String Stack Removal

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.75 seconds
Memory limit:	256 megabytes

El Bandito Inofensivo gives a set of words S and a stack, you are asked to perform the following operations:

- operation of type 0: add the character c on top of the stack
- operation of type 1: pop the character from the top of the stack
- operation of type 2: insert the word W in S
- operation of type 3: erase the word W from S

We say that a word S of length N is on top of the stack if the suffix (we see the stack as a string, where the last element of the string is the top of the stack) of length N of the stack is equal to the string S. If there is a word from the set S at the top of the stack, it is removed instantly.

Initially, both the stack and the set of words are empty. You are given Q operations of the types above and you are asked to print the character on top of the stack after each of the Q operations. The character on top of the empty stack is considered to be the character $'?'$.

Input

First line of the input contains the number Q ($1 \leq Q \leq 1.5 * 10^5$). The next Q lines describe the Q operations. Each of the lines contain a number op, representing the type of operation we have to perform:

- if $op = 0$, then the line also contains the character c. This line describes an operation of type 0.
- if $op = 1$, the line describes an operation on type 1. This operation will never show up when the stack is empty.
- if $op = 2$, then the line also contains the string W. This line describes an operation of type 2.
- if $op = 3$, then the line also contains the string W. This line describes an operation of type 3. This operation will never involve a word which is not in S.

For tests worth 13 points, $1 \leq Q \leq 100$.

For tests worth 38 more points, there is no operation of type 1 and all the operations of type 0 are performed after all the operations of type 2 and 3.

The sum of lengths of all words inserted in S is at most 10^5.

The number of operations of type 0 is at most 10^5.

It is guaranteed that, at any moment of time, S does not contain two words such that one is a suffix of the other.

Output

The only line of the output will contain a string S of length Q, where the $i-th$ character from S represents the character on top of the stack after the $i-th$ operation. If after the $i-th$ operation the stack is empty, then the $i-th$ character in S will be the character $'?'$.

Example

standard input	standard output
10 0 a 0 b 2 bbd 0 b 0 b 1 2 ab 0 d 3 ab 0 b	abbbbbbaab

Note

The evolution of the stack and the set of string after each operation:

After step 1: "a"

After step 2: "ab"

After step 3: "ab", ["bbd"]

After step 4: "abb", ["bbd"]

After step 5: "$abbb$", ["bbd"]

After step 6: "abb", ["bbd"]

After step 7: "abb", ["bbd"ab"]

After step 8: "$abbd$" -> "a", ["bbd"ab"]

After step 9: "a", ["bbd"]

After step 10: "ab", ["bbd"]