

Paper Code: BCADSC 2.6

Paper Title: Data Structures Lab

Teaching Hours: 3 Hrs / Week

Marks: Th-40+IA-10

Credits: 1

Section A:

1. Write a C program to demonstrate the Dynamic Memory Allocation for Structure by reading and printing 'n' employee details.
2. Write a C program to read one dimensional array, and print sum of all elements along with inputted array elements using Dynamic Memory Allocation.
3. Write a program that takes a file as an argument and counts the total number of lines. Lines are defined as ending with a newline character. Program usage should be count filename.txt and the output should be the line count.
4. Write a C program to find n Fibonacci numbers using recursion.
5. Write a C program to search for an element in an array using Sequential search
6. Write a C program to sort a list of N elements using Bubble sort Technique
7. Write a C program to sort a list of N elements using Insertion sort Technique
8. Write a C program to demonstrate the working of stack of size N using an array. The elements of the stack may assume to be of type integer or real, the operations to be supported are 1. PUSH 2. POP 3. DISPLAY. The program should print appropriate messages for STACK overflow, Under flow and empty, use separate functions to implement the same.

Section B:

1. Write a C program to search for an element in an array using Binary search
2. Write a C program to sort a list of N elements using Selection Sort Technique
3. Write a C program to sort a list of N elements using Merge sort Technique
4. Write a C program to read a text file and convert the file contents in capital (upper-case) and write the contents in an output file.
5. Write a C program to convert a given infix expression into its postfix Equivalent, Implement the stack using an array
6. Write a C program to simulate the working of an ordinary Queue using an array. Provide the operations QINSERT, QDELETE and QDISPLAY. Check the Queue status for empty and full detect these cases
7. Using dynamic variables and pointers Write a C program to construct a singly linked list consisting of the following information in each node; Roll – No (Integer), Name (Character string). The operations to be supported are:
 - LINSERT Inserting a node in the front of the list
 - LDELETE Deleting the node based on Roll –No
 - LSEARCH Searching a node based on Roll-No
 - LDISPLAY displaying all the nodes in the list