MOBILE APPLICATION DEVELOPMENT LAB

(CSL 4341)

DEPARTMENT OF COMPUTER SCIENCES

BAHRIA UNIVERSITY, ISLAMABAD CAMPUS

CLASS: BS (CS) – 6B

SESSION: SPRING 2025

SUBMITTED TO: MOHSIN JAVED BUTT

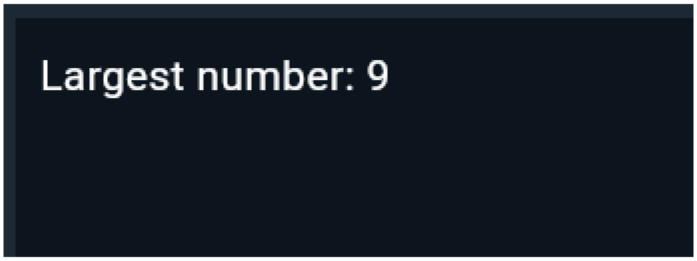
SUBMITTED BY: AREEBA AMIN (01-134222-028)

SUBMISSION DATE: 3RD March, 2025

TASK 1:

Find the largest number in a given list.

```
int findLargest(List<int> nums) {
  int largest = nums[0];
  for (int i = 1; i < nums.length; i++) {
    if (nums[i] > largest) {
      largest = nums[i];
    }
  }
  return largest;
}
void main() {
  List<int> numbers = [3, 8, 2, 5, 9, 1];
  print("Largest number: ${findLargest(numbers)}");
}
```

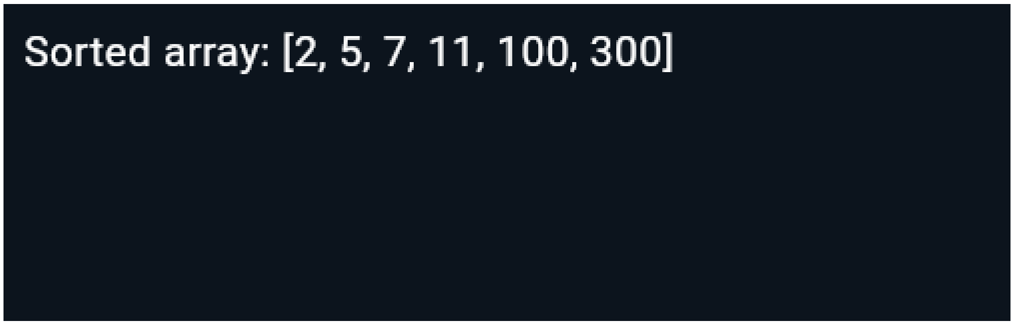


```
Largest number: 9
```

TASK 2:

Use merge sort to sort a List.

```
void merge(List<int> arr, int left, int mid, int right) {
  int n1 = mid - left + 1;
  int n2 = right - mid;
  List<int> L = List.filled(n1, 0);
  List<int> R = List.filled(n2, 0);
  for (int i = 0; i < n1; i++) L[i] = arr[left + i];
  for (int j = 0; j < n2; j++) R[j] = arr[mid + 1 + j];
  int i = 0, j = 0, k = left;
  while (i < n1 && j < n2) {
    if (L[i] <= R[j]) {
      arr[k] = L[i];
      i++;
    } else {
      arr[k] = R[j];
      j++;
    }
    k++;
  }
  while (i < n1) {
    arr[k] = L[i];
    i++;
    k++;
  }
  while (j < n2) {
    arr[k] = R[j];
    j++;
    k++;
  }
}
void mergeSort(List<int> arr, int left, int right) {
  if (left < right) {
    int mid = left + (right - left) ~/ 2;
    mergeSort(arr, left, mid);
    mergeSort(arr, mid + 1, right);
    merge(arr, left, mid, right);
  }
}
void main() {
  List<int> arr = [100, 11, 2, 5, 300, 7];
  mergeSort(arr, 0, arr.length - 1);
  print("Sorted array: $arr");
}
```



```
Sorted array: [2, 5, 7, 11, 100, 300]
```

Task 3:

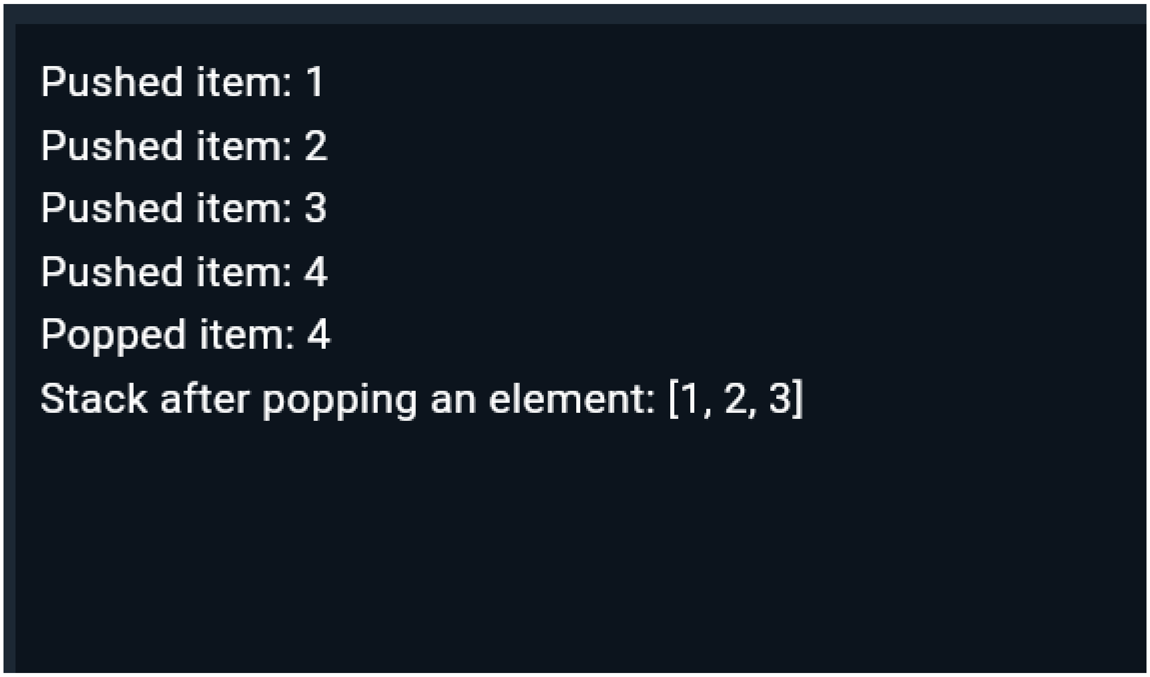
Implement a Stack from Scratch.

```
class Stack {
  List<String> _stack = [];

  void push(String item) {
    _stack.add(item);
    print("Pushed item: $item");
  }

  String pop() {
    if (_stack.isEmpty) {
      return "Stack is empty";
    }
    return _stack.removeLast();
  }

  bool isEmpty() {
    return _stack.isEmpty;
  }
}
void main() {
  Stack stack = Stack();
  stack.push("1");
  stack.push("2");
  stack.push("3");
  stack.push("4");
  print("Popped item: ${stack.pop()}");
  print("Stack after popping an element: ${stack._stack}");
}
```



```
Pushed item: 1
Pushed item: 2
Pushed item: 3
Pushed item: 4
Popped item: 4
Stack after popping an element: [1, 2, 3]
```