| **DATA STRUCTURES LABORATORY** | | | |
|---|---|---|---|
| **(Effective from the academic year 2018 -2019)** | | | |
| **SEMESTER – III** | | | |
| **Course Code** | **18CSL38** | **CIE Marks** | 40 |
| **Number of Contact Hours/Week** | 0:2:2 | **SEE Marks** | 60 |
| **Total Number of Lab Contact Hours** | 36 | **Exam Hours** | 03 |
| **Credits – 2** | | | |

**Course Learning Objectives:** This course (18CSL38) will enable students to:

This laboratory course enable students to get practical experience in design, develop, implement, analyze and evaluation/testing of

- Asymptotic performance of algorithms.
- Linear data structures and their applications such as stacks, queues and lists
- Non-Linear data structures and their applications such as trees and graphs
- Sorting and searching algorithms

**Descriptions (if any):**

- Implement all the programs in 'C / C++' Programming Language and Linux / Windows as OS.

**Programs List:**

| 1. | Design, Develop and Implement a menu driven Program in C for the following array operations. <br>     a. Creating an array of N Integer Elements <br>     b. Display of array Elements with Suitable Headings <br>     c. Inserting an Element (ELEM) at a given valid Position (POS) <br>     d. Deleting an Element at a given valid Position (POS) <br>     e. Exit. <br> Support the program with functions for each of the above operations. |
|---|---|
| 2. | Design, Develop and Implement a Program in C for the following operations on Strings. <br>     a. Read a main String (STR), a Pattern String (PAT) and a Replace String (REP) <br>     b. Perform Pattern Matching Operation: Find and Replace all occurrences of PAT in STR with REP if PAT exists in STR. Report suitable messages in case PAT does not exist in STR <br> Support the program with functions for each of the above operations. Don't use Built-in functions. |
| 3. | Design, Develop and Implement a menu driven Program in C for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX) <br>     a. Push an Element on to Stack <br>     b. Pop an Element from Stack <br>     c. Demonstrate how Stack can be used to check Palindrome <br>     d. Demonstrate Overflow and Underflow situations on Stack <br>     e. Display the status of Stack <br>     f. Exit <br> Support the program with appropriate functions for each of the above operations |
| 4. | Design, Develop and Implement a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, *, /, % (Remainder), ^ (Power) and alphanumeric operands. |
| 5. | Design, Develop and Implement a Program in C for the following Stack Applications <br>     a. Evaluation of Suffix expression with single digit operands and operators: +, -, *, /, %, ^ <br>     b. Solving Tower of Hanoi problem with n disks |

| | |
|---|---|
| 6. | Design, Develop and Implement a menu driven Program in C for the following operations on Circular QUEUE of Characters (Array Implementation of Queue with maximum size MAX)<br>    a.  Insert an Element on to Circular QUEUE<br>    b.  Delete an Element from Circular QUEUE<br>    c.  Demonstrate Overflow and Underflow situations on Circular QUEUE<br>    d.  Display the status of Circular QUEUE<br>    e.  Exit<br>Support the program with appropriate functions for each of the above operations |
| 7. | Design, Develop and Implement a menu driven Program in C for the following operations on Singly Linked List (SLL) of Student Data with the fields: *USN, Name, Programme, Sem, PhNo*<br>    a.  Create a SLL of N Students Data by using *front insertion*.<br>    b.  Display the status of SLL and count the number of nodes in it<br>    c.  Perform Insertion / Deletion at End of SLL<br>    d.  Perform Insertion / Deletion at Front of SLL(Demonstration of stack)<br>    e.  Exit |
| 8. | Design, Develop and Implement a menu driven Program in C for the following operations on Doubly Linked List (DLL) of Employee Data with the fields: *SSN, Name, Dept, Designation, Sal, PhNo*<br>    a.  Create a DLL of N Employees Data by using *end insertion*.<br>    b.  Display the status of DLL and count the number of nodes in it<br>    c.  Perform Insertion and Deletion at End of DLL<br>    d.  Perform Insertion and Deletion at Front of DLL<br>    e.  Demonstrate how this DLL can be used as Double Ended Queue.<br>    f.  Exit |
| 9. | Design, Develop and Implement a Program in C for the following operationson Singly Circular Linked List (SCLL) with header nodes<br>    a.  Represent and Evaluate a Polynomial $P(x,y,z) = 6x^2y^2z-4yz^5+3x^3yz+2xy^5z-2xyz^3$<br>    b.  Find the sum of two polynomials POLY1(x,y,z) and POLY2(x,y,z) and store the result in POLYSUM(x,y,z)<br>Support the program with appropriate functions for each of the above operations |
| 10. | Design, Develop and Implement a menu driven Program in C for the following operations on Binary Search Tree (BST) of Integers .<br>    a.  Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2<br>    b.  Traverse the BST in Inorder, Preorder and Post Order<br>    c.  Search the BST for a given element (KEY) and report the appropriate message<br>    d.  Exit |
| 11. | Design, Develop and Implement a Program in C for the following operations on Graph(G) of Cities<br>    a.  Create a Graph of N cities using Adjacency Matrix.<br>    b.  Print all the nodes reachable from a given starting node in a digraph using DFS/BFS method |
| 12. | Given a File of N employee records with a set K of Keys (4-digit) which uniquely determine the records in file F. Assume that file F is maintained in memory by a Hash Table (HT) of m memory locations with L as the set of memory addresses (2-digit) of locations in HT. Let the keys in K and addresses in L are Integers. Design and develop a Program in C that uses Hash function H: K →L as H(K)=K mod m (remainder method), and implement hashing technique to map a given key K to the address space L. Resolve the collision (if any) using linear probing. |
| **Laboratory Outcomes**: The student should be able to: | |

| |
|---|
| • Analyze and Compare various linear and non-linear data structures |
| • Code, debug and demonstrate the working nature of different types of data structures and their applications |
| • Implement, analyze and evaluate the searching and sorting algorithms |
| • Choose the appropriate data structure for solving real world problems |

**Conduct of Practical Examination:**

- Experiment distribution
  - o For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.
  - o For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.
- Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.
- Marks Distribution *(Courseed to change in accoradance with university regulations)*
  - c) For laboratories having only one part – Procedure + Execution + Viva-Voce: 15+70+15 = 100 Marks
  - d) For laboratories having PART A and PART B
    - i. Part A – Procedure + Execution + Viva = 6 + 28 + 6 = 40 Marks
    - ii. Part B – Procedure + Execution + Viva = 9 + 42 + 9 = 60 Marks