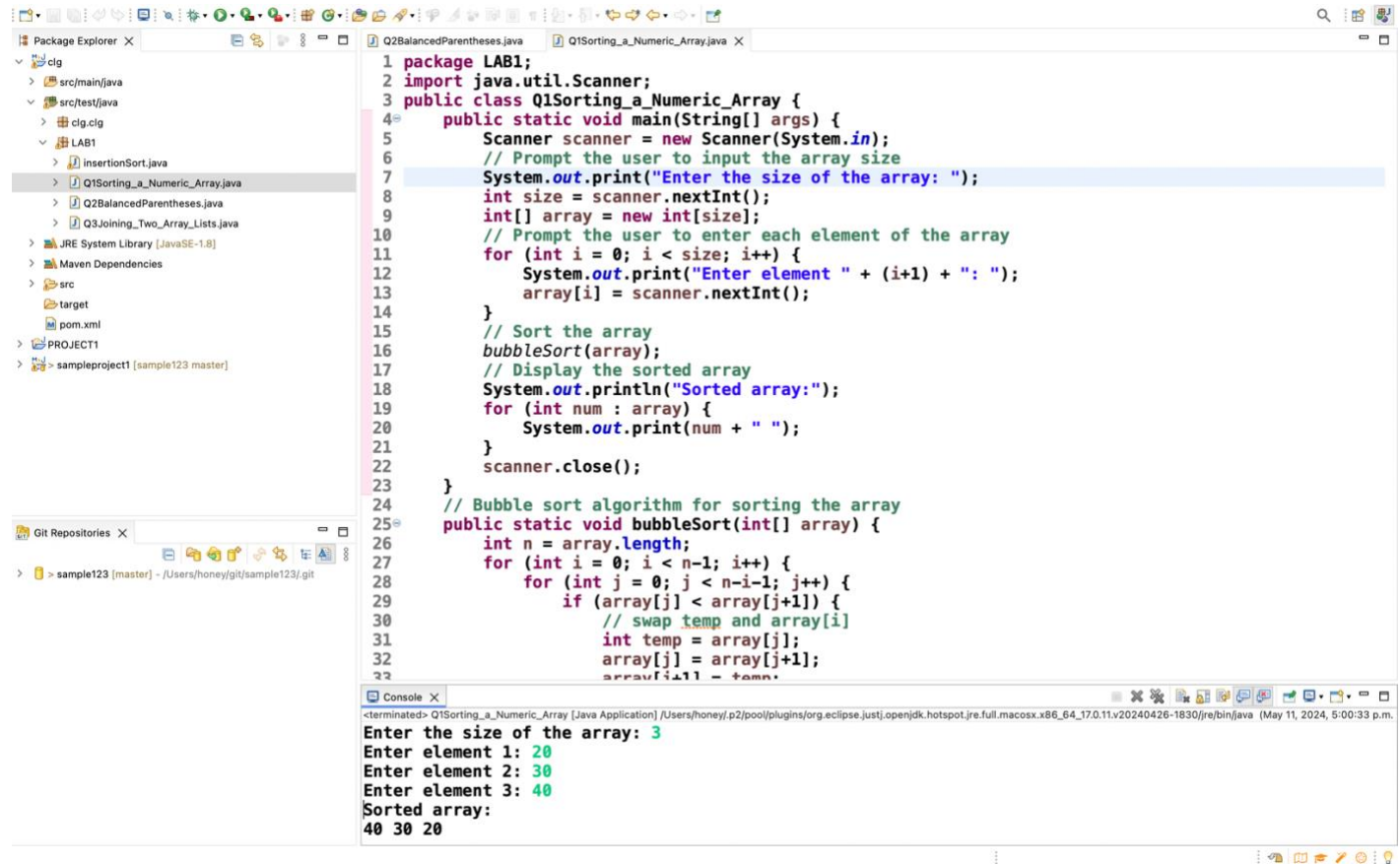


Q1Sorting a Numeric Array

Program and output



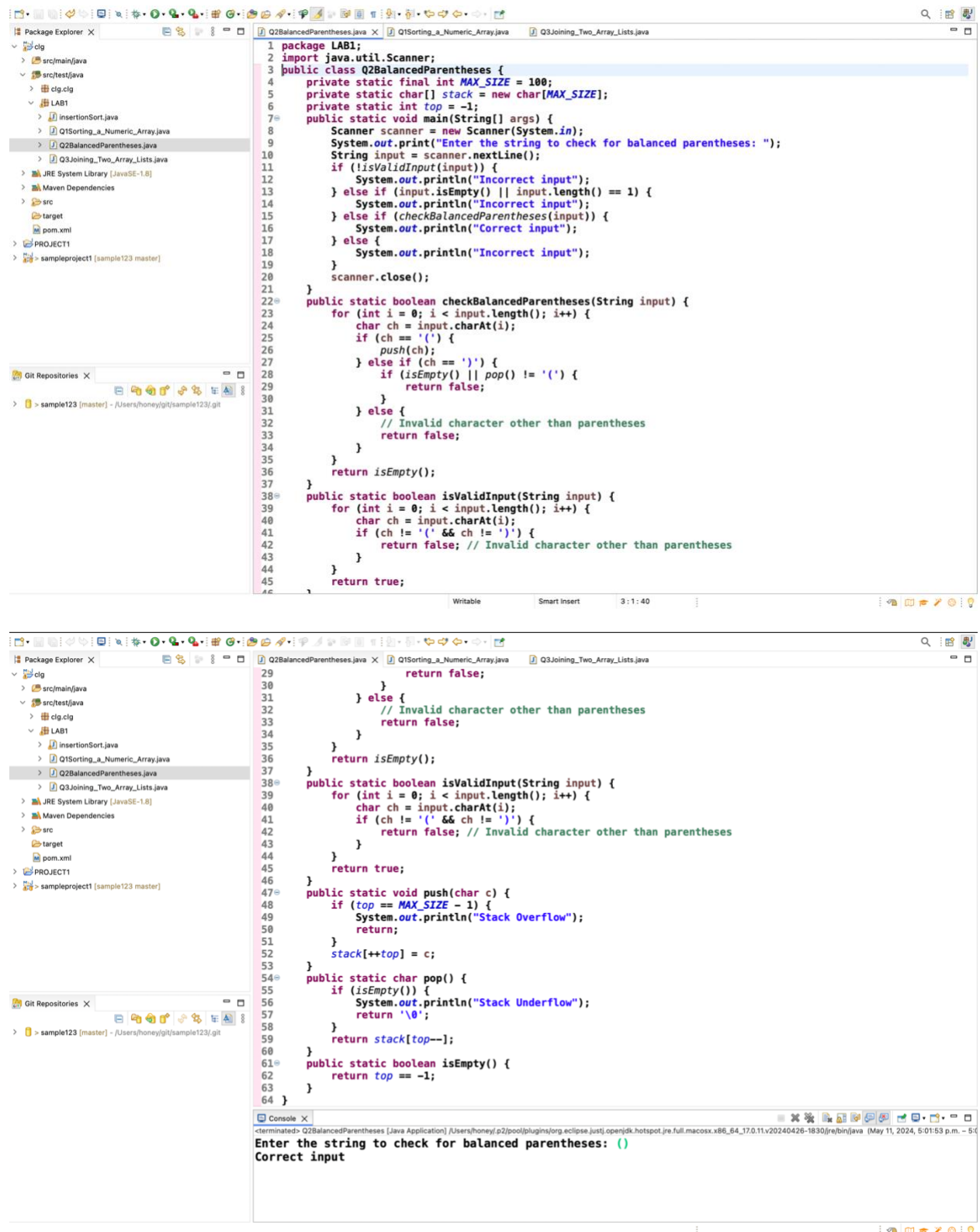
```
1 package LAB1;
2 import java.util.Scanner;
3 public class Q1Sorting_a_Numeric_Array {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         // Prompt the user to input the array size
7         System.out.print("Enter the size of the array: ");
8         int size = scanner.nextInt();
9         int[] array = new int[size];
10        // Prompt the user to enter each element of the array
11        for (int i = 0; i < size; i++) {
12            System.out.print("Enter element " + (i+1) + ": ");
13            array[i] = scanner.nextInt();
14        }
15        // Sort the array
16        bubbleSort(array);
17        // Display the sorted array
18        System.out.println("Sorted array:");
19        for (int num : array) {
20            System.out.print(num + " ");
21        }
22        scanner.close();
23    }
24    // Bubble sort algorithm for sorting the array
25    public static void bubbleSort(int[] array) {
26        int n = array.length;
27        for (int i = 0; i < n-1; i++) {
28            for (int j = 0; j < n-i-1; j++) {
29                if (array[j] < array[j+1]) {
30                    // swap temp and array[i]
31                    int temp = array[j];
32                    array[j] = array[j+1];
33                    array[j+1] = temp;
34                }
35            }
36        }
37    }
38 }
```

