C++ Fundamentals – Exam (8 May 2021)

3. Machine Commands

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**.

List of operation command:

- **`{number}`** – Inserting a number: **`number`** is a single integer number (can be positive, negative or **`0`**). Inserts a number **at the end of the sequence** (appends to the end of the sequence).

- **`sum`** - Removes the last two numbers in the sequence, calculates their sum and adds it back to the end of the sequence.

- **`subtract`** - Removes the last element from the sequence **`(a)`** then removes the next last element **`(b)`**. Subtracts the two **`(a - b)`** and adds the result to the end of the sequence.
  Example: If the sequence is **`(1, 4, 7)`**, the command **`subtract`** will remove **7** and **4**, calculate **`7 - 4`** and add it back to the sequence. The result sequence will be **`(1, 3)`**.

- **`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.
  Example: 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 result sequence will be **`(1, 47)`**.

- **`discard`** - Removes the last element from the sequence.

- **`end`** – End of program.

Write a program which executes the operations described above and prints the final sequence of numbers 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 **`{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 – there will always be enough numbers in the sequence for an operation to be executed correctly. There will always be at least 1 number in the final sequence.

Examples

Input	Output	Comments
1 4 sum end	5	Inserting: 1 => 1 Inserting: 4 => 1 4 Summing the last two numbers: 4 + 1 = 5 => 5 End of program. The result output is: 5
1 4 subtract 7 end	3 7	Inserting: 1 => 1 Inserting: 4 => 1 4 Subtracting the last two numbers: 4 - 1 = 3 => 3 Inserting: 7 => 3 7 End of program. The result output is: 3 7
3 10 1 -1 sum concat sum end	103	Inserting: 3 => 3 Inserting: 10 => 3 10 Inserting: 1 => 3 10 1 Inserting: -1 => 3 10 1 -1 Summing the last two numbers: -1 + 1 = 0 => 3 10 0 Contatinating the last two numbers: 100 => 3 100 Summing the last two numbers: 100 + 3 => 103 End of program. The result output is: 103