# Lab: Lists

Tasks for exercise in class and for homework to the course ["Programming Fundamentals and Unit Testing" @ SoftUni](https://softuni.bg/trainings/4256/programming-fundamentals-and-unit-testing-september-2023).

Test your tasks in the Judge system: [https://judge.softuni.org/Contests/4429](https://judge.softuni.org/Contests/4429/Lists-Lab)

## Change List

Write a program that:

* **Reads a list of integers** from the console and receives **commands** to **manipulate the list**
* Then until you receive **"end"**, you will receive different **commands:**
  + **"**Delete {element}**"** – delete all elements in the array, which are equal to the given element
  + **"**Insert {element} {position}**"** – insert the element at the given position
* When you receive the **"end"** command, print the **final state** of the list (**separated by spaces**)

## Example Input / Output

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 2 3 4 5 5 5 6  Delete 5  Insert 10 1  Delete 5  end | 1 10 2 3 4 6 |
| 20 12 4 319 21 31234 2 41 23 4  Insert 50 2  Insert 50 5  Delete 4  end | 20 12 50 319 50 21 31234 2 41 23 |

## List of Products

Write a program that:

* Read an integer number **n** and **n lines of products**
* Print a **numbered list** of all the products **ordered by name**

## Example Input / Output

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  Potatoes  Tomatoes  Onions  Apples | 1.Apples  2.Onions  3.Potatoes  4.Tomatoes |
| 5  Carrots  Artichokes  Beans  Eggplants  Peppers | 1.Artichokes  2.Beans  3.Carrots  4.Eggplants  5.Peppers |

## Remove Negatives and Reverse

Write a program that:

* Read a **list of integers**
* **Remove all negative numbers** from it
* Print the remaining elements in **reversed order**
* If there are no elements left in the list, print "**empty**"

## Example Input / Output

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

## List Manipulation Basics

Write a program that:

* Reads a list of integers
* Then until you receive **"end"**, you will receive 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
* When you receive the **"end"** command, print the **final state** of the list (**separated by spaces**)

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

## Example Input / Output

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 19 2 53 6 43  Add 3  Remove 2  RemoveAt 1  Insert 8 3  end | 4 53 6 8 43 3 |
| 23 1 456 63 32 87 9 32  Remove 5  Add 1  Insert 14 2  RemoveAt 3  Add 34  end | 23 1 14 63 32 87 9 32 1 34 |

## List Manipulation Advanced

Write a program that:

* Reads a list of integers
* Then until you receive **"end"**, you will receive different **commands:**
  + **"Contains {number}"** – check if the list contains the number and if so - print **"Yes**", **otherwise** print **"No such number"**
  + **"PrintEven"** – print **all the even numbers**, **separated by a space**
  + **"PrintOdd"** – print **all the odd numbers, separated by a space**
  + **"GetSum"** – print the **sum of all the numbers**
  + **"Filter {condition} {number}"** – left in the list all the numbers that **fulfill the given condition**. The condition will be either '**<**', '**>**', "**>=**","**<=**"
* When you receive the **"end"** command, print the **final state** of the list (**separated by spaces**)

## Example Input / Output

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 34 678 67 5 563 98  Contains 23  PrintOdd  GetSum  Filter >= 21  end | No such number  5 67 5 563  1450  34 678 67 563 98 |
| 2 13 43 876 342 23 543  Contains 100  Contains 543  PrintEven  PrintOdd  GetSum  Filter >= 43  Filter < 100  end | No such number  Yes  2 876 342  13 43 23 543  1842  43 876 342 543 2 13 43 23 |