Console Output:

```
<terminated> Q1Sorting_a_Numeric_Array [Java Application] /Users/honeyj/p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_17.0.11.v20240426-1830/jre/bin/java (May 11, 2024, 5:00:33 p.m.)
Enter the size of the array: 3
Enter element 1: 20
Enter element 2: 30
Enter element 3: 40
Sorted array:
40 30 20
```

Q2 Implementing a Stack as an Array

Program and output



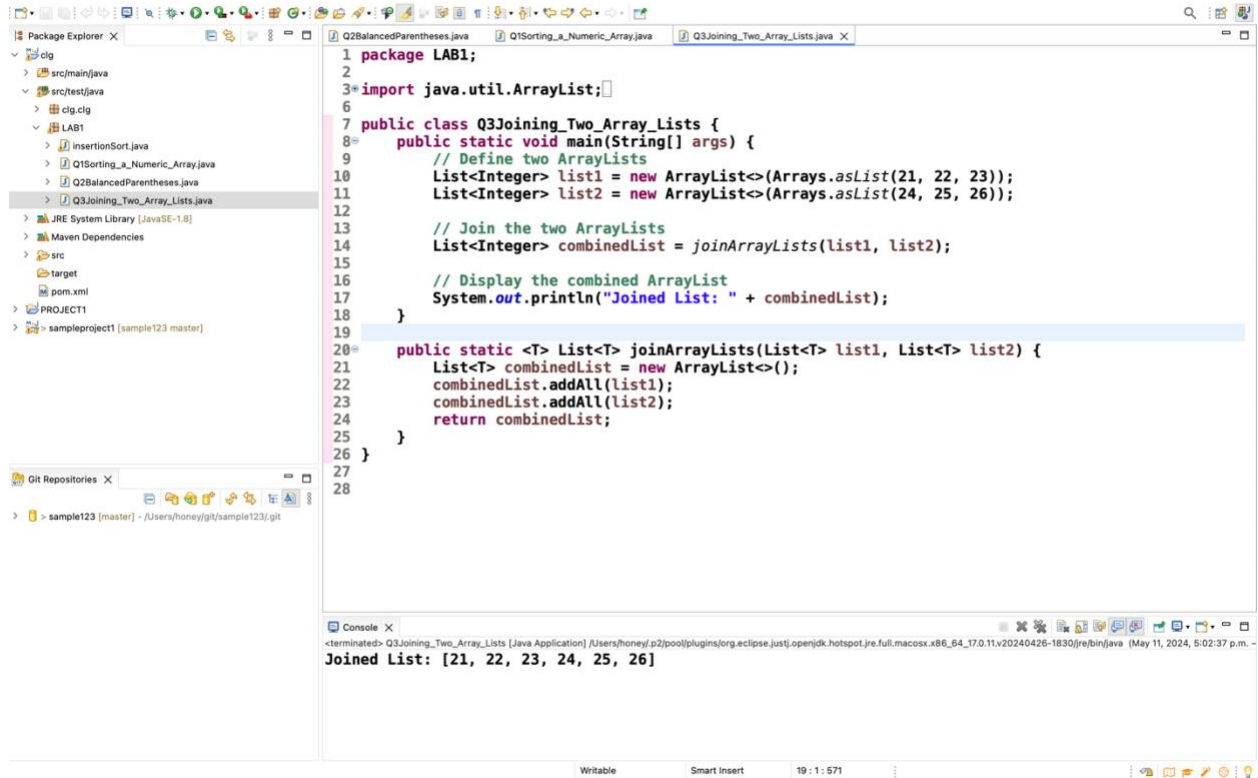
```
1 package LAB1;
2 import java.util.Scanner;
3 public class Q2BalancedParentheses {
4     private static final int MAX_SIZE = 100;
5     private static char[] stack = new char[MAX_SIZE];
6     private static int top = -1;
7     public static void main(String[] args) {
8         Scanner scanner = new Scanner(System.in);
9         System.out.print("Enter the string to check for balanced parentheses: ");
10        String input = scanner.nextLine();
11        if (!isValidInput(input)) {
12            System.out.println("Incorrect input");
13        } else if (input.isEmpty() || input.length() == 1) {
14            System.out.println("Incorrect input");
15        } else if (checkBalancedParentheses(input)) {
16            System.out.println("Correct input");
17        } else {
18            System.out.println("Incorrect input");
19        }
20        scanner.close();
21    }
22    public static boolean checkBalancedParentheses(String input) {
23        for (int i = 0; i < input.length(); i++) {
24            char ch = input.charAt(i);
25            if (ch == '(') {
26                push(ch);
27            } else if (ch == ')') {
28                if (isEmpty() || pop() != '(') {
29                    return false;
30                }
31            } else {
32                // Invalid character other than parentheses
33                return false;
34            }
35        }
36        return isEmpty();
37    }
38    public static boolean isValidInput(String input) {
39        for (int i = 0; i < input.length(); i++) {
40            char ch = input.charAt(i);
41            if (ch != '(' && ch != ')') {
42                return false; // Invalid character other than parentheses
43            }
44        }
45        return true;
46    }
47    public static void push(char c) {
48        if (top == MAX_SIZE - 1) {
49            System.out.println("Stack Overflow");
50            return;
51        }
52        stack[++top] = c;
53    }
54    public static char pop() {
55        if (isEmpty()) {
56            System.out.println("Stack Underflow");
57            return '\0';
58        }
59        return stack[top--];
60    }
61    public static boolean isEmpty() {
62        return top == -1;
63    }
64 }
```

