Statement

You have to simulate a text editor with an integrated clipboard manager which can handle the following operations:

- `TYPE <text>`: insert `<text>` after the current cursor position, where `<text>` is at most `50` characters (letters, digits, whitespaces, and the symbols `.,?!-`). Then, the cursor moves to the end of the inserted text.
- `SELECT <start_index> <end_index>`: select the characters of the current text `[text[start_index] ... text[end_index]]` (inclusive indices, 0-based) of length `end_index - start_index + 1`. The indices will be always valid. Then, the cursor moves to the end of the selected area.
- `MOVE_CURSOR <offset>`: change the cursor's position. The `<offset>` specifies the direction and the value change - a negative value moves the cursor to the left, and a positive value moves it to the right. The cursor can be located between the beginning of the string (before the first character), or the end of the string (after the last character) - the cursor should always be within the text bounds. If the `<offset>` is larger/shorter than the text boundaries, the cursor will move in that direction as much as it can. If there is a selected area, it should be cleared after this operation.
- `COPY`: add the text from the selected area to the clipboard. If no area is selected, do nothing.
- `PASTE <steps_back>`: insert the copied text from the clipboard. The number `<steps_back>` indicates which of the stored copied texts to insert. If `<steps_back>` is `1` then insert the last copied text, if it is `2` then insert the text copied before the last, and so on. If `<steps_back>` exceeds the amount of the clipboard history size, ignore this operation. The cursor moves to the end of the inserted text.
- `PASTE`: do the same as if the operation is `PASTE 1`.
- `EXIT`: finish the execution.

Note: If a selected area is not empty and the next operation is either `TYPE` or `PASTE` then the following process has to be performed before the `TYPE` or `PASTE` operation:

1. Delete the selected text.
2. Insert the new text in place of the deleted text.
3. The cursor should move to the end of the new text.

You will receive multiple operations as an input, an array of strings where each is an operation type described above. Your code has to wait for inputs, parse them, and show the processed text after performing the given operations. Then, it has to wait for more inputs until the command "EXIT" is received.

Example

- For an input `"TYPE Barcelona is located in Cactus", "SELECT 0 8", "COPY", "SELECT 24 29", "COPY", "PASTE 2", "SELECT 0 8", "PASTE", "MOVE_CURSOR 24", "TYPE !"`, the output should be `"Cactus is located in Barcelona!"`.
    - Initially, the text is empty,
    - After the `"TYPE Barcelona is located in Cactus"` operation, the text is `"Barcelona is located in Cactus|"` (where the `|` symbol represents the cursor position),
    - After the `"SELECT 0 8"` operation, the selected text is `"Barcelona"`, the cursor is moved to the end of the selected area `"Barcelona| is located in Cactus"`,
    - After the `"COPY"` operation, the clipboard is `["Barcelona"]`, the cursor doesn't move, and the selected area stays the same,
    - After the `"SELECT 24 29"` operation, the selected text is `"Cactus"`, the cursor is moved to the end of the selected area `"Barcelona is located in Cactus|"`,
    - After the `"COPY"` operation, the clipboard is `["Barcelona", "Cactus"]`, the cursor doesn't move, and the selected area stays the same,
    - After the `"PASTE 2"` operation, since there is a selected text, it is deleted and replaced with the text `"Barcelona"` because it is `2` positions back in the clipboard, the cursor stays after the inserted text, so the text is `"Barcelona is located in Barcelona|"`,
    - After the `"SELECT 0 8"` operation, the selected text is `"Barcelona"`, the cursor is moved to the end of the selected area `"Barcelona| is located in Barcelona"`,
    - After the `"PASTE"` operation, since there is a selected text, it is deleted and replaced with the text `"Cactus"` because it is the last copied text in the clipboard, the cursor stays after the inserted text, so the text is `"Cactus| is located in Barcelona"`,
    - After the `"MOVE_CURSOR 24"`, the cursor moves `24` symbols forward, so the text is `"Cactus is located in Barcelona|"`,
    - After the `"TYPE !"` operation, the text is `"Cactus is located in Barcelona!|"`,
    - So the final string is `"Cactus is located in Barcelona!"`,
    - Then, it prints the current text.
- Finally, you can keep sending more commands or finish the execution with the command `"EXIT"`.

Input/Output

- **[input] array.string operations**

  A sequence of operations with double quotes as an operation delimiter, and a comma to separate them. They will satisfy the format described above.

- **[output] string**

  The resulting text after performing the sequence of operations.

Test

Input:

```
 "TYPE We hate pointers",
 "MOVE_CURSOR -3",
 "SELECT 3 6",
 "TYPE love"]
```

Expected Output:

```
"We love pointers"
```