# Lab: Exceptions and Error Handling

Tasks for exercise in class and for homework to the course ["Programming Advanced for QA" @ SoftUni](https://softuni.bg/trainings/4257/programming-advanced-for-qa-november-2023)

Test your tasks in the Judge system: [https://judge.softuni.org/Contests/4491](https://judge.softuni.org/Contests/4491/Exceptions-and-Error-Handling-Lab)

## Square Root

Write a program that reads an integer **number** and **calculates** and **prints** its **square** **root**.

* If the number is negative, print "**Invalid number.**"
* In all cases finally, print "**Goodbye.**"

Use **try-catch-finally**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 9 | 3  Goodbye. |
| -1 | Invalid number.  Goodbye. |

## Enter Numbers

Write a method **ReadNumber(int start, int end)** that enters an integer number in a given range (**start…end**), **excluding** the numbers **start** and **end**. If an **invalid number** or a **non-number** text is entered, the method should **throw an exception**. Using this method write a program that enters **10 numbers**: **a1, a2, … a10, such that 1 < a1 < … < a10 < 100**. If the user enters an invalid number, continue while there are 10 valid numbers entered. Print the array elements, separated with **comma and space "**, **"**.

### Hints

* When the entered input holds a non-integer value, print: **"Invalid Number!"**
* When the entered input holds an integer that is out of range, print:

"**Your number is not in range {currentNumber} - 100!**"

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  3  4  5  6  7  8  9  10  11 | 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 |
| 1  2  1  3  p  4  5  6  7  8  9  10  11 | Your number is not in range (1 - 100)  Your number is not in range (1 – 100)  Invalid Number!  2, 3, 4, 5, 6, 7, 8, 9, 10, 11 |

## Sum of Integers

You will receive a sequence of **elements of different types**, separated by **space**. Your task is to calculate the sum of all valid integer numbers in the input. Try to add each element of the array to the sum and **write** **messages** for the possible **exceptions** while processing the element:

* If you receive an **element**, which is **not in the correct format** **(FormatException)**:  
  **"The element '{element}' is in wrong format!"**
* If you receive an **element**, which is **out of the integer** **type range (OverflowException)**:  
  **"The element '{element}' is out of range!"**

After each processed element add the following message:

**"Element '{element}' processed - current sum: {sum}"**

At the end print the total sum of all integers:

**"The total sum of all integers is: {sum}"**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2147483649 2 3.4 5 invalid 24 -4 | The element '2147483649' is out of range!  Element '2147483649' processed - current sum: 0  Element '2' processed - current sum: 2  The element '3.4' is in wrong format!  Element '3.4' processed - current sum: 2  Element '5' processed - current sum: 7  The element 'invalid' is in wrong format!  Element 'invalid' processed - current sum: 7  Element '24' processed - current sum: 31  Element '-4' processed - current sum: 27  The total sum of all integers is: 27 |
| 9876 string 10 -2147483649 -8 3 4.86555 8 | Element '9876' processed - current sum: 9876  The element 'string' is in wrong format!  Element 'string' processed - current sum: 9876  Element '10' processed - current sum: 9886  The element '-2147483649' is out of range!  Element '-2147483649' processed - current sum: 9886  Element '-8' processed - current sum: 9878  Element '3' processed - current sum: 9881  The element '4.86555' is in wrong format!  Element '4.86555' processed - current sum: 9881  Element '8' processed - current sum: 9889  The total sum of all integers is: 9889 |