B. P. MANDAL COLLEGE OF ENGINEERING
MADHEPURA (BIHAR)

Department of Computer Science and Engineering

Laboratory manual for Course “Data Structure & Algorithms”
for 2nd Year B. Tech. Students

LIST OF EXPERIMENTS

Course Code : PCC CS 301

Course Title : Data Structure & Algorithms

HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS :

SOFTWARE REQUIREMENTS :

Dev C++ / code::blocks / VScode
GNU Compiler
GNU Debugger

DOs and DON‛Ts in Laboratory

Pre-Lab Session Instructions

1. Students should carry lab manual book to every lab session.
2. Be in time and adhere to the institution’s rules and maintain the decorum.
3. Must sign in the log register provided.
4. Make sure to occupy the allocated system and answer the attendance.

In- Lab Session Instructions

1. Follow the instructions on the allotted exercises.
2. Show the program and results to the instructors on completion of experiments.
3. Copy the program and results in the Lab record.
4. Prescribed textbooks and class notes can be kept ready for reference if required.

THE STUDENTS SHOULD NOT

1. Bring mobile phones or any other electronic gadgets to the lab.
2. Go out of the lab without permission.
3. Bring their bags inside the lab.
4. Leave their systems powered on after the class.
5. Do not handle computer for security without reading the instructions/Instruction manuals.
6. Refer Help for security options.
7. Go through Internet options whenever necessary.
8. Strictly observe the instructions given by the teacher/Lab Instructor.

General instructions for the exercises in the lab :

- Academic honesty is required in all your work. You must solve all programming assignments entirely on your own, except where group work is explicitly authorized. This means you must not take, neither show, give or otherwise allow others to take your program code, problem solutions, or other work.
- The programs should meet the following criteria:
  - Programs should be interactive with appropriate prompt messages, error messages if any, and descriptive messages for outputs.
  - Programs should perform input validation (Data type, range error, etc.) and give appropriate error messages and suggest corrective actions.
  - Comments should be used to give the statement of the problem and every function should indicate the purpose of the function, inputs and outputs.
  - Statements within the program should be properly indented.
  - Use meaningful names for variables and functions.
  - Make use of constants and type definitions wherever needed.
- Questions for lab tests and examinations are not necessarily limited to the questions in the manual, but may involve some variations and/or combinations of the questions.

Objectives of this laboratory:

1. To impart the basic concepts of data structures and algorithms.
2. To understand concepts about searching and sorting techniques.
3. To understand basic concepts about stacks, queues, lists, trees and graphs.
4. To enable learners to write algorithms for solving problems with the help of fundamental data structures.

Outcome:

1. For a given algorithm, student will able to analyze the algorithms to determine the time and computation complexity and justify the correctness.
2. For a given Search problem (Linear Search and Binary Search), student will able to implement it.
3. For a given problem of Stacks, Queues and linked list, student will able to implement it and analyze the same to determine the time and computation complexity.
4. Student will able to write an algorithm Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort and compare their performance in term of Space and Time complexity.
5. Student will able to implement Graph search and traversal algorithms and determine the time and computation complexity.

Lab 1. Date:

Review of C programming basic concepts (Arrays and Functions)

Problem Statements:

AIM 1:

Find the sum of given n numbers using a 1D array. Use a function to find the sum of elements in the array and main program to display the sum.

AIM 2:

Sort given list of n integers into ascending order using selection sort. Use a function to sort.

AIM 3:

Implement Linear Search on a list of integers.

AIM 4:

Implement Binary search on a list of integers.

AIM 5:

Find the sum of 2 matrices using a function named add with suitable parameters.

AIM 6:

Find the biggest in a list of numbers using a function.

AIM 8:

Write a program in C to

1. Demonstrate passing pointers to a function.
2. Demonstrate Returning pointer from a function.

Lab 2. Date.

Review of C programming basic concepts (Recursion)

Problem Statements:

1. Find the factorial of a given number.
2. Find GCD of 2 numbers.
3. Display Fibonacci series upto n^{th} term.
4. Simulate the working of Tower of Hanoi for n disks. Print the number of moves.
5. Convert a given decimal number to binary.
6. Find product of two natural numbers.

Lab 3. Date.

Stacks I

Problem Statements:

1. Implement Stack using an array. The elements of the stack are characters. Include functions push ,pop and display. Also include functions for checking error conditions such as underflow and overflow.
2. Determine if a given string is palindrome using stack.
3. Convert a given decimal number to binary.
4. Check if the given parenthesized expression has properly matching open and closing parenthesis.

Lab 4. Date.

Stacks II

Problem Statements:

1. Evaluate the given postfix expression.
2. Evaluate the given prefix expression.
3. Convert a fully parenthesized expression from infix to postfix.
4. Convert a fully parenthesized infix expression to prefix.
5. Implement two stacks in an array.
6. To convert a prefix expression to postfix.

Lab 5. Date:

Queues II

Problem Statements:

1. Implement a queue of characters. Include functions insertq,deleteq and displayq.
2. Implement a circular queue of Strings. Include functions insert,delete and display.
3. Implement two circular queues of integers in a single array.
4. Implement an ascending priority queue.

Lab 6. Date:

Linked List I

Problem Statements:

1. Implement stack using singly linked list.
2. Implement queues using singly linked lists with header nodes.
3. Two singly circular linked lists containing header nodes contain char data which are already sorted. Create a new list so that the final list is sorted after merging them.
4. Implement Union using singly circular linked list.
5. Implement Intersection using doubly linked list.

Lab 7. Date:

Linked List II

Problem Statements:

1. Given two polynomials, write a program to perform the following operations on a singly circular linked list with header node. Use a menu driven approach to input two polynomials, add, subtract, multiply and display the result.
2. Add two long positive integers represented using a circular doubly linked list with header node.
3. Reverse a doubly linked list containing words in the data field.

Lab 8. Date:

Trees I

Problem Statements:

1. Create a simple binary tree using recursion.
2. Create a tree, traverse and search for an item.
3. Display inorder, preorder and postorder traversal on the binary tree created using iteration and recursion
4. Check if two trees are equal.

Lab 9. Date:

Trees II

Problem Statement:

1. Write a program to insert, delete in a BST
2. Write program to find the depth, height, number of leaves, nodes in a BST
3. Implement level order traversal in a BST
4. Create a tree for postfix expression and evaluate it

Lab 10. Date:

Graphs

Problem Statement

1. Write a program to implement Depth First Search on a graph.
2. Write a program to implement Breadth First Search on a graph.
3. Write a program to sort numbers using merge sort.
4. Write a program to sort numbers using insertion sort.