Count Strings

A regular expression is used to describe a set of strings. For this problem the alphabet is limited to 'a' and 'b'.

We define \( R \) to be a valid regular expression if:

1) \( R \) is "a" or "b".
2) \( R \) is of the form \( (R_1 R_2) \), where \( R_1 \) and \( R_2 \) are regular expressions.
3) \( R \) is of the form \( (R_1 | R_2) \) where \( R_1 \) and \( R_2 \) are regular expressions.
4) \( R \) is of the form \( (R_1^*) \) where \( R_1 \) is a regular expression.

Regular expressions can be nested and will always have two elements in the parentheses. ('*' is an element, '|' is not; basically, there will always be pairwise evaluation) Additionally, '*' will always be the second element; '('*a')' is invalid.

The set of strings recognized by \( R \) are as follows:

1) If \( R \) is "a", then the set of strings recognized = \( a \).
2) If \( R \) is "b", then the set of strings recognized = \( b \).
3) If \( R \) is of the form \( (R_1 R_2) \) then the set of strings recognized = all strings which can be obtained by a concatenation of strings \( s_1 \) and \( s_2 \), where \( s_1 \) is recognized by \( R_1 \) and \( s_2 \) by \( R_2 \).
4) If \( R \) is of the form \( (R_1 | R_2) \) then the set of strings recognized = union of the set of strings recognized by \( R_1 \) and \( R_2 \).
5) If \( R \) is of the form \( (R_1^*) \) then the the strings recognized are the empty string and the concatenation of an arbitrary number of copies of any string recognized by \( R_1 \).

Task
Given a regular expression and an integer, \( L \), count how many strings of length \( L \) are recognized by it.

Input Format
The first line contains the number of test cases \( T \). \( T \) test cases follow.
Each test case contains a regular expression, \( R \), and an integer, \( L \).

Constraints

- \( 1 \leq T \leq 50 \)
- \( 1 \leq |R| \leq 100 \)
- \( 1 \leq L \leq 10^9 \)

- It is guaranteed that \( R \) will conform to the definition provided above.

Output Format
Print \( T \) lines, one corresponding to each test case containing the required answer for the corresponding test case. As the answers can be very big, output them modulo \( 10^9 + 7 \).

Sample Input

```
3
((ab)|(ba)) 2
((a|b)*) 5
((a*)(b(a*))))) 100
```
Sample Output

2
32
100

Explanation

For the first case, the only strings recognized are "ab" and "ba". Of the 4 possible strings of length 2, 2 of them fit that expression.

For the second case, the RegEx recognizes any string of any length containing only a's and b's. The number of strings of length 5 recognized by this expression is \(2^5 = 32\).

For the third case, the RegEx recognizes any string having one b, preceded and followed by any number of a's. There are 100 strings of length 100 which have a single b in them.