**Uncle Grandpa and the home-made text editor**

**Note: this question will be graded for 1%.**

Hi there, it’s Uncle Grandpa again. After having you created a simple calculator for him last week, he makes another “strange” request to you this week: Create a simple text editor for him. The text editor will take in a string . The text editor will use a cursor to navigate. It also has a string to support copy-paste operations.

The image shows 6 possible positions of the cursor for string . Position is to the left of the first character, and position is to the right of the last character. In the beginning, the cursor is at position 0.

It should support the following 6 operations:

* : increase the position of the cursor by . If the cursor is already at the last position, nothing changes.
* : decrease the position of the cursor by **.** If the cursor is already at the first position, nothing changes.
* : set value to the substring of length starting from the character **immediately to the right** ofthe cursor (If there are not enough letters after the cursor, the substring shall end at the last character of the string).
* : insert the string to the position **immediately to the left** the cursor. Set to empty. After this operation, the position of the cursor will be increased by
* : insert character ( is either a lower-case or upper-case English letter) to the position **immediately to the left** of the cursor. After this operation, the position of the cursor will be increased by
* : delete the character **immediately to the left** of the cursor. If the cursor is at position , nothing changes. After this operation, the position of the cursor will be decreased by

Please help him! He promises to treat you with KFC, McDonald’s, Texas Chicken, Popeyes if you can help him! (Well, Uncle Grandpa is a fictional character and so are his promises).

**Note:** For the purpose of learning, you are only allowed to use Linked-List in any form (including Java’s API or implementation from scratch). Usage of other data structures is not allowed (including array, Stack, Queue, ArrayList … but excluding String ~~and StringBuilder~~), and will result in **0 mark**.

## Input

The first line contains a string  **(** consists of only lower-case and upper-case English letters

The second line is a single integer  – the number of operation you need to process

The next lines, each line contains the description of one operation of the 6 types described above.

**It’s guaranteed that the total length of copied string will be no more than 4**

## Output

The string after processing all operations.

## Examples

|  |  |
| --- | --- |
| Input (editor1.in) | Output (editor1.out) |
| UncleG  9  R  C 6  P  R  C 2  P  I A  L  B | UncleGncAcleG |

## Explanation:

show the current position of the cursor

After 0th step: “| Buffer:

After 1st step: “ Buffer:

After 2nd step: “ Buffer: *(only 5 characters after cursor)*

After 3rd step: “ Buffer:

After 4th step: “ Buffer:

After 5th step: “ Buffer:

After 6th step: “ Buffer:

After 7th step: “ Buffer:

After 8th step: “ Buffer:

After 9th step: “ Buffer:

## Note:

1. A skeleton file has been given to help you. You should not create a new file or rename the file provided. You should develop your program using this skeleton file.
2. You are free to define your own helper methods and classes (or remove existing ones) if it is suitable but you must put all the new classes, if any, in the same skeleton file provided

## Skeleton File

You are given the skeleton file Editor.java. You should see the following contents when you open the file:

|  |
| --- |
| /\*\*  \* Name :  \* Matric. No :  \*/  import java.util.\*;  public class Editor {  private void run() {  //implement your "main" method here  }  public static void main(String[] args) {  Editor newEditor = new Editor();  newEditor.run();  }  } |