Academy of Technology
Data Structure and Algorithms Lab (PCC CS391)

1. Write a menu driven program to perform the following operations on an array.
   (a) ***update*** an element ***x*** at position ***k*** in the array.
   (b) ***insert*** an element ***x*** at position ***k*** in the array.
   (c) ***search*** an element ***x*** from the array.
   (d) ***remove*** an element ***x*** from the array.
   (e) ***display*** an element ***x*** from the array.

2. Write a menu driven program in C/C++ to implement a **Stack** using array and perform the following operations.
   (a) ***isFull()*** function to check whether the ***Stack*** is full or not.
   (b) ***isEmpty()*** function to check whether the ***Stack*** is empty or not.
   (c) ***peek()*** function to read the stack top element without deleting it.
   (d) ***push(item)*** function to insert an element ***item*** in the ***Stack***.
   (e) ***pop()*** function to read and remove an element from the ***Stack***.
   (f) ***display()*** function to display the entire stack.

3. Write a menu driven program in C or C++ to implement a ***Queue*** using array and perform the following operations.
   (a) ***isFull()*** function to check whether the ***Queue*** is full or not.
   (b) ***isEmpty()*** function to check whether the ***Queue*** is empty or not.
   (c) ***insert(item)*** function to insert an element ***item*** in the ***Queue***.
   (d) ***delete()*** function to read and remove an element from the ***Queue***.
   (e) ***display()*** function to display the entire ***Queue***.

4. Write a menu driven program in C or C++ to implement a ***Circular Queue*** using array and perform the following operations.
   (a) ***isFull()*** function to check whether the ***Circular Queue*** is full or not.
   (b) ***isEmpty()*** function to check whether the ***Circular Queue*** is empty or not.
   (c) ***insert(item)*** function to insert an element ***item*** in the ***Circular Queue***.
   (d) ***delete()*** function to read and remove an element from the ***Circular Queue***.
   (e) ***display()*** function to display the entire ***Circular Queue***.

5. Assume that there are only four operators ($*$, $/$, $+$, $-$,) in an ***infix*** expression and operand is single digit only. Write a C/C++ program to Convert Infix to Postfix Expression using Stack.

6. Assume that there are only four operators ($*$, $/$, $+$, $-$,) in a ***postfix*** expression and operand is single digit only. Write a C/C++ program to evaluate a given postfix expression.

7. Write a menu driven program in C or C++ to perform the following operations on **single linked list**.
   (a) ***insert*** a node at the beginning of the list.

(b) ***insert*** a node at the end of the list.

(c) ***deletet*** a node from the beginning of the list.

(d) ***delete*** a node at the end of the list.

(e) ***display*** the whole list.

8. Write a menu driven program in C or C++ to perform the following operations on **single linked list**.

   (a) ***insert*** a node at the beginning of the list.

   (b) ***delete*** a node at the end of the list.

   (c) ***display*** the whole list.

   (d) ***search*** an element ***x*** in the list.

   (e) ***reverse*** the list.

9. Write a menu driven program in C or C++ to implement a ***stack*** using **single linked list** and perform the following operations.

   (a) ***isEmpty()*** is to check whether the stack is empty or not.

   (b) ***push()*** is to insert an item in the stack.

   (c) ***pop()*** is to delete an item from the stack.

   (d) ***display()*** is to show the entire stack.

10. Write a menu driven program in C or C++ to implement a ***queue*** using **single linked list** and perform the following operations.

    (a) ***isEmpty()*** is to check whether the queue is empty or not.

    (b) ***insert()*** is to insert an item in the queue.

    (c) ***del()*** is to delete an item from the queue.

    (d) ***display()*** is to show the entire queue.

11. Write a menu driven program in C or C++ to implement a ***priority queue*** using **single linked list** and perform the following operations.

    (a) ***isEmpty()*** is to check whether the queue is empty or not.

    (b) ***insert()*** is to insert an item in the queue.

    (c) ***del()*** is to delete an item from the queue.

    (d) ***display()*** is to show the entire queue.

12. Write a menu driven program in C or C++ to perform the following operations on **circular linked list**.

    (a) ***insert*** a node at the beginning of the list.

    (b) ***delete*** a node at the end of the list.

    (c) ***display*** the whole list.

13. Write a menu driven program in C or C++ to perform the following operations on **circular linked list**.

    (a) ***insert*** a node at the end of the list.

    (b) ***deletet*** a node from the beginning of the list.

    (c) ***display*** the whole list.

14. Write a program in C or C++ to add two polynomials using **linked list**.

15. (a) Write a program in C or C++ to sort a given array using **selection sort** algorithm. Show the number of comparison required for a given input.

    (b) Write a program to find the n^{th} Fibonacci number using tail recursion.

16. (a) Write a program in C or C++ to sort a given array using **bubble sort** algorithm. Show the number of comparison required for a given input.

    (b) Write a program to find the factorial of a number using tail recursion.

17. (a) Write a program in C or C++ to sort a given array using **insertion sort** algorithm. Show the number of comparison required for a given input.

    (b) Write a program to implement **Tower of Hanoi** Problem.

18. Write a program in C or C++ to sort a given array using **Merge sort** algorithm.

19. Write a program in C or C++ to sort a given array using **Quick sort** algorithm.

20. Write a program in C or C++ to sort a given array using **Heap sort** algorithm.

21. Write a menu driven program in C or C++ to perform the following operations on **Binary Search Tree.**

    (a) ***insert*** a node.

    (b) ***inorder*** traversal.

    (c) ***preorder*** traversal.

    (d) ***search*** an given ***key***.

    (e) Find the smallest element.

22. Write a menu driven program in C or C++ to perform the following operations on **Binary Search Tree.**

    (a) ***insert*** a node.

    (b) ***inorder*** traversal.

    (c) ***postorder*** traversal.

    (d) Find the largest element.

    (e) Count the total number of nodes.

23. Write a menu driven program in C or C++ to perform the following operations on **Binary Search Tree.**

(a) ***insert*** a node.

(b) ***inorder*** traversal.

(c) Count the total number of external nodes.

(d) Count the total number of internal nodes.

(e) Determine the height of the tree.

24. Write a menu driven program in C or C++ to perform the following operations on **Binary Search Tree.**

    (a) Create a binary search tree

    (b) ***inorder*** traversal.

    (c) Delete items from the tree.