

# Binary String

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            **2 seconds**  
Memory limit:         **1024 megabytes**

You are given a binary string  $a_0a_1a_2 \dots a_{n-1}$  arranged on a cycle. Each second, you will change every 01 to 10 simultaneously. In other words, if  $a_i = 0$  and  $a_{(i+1) \bmod n} = 1$ , you swap  $a_i$  and  $a_{(i+1) \bmod n}$ . For example, we will change 100101110 to 001010111.

You need to answer how many different strings will occur in infinite seconds, modulo 998244353.

Note: Two strings  $a_0a_1 \dots a_{n-1}$  and  $b_0b_1 \dots b_{n-1}$  are different if there exists an integer  $i \in \{0, 1, \dots, n-1\}$  such that  $a_i \neq b_i$ . Thus, the cyclic shifts of a string may be different from the original string.

## Input

The first line contains an integer  $T$  ( $1 \leq T \leq 10^6$ ) – the number of test cases.

For each test case, the first line contains a binary string  $a_0a_1 \dots a_{n-1}$  ( $a_i \in \{0, 1\}$ ).

It is guaranteed that the sum of lengths of strings over all test cases does not exceed  $10^7$ .

## Output

For each test case, output one integer representing the answer in one line.

## Example

standard input	standard output
3	1
1	3
001001	9
0001111	