Bahria University, Islamabad Campus

Department of Computer Science

Mobile App Development Lab

CSL 341

Name: Syed Muhammad Ali Raza Naqvi

Enrollment Number: 01-134222-149

Section: BS(CS) – 6(B)

Instructor: Mr. Mohsin Javed Butt

Lab Journal # 02

Github Repository Link:

https://github.com/alirnaqvi/Mobile-Application-Development-Lab

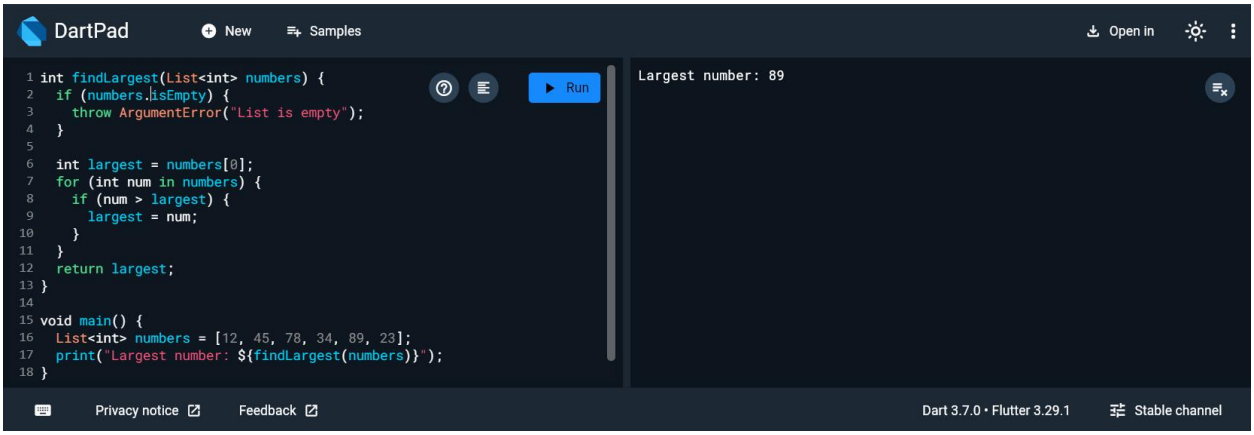
Task # 01:

Solution:

Code:

```
int findLargest(List<int> numbers) {
  if (numbers.isEmpty) {
    throw ArgumentError("List is empty");
  }

  int largest = numbers[0];
  for (int num in numbers) {
    if (num > largest) {
      largest = num;
    }
  }
  return largest;
}

void main() {
  List<int> numbers = [12, 45, 78, 34, 89, 23];
  print("Largest number: ${findLargest(numbers)}");
}
```

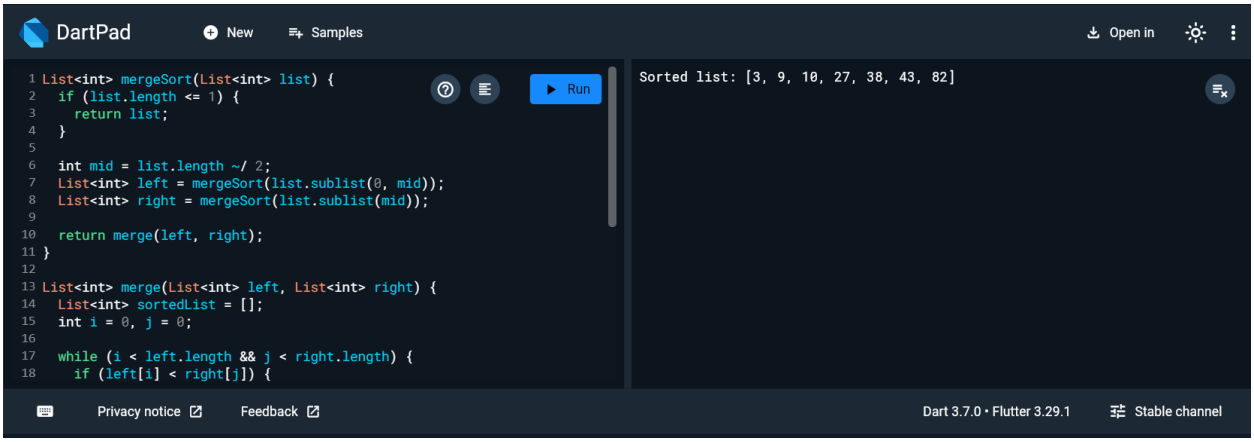
Output:



Task # 02:
Solution:
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> sortedList = [];
  int i = 0, j = 0;

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

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

  return sortedList;
}

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

Output:




```
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> sortedList = [];
  int i = 0, j = 0;

  while (i < left.length && j < right.length) {
    if (left[i] < right[j]) {
```

Task # 03:

Solution:

Code:

```
class Stack<T> {
  List<T> _items = [];

  void push(T item) {
    _items.add(item);
  }

  T? pop() {
    if (isEmpty()) {
      print("Stack is empty");
      return null;
    }
    return _items.removeLast();
  }

  T? peek() {
    if (isEmpty()) {
      print("Stack is empty");
      return null;
    }
    return _items.last;
  }

  bool isEmpty() {
    return _items.isEmpty;
  }
```

```
  int size() {
    return _items.length;
  }
}

void main() {
  Stack<int> stack = Stack<int>();
  stack.push(10);
  stack.push(20);
  stack.push(30);

  print("Top element: ${stack.peek()}");
  print("Popped element: ${stack.pop()}");
  print("Stack size: ${stack.size()}");
}
```

Output: