

Problem D - Theory of Magic II

While spells are a convenient way to cast magic on the fly, more elaborate applications of magic are dependent on *scrolls* which are sophisticated magic commands pre-written on special paper.

Writing a scroll is much like programming a modern computer, in the sense that one has to be very precise with every single character written. To make things worse, most scrolls are written in a super compact and indecipherable representation that somewhat resembles the Brainfuck language, as special magical paper is quite expensive.

As a magic theoretician, Raunak is not as proficient at writing scrolls and he is tired of having mistakes slowing down his work. After enough frustration, he has asked you for help.

Specifically parentheses are of utmost important in the language of magical scrolls, and Raunak has problem balancing them correctly! He has asked you invent a charm that will automatically detect whether parentheses are balanced or not every time he writes to a scroll, so that he can avoid making these mistakes.

Input

The input starts with one integer, T , denoting the number of test cases.

Each test case starts with an integer, $1 \leq n \leq 100,000$, the number of times Raunak writes to his scroll.

The next line contains a single string S ($1 \leq |S| \leq 100,000$) consisting solely of the characters `()*<>+-&`.

After that, n lines follow, each containing a single integer $1 \leq k \leq |S|$ followed by one of the valid characters described above, indicating that Raunak has written that character to the k -th position of the string, overwriting any previous character there. Positions in the string are 1-indexed in this problem.

Output

After each write, output a single word **Yes** if the parentheses are balanced, and **No** if they are not balanced.

A string is considered to contain balanced parentheses if every `(` can be matched one-to-one to a corresponding `)` **to the right** such that no `)` is unmatched and the substring between any `(` and its matching `)` is balanced; a string with no parentheses is considered balanced.

Sample Input

```
1
5
(((*)>+)()())+
1 )
1 (
1 *
7 (
8 )
```

Sample Output

```
No
No
Yes
No
Yes
```
