

Problem 3 – Text Transformer

Nakov is a good lecturer, but sometimes he drinks beer during lectures and gets drunk. When he gets drunk, instead of writing normally, he produces an incomprehensible tsunami of words and symbols. When he gets sober, he has no idea what he has written just a few hours before, so he's giving away 100 exam points to those of our Advanced C# students who can write a program to decipher his gibberish. The good news – you know how Nakov thinks when he's drunk! There is a specific algorithm his brain follows when too much beer is introduced into his blood stream.

He's typing symbols and every once in a while he presses enter to go to a new line. Your first task is to **collect all the pieces of text** into a single string and **replace multiple whitespaces (e.g. " "), with a single one (" ")**. The important pieces of data are stored between special symbols which are the following: a **dollar sign ('\$') with weight of 1**, a **percentage sign ('%') with weight of 2**, **ampersand ('&') with weight of 3** and a **single quote (') with weight of 4**. You need to retrieve all **non-empty strings that are contained between two identical special symbols**. **Special symbols aren't allowed inside these strings!** A special symbol can be part of only one string, e.g. in "\$abc\$def\$" only the left string should be captured ("abc\$"). Then, for each **even** symbol in the captured string (positions 0, 2, etc.), you need to **add** the **weight** of the surrounding special symbol to the **ASCII code** of the current symbol. For each **odd** symbol (positions 1, 3, etc.), you need to **subtract** the weight of the special symbol from the ASCII code of the current symbol. When you're done, just **print all resulting strings on the console (on a single line, separated by a space)**. That's it! Check out the example for a more thorough explanation of the process.

Input

- The input data should be read from the console.
- It consists of an unknown number of lines, containing various symbols from the ASCII table.
- The input ends with the keyword "burp".
- The input data will always be valid and in the format described. There is no need to check it explicitly.

Output

- The output should be printed on the console. It should consist of **exactly one line**.
- Print the resulting string as described above on a single line. All decoded pieces should be separated from each other by a single space.

Constraints

- The count of **input lines** will be in the range [1 ... 125 000].
- Each input line will contain ASCII symbols and will have a length in the range [1 ... 20].
- Allowed working time for your program: 0.1 seconds. Allowed memory: 20 MB.

Examples

Input	Output	Comments
Gv \$ Rp e uT o h \$ ef f' 'j\$g ij? burp	SoftUni	<p>The input (after consolidating the lines) is: Gv\$RpeuToh\$ ef f' 'j\$gij?</p> <p>Replace multiple whitespaces with a single one: Gv\$RpeuToh\$ ef f' 'j\$gij?</p> <p>Look for a sequence of symbols surrounded by the same special character. There is one such string, the sequence is shaded grey.</p> <p>Start transofrming the characters ('\$' has a weight of 1):</p> <p>R (position 0, even): 'R' (82) + '\$' (1) = 83 (S)</p> <p>p (position 1, odd): 'p' (112) - '\$' (1) = 111 (o)</p> <p>e (position 2, even): 'e' (101) + '\$' (1) = 102 (f)</p> <p>u (position 3, odd): 'u' (117) - '\$' (1) = 116 (t)</p> <p>T (position 4, even): 'T' (84) + '\$' (1) = 85 (U)</p> <p>o (position 5, odd): 'o' (111) - '\$' (1) = 110 (n)</p> <p>h (position 6, even): 'h' (104) + '\$' (1) = 105 (i)</p> <p>The resulting text is: SoftUni</p>