Robin

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

There are n people standing in a row. The i-th person has a_i coins.

Beggar Robin wants to help everyone become more equal. In one operation, he chooses two people and distributes their money equally between them. Formally, he chooses two positions i and j such that $1 \le i \le j \le n$, computes $x = \frac{a_i + a_j}{2}$, and then sets $a_i := x$ and $a_j := x$.

But there is one problem: coins are indivisible. If $a_i + a_j$ is odd, Robin cannot split one coin into two halves. In that case, Robin happily takes this one coin as his salary, and distributes the rest equally.

Robin is very hungry, so he wants to earn his salary (i.e. to take at least one coin) as soon as possible.

For an array $a_1, a_2, \ldots, a_n$ of positive integers, define $f(a_1, a_2, \ldots, a_n)$ as the minimum number of operations Robin needs to perform to earn his salary. If Robin can never earn his salary, then $f(a_1, a_2, \ldots, a_n) = 0$.

You are given an array $a_1, a_2, \ldots, a_n$. Your task is to calculate

$$\sum_{l=1}^{n} \sum_{r=l}^{n} f(a_l, \ldots, a_r)^2$$

modulo 998 244 353.

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases.

The first line of each test case contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the length of the array.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the numbers of coins the people have.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output a single integer — the sum of squared values of f over all subarrays of the given array, modulo 998 244 353.

Example

standard input	standard output
4 3 2 18 2 10 1 2 3 4 5 6 7 8 9 10 5 20 36 100 36 20 2 1000000000 999999999	25 45 174 1