Aror University of Art, Architecture, Design & Heritage Sukkur

Department of AI-Multimedia and Gaming

Lab 01: Fundamentals of array data structure — Date:

Subject: Data Structure (CSC221), Fall 2024 — Instructor: Abdul Ghafoor

Lab objectives: Develop skills in array data structure, string operations, and algorithmic problem-solving, focusing on searching, inserting, deleting, and comparing.

Lab Task 01: Array Basics and Traversal

Objective: Learn how to declare, initialize, and traverse an array.

Exercise 1: Declare an integer array of size 6 and initialize it with the values {5, 15, 25, 35, 45, 55}. Write a program to print all elements of the array.

Expected output

```
5 15 25 35 45 55
```

Exercise 2: Modify the program to print each element with its corresponding index.

Expected output

```
Element at index 0: 5
Element at index 1: 15
Element at index 2: 25
Element at index 3: 35
Element at index 4: 45
Element at index 5: 55
```

Lab Task 2: Insertion in Arrays

Objective: Practice inserting elements at different positions in an array.

Exercise 1: Insert an element at the start of the array.

- Start with the array {12, 24, 36, 48, 60}.
- Insert the value 6 at the beginning.

Expected output

```
6 12 24 36 48 60
```

Exercise 2: Insert an element in the middle of the array.

- Start with the array {100, 200, 300, 400, 500}.
- Insert the value 250 at the 2nd index.

Expected output

```
100 200 250 300 400 500
```

Exercise 3: Insert an element at the end of the array.

- Start with the array {3, 6, 9, 12, 15}.
- Insert the value 18 at the end.

Expected output

```
3 6 9 12 15 18
```

Lab Task 3: Deletion in Arrays

Objective: Learn how to delete elements from an array.

- Exercise 1: Delete an element from a specific position.
- Start with the array {8, 16, 24, 32, 40}.
- Delete the element at index 2.

Expected output

```
8 16 32 40 0
```

Lab Task 4: Searching in Arrays

Objective: Practice different search techniques in arrays.

- Exercise 1: Search for an element by its index.
- Start with the array {13, 26, 39, 52, 65}.
- Search for the element at index 4.

Expected Output:

```
Element at index 4: 65
```

Exercise 2: Implement linear search to find an element by its value.

- Start with the array {4, 6, 2, 8, 10}.
- Search for the value 8.

Expected Output:

```
Element 8 found at index 3
```

Exercise 3: Implement binary search on a sorted array.

- Start with the sorted array {11, 22, 33, 44, 55}.
- Search for the value 33.

Expected Output:

```
Element 33 found at index 2
```

Lab Task 05: Reverse Array

Write a function **reverseArray** that takes an array of integers as input and returns a new array that is the reverse of the input array.

Input

```
1 3 5 7 9
```

Expected output

```
9 7 5 3 1
```

Lab Task 06: Palindrome

A **palindrome** is a sequence of characters that reads the same forward and backward. In the context of strings, a palindrome is a word, phrase, or sequence that remains unchanged when its characters are reversed. For example, "madam" and "racecar" are palindromes, while "hello" is not.

Task: Given a string, check if it is a palindrome or not. A string is considered a palindrome if it reads the same forward and backward.

Hint: The string str is converted to a array of characters using the .toCharArray() method.

Input string

```
radar
```

Expected output

```
Palindrome
```

Input string

```
apple
```

Expected output

```
Not Palindrome
```

Lab Task 07: Anagram Check

Given two strings, determine if they are anagrams of each other. Two strings are considered anagrams if they contain the same characters in the same frequencies but possibly in a different order.

Input

```
str1 = "listen"
str2 = "silent"
```

Expected output

```
Anagram
```

Input string

```
str1 = "hello"
str2 = "world"
```

Expected output

```
Not Anagram
```

Submission Guidelines

1. **File Format**:
    - Submit all your lab exercises as individual Java files (.java).
    - Each Java file should correspond to a specific lab task.
2. **Output Snapshots**:
    - For each lab exercise, include a screenshot of the output showing that your code runs successfully.
    - Save the screenshots as image files (.png or .jpg), and ensure they are clearly named to match the corresponding Java files.
3. **Folder Structure**:
    - Create a single folder for your submission.
    - Name the folder as follows: *[Your Name]_Lab[Lab Number]_[Course Name].*
        - **Example**: *Shoaib_Lab01_DataStructure*
4. **Organization**:
    - Place all Java files and corresponding output snapshots in the folder.
    - Ensure that the files are named clearly and consistently to avoid any confusion.
5. **Compress the Folder**:
    - Compress the folder into a .zip file.
6. **Submission on Google Classroom**:
    - Log in to Google Classroom and navigate to the assignment for this lab.
    - Upload the .zip file containing your folder to the assignment submission area.
    - Double-check that your submission has been uploaded correctly.
7. Due Date: