12214994_Saurabh Rana

Title

Array Addition and Indexed Access with Exception Handling

Question No : 1 / 1

Problem Statement

Write a **C# console application** that reads **two arrays of integers** from the user, adds corresponding elements from both arrays to create a **third array**, and then prompts the user to input an **index** to access an element from the resultant array.

The program must include **error handling** for potential exceptions such as **FormatException** and **IndexOutOfRangeException**.

Write the solution within the **Program.cs** file.

Requirements

1. Input Handling

- The program prompts the user to enter elements for **two separate arrays** (`inputArray1` and `inputArray2`)
- Each input string is **space-separated** and converted into integer arrays (`numbers1` and `numbers2`)

2. Validation

- Check whether both arrays (`numbers1` and `numbers2`) have the **same length**
- If lengths differ, display an error message and terminate the program

3. Array Addition

- Create a third array (`sumArray`)
- Each element of `sumArray` should be the **sum of corresponding elements** from `numbers1` and `numbers2`

---

4. Index Access

- Prompt the user to input an **index**
- Display the element at the specified index from `sumArray`

---

5. Error Handling

The program must handle the following exceptions using **try-catch** blocks:

- **FormatException**
  - Occurs when non-integer values are entered
- **IndexOutOfRangeException**
  - Occurs when the entered index is outside the bounds of `sumArray`

---

6. Output

- Based on user input and processing results, print the element at the specified index or appropriate error messages

---

Input Format

- First line: Space-separated integers for **first array**
- Second line: Space-separated integers for **second array**
- Third line: Integer value representing the **index**

Output Format

- If index is valid:
- `Element at index {index} in the sum array: {value}`

- If index is out of range:
- `Error: Index out of range for the sum array.`
- `Exception Message: {errorMessage}`
- If non-integer values are entered:
- `Error: Invalid input format. Please enter integers only.`
- `Exception Message: {errorMessage}`
- If array lengths do not match:
- `Error: Arrays must have the same length for addition.`

Test Cases

Test Case ID	Input	Expected Output

TC01	`10 20 30 40`
	`40 30 20 10`
	`2`
		Element at index 2 in the sum array: 50

TC02	`11 12`
	`10 12`
	`3`
		Error: Index out of range for the sum array.
		Exception Message: Index was outside the bounds of the array.

TC03	`1 2 3 1`
	`3 2 1 a`
	`2`
		Error: Invalid input format. Please enter integers only.
		Exception Message: Input string was not in a correct format.

TC04	`1 2 3 a`	Error: Invalid input format. Please enter integers only. Exception Message: Input string was not in a correct format.

TC05	`1 2 3`
	`1 2`
	`2`
		Error: Arrays must have the same length for addition.

Sample Input 1

```
10 20 30 40
40 30 20 10
2
```

Sample Output 1

```
Element at index 2 in the sum array: 50
```

Sample Input 2

```
11 12
10 12
3
```

Sample Output 2

```
Error: Index out of range for the sum array.
Exception Message: Index was outside the bounds of the array.
```

Sample Input 3

```
1 2 3 1
3 2 1 a
2
```

Sample Output 3

```
Error: Invalid input format. Please enter integers only.
Exception Message: Input string was not in a correct format.
```

Sample Input 4

```
1 2 3 a
```

Sample Output 4

```
Error: Invalid input format. Please enter integers only.
Exception Message: Input string was not in a correct format.
```

Sample Input 5

```
1 2 3
1 2
2
```

Sample Output 5

```
Error: Arrays must have the same length for addition.
```

C# Solution (Program.cs)

```
using System.Collections;
```

```
class InvalidLength: Exception
{
    public InvalidLength(string Message) : base(Message) { }
}
class Program
{
    static void Main()
    {
        try{
            ArrayList arr1 = new ArrayList();
            string input1 = Console.ReadLine()??"";
            string[] values1 = input1.Split(' ');
            for(int i = 0; i < values1.Length; i++)
            {
                arr1.Add(Convert.ToInt32(values1[i]));
            }

            ArrayList arr2 = new ArrayList();
            string input2 = Console.ReadLine()??"";
            string[] values2 = input2.Split(' ');
            for(int i = 0; i < values2.Length; i++)
            {
                arr2.Add(Convert.ToInt32(values2[i]));
            }
            if (arr1.Count != arr2.Count)
            {
                throw new InvalidLength("Error: Arrays must have the same length for addition.");
            }

            ArrayList sumArray = new ArrayList();
            for (int i = 0; i < arr1.Count; i++)
            {
                sumArray.Add((int)arr1[i] + (int)arr2[i]);
            }

            int index = Convert.ToInt32(Console.ReadLine());

            // Output
            Console.WriteLine($"Element at index {index} in the sum array: {sumArray[index]}");

        }
        catch (FormatException ex)
        {
            Console.WriteLine("Error: Invalid input format. Please enter integers only.");
            Console.WriteLine("Exception Message: " + ex.Message);
        }
        catch (IndexOutOfRangeException ex)
        {
            Console.WriteLine("Error: Index out of range for the sum array.");
            Console.WriteLine("Exception Message: " + ex.Message);
```

```
        }
        catch (InvalidLength ex)
        {
            Console.WriteLine(ex.Message);
        }




    }
}
```