E. Evaluate Expression

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Moca is learning how to evaluate a mathematical expression, and she is only able to evaluate a complicated expression step by step. But she is very good at counting, so she is curious about how many different orders evaluate a given expression.

In this problem, you only need to consider some simple mathematical expressions consisting of $0, 1, 2, 3, 4, 5, 6, 7, 8, 9,$ + (addition), * (multiplication) and ( ). Multiplication has a higher priority than addition. Parentheses ( ) can be used to indicate an alternative order. In each step of the evaluation, Moca can choose an operation (addition or multiplication) and replace it with the corresponding result. Notice that Moca can not break the original priority, which may cause her to evaluate an incorrect result. For example, Moca can not choose to evaluate the middle addition in $1 \times 1 + 1 \times 1$ and the middle multiplication in $(1 + 1) \times (1 + 1)$. If many different operations can be evaluated valid, she can choose any one of them.

Moca is very curious. Can you help her count the number of different valid orders to evaluate the given mathematical expression? Since the answer may be very large, you only need to print the answer modulo 998244353.

Input

The first line contains one integer T $(1 \le T \le 100)$ – the number of testcases.

For each testcase, there is only one line contains a valid expression $expr$ $(1 \le |expr| \le 10000)$ consisting of $0, 1, 2, 3, 4, 5, 6, 7, 8, 9,$ +, * and ( ). All the numbers that appear are not greater than 9 (only one digit). All the + that appear are binary operations ($+1$ is invalid).

It is guaranteed that the sum of lengths of all expressions does not exceed 20000.

Output

For each testcase, print the number of valid evaluation order modulo 998244353 in one line.

Example

input Copy

```
3
1+1+1+1
1*2*3+4*5
1+(2*3*4+5+6)*6+7*8*9
```

output Copy

```
6
6
784
```

Note

In the first example, $1 + 1 + 1 + 1$ has 6 valid evaluation orders:

1. $\underline{1 + 1} + 1 + 1 \to \underline{2 + 1} + 1 \to \underline{3 + 1} \to 4$
2. $\underline{1 + 1} + 1 + 1 \to 2 + \underline{1 + 1} \to \underline{2 + 2} \to 4$
3. $1 + \underline{1 + 1} + 1 \to \underline{1 + 2} + 1 \to \underline{3 + 1} \to 4$
4. $1 + \underline{1 + 1} + 1 \to 1 + \underline{2 + 1} \to \underline{1 + 3} \to 4$
5. $1 + 1 + \underline{1 + 1} \to 1 + \underline{1 + 2} \to \underline{1 + 3} \to 4$
6. $1 + 1 + \underline{1 + 1} \to \underline{1 + 1} + 2 \to \underline{2 + 2} \to 4$

In the second example, $1 \times 2 \times 3 + 4 \times 5$ has 6 valid evaluation orders:

1. $\underline{1 \times 2} \times 3 + 4 \times 5 \to \underline{2 \times 3} + 4 \times 5 \to 6 + \underline{4 \times 5} \to \underline{6 + 20} \to 26$
2. $\underline{1 \times 2} \times 3 + 4 \times 5 \to 2 \times 3 + \underline{4 \times 5} \to \underline{2 \times 3} + 20 \to \underline{6 + 20} \to 26$

2021 Shandong Provincial Collegiate Programming Contest

Finished

Practice

→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ Clone Contest to Mashup

You can clone this contest to a mashup.

Clone Contest

→ Submit?

Language: GNU G++11 5.1.0

Choose file: 浏览... 未选择文件。

Submit

→ Contest materials

- Statements (en)

3. $1 \times \underline{2 \times 3} + 4 \times 5 \to \underline{1 \times 6} + 4 \times 5 \to 6 + \underline{4 \times 5} \to \underline{6 + 20} \to 26$
4. $1 \times \underline{2 \times 3} + 4 \times 5 \to 2 \times 3 + \underline{4 \times 5} \to \underline{2 \times 3} + 20 \to \underline{6 + 20} \to 26$
5. $1 \times 2 \times 3 + \underline{4 \times 5} \to \underline{1 \times 2} \times 3 + 20 \to \underline{2 \times 3} + 20 \to \underline{6 + 20} \to 26$
6. $1 \times 2 \times 3 + \underline{4 \times 5} \to 1 \times \underline{2 \times 3} + 20 \to \underline{1 \times 6} + 20 \to \underline{6 + 20} \to 26$