# Lab: Vectors, Lists and Iterators

This document defines the exercises for the ["C++ Fundamentals" course @ Software University](https://softuni.bg/trainings/4263/cpp-fundamentals-november-2023).

Please submit your solutions (source code) of all below-described problems in [Judge](https://judge.softuni.org/Contests/4484/Vectors-Lists-and-Iterators-Lab).

## Remove Negatives and Reverse

Write a program that:

* Read a **sequence of numbers**, separated with single space
* **Remove all negative numbers** from given **sequence**
* Print the remaining elements in **reversed order**

**Note**: In case of no elements left in the list, print "empty".

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 10 -5 7 9 -33 50 | 50 9 7 10 |
| 7 -2 -10 1 | 1 7 |
| -1 -2 -3 | empty |

## Products

Write a program that:

* Read an **integer number N (count of the products)** from the first line of the console
* Read **N lines of products (string)**
* Print a **numbered list** of all the products **ordered by name**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  Potatoes  Tomatoes  Onions  Apples | 1.Apples  2.Onions  3.Potatoes  4.Tomatoes |
| 3  Orange  Grape  Strawberry | 1.Grape  2.Orange  3.Strawberry |

## Gauss' Trick

Write a program that:

* Read a **sequence of numbers**, separated with single space
* **Sum** all **numbers in a list** in the following order:

**first + last, first + 1 + last - 1, first + 2 + last - 2, … first + n, last – n**

* Print **resulting sequence**



### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 2 3 4 5 | 6 6 3 |
| 1 2 3 4 | 1. 5 |

## Merging Sequences

Write a program that:

* Read **two sequences with numbers** from the first two lines of the console
* Create a **result sequence that contains the numbers from both of the lists**
* The **first element should be from the first list**, the **second from the second list** and so on
* Print the **resulting merged sequence**

**Note**: If the length of the two lists is not equal, just add the remaining elements at the end of the list.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 5 2 43 12 3 54 10 23  76 5 34 2 4 12 | 3 76 5 5 2 34 43 2 12 4 3 12 54 10 23 |
| 76 5 34 2 4 12  3 5 2 43 12 3 54 10 23 | 76 3 5 5 34 2 2 43 4 12 12 3 54 10 23 |

## Manipulations

Write a program that:

* Read a list of integers from the first line of the console
* Then until you receive **"end"**, you will be given different **commands:**
* **Add {number}** –add a number to the end of the list
* **Remove {number}** – remove a number from the list
* **RemoveAt {index}** – remove a number at a given index
* **Insert {number} {index}** – insert a number at a given index

**Note**: All the indices will be valid!

* When you receive the **"end"** command, print the **final state** of the list (**separated by spaces**).

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 19 2 53 6 43  Add 3  Remove 2  RemoveAt 1  Insert 8 3  end | 4 53 6 8 43 3 |
| 12 34 100 1 45 2 8  Add 30  Remove 12  Remove 3  RemoveAt 3  Insert 2 3  end | 34 100 1 2 2 8 30 |