

Champernowne Substring

Input file: **standard input**
Output file: **standard output**
Time limit: 10 seconds
Memory limit: 2048 megabytes

The Champernowne string is an infinite string formed by concatenating the base-10 representations of the positive integers in order.

It begins 1234567891011121314...

It can be proven that any finite string of digits will appear as a substring in the Champernowne string at least once.

Given a string of digits and question marks, compute the smallest possible index that this string could appear as a substring in the Champernowne string by replacing each question mark with a single digit from 0 to 9. Each question mark can map to a different digit. Since this index can be large, print it modulo 998 244 353.

Input

The first line of input contains a single integer t ($1 \leq t \leq 10$), which is the number of test cases.

Each of the next t lines contains a string s ($1 \leq |s| \leq 25$) consisting of digits 0 to 9 or question marks.

Output

Output t lines. For each test case in order, output a single line with a single integer, which is the smallest possible index where the string could appear as a substring in the Champernowne string, modulo 998 244 353.

Example

standard input	standard output
9	11
0	7
???1	14
121	10
1?1?1	314159
??5?54?50?5?505?65?5	796889014
000000000000	7777
?2222222	8058869
?3????????9??8????????1??0	38886
9?9??0?????????????2	