UCS406 Data Structures and Algorithms

Lab Assignment-I

1) Develop a Menu driven program to demonstrate the following operations of Arrays

——MENU———-
1.CREATE
2.DISPLAY
3.INSERT
4.DELETE
5.SEARCH
6.EXIT

2) Design the logic to remove the duplicate elements from an Array and after the deletion the array should contain the unique elements.

3) Predict the Output of the following program

```
int main()
{
  int i;
  int arr[5] = {1};
  for (i = 0; i < 5; i++)
    printf("%d ", arr[i]);
  return 0;
}
```

4) Implement the logic to

i Reverse the elements of an array
ii Find the matrix multiplication
iii Find the Transpose of a Matrix

5) Implement the Binary search algorithm regarded as a fast search algorithm with run-time complexity of O(log n) in comparison to the Linear Search.

6) Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order. Code the Bubble sort with the following elements:

64	34	25	12	22	11	90

7) Design the Logic to Find a Missing Number in a Sorted Array.

Assignment 1_A

1. Space required to store any two-dimensional array is $number\ of\ rows \times number\ of\ columns$. Assuming array is used to store elements of the following matrices, implement an efficient way that reduces the space requirement.
    (a) Diagonal Matrix.
    (b) Tri-diagonal Matrix.
    (c) Lower triangular Matrix.
    (d) Upper triangular Matrix.
    (e) Symmetric Matrix
2. Write a program to implement the following operations on a Sparse Matrix, assuming the matrix is represented using a triplet.
    (a) Transpose of a matrix.
    (b) Addition of two matrices.
    (c) Multiplication of two matrices.
3. Write a program to find sum of every row and every column in a two-dimensional array.
4. Write a program to find a saddle point in a two-dimensional array. A saddle point in a numerical array is a number that is larger than or equal to every number in its column, and smaller than or equal to every number in its row.
5. https://www.interviewbit.com/problems/spiral-order-matrix-i/
6. https://www.interviewbit.com/problems/spiral-order-matrix-ii/

Lab Assignment 2
Single Linked List

1. Develop a menu driven program for the following operations of on a Singly Linked List.
   (a) Insertion at the beginning.
   (b) Insertion at the end.
   (c) Insertion in between (before or after a node having a specific value, say 'Insert a new Node 35 before/after the Node 30').
   (d) Deletion from the beginning.
   (e) Deletion from the end.
   (f) Deletion of a specific node, say 'Delete Node 60').
   (g) Search for a node and display its position from head.
   (h) Display all the node values.
2. Write a program to count the number of occurrences of a given key in a singly linked list and then delete all the occurrences. For example, if given linked list is 1->2->1->2->1->3->1 and given key is 1, then output should be 4. After deletion of all the occurrences of 1, the linked list is 2->2->3.
3. Write a program to find the middle of a linked list.
   https://www.geeksforgeeks.org/write-a-c-function-to-print-the-middle-of-the-linked-list/
4. Write a program to reverse a linked list.
   https://www.geeksforgeeks.org/reverse-a-linked-list/

Additional Questions:

- https://www.interviewbit.com/problems/reverse-link-list-ii/
- https://www.interviewbit.com/problems/rotate-list/
- https://www.geeksforgeeks.org/adding-two-polynomials-using-linked-list/
- https://www.geeksforgeeks.org/write-a-function-to-get-the-intersection-point-of-two-linked-lists/

Lab Assignment 3
Doubly and Circular Linked List

1. Develop a menu driven program for the following operations of on a Circular as well as a Doubly Linked List.
   (a) Insertion anywhere in the linked list (As a first node, as a last node, and after/before a specific node).
   (b) Deletion of a specific node, say 'Delete Node 60'. That mean the node to be deleted may appear as a head node, last node or a node in between.
   (c) Search for a node.
2. Display all the node values in a circular linked list, repeating value of head node at the end too. For example, if elements present in the circular linked list starting from head are 20 → 100 → 40 → 80 → 60, then output should be displayed as 20 100 40 80 60 20.
3. Write a program to find size of
   (a) Doubly Linked List.
   https://www.geeksforgeeks.org/program-find-size-doubly-linked-list/
   (b) Circular Linked List.
   https://www.geeksforgeeks.org/count-nodes-circular-linked-list/
4. Write a program to check if a doubly linked list of characters is palindrome or not.
   https://www.geeksforgeeks.org/check-doubly-linked-list-characters-palindrome-not/
5. Write a program to check if a linked list is Circular Linked List or not.
   https://www.geeksforgeeks.org/check-if-a-linked-list-is-circular-linked-list/

