

Problem K. Text Editor

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 150 mebibytes

This is a run-twice problem: your solution will be executed twice on each test. See the rest of the statement and the input format section for more details.

Your task is to implement a simple text editor that works with strings of characters with values in range 33–126 and implements an interface described below. The length of the string is guaranteed to be at most $2^{63} - 1$ at all times. All parameters of the commands are 0-indexed and all intervals are semi-open (thus, whenever the command accepts ℓ and r , ℓ is included while r is excluded) and guaranteed to be nonempty. All positions and intervals are within bounds. The clipboard is initially empty.

- **insert** p str . This command inserts string str to the p -th position of the file. The total length of all str is guaranteed to be at most 10^6 .
- **erase** ℓ r . This command erases the contents of the file from ℓ -th to r -th character.
- **print** ℓ r . After this command you should print the contents of the file from ℓ -th to r -th character. It is guaranteed that the total length of the output for all **print** commands does not exceed 10^6 .
- **copy** ℓ r . This command copies the contents of the file from ℓ -th to r -th character to the clipboard.
- **cut** ℓ r . This command cuts the contents of the file from ℓ -th to r -th character to the clipboard.
- **paste** p . This command inserts the contents of the clipboard to the p -th position of the file. It is guaranteed that empty clipboard is never pasted.
- **serialize**. After this command you should write the contents of the file (but not the clipboard) to the output, not exceeding 10^7 characters and using only characters of the range 33–126. The output string does not have to represent the contents of the file exactly, but your program should be able to recreate it in response to the command **deserialize**. You are also required to print **at least one** character. This command will be issued only once, as the last command on the first run.
- **deserialize** str . Here str is guaranteed to be the exact string that was printed in response to the **serialize** command. This command will be issued only once, as the first command on the second run.
- **undo**. After this command you should rollback the changes made by the last command changing the contents of the file (thus, **insert**, **erase**, **cut**, **paste** or **deserialize**), leaving the state of the clipboard intact. If there is no previous version, the command should do nothing.
- **redo**. Undo the last undo that is not yet undone. If there is no next version, the command should do nothing. Note that as in most text editors, the commands changing the state of the file also “erase” all next versions, but other commands (for example, **copy**) don't.

Input

Input formats for both runs are the same.

The first line contains the only integer n ($1 \leq n \leq 10^4$) representing the number of commands to the text editor.

The next n lines contain commands in the format described above.

Output

Print answers to all **print** and **serialize** commands on separate lines.

Examples

| <i>standard input</i> | <i>standard output</i> |
|---|---|
| 17 insert 0 abcdef print 0 6 erase 4 5 print 0 5 copy 0 3 paste 1 print 0 8 cut 2 4 print 0 6 undo print 0 8 paste 6 print 0 10 redo redo print 0 10 serialize | abcdef abcdf aabcbcdf aabcdf aabcbcdf aabcbcbcdf aabcbcbcdf serialize:aabcbcbcdf |
| 2 deserialize serialize:aabcbcbcdf print 0 10 | aabcbcbcdf |