

C++ Fundamentals: Judge Assignment 2 (JA2)

The following tasks should be submitted to the SoftUni Judge system, which will be open starting **Saturday, 9 December 2017, 10:00** (in the morning) and will close on **Saturday, 23 December 2017, 23:59**. Submit your solutions here: <https://judge.softuni.bg/Contests/Compete/Index/878>.

After the system closes, you will be able to “Practice” on the tasks – however the “Practice” results are NOT considered in the homework evaluation.

For this assignment, 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**.

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

Task 3 – Machine (JA2-Task-3-Machine)

You are tasked with implementing software for a computational machine, which does basic operations with integer numbers. The machine has memory in the form of a sequence of numbers. When it does an operation, it takes (removes) numbers from the end of the sequence, calculates the result using those numbers and puts it back at the end of the sequence. There are also some operations which only add at the end or only remove from the end of the sequence. Here's a list:

- Inserting a number. Syntax: an integer number, (can be positive, negative or **0**). Inserts a number at the end of the sequence (i.e. appends to the end of the sequence).
- Summing last two numbers. Syntax: **sum**. Removes the last two numbers in the sequence, calculates their sum and adds it to the end of the sequence.
- Subtracting last two numbers. Syntax: **subtract**. Removes the last element from the sequence and calls it **a**. Removes the next last element (again) and calls it **b**. Subtracts the two, i.e. **a - b** and adds the result to the end of the sequence.
E.g. if the sequence is **(1, 4, 7)**, then **subtract** will remove **7** and **4**, calculate **7 - 4** and add it to the sequence – the resulting sequence will be **(1, 3)**
- Concatenation of last two numbers. Syntax: **concat**. Concatenates the last two elements in the sequence (as if they were strings), in the order they were added to it, evaluates the result to an integer and adds it to the end of the sequence.
E.g. if the sequence is **(1, 4, 7)**, then **concat** will remove **7** and **4**, concatenate **4** and **7**, resulting in **47**, and add it to the sequence – the resulting sequence will be **(1, 47)**
- Discarding the last number. Syntax: **discard**. Removes the last element from the sequence.
- Ending the computations. Syntax: **end**. Stops the machine (ends the program)

Write a program which does the operations described above and prints out the final sequence of numbers, from first to last (i.e. in the order in which they were added).

Input

Two or more lines, each indicating an operation to be done with the machine (note that a line containing a single integer is the "Inserting a number" operation). The final line will not contain numbers and will only contain the string "end"

Output

One or more lines, each containing a single integer, representing the numbers in the final sequence.

Restrictions

There will be no more than **50** lines of operations in the input. The concat operation will never be done when the last element in the sequence is negative. All operations will be valid – i.e. the input data is such that there will always be enough numbers in the sequence for an operation to be executed correctly. The input will be such that there will be at least 1 number in the final sequence

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
1	5

4 sum end	
1 4 subtract 7 end	3 7
3 10 1 -1 sum concat sum end	103