# Stacks and Queues – Lab

Submit your solutions here: <https://judge.softuni.org/Contests/4618/Stacks-and-Queues-Lab>

# Working with Stacks

## Reverse String

Write a program that:

* **Reads** an **input string**
* **Prints** the input string in a reversed order

**Note:** The words are reverted and the letters inside each word are reverted as well.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| I Love C++ | ++C evoL I |
| Stacks and Queues | seueuQ dna skcatS |

## Stack Sum

Write a program that:

* **Reads** an **input of integer numbers** and **adds** them to a **stack**
* **Reads commands** until **"end"** is received
* **Prints** the **sum** of the remaining elements of the **stack**

### Input

* On the **first** **line,** you will receive **an array of integers.**
* On the **next** **lines**, until the "**end**" command is given, you will receive **commands** – a **single** **command** and **one** or **two** numbers after the **command, depending** on what **command** you are given.
* If the **command** is "**add**", you will **always** receive **exactly two** numbers after the command which you need to **add** to the **stack.**
* If the **command** is "**remove**", you will **always** receive **exactly** **one** number after the command which represents the **count** of the numbers you need to **remove** from the **stack.** If there are **not enough elements** skip the command.

### Output

* When the **command** "**end**" is received, you need to print the sum of the remaining elements in the stack.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 2 3 4  add 5 6  remove 3  end | 6 |
| 3 5 8 4 1 9  add 19 32  remove 10  add 89 22  remove 4  remove 3  end | 16 |

## Simple Calculator

Write a program that:

* Reads an **input string (expression)** containing: **numbers**, **+** and **-**
* **Evaluate the expression** with only addition (+) and subtraction (-)
* **Print the result**

**Note**: There will not be any parentheses.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 + 5 + 10 - 2 - 1 | 14 |
| 2 - 2 + 5 | 5 |

## Matching Brackets

Write a program that:

* Reads an **input string** (arithmetic expression with brackets)
* Find all **sub-expressions**
* Print each **sub-expression on separate line**

**Note:** The input expression will always be a correct mathematical expression.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 + (2 - (2 + 3) \* 4 / (3 + 1)) \* 5 | (2 + 3)  (3 + 1)  (2 - (2 + 3) \* 4 / (3 + 1)) |
| (2 + 3) - (2 + 3) | (2 + 3)  (2 + 3) |

# Working with Queues

## Print Even Numbers

Write a program that:

* **Reads** an array of **integers** and **adds** them to a **queue.**
* **Prints** the **even** numbers **separated** by "**,** ".

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 2 3 4 5 6 | 2, 4, 6 |
| 11 13 18 95 2 112 81 46 | 18, 2, 112, 46 |

## Supermarket

Write a program that:

* **Reads** an **input string** consisting of a **name** and **adds** it to a **queue** until "**End**" is received.
* If you receive "**Paid**", **print** every customer name and empty the queue, otherwise, we receive a client and we have to add him to the queue.
* When we receive **"End",** we have to print the count of the remaining people in the queue in the format: "**{count} people remaining."**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Liam  Noah  James  Paid  Oliver  Lucas  Logan  Tiana  End | Liam  Noah  James  4 people remaining. |
| Amelia  Thomas  Elias  End | 1. people remaining. |

## Hot Potato

Write a program that:

* Simulates the game of **Hot Potato**
* This is the rules:
  + Hot potato is a game in which **children form a circle and start passing a hot potato**.
  + The counting starts with **the first kid**
  + **Every nth toss, the child left with the potato leaves the game**
  + When a **kid leaves** the game, it passes the **potato along**
  + This continues **until there is only one kid left**
* **Print every kid that is removed from the circle**
* In the end, **print the kid that is left last**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Alva James William  2 | Removed James  Removed Alva  Last is William |
| Lucas Jacob Noah Logan Ethan  10 | Removed Ethan  Removed Jacob  Removed Noah  Removed Lucas  Last is Logan |
| Carter Dylan Jack Luke Gabriel  1 | Removed Carter  Removed Dylan  Removed Jack  Removed Luke  Last is Gabriel |