Problem 2 - String Rearrangement

Write a C program that reads a line from the console and extracts all text pieces inside curly brackets { }. The program should then read commands line by line and execute them on the resulting text.

For example, we are given the line

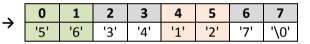
aide na moreto | detelini {12345} | saxi {q | konf} | h {67}

We are looking for text inside curly brackets. Moreover, the opening and closing curly bracket should be inside a block together. A block is a sequence of characters, separated by | from other blocks. There will always be a maximum of one sequence, enclosed in curly brackets { } inside a block. The only sequences that match these rules are 12345 and 67 (notice how we skip q | konf, because it is not inside a single block). We concatenate them and the result is:

1234567

We then start reading commands, each at separate line until we read "end". Each command will be in the format swap {positionA} {positionB} {size}. We must swap size bytes from positionA with size bytes from positionB. For example, given the command **swap 0 4 2**, we should do the following:

0	1	2	3	4	5	6	7
'1'	'2'	'3'	'4'	'5'	'6'	'7'	'\0'



We do this for each command and stop once "end" is entered. We print the resulting string on the console -5634127.

Input

The input will be read from the standard input (console).

- On the first line, you will receive the text.
- Every next line will hold a command in the format "swap {positionA} {positionB} {size}".
- Command "end" denotes end of input.

Output

The output should be printed on the console. Print the string after all commands have been executed. If a command holds invalid parameters (e.g. position -10) or the position + size would exceed the buffer, an error message should be printed - "Invalid command parameters".

Constraints

- The input text will be a string of **unknown size**.
- All character sequences inside curly brackets {} will be in the range [0...64].
- There won't be any nested curly brackets (e.g. { t {e}) inside a single block.
- There will be a maximum of **one valid curly brackets sequence** inside a single block.
- The positions and size arguments in each command can be invalid (e.g. size = -1).
- The program should display no memory leaks, buffer overflows or dangling pointer anomalies.

Examples

Input	Output
<pre>aide na moreto pendel{12345} saxi{q be} h{67} swap 0 4 2 end</pre>	5634127





















Input	Output
<pre>{text} {} }{alphabet} swap -5 2 3 swap 0 4 20 end</pre>	Invalid command parameters Invalid command parameters textalphabet

















