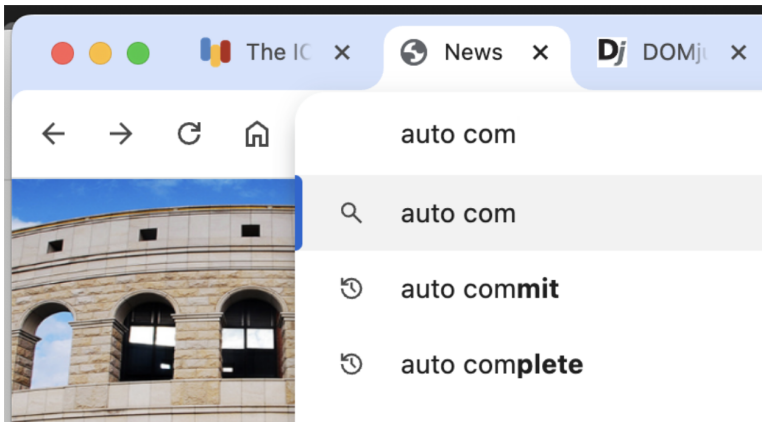
Auto Complete

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	1024 megabytes



You are designing a snazzy new text editor, and you want to add a nifty auto-complete feature to help users save time. Here is how it will work: if a user types "`App`", your editor will magically suggest the word "`Application`"! Even better, users can personalize the words that auto-complete in your editor.

Your editor will support 4 kinds of operations (Let's say the current text in your editor is t):

1. Add an auto complete pattern p_i.
2. Delete an auto complete pattern p_i.
3. Append a string s to the end of t.
4. Delete c characters from the end of t. Note that if c is larger then the length of t, delete all the characters from t.

After each action, your editor should suggest an auto-complete candidate i that matches the following criteria:

1. The string p_i has a prefix equal to t.
2. If there are multiple p_i, pick the longest one.
3. If there are still multiple p_i, pick the one with the smallest lexicographic order.
4. If there are still multiple p_i, pick the one with the smallest ID.

To simplify the question, for each action, print the suggested auto complete pattern ID. If there's no match, print `-1`.

For example, let us say we have three candidates: "`alice`", "`bob`", and "`charlie`", with ID 1, 2, and 3. At first, there is nothing on the screen, so "`charlie`" (3) should be suggested because it is the longest. Then, let us say the user types "`b`". You should suggest "`bob`" (2) because it is the only one that starts with "`b`". Finally, let us say the user type "`body`". You should print `-1` because there is no matched pattern.

Input

The first line contains an integer n, followed by n lines, with each line containing an action.

There are four types of actions:

1. `add` i p_i
2. `delete` i
3. `append` s
4. `backspace` c

The `add` actions are followed by an integer i and a pattern p_i, which means the user wants to add a pattern with ID i. The `delete` actions are followed by an integer i, which means the user wants to delete p_i from the pattern set. The `append` actions are followed by a string s, which means the user appends s to the end of t. The `backspace` actions are followed by an integer c, which means the user deletes c characters from the end of t. All the parameters are splitted by a single space character.

- $1 \leq n \leq 10^6$
- The total number of characters in all p_i and s does not exceed 2×10^6.
- $1 \leq c \leq 2 \times 10^6$
- The strings p_i and s may contain any printable characters, excluding all space characters (ASCII number in the range from 33 to 126).
- The ID i for each `add` operation is unique.
- The ID i for each `delete` is guaranteed to be valid.
- Each ID i satisfies $0 \leq i \leq n$.

Output

The program should output n lines. For each action, output an integer i, which means that after the action, p_i is the suggested auto complete candidate. If there is no p_i that matches the requirement, output `-1`.

Examples

standard input	standard output
6 add 1 pattern1_alice add 2 pattern2_bob add 3 pattern3_charlie append pattern append 2_bobabc backspace 3	1 1 3 3 -1 2
6 append pattern add 1 pattern1_alice____ add 2 pattern2_bob______ add 3 pattern3_charlie__ delete 1 delete 2	-1 1 1 1 2 3