**Lab Report: Array Implementation and For Loop Practice**

Lab Title:

Array Implementation and For Loop Practice

Name: Naldrelle Yuan M. Briones

Course: BSECE – 1

Date Performed: November 16, 2024

1. Objectives:

The objective of this lab is to practice working with arrays and for loops in Java. Students will learn how to declare, initialize, and manipulate arrays, as well as implement for loops to process the data in arrays. This activity focuses on solving engineering-related problems using arrays.

1. **Problem Statement:**
   1. **Sum and Average of an Array**
      * **Task:** Write a Java program that:
        + Declare an array of integers with 10 elements.
        + Prompts the user to input 10 numbers to fill the array.
        + Computes and prints the sum and average of the array elements.
   2. **Maximum and Minimum Element Finder**
      * **Task**: Write a Java program that:
        + Declares an array of integers with 15 elements.
        + Prompts the user to input 15 numbers to fill the array.
        + Finds and prints the maximum and minimum element in the array using a for loop.
   3. **Reverse an Array**
      * **Task**: Write a Java program that:
        + Declares an array of 5 floating-point numbers.
        + Prompts the user to input 5 numbers.
        + Prints the array in reverse order using a for loop.
   4. **Frequency Counter**
      * **Task**: Write a Java program that:
        + Declares an array of integers with 10 elements.
        + Prompts the user to input 10 integers between 1 and 100.
        + Finds and prints how many times a particular number appears in the array. The number to be checked is also input by the user.
2. Pseudocode / Algorithm:

Main Function:

1. Initialize loop as true

2. While loop is true:

a. Display menu to the user

b. Get user's choice

c. Switch on user's choice:

- Case "1": Call sumAndAverageAndFrequencyCounter()

- Case "2": Call findMaxAndMin()

- Case "3": Call reverseArray()

- Default: Display "Invalid choice"

d. Ask if user wants to perform another operation

e. If user chooses "No", set loop to false

sumAndAverageAndFrequencyCounter Function:

1. Initialize array numbers with 10 elements

2. For i from 0 to 9:

a. Prompt user to enter number

b. Store number in numbers[i]

3. Initialize sum as 0

4. For each number in numbers:

a. Add number to sum

5. Calculate average as sum / 10

6. Display sum and average

7. Initialize frequencyMap as empty map

8. For each number in numbers:

a. Increment frequency of number in frequencyMap

9. Display frequency count of each number

findMaxAndMin Function:

1. Initialize array numbers with 10 elements

2. For i from 0 to 9:

a. Prompt user to enter number

b. Store number in numbers[i]

3. Initialize max and min as numbers[0]

4. For i from 1 to 9:

a. If numbers[i] > max, set max to numbers[i]

b. If numbers[i] < min, set min to numbers[i]

5. Display max and min

reverseArray Function:

1. Initialize array numbers with 10 elements

2. For i from 0 to 9:

a. Prompt user to enter number

b. Store number in numbers[i]

3. For i from 0 to floor(n/2):

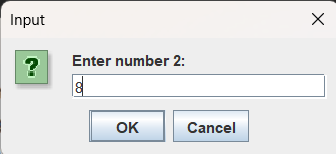
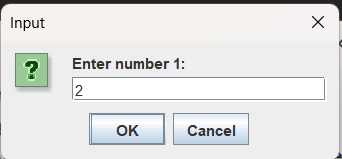
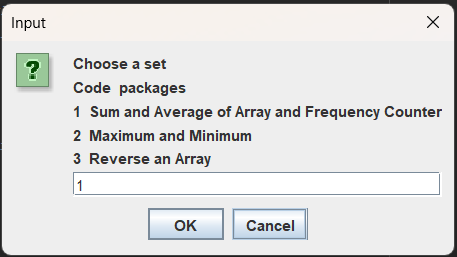
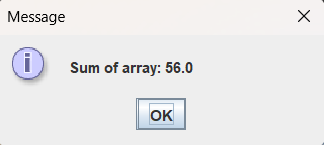
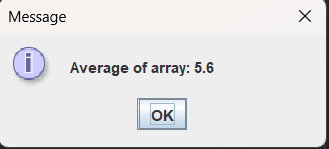
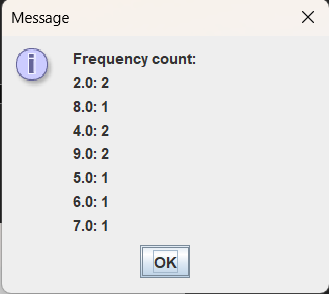
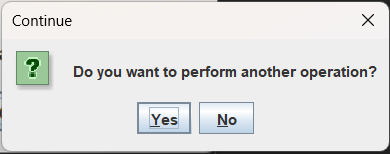
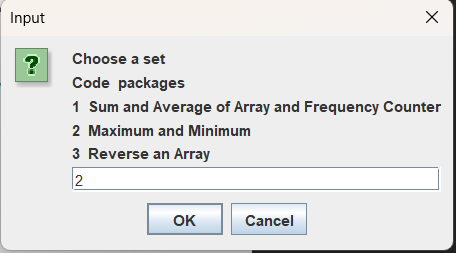
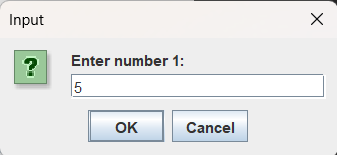
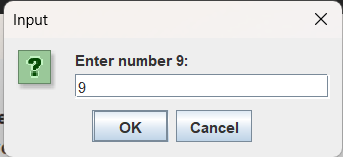
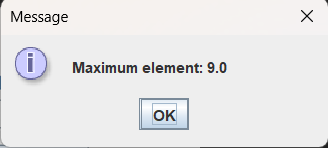
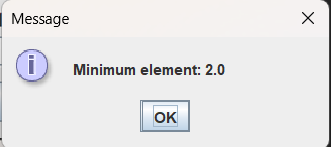
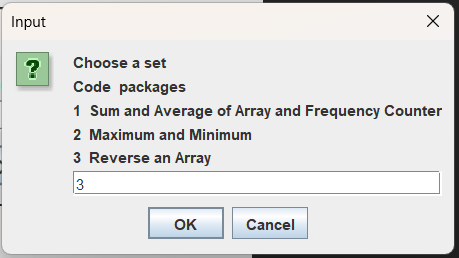
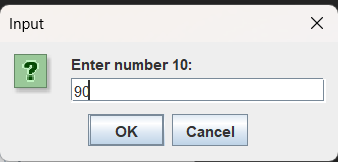
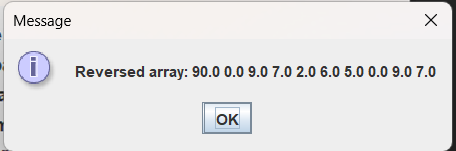
a. Swap numbers[i] with numbers[n-1-i]

