

## **Subject: Data Structure and Algorithm ( CT 552)**

Total Marks: 50    Labsheet and Ouptut: 40(10\*4=40)

Attendance: 10(10\*1=10)

INITIAL:

- (i) Theory
- (ii) Code of implementation is required

FINAL:

- (i) Output

### **Lab 1: Understanding the Stack**

Theory:

- a. Explain Data structure, basic operations of data structures, ADT and it's necessity.
- b. Explain the algorithm for push and pop operation in stack.

Implementation:

- a. Implement the operation of Stack using C or C++.

### **Lab2: Application of stack**

Theory:

- a. Explain the algorithm for infix to postfix conversion with its evaluation and show example.
- b. Explain the algorithm for infix to ~~postfix~~ **Prefix** conversion with its evaluation and show example.

Implementation:

- a. Reverse the number and string using concept of stack. (example: 1234=> 4321 and apple=>elppa) using C or C++. You have done same program using c without using stack. How can this be different from stack concept? Is it easy then how explain?

### **Lab3: Understanding the Queue**

Theory:

- a. Explain the algorithm for the operation of queue.
- b. What are the difference between stack and queue?

Implementation

- a. Implement the operation of Queue using C or C++.

### **Lab4: Application of Queue**

Theory:

- a. Are there any drawbacks of Queue? If so explain.
- b. Can you always insert elements in queue?

Implementation:

- a. Find the sum of the first n number of queue. Ask n number to the user and specify the size (MAX) of queue as well. What happen if  $n > \text{MAX}$ ? And if problem arises how you solve it?

### **Lab5: Understanding the Circular Queue**

Theory:

- a. Explain the algorithm of Circular Queue.
- b. Why circular Queue is necessary?

Implementation

- a. Implement the circular queue using C or C++.

## **Lab6: Understanding the Linked list**

Theory:

- a. Explain static and dynamic implementation of list with suitable example.
- b. Why linked list come into existence?
- c. Explain the algorithm for insertion and deletion of a node after a given node in singly linked list with suitable example.

Implementation

- a. Implement the insertion of new node at the beginning of singly Linked using C or C++.

## **Lab7: Understanding the Linked list**

Theory

- a. Explain the algorithm for insertion and deletion of a node after a given node in doubly linked list with suitable example.
- b. What are the merits and demerits of linked list and static list?

Implementation

- a. Implement the insertion of new node at the beginning of doubly Linked using C or C++.

## **Lab8: Understanding the Recursion with Application**

Theory:

- a. Explain algorithm of Fibonacci with its tree diagram for f(5)
- b. Explain algorithm of TOH with its tree diagram for 3 disks

Implementation

- a. Implement TOH using Recursion.

## **Lab9: Understanding Tree and sorting**

Theory:

- a. Explain about the insertion and deletion in binary trees.
- b. Mention some applications of Binary Trees.
- c. Explain about the importance of merge sort with its algorithm and example.

Implementation:

- a. Implement any one tree (AVL or B Tree or Red Black Tree).
- b. Implement any one sorting(merge sort, insertion, selection, bubble)

## **Lab10: understanding Searching and Graph**

Theory:

- a. Explain about the binary search.
- b. Explain about hasing with the collision resolution techniques.
- c. Consider a hash table of size = 10. Using double hashing, insert the keys 72,27, 36, 24, 63, 81, 92, and 101 into the table. Take  $h1 = (k \bmod 10)$  and  $h2 = (k \bmod 8)$ .

Implement:

- a. Implement Binary Search using C or C++.
- b. Implement either DFS or BFS or Dijkstra's algorithm using C or C++.

**Compiled By: Er. Bishwas Pokharel(Lecturer)**