Mobile Application Development Lab

CSL-341

Lab Journal 2

Atika Abid
01-134222-032
BS(CS) – 6B

Department of Computer Science
BAHRIA UNIVERSITY ISLAMABAD

Table of Contents

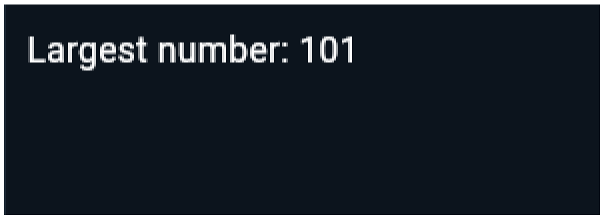
TASK 1:

Find the largest number in a given list.

CODE:

```
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 = [37, 68, 42, 15, 96, 101];
 print("Largest number: ${findLargest(numbers)}");
}
```

OUTPUT:



```
Largest number: 101
```

TASK 2:

Use merge sort to sort a List.

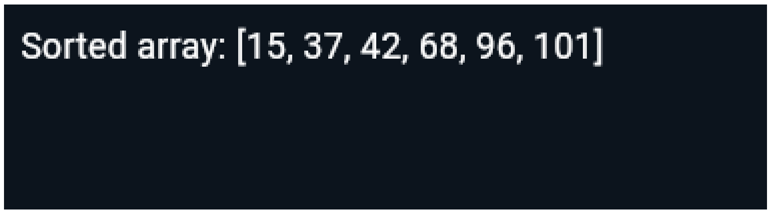
CODE:

```
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 = [37, 68, 42, 15, 96, 101];
 mergeSort(arr, 0, arr.length - 1);
 print("Sorted array: $arr");
}
```

OUTPUT:



```
Sorted array: [15, 37, 42, 68, 96, 101]
```

TASK 2:

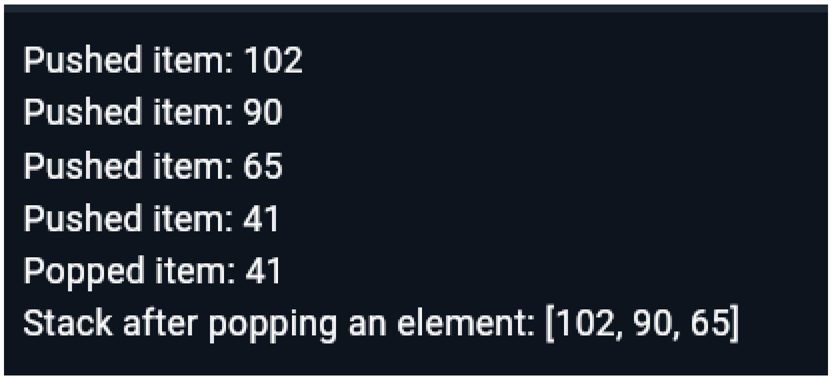
Implement a Stack from Scratch.

CODE:

```
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("102");
 stack.push("90");
 stack.push("65");
 stack.push("41");
 print("Popped item: ${stack.pop()}");
 print("Stack after popping an element: ${stack._stack}");
}
```

OUTPUT:



```
Pushed item: 102
Pushed item: 90
Pushed item: 65
Pushed item: 41
Popped item: 41
Stack after popping an element: [102, 90, 65]
```