GUJARAT TECHNOLOGICALUNIVERSITY
MASTERS IN COMPUTERAPPLICATIONS
Semester–II

Subject Name: Data Structures (DS)
Subject Code: 3620002

Practical List

C programming language to performs followings :

1. Create a Structure with following Data Members:
    1. Integer Array
    2. Size of the Array
    Sort the Array using various Sorting algorithms such as (i) Selection Sort (ii) Bubble Sort (iii) Two-way Merge Sort (iv) Insertion Sort (v) Quick Sort (vi) Radix Sort (vii) Heap Sort and (viii) Shell Sort. And store the sorted Array in a text file.

2. Create a Structure with following Data Members:
    1. Integer Array
    2. Size of the Array
    Search an element in Array using Linear (Sequential) Search and Binary Search. And Display result in file. For Sequential Search, assume (a) Unordered Array, and (b) Ordered Array and develop programs accordingly.

3. Create a structure with following Data members:
    1. Integer Array
    2. Size of the Array
    Perform the following operations on stack using user-defined functions:
    1. Push
    2. Pop
    3. Isempty
    4. Isfull
    5. Peep
    Create a file which stores all values of Array through Stack. Has it reversed the order of the elements of the Array? Why?

4. Create a user-defined structure with the following data members:
    1. A Data
    2. A link to the next node
    Perform the following operations on stack using user-defined functions:
    1. Push
    2. Pop
    3. Isempty

4. Isfull
5. Peep
Create a file which stores all values of list.

5. Write a program to convert an infix arithmetic expression (parenthesize / unparenthesized) into postfix notation.

6. Write a program to evaluate a postfix expression.

7. Create a structure with following Data members:
   1. Integer Array
   2. Size of the Array
   Search an element in a given list using Binary search by recursion. And Display result in file.

8. Create a structure with following Data members:
   1. Integer Array
   2. Size of the Array
   Perform the following operations on Simple queue using user-defined functions:
   1. Insert an element
   2. Remove an element
   3. Display
   4. Isfull
   5. Isempty
   Create a file which stores all values of Array.

9. Create a user-defined structure with the following data members:
   1) A Data
   2) A link to the next node
   Perform the following operations on Simple queue using user-defined functions:
   1. Insert an element
   2. Remove an element
   3. Display
   4. Isfull
   5. Isempty
   Create a file which stores all values of list.

10. Create a structure with following Data members:
    1. Integer Array
    2. Size of the Array
    Perform the following operations on Circular queue using user-defined functions:
    1. Insert an element
    2. Remove an element
    3. Display
    4. Isfull
    5. Isempty
    Create a file which stores all values of Array.

11. Create a user-defined structure with the following data members:
    1. A Data

2. A link to the next node
Perform the following operations on Circular queue using user-defined functions:
1. Insert an element
2. Remove an element
3. Display
4. Isfull
5. Isempty
Create a file which stores all values of list.

12. Create a user-defined structure with the following data members:
1. A Co-efficient
2. A Exponent
3. A link to the next node
Perform the following operations on Singly list using user-defined functions:
1. Create
2. Display
3. Addition
4. Multiplication
Create a file which stores all values of list.

13. Create a user-defined structure with the following data members:
1. A Data
2. A link to the next node

Perform the following operations on list using user-defined functions:
1. Create a list
2. Traverse the whole list\
3. Delete first node
4. Delete last node
5. Delete a node before specified data
6. Insert at first position
7. Insert at last position
8. Insert a node before specified data
9. Insert a node at specified position
10. Count
11. Copy
12. Merge two list
13. Reverse
14. Search
15. Sort
Create a file which stores all values of list.

14. Create a user-defined structure with the following data members:
1. A Data
2. A link to the next node

Perform the following operations on Circular list using user-defined functions:
1. Create a list

2. Traverse the whole list\
3. Delete first node
4. Delete last node
5. Delete a node before specified data
6. Insert at first position
7. Insert at last position
8. Insert a node before specified data
9. Insert a node at specified position
10. Count
11. Copy
12. Merge two list
13. Reverse
14. Search
15. Sort
Create a file which stores all values of list.

15. Create a user-defined structure with the following data members:
    1. A Data
    2. A link to the next node
    3. A link to the previous node

    Perform the following operations on the doubly-linked list using user-defined functions:
    1. Create a list
    2. Traverse the whole list\
    3. Delete first node
    4. Delete last node
    5. Delete a node before specified data
    6. Insert at first position
    7. Insert at last position
    8. Insert a node before specified data
    9. Insert a node at specified position
    10. Count
    11. Copy
    12. Merge two list
    13. Reverse
    14. Search
    15. Sort
    Create a file which stores all values of list.

16. Create a user-defined structure with the following data members:
    1. A Data
    2. A link to the next node
    3. A link to the previous node

    Perform the following operations on doubly-linked Circular list using user defined functions:
    1. Create a list
    2. Traverse the whole list\
    3. Delete first node

4. Delete last node
5. Delete a node before specified data
6. Insert at first position
7. Insert at last position
8. Insert a node before specified data
9. Insert a node at specified position
10. Count
11. Copy
12. Merge two list
13. Reverse
14. Search
15. Sort
Create a file which stores all values of list.

17.
Write a program to represent an undirected graph using the adjacency matrix to implement the graph and your program be menu driven allowing the user the following options:
1. Create graph
2. Insert an edge
3. Print Adjacency Matrix
4. List all vertices that are adjacent to a specified vertex.
5. Print out vertices using depth first search
6. Print out vertices using breadth first search
7. Exit program

18. Create a user-defined structure with the following data members:
1. A Data
2. A link to the Left child
3. A link to the Right child

Perform the following operations on Binary Search Tree using recursion:
1. Create
2. Traverse (Inorder, Preorder, Postorder)
3. Insert
4. Delete
5. Search
Create a file which stores all values of traversal.

*****