

CyberChallenge.IT 2025

Programming Test

Aliens [80 points]

Problem Statement

The CyberChallenge.IT space agency managed to intercept messages from an alien species that communicates not that efficiently. It took years and efforts to reverse engineer their language, but they finally made it. The aliens never use words directly: they build their messages by using operations; adding, deleting, swapping and rotating characters.

They send hundreds of operations a day, which are carefully collected continuously, but the space team is so tired of doing all the work manually. They need someone who can make the entire decoding automatic. Someone who can help them build messages starting from operations. Can you be that one?

Problem Details

You start from an empty string and process an **ordered** series of operations. Each operation can be:

- **add** *c*: appends character *c* at the end of the string
- **del**: removes last character from the string (if any)
- **swap** *a* *b*: swaps character *a* with character *b*, meaning that all occurrences of character *a* are replaced with character *b*, and all occurrences of character *b* are replaced with character *a*
- **rot** *x*: applies **rot-*x*** to the current string

Notes:

- each operation is only applied to the string as it is before the operation itself, and it does not influence the characters added after the operation is executed
- the string below represents the alphabet of allowed characters:

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

- when performing **rot** operations, characters are rotated following the order in the alphabet, rotating to the right. For example, performing **rot-6** on character **y** outputs character **E** (because **E** is 6 places to the right of **y** in the alphabet), and performing **rot-3** on character **8** outputs character **b** (because after the end the alphabet restarts from **a**).

Input

The input consists of $N + 1$ lines:

- Line 1: an integer, N , the number of operations
- Line 2, ..., $N + 1$: the operations, one per line

Output

The output consists of 1 line, containing the string generated by executing the input operations.

Scoring

Your program will be tested on a number of testcases grouped in subtasks. In order to obtain the score associated to a subtask, you need to correctly solve all its testcases.

- **Subtask 1** [20 points]: $1 \leq N \leq 10^3$, using only `add` and `del` operations
- **Subtask 2** [40 points]: $1 \leq N \leq 10^4$, using all operations
- **Subtask 3** [20 points]: $1 \leq N \leq 10^6$, using all operations

Examples

INPUT	OUTPUT
12 del add G add q add F del swap q h add Y add y rot 21 swap a F add j swap 1 j	jC1T1

Explanation

Let's go through each operation. We start from an empty string.

- The first line deletes the last character of the string, but there isn't any, so the operation does nothing
- The second line adds the character `G`, so now the string is: `G`
- The third line adds the character `q`, so now the string is: `Gq`
- The fourth line adds the character `F`, so now the string is: `GqF`
- The fifth line deletes the last character, so now the string is: `Gq`
- The sixth line swaps the character `q` with the character `h`. There is one occurrence of `q`, that will become `h`, but there is no `h`, so the string becomes: `Gh`
- The seventh line adds the character `Y`, so now the string is: `GhY`
- The eighth line adds the character `y`, so now the string is: `GhYy`
- The ninth line performs `rot-21` on the string, so the string now is: `1CjT`
- The tenth line swaps the character `a` with the character `F`, but there is no `a` and no `F`, so the string stays the same: `1CjT`
- The eleventh line adds the character `j`, so now the string is: `1CjTj`
- The twelfth line swaps the character `1` with the character `j`, so the string now is: `jC1T1`