Console Output:

```
<terminated> Q2BalancedParentheses [Java Application] /Users/honey/p2/pool/plugins/org.eclipse.justi.openjdk.hotspot.jre.full.macosx.x86_64_17.0.11.v20240426-1830/jre/bin/java (May 11, 2024, 5:01:53 p.m. - 5:01:53 p.m.)
Enter the string to check for balanced parentheses: ()
Correct input
```

Q.3 - Joining Two Array Lists

Program and output



```
1 package LAB1;
2
3 import java.util.ArrayList;
4
5
6
7 public class Q3Joining_Two_Array_Lists {
8     public static void main(String[] args) {
9         // Define two ArrayLists
10        List<Integer> list1 = new ArrayList<>(Arrays.asList(21, 22, 23));
11        List<Integer> list2 = new ArrayList<>(Arrays.asList(24, 25, 26));
12
13        // Join the two ArrayLists
14        List<Integer> combinedList = joinArrayLists(list1, list2);
15
16        // Display the combined ArrayList
17        System.out.println("Joined List: " + combinedList);
18    }
19
20    public static <T> List<T> joinArrayLists(List<T> list1, List<T> list2) {
21        List<T> combinedList = new ArrayList<>();
22        combinedList.addAll(list1);
23        combinedList.addAll(list2);
24        return combinedList;
25    }
26 }
27
28
```

Console

<terminated> Q3Joining_Two_Array_Lists [Java Application] /Users/honey/p2/pool/plugins/org.eclipse.justi.openjdk.hotspot.jre.full.macosx.x86_64.17.0.11.v20240426-1830/jre/bin/java (May 11, 2024, 5:02:37 p.m. -

Joined List: [21, 22, 23, 24, 25, 26]