



ACADEMY OF TECHNOLOGY

Lab Assignment (Practice)

Paper name: Data Structure and Algorithm

Code: PCC-CS391

Discipline: CSE

Semester: 3rd

Time: 2 Hours

Date: January 4, 2022

- Write a menu driven program to perform the following operations on an array.
 - update* an element *x* at position *k* in the array.
 - insert* an element *x* at position *k* in the array.
 - search* an element *x* from the array.
 - remove* an element *x* from the array.
 - display* an element *x* from the array.
- Given two polynomials represented by two arrays, write a program that adds given two polynomials.
- Write a menu driven program to implement a stack using array and perform the following operations.
 - isFull()* function to check whether the stack is full or not.
 - isEmpty()* function to check whether the stack is empty or not.
 - peek()* function to read the stack top element without deleting it.
 - push(item)* function to insert an element *item* in the stack.
 - pop()* function to read and remove an element from the stack.
 - display()* function to display the entire stack.
- Write a menu driven program in C or C++ to implement a *Queue* using array and perform the following operations.
 - isFull()* function to check whether the *Queue* is full or not.
 - isEmpty()* function to check whether the *Queue* is empty or not.
 - insert(item)* function to insert an element *item* in the *Queue*.
 - delete()* function to read and remove an element from the *Queue*.
 - display()* function to display the entire *Queue*.
- Write a menu driven program in C or C++ to implement a *Circular Queue* using array and perform the following operations.
 - isFull()* function to check whether the *Circular Queue* is full or not.
 - isEmpty()* function to check whether the *Circular Queue* is empty or not.
 - insert(item)* function to insert an element *item* in the *Circular Queue*.
 - delete()* function to read and remove an element from the *Circular Queue*.
 - 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 insert 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) *delete* 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*.
-