4. Display reversed array

1. Java Code:

https://github.com/DotYohan/LABORATORY-3/blob/DotYohan-Lab3.1/Lab31.java

5. Sample Output:

1. Test Cases:

|  |  |  |
| --- | --- | --- |
| Test Cases | Input | Expected Output |
| sumAndAverageAndFrequencyCounter | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] | Sum: 55, Average: 5.5, Frequency: {1:1, 2:1, ..., 10:1} |
| sumAndAverageAndFrequencyCounter | [5, 5, 5, 5, 5, 5, 5, 5, 5, 5] | Sum: 50, Average: 5, Frequency: {5:10} |
| sumAndAverageAndFrequencyCounter | [1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 8.5, 9.5, 10.5] | Sum: 60.5, Average: 6.05, Frequency: {1.5:1, 2.5:1, ..., 10.5:1} |
| findMaxAndMin | [3, 1, 4, 1, 5, 9, 2, 6, 5, 3] | Max: 9, Min: 1 |
| findMaxAndMin | [10, 9, 8, 7, 6, 5, 4, 3, 2, 1] | Max: 10, Min: 1 |
| findMaxAndMin | [1.1, 2.2, 3.3, 4.4, 5.5, 6.6, 7.7, 8.8, 9.9, 10.1] | Area of Circle Segment = 4.19 |
| Calculate missing side of a triangle | Side a: 3, Side b: 4, Angle C: 90 | Max: 10.1, Min: 1.1 |
| reverseArray | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] | Reversed Array: [10, 9, 8, 7, 6, 5, 4, 3, 2, 1] |
| reverseArray | [10, 20, 30, 40, 50, 60, 70, 80, 90, 100] | Reversed Array: [100, 90, 80, 70, 60, 50, 40, 30, 20, 10] |
| reverseArray | [1.1, 2.2, 3.3, 4.4, 5.5, 6.6, 7.7, 8.8, 9.9, 10.1] | Reversed Array: [10.1, 9.9, 8.8, 7.7, 6.6, 5.5, 4.4, 3.3, 2.2, 1.1] |

1. Discussion:

During our recent lab exercises in Java programming, we explored functionalities such as calculating the sum, average, and frequency of array elements, finding maximum and minimum values, and reversing arrays. These tasks provided invaluable learning experiences, particularly in array manipulation, basic statistics, and algorithm efficiency. The use of HashMaps for frequency counts was especially enlightening, showcasing the power of data structures. Despite some challenges, such as handling edge cases, ensuring correct input handling, and managing indices for array reversal, these exercises significantly enhanced my understanding and problem-solving skills. They emphasized the importance of efficient algorithms and memory management, and I now feel more confident in tackling complex programming tasks. Overall, these lab sessions were instrumental in deepening my knowledge and improving my debugging capabilities, setting a solid foundation for future programming endeavors.

1. Conclusion:

In conclusion, our lab exercises on Java programming were highly educational, covering key areas like array manipulation, statistical calculations, and algorithm efficiency. We tackled tasks such as calculating sums, averages, and frequencies, identifying maximum and minimum values, and reversing arrays. These activities enhanced our understanding of programming concepts and practical problem-solving. Despite challenges like input validation and edge case handling, we significantly improved our skills and confidence in managing complex programming tasks. The lab sessions provided a solid foundation for further exploration in Java, reinforcing our technical abilities and debugging proficiency.