Unit - 1

1) Assume a 4 byte storage is required to hold each element of array a[12][15]. Assume that the storage location for the array begins from 1000. Show the actual address of the element a[6][5] for both row major order and column major order.

2) Consider a sparse matrix is given to function. Write a function to find transpose of sparse matrix and display transpose of matrix
3) A company maintains an array of employees which stores employee name, employee id and salary. The array is always kept in **sorted order according to employee ID**. Design a C function which will search and display the details of a particular employee when the employee ID is specified.

Unit - 2

1) Consider the following dequeue where DEQUE is allocated 6 memory cells. FRONT=2 and REAR=5. QUEUE contains elements as LONDON, BERLIN, ROME, PARIS. Describe the queue, including FRONT & REAR as following operation take place

   a) Athen is added from FRONT 2) Two Cities are deleted from REAR 3)Madrid is added from Front. 4) Moscow is added from Rear 5) A city is deleted from left.6) Oslo is added from front.
2) Write a program to implement two stacks in an array. Stack 1 grows from left to right. Stack 2 grows from right to left. Write appropriate push, pop and display functions for both stacks.
3) Write an algorithm/program to reverse a stack using queue. Write appropriate functions for stack and queue.
4) An input text string comprises of the following a) Letters, b) Digits c) Special characters.

5) Write a function that accepts a text string from user and stores individual characters in three different stacks, ie L(for letters), D(for digits) and SC(for special characters). The function should return control as soon as '~' is encountered. Use the concept of multiple stack implementation in memory to implement three stacks
6) Write the prefix and postfix form for: A+B*(C-D)/(E-F) .
7) Write an algorithm to convert Infix expression into postfix expression.

8) Consider a Stack ADT is already created. Write a function to check whether the parentheses are balance or not.

   Balanced Parentheses : ( ( ) ( ) ( ( ) ) )

   Unbalanced Parentheses: ( ( ) ( ) ( ( ) )

9) Given a string s representing an expression containing various types of brackets: {}, (), and [], the task is to determine whether the brackets in the expression are balanced or not.
10) Write a program to reverse a stack using recursion, without using any loop.

10) Write a program to reverse a stack using recursion, without using any loop.
11) Given a stack with **push()**, **pop()**, and **empty()** operations, The task is to delete the **middle** element of it without using any additional data structure.
12) Implement a stack using queues. The stack should support the following operations:
    1. **Push(x)**: Push an element onto the stack.
    2. **Pop()**: Pop the element from the top of the stack and return it.

Jnit - 3

1) Consider a linked list "L1"which stores numbers. Write a function to create a new linked "L2" which stores only the odd numbers present in L1. Display linked list L2.
2) Write a function to merge two sorted linked list. The merged list should be sorted.
3) Consider a double linked list. Write a function to reverse a double linked list.
4) Consider a linked list stores numbers. Write a function to re-arrange the list in such a manner so that it contains minimum number than maximum number followed by second minimum and the second maximum.

   Example:- Original LL:- 1->3-> 5-> 7-> 2-> 9

   Rearranged LL : - 1-> 9-> 2-> 7-> 3-> 5

5) Write a program to reverse a circular linked list.

6) Write a function to show whether the number stored in the Single Linked List(1 digit in each node) is a palindrome number or not.
7) Given a linked list and a **key**, the task is to check if **key** is present in the linked list or not.

   Linked List : 5->6->90->80->18->65

   Key1 : 88 and Key2 : 18

8) Given a linked list sorted in **non-decreasing order.** Return the list by **deleting the duplicate nodes** from the list. The returned list should also be in **non-decreasing order.**

   Linked List :- 12->13->13->17->18->18->56->89->89

Unit - 4

1) Write a non-recursive function to print the preorder traversal on a binary tree .
2) Write a function to find height of a binary tree.
3) Create an AVL Tree from the nodes given below.

   56, 78, 80, 45, 48, 90, 85, 100, 95.

   Delete nodes 100 and 78

4) Write a function to find in-degree and out-degree of node, if the adjacency matrix is given.
5) Consider the following sequence list of a binary tree in inorder and preorder

   Preorder:-G,B,Q,A,C,K,F,P,D,E,R,H

   Inorder:- Q,B,K,C,F,A,G,P,E,D,H,R

   Draw the binary tree.

6) Draw B Tree of order 3 with the sequence of node values given below

   1, 5 19, 24, 7, 4, 6, 20, 22, 78, 8, 11,20, 92,85,36,44

7) Write a function to find whether the tree is balanced or not.
8) What is height balanced tree? Explain what do you mean by balance factor. Construct a height balanced binary search tree from the following sequence of integers 40,30,10,5,50,60,80,70 .

9) Draw the binary search tree resulting from the insertion of keys: 50,72,96,966,1074,26,12,11,92,10,25,51,21.

   i) Traverse the tree in postorder and preorder

   ii) Convert the tree to an inorder threaded binary tree.

10) Write a function to find source and sink nodes in a graph. The graph is represented as adjacency matrix.
11) Construct a B+ tree of order 5. The numbers are inserted in the sequence given below:

    80, 40, 15, 25, 30, 90, 35, 50, 60, 70

12) Given a binary tree, find its minimum depth. The minimum depth is the number of nodes along the shortest path from the root node down to the nearest leaf node.

    10,20,30,40,50,60

Unit - 5

1) Consider a hash table with size=11.

   Using double hashing insert keys 27, 72, 63, 42, 36, 18, 29, 101. Consider $h_1$=kmod10 and $h_2$=kmod8. Use suitable collision resolution technique.

2) If the following sequence of numbers is to be sorted using quick sort, then show iterations of sorting process

   42,34,75,23,21,18,90,67,78

3) Consider the following 4 digit employee numbers

   9614, 5882, 6713, 4409, 1825

   Find 2-digit hash address of each number using

   a) Division method m=97 b)mid square method c) folding method

4) A certain sorting technique was applied to the following data set,
   81,72,63,45,27,36
   After 2 passes, the rearrangement of dataset is as given below,
   27, 36, 80, 72, 63, 54, 90
   Identify the sorting algorithm applied. Justify.
5) How is pivot selected in quick sort. What is the best method to select pivot.
6) Sort the following numbers with the help of heap sort.

   30, 60, 51, 40, 15, 95,89,36

   Explain the worst case complexity of heap sort.

7) Consider a hash table of size 11. It uses linear probing with following hash function f(x)=(5x+4)%11. Show the resultant hash table if we insert the following key in sequence:

   3, 9, 1, 2, 14, 25, 12

   What is the drawback of using this method, how can it be resolved