ACADEMY OF TECHNOLOGY

Lab Assignment (Practice)

Paper name: Data Structure and Algorithm

Code: PCC-CS391 **Semester: 3rd**

Discipline: CSE **Time: 2 Hours**

Date: January 4, 2022

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. Given two polynomials represented by two arrays, write a program that adds given two polynomials.
3. Write a menu driven program 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.
4. 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***.
5. 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***.

6. Write a program to find the factorial of a number using tail recursion.
7. Write a program to find the n^{th} Fibonacci number using tail recursion.
8. Write a program to implement Tower of Hanoi Problem.
9. Write a program to sort a given array using selection sort algorithm.
10. Write a program to sort a given array using bubble sort algorithm.
11. Write a program to inset an item in a given sorted array.
12. Write a program to sort a given array using insertion sort algorithm.
13. Write a program to count the frequency of each distinct character in a given string.
14. Write a program to sort a given array using ***count sort*** algorithm.
15. Write a program to sort a given array using ***radix sort*** algorithm.
16. Write a program in C or C++ to sort a given array using ***merge sort*** algorithm.
17. Write a program in C or C++ to sort a given array using ***quick sort*** algorithm.
18. Write a program in C or C++ to search an item in a given array using linear search algorithm.
19. Write a program in C or C++ to search an item in a given sorted array using
    (a) iterative binary search algorithm.
    (b) recursive binary search algorithm.
20. 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) ***display*** the whole list.
    (d) ***search*** an element x in the list.
21. 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.
22. 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.

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

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

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

24. Write a menu driven program in C or C++ to perform the following operations on double 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.

25. 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) ***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.

26. 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) ***postorder*** traversal.

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