Bitwise

After seeing an ad for the Banana trading app, you decided that you want to try it out! Upon opening up the app, you realized that it is not so simple…

After finishing the 5,000-page tutorial, you finally understood how the app works:

- There are 3 types of *commands*: `AND val/OR val/XOR val`.
- If you currently have k *Coin bits*, after *executing a command*, you will have k `AND/OR/XOR` *val Coin bits*, where `AND/OR/XOR` are the bitwise operations corresponding to the *command* you used.
- Suppose there are a list of N *commands* for today's session. You can *surf the wave* by going through each *commands* in order from first to last. At each *command*, you can choose to either *execute that command*. There are 2^N different ways to *surf*.

At the beginning, the *command* list is empty. You will be using the Banana app for Q days. Everyday, one of these 3 types of events will occur:

- `TYPE val` (`TYPE` $\in$ {"`AND`", "`OR`", "`XOR`"}) - the *command* `TYPE val` is appended to the *command* list.
- `REMOVE` - the last *command* added is now removed from the list.
- `QUERY val` - You have 0 *Coin bits* at the beginning of the session. You want to count the number of ways to *surf* on today's *command* list such that you will end up with `val` *Coin bits* at the end of the session.

Input

- The first line contains integer Q - the number of days you use the Banana app.
- Each of the next Q lines contains an event.

Output

For each `QUERY`-type event, print the answer to that query to a line. As the result may be large, you should print the answer modulo $1,000,000,007$.

Constraints

- $Q \leq 10,000$
- $val < 2^{14}$

Subtask

- Subtask 1 (25%): $Q \leq 500$.
- Subtask 2 (25%): There are no `AND` và `OR` *commands*.
- Subtask 3 (25%): There are no `AND` và `XOR` *commands*.
- Subtask 4 (25%): No additional constraints

Sample

Input

```
8
XOR 5
XOR 2
QUERY 7
AND 1
QUERY 1
REMOVE
OR 2
QUERY 2
```

Output

```
1
2
3
```