Balanced Binary String

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 512 megabytes

Chiaki has a string s of length n, consisting of '0', '1' and '?'.

A circular substring $s(i, l)$ of $s = s_1 s_2 \dots s_n$ is string $\begin{cases} s_i s_{i+1} \dots s_{i+l-1} & i+l-1 \le n \\ s_i s_{i+1} \dots s_n s_1 s_2 \dots s_{i+l-1-n} & i+l-1 > n \end{cases}$.

A binary string s of length n is *balanced* if for every two circular substrings $s(i, l)$ and $s(j, l)$ $(1 \le i, j, l \le n)$, the number of 1's in $s(i, l)$ and $s(j, l)$ differ at most by one. For example, 101 and 11010110 are balanced, while 1100 and 1010110110 are not balanced.

Chiaki would like to know the number of ways to replace every '?' to '0' or '1' in her string to make it balanced. Since this number may be very large, you are only asked to calculate it modulo $10^9 + 7$.

Input

There are multiple test cases. The first line of input contains an integer T, indicating the number of test cases. For each test case:

The first line contains a nonempty string s $(1 \le |s| \le 1024)$ consisting of '0', '1' and '?'.

It is guaranteed that the sum of $|s|$ over all test cases does not exceed 1024.

Output

For each test case, output an integer denoting the number of ways.

Example

standard input	standard output
10 ? ?? ??1 ???0 ????1 ?????0 ??????1 ???????0 ????????1 ?????????0	2 4 4 6 11 11 22 22 31 32