Data Structures & Algorithms

Assignment

Submitted by:

Bisma Rathore

Roll Number: **2k24/CSE/43**

Department of Computer Science

Assignment Questions:

1. Write a program to reverse an array using stack data structure.
2. Write a program to match the parentheses stored in a string using stack data structure.
3. Write a program to calculate the sum of all integer elements in an integer array by implementing a recursive sum method/function.

Programs:

1. Program to Reverse an Array using Stack

```
import java.util.*;
public class ReverseArrayUsingStack{
public static void main(String[] args){
int[] array = {10, 20, 30, 40, 50};
      System.out.println("Original Array: " + Arrays. toString(array));
reverseArray(array);
      System.out.println("Reversed Array: " + Arrays. toString(array));
   }

// Method to reserve array using stack

   public static void reverseArray(int[] arr) {
Stack<Integer> stack = new Stack<>();
// Push all elements into the stack
  for (int num : arr) {
  stack.push(num);
}
```

```
// Pop elements back into the array to reverse
  for (int i = 0; i < arr .length; i++) {
  arr[i] = stack. pop();
}
   }
}
```

Output:

Original Array: [10, 20, 30, 40, 50]

Reversed Array: [50, 40, 30, 20, 10]

2. Program to Match Parentheses using Stack

```
import java.util.*;
 public class ParenthesesMatcher {
public static void main(String[] args){
      String expression = "{[()]}";
      System.out.println("Expression: " + expression);
      System.out.println("Is balanced? " + isBalanced(expression));
   }

// Method to check balanced parentheses

   public static boolean isBalanced(String expr) {
Stack<Character> stack = new Stack<>();
  for (char ch : expr .toCharArray()) {
if (ch == '(' || ch == '{' || ch == '[' ){
    stack.push(ch);
     } else if (ch == ')' || ch == '}' || ch == ']') {
```

```
    if (stack. isEmpty())
    return false;
            char top = stack.pop();
          if (!matches(top, ch))
   return false;
         }
      }
      return stack.isEmpty();
  }

// Helper method to match corresponding brackets

   public static boolean matches(char open, char close){
  return (open == '(' && close == ')') ||
    (open == '{' && close == '}') ||
           (open == '[' && close == ']');
  }
}
```

Output:

Expression: {[()]}

Is balanced? true

3. Program to Calculate Sum of Array Elements using Recursion (Java)

```
public class RecursiveArraySum {
 public static void main(String[] args) {
int[] numbers = {5, 10, 15, 20, 25};
 int sum = recursiveSum(numbers, 0);
```

```
        System.out.println("Array: " + java.util.Arrays .toString(numbers));
        System.out.println("Sum of Array Elements: " + sum);
    }

// Recursive function to calculate sum

    public static int recursiveSum(int[] arr, int index) {
      if (index == arr.length)
return 0;
        return arr[index] + recursiveSum(arr, index +1);
    }
}
```

Output:

```
Array: [5, 10, 15, 20, 25]
Sum of Array Elements: 75
```