edU, or ed with multiple Undo

Design of algorithms and data structures course
Politecnico di Milano 2020

Description

The project consists of implementing a simple text editor. The editor considers a document as a sequence of lines, of arbitrary size, numbered from 1.
The editor interface consists of text commands, terminated by the enter command. Commands can be followed by a portion of text, consisting of one or more lines, terminated by a character . (period) that appears as the only character on the next line. The commands are made up by a single letter, optionally preceded by one or two integers.
The interface of the editor is inspired by that of the traditional editor ed(Unix).
In some commands, the integers that appear in them represent specific addresses. More precisely, an address a number n, expressed in decimal, indicating the address of the n-th line; the first line of the text has address 1.
The supported commands are as follows, with the convention that ind1,ind2 indicate two addresses such that ind1 <= ind2 and round brackets are introduced to simplify the reading of this text, but not included in the command:

- **(ind1,ind2)c**
  Change the text on the lines between ind1 and ind2 (extremes included). The text following the command must be made of a number of rows equal to ind2-ind1+1. ind1 must be either an address present in the text, or the first address after the last line present in the text (or 1 if the text is empty).
- **(ind1,ind2)d**
  Erases the lines between ind1 and ind2 (including extremes), moving upwards the lines following that of ind2 address (if any). Deleting a line that does not exist in the text has no effect.
- **(ind1,ind2)p**
  Prints the lines between ind1 and ind2, included extremes. Where it is not present a line in the text at the location to be printed a line is printed containing only the character '.' (period).
- **(number)u**
  Undo a number of commands (c or d) equal to the one specified in round brackets (where number is an integer strictly greater than zero). A sequence of consecutive undo commands deletes a number of steps equal to the sum of the steps specified in each of them. If the number of deleted commands is higher than the executed commands, all steps are canceled. The execution of a command to modify the text (c, d) after a undo deletes definitely the deleted command (they can't be recovered with a redo). Note that in the number of commands to delete also counts the commands that do not have a effect (e.g. deleting a block of rows that do not exist).
- **(number)r**
  Cancel the effect of an undo for a number of commands equal to the one specified in round brackets (note that number is an integer strictly greater than zero). We then have that a sequence of commands of the type:

10u
5r
Is equivalent to the 5u command alone. Similarly, the sequence:
12u
3r
2u
7r
Is equivalent to only the 4u command. If the number of commands of redo is higher than those currently deleted, it is made the maximum number of possible redos.

- **q**
  Finish running the editor.

A line of text provided at the input to the editor can contain a maximum of 1024 characters. Assume that the editor is given only correct commands (it is not necessary to verify their correctness).
For example, it can't be given a c command where ind1 > ind2, or ind1 < 1. Similarly for other commands. Attention should be paid to the fact that, in some cases, the command is allowed, but simply has no effect.

Input example:

```
1.2c
first line
second line
.
2.3c
new second line
third line
.
1.3p
1.1c
new first line
.
1.2p
2.2d
4.5p
1.3p
4.5d
1.4p
3u
1.6p
1r
1.3p
q
```

The previous sequence of commands gives the following output by the editor:
first line
new second line
third line
new first line
new second line
.
.
new first line
third line
.
new first line
third line
.
.
first line
new second line
third line
.
.
.
new first line
new second line
third line