Mobile Application Development Lab

CSL-341

Lab Journal

Student Name: Fawad Naveed
Enrollment No: 01-134222-049
Class and Section: 6-B

Department of Computer Science
BAHRIA UNIVERSITY ISLAMABAD

Table of Contents

Task 1:

Problem Statement:

Find the Largest Number in a List

Code

```
void main() {
  List<int> numbers = [12, 45, 78, 23, 56, 89, 10];
  int largest = numbers.reduce((a, b) => a > b ? a : b);

  print("The largest number is: $largest");
}
```

Task 2:

Problem Statement:

Use merge sort to sort a List.

Code:

```
List<int> mergeSort(List<int> list) {
  if (list.length <= 1) {
    return list;
  }

  int mid = list.length ~/ 2;
  List<int> left = mergeSort(list.sublist(0, mid));
  List<int> right = mergeSort(list.sublist(mid));

  return merge(left, right);
}

List<int> merge(List<int> left, List<int> right) {
  List<int> result = [];
  int i = 0, j = 0;

  while (i < left.length && j < right.length) {
    if (left[i] < right[j]) {
      result.add(left[i]);
      i++;
    } else {
      result.add(right[j]);
      j++;
    }
  }

  result.addAll(left.sublist(i));
  result.addAll(right.sublist(j));

  return result;
}

void main() {
  List<int> numbers = [38, 27, 43, 3, 9, 82, 10];
  print("Sorted List: ${mergeSort(numbers)}");
}
```

Task 3:

Problem Statement:

Implement a Stack from Scratch.

Code:

```
class Stack<T> {
  List<T> _stack = [];
  void push(T item) {
    _stack.add(item);
  }

  T? pop() {
    if (_stack.isNotEmpty) {
      return _stack.removeLast();
    }
    return null;
  }

  T? peek() {
    if (_stack.isNotEmpty) {
      return _stack.last;
    }
    return null;
  }

  bool isEmpty() {
    return _stack.isEmpty;
  }

  void display() {
    print("Stack: $_stack");
  }
}

void main() {
  Stack<int> stack = Stack<int>();

  stack.push(10);
  stack.push(20);
  stack.push(30);

  stack.display(); // Output: Stack: [10, 20, 30]

  print("Popped: ${stack.pop()}"); // Output: Popped: 30
  stack.display();
}
```