

National University of Computer and Emerging Sciences



Laboratory Manual

for

Data Structures Lab

Course Instructor	Miss Arooj Khalil
Lab Instructor(s)	Mr. Dilawar Shabbir Mr. Sohaib Ahmad
Section	SE-3A
Semester	Fall 2022

Department of Computer Science

FAST-NU, Lahore, Pakistan

Objectives:

In this lab, students will practice:

1. Recursion.
2. Recursion and LinkedList.
3. Recursion and Stack.

Recursion**Task No 1:**

Implement a **recursive** global function `stringCompare` which compares two character strings recursively and:

- a). returns 0 if the two strings are equal.
- b). If the character of the first string at the index, where the first mismatch occurred, is greater in ASCII value; then it returns 1
- c). else it returns -1.

int stringCompare (char const* string1, char const* string2);

Task No 2:

Implement a **recursive** global function to find the Fibonacci number.

int Fibonacci(int n);

Recursion and LinkedList**Task No 3:**

Use your linked list implementation for the following recursive tasks. Use only **recursion** to implement these operations.

1. Implement a recursive member function `recursivePrint` which prints the singly linked in reverse order.

void recursivePrint() const;

2. Implement a recursive member function "length" which recursively finds the length of the linked list.

int length() const;

3. Implement a recursive member function "isSorted" which recursively checks whether the linked list is sorted (ascendingly).

bool isSorted() const;

4. Implement a function `deleteAll` which recursively deletes all nodes of the linked list.

void deleteAll();

5. Create a main function with the following instructions:

- a. Compare "ab" and "abC". Print the result.
- b. Compare "abc" and "ab". Print the result.
- c. Compare "abc" and "abc". Print the result.
- d. Find Fibonacci for n=10
- e. Insert at head of your singly linked list: 10, 9, 7, 5.
- f. Call `recursivePrint` function.
- g. Print the output of `isSorted`.
- h. Print the length of linked list.
- i. Call `deleteAll` function.
- j. Print the length of linked list.

Recursion and Stack

Task No 4:

Use your stack implemented in the previous session and write a program to reverse a stack using recursion, without using any loop.

```
void reverse(stack<int>& st);
```

Input: elements present in stack from top to bottom 1 2 3 4

Output: 4 3 2 1

Input: elements present in stack from top to bottom 1 2 3

Output: 3 2 1

Hint:

Hold all values in Function Call Stack until the stack becomes empty. When the stack becomes empty, insert all held items one by one at the bottom of the stack.

Task No 5:

Now, sort your **stack** using recursion.

Input: elements present in stack from top to bottom -3 14 18 -5 30

Output: 30 18 14 -3 -5

Hint:

Hold all values in Function Call Stack until the stack becomes empty. When the stack becomes empty, insert all held items one by one in sorted order. and then print the stack