

C++ Fundamentals: Exam Preparation

The following tasks should be submitted to the SoftUni Judge system, which will be open starting Tuesday, 9 January 2018, 18:00 (in the afternoon) and will close on Saturday, 13 January 2018, 23:59. Submit your solutions here:

<https://judge.softuni.bg/Contests/Compete/Index/916>.

For this exam, the code for each task should be a single C++ file, the contents of which you copy-paste into the Judge system.

Please be mindful of the strict input and output requirements for each task, as well as any additional requirements on running time, used memory, etc., as the tasks are evaluated automatically and not following the requirements strictly may result in your program's output being evaluated as incorrect, even if the program's logic is mostly correct.

You can use C++03 and C++11 features in your code.

Unless explicitly stated, any integer input fits into **int** and any floating-point input can be stored in **double**. On the Judge system, a C++ **int** is a **32-bit** signed integer and a C++ **double** is a **64-bit** IEEE754 floating point number.

NOTE: the tasks here are NOT ordered by difficulty level.

Task 2 – Sequence (Task-2-Copy-Paste)

You are given a text in the form of words, separated by single spaces, where a word is a sequence of English letters (**a-z** and **A-Z**), as well as copy and paste operations on that text. Copy and paste operations take indices in the text as parameters to define on what part of the text they operate.

Copy operations are defined by a **from** index and a **to** index in the text. Any word which intersects the [**from**, **to**] range (both inclusive) is copied (as well as any spaces between those words). That means if the **from** or **to** index are inside a word, that word is also copied.

For example, if we do **copy 4 6** on the following text (note the indices above the text):

0	1	2	3	4	5	6	7	8	9	10
H	e	l	l	o		W	o	r	l	d

the copied text will be “**Hello World**”, i.e. both words will be copied. On the other hand, if we do **copy 2 3**, only the word “**Hello**” will be copied.

Each copy operations puts the copied text as a new entry in the clipboard, which can later be used for pasting by the paste operation.

Copy operations will always have their parameters “inside” words, i.e. copy parameters will never be indices of spaces.

Paste operations are defined by a single **position** index. Paste operations take the latest entry in the clipboard and insert it at the **position** index – meaning that any symbols currently at the position index are “pushed to the right” – and then remove that entry from the clipboard.

If there are no entries in the clipboard (either because all have been already pasted or because none have been created), the paste operation does nothing. The position index can be the index of a space in the text, or of a letter in a word.

The position parameter will be an integer number between **0** and one less than the length of the text (inclusive), i.e. the position parameter will never be equal to the length of the text.

Paste operations **must keep the format** of the text the same as in the input – i.e. the **text must remain a sequence of words, separated by single spaces**:

- If the **position** index is inside a word, the text is pasted “as is”. Because the copy operation does not copy spaces at the edges, the resulting text will have no duplicate spaces regardless of whether the paste is done at the first symbol of the word, last symbol of the word, or some symbol in between
- If the position index is the index of a space (i.e. is not inside a word), then the paste operation should result in the copied text appearing at the position of the space, surrounded by single spaces on each side (i.e. we replace the space with a word surrounded by one space on each side)

For example, if we do **copy 2 3** on the text in the previous example, then do **paste 5**, the modified text will be “**Hello Hello World**”. If we then do **copy 6 23** and **paste 2**, the resulting text will be “**HeHello Worldllo Hello World**”.

Write a program which reads a line of text, and then reads and executes the above-described operations until the string **end** is entered.

Input

The first line of the standard input will contain the initial text.

Each of the next lines will either contain an operation/command (copy or paste), or the string **end** (after which there will be no more input).

Copy operations have the following syntax: the string **copy**, and two integers – the **to** and **from** parameters respectively (separated by single spaces).

Paste operations have the following syntax: the string **paste** and a single integer – the **position** parameter.

Output

A single line, containing the modified text after the end command has been read. The resulting text must remain a sequence of words separated by single spaces.

Restrictions

The text will contain no more than **100** words and no word will be more than **10** symbols long.

There will be no more than **50** copy operations and **50** paste operations.

The total running time of your program should be no more than **0.1s**

The total memory allowed for use by your program is **16MB**

Example I/O

Example Input	Expected Output
some text copy 0 1 paste 4 copy 2 5 paste 1 end	ssome someome some text
Hello World copy 2 3 paste 5 copy 6 23 paste 2 end	HeHello Worldllo Hello World