Additional Questions:

- https://www.geeksforgeeks.org/split-a-circular-linked-list-into-two-halves/
- https://www.geeksforgeeks.org/remove-all-even-parity-nodes-from-a-doubly-and-circular-singly-linked-list/
- https://www.geeksforgeeks.org/reverse-doubly-linked-list-groups-given-size/
- https://www.geeksforgeeks.org/correct-the-random-pointer-in-doubly-linked-list/
- https://www.geeksforgeeks.org/construct-a-doubly-linked-linked-list-from-2d-matrix/?ref=rp

Lab Assignment 4

Stacks

1. Develop a menu driven program demonstrating the following operations on a Stack: push(), pop(), isEmpty(), isFull(), display(), and peek().
   Note: Use either arrays or linked list to implement stack.
2. Given a String, Reverse it using STACK. For example “data structure” should be output as “erutcurtsatad.”
3. Write a program that checks if an expression has balanced parentheses.
   https://www.geeksforgeeks.org/check-for-balanced-parentheses-in-an-expression/
4. Write a program to convert an Infix expression into a Postfix expression.
   https://www.geeksforgeeks.org/stack-set-2-infix-to-postfix/
5. Write a program for the evaluation of a Postfix expression.
   https://www.geeksforgeeks.org/stack-set-4-evaluation-postfix-expression/

Additional Questions:

- https://www.interviewbit.com/problems/nearest-smaller-element/
- https://www.geeksforgeeks.org/design-a-stack-that-supports-getmin-in-o1-time-and-o1-extra-space/
- https://www.codechef.com/UCSD2020/problems/DSLA6
- https://www.codechef.com/problems/BEX

Lab Assignment 5

Queues

1. Develop a menu driven program demonstrating the following operations on simple Queues: enqueue(), dequeue(), isEmpty(), isFull(), display(), and peek().
   Note: Use either arrays or linked list to implement queue.
2. Develop a menu driven program demonstrating the following operations on Circular Queues: enqueue(), dequeue(), isEmpty(), isFull(), display(), and peek().
   Note: Use arrays to implement circular queue.
3. Write a program interleave the first half of the queue with second half.
   https://www.geeksforgeeks.org/interleave-first-half-queue-second-half/
4. Write a program to find first non-repeating character in a string using Queue.
   https://www.geeksforgeeks.org/queue-based-approach-for-first-non-repeating-character-in-a-stream/

Additional Questions:

- https://www.geeksforgeeks.org/interesting-method-generate-binary-numbers-1-n/
- https://www.geeksforgeeks.org/sorting-queue-without-extra-space/
- https://www.geeksforgeeks.org/program-page-replacement-algorithms-set-2-fifo/
- https://www.geeksforgeeks.org/lru-cache-implementation/

Lab Assignment 6
Sorting Algorithms

1. Write a program to implement following sorting techniques:
   a. Selection Sort
   b. Insertion Sort
   c. Bubble Sort
   d. Merge Sort
   e. Quick Sort
   f. Counting Sort

2. A slightly improved selection sort.
   https://www.geeksforgeeks.org/sorting-algorithm-slightly-improves-selection-sort/?ref=rp

Lab Assignment 7

Hashing

1. Write a program to determine the most frequent element an array. If there are multiple elements that are appearing the maximum number of times, then print any one of them.
   https://www.geeksforgeeks.org/frequent-element-array/
2. Write a program that, given an array $A[]$ of n numbers and another number x, determines whether or not there exist two elements in $A[]$ whose sum is exactly x.
   https://www.geeksforgeeks.org/given-an-array-a-and-a-number-x-check-for-pair-in-a-with-sum-as-x/
3. You are provided with two arrays. Write a program to find numbers which are present in first array, but not present in the second array.
   https://www.geeksforgeeks.org/find-elements-present-first-array-not-second/
4. Write a program that creates union and intersection lists from the two Linked Lists given as an input. The union list contains unique elements present in both the input Linked Lists and intersection list contains common elements present in the given lists. Order of elements in output lists doesn't matter.
   https://www.geeksforgeeks.org/union-intersection-two-linked-lists-set-3-hashing/

Additional Questions:

- https://www.codechef.com/problems/AUHASH
- https://www.interviewbit.com/problems/valid-sudoku/

Lab Assignment 8
Binary Search Trees and Heaps

1. Write a program for binary search tree (BST) having functions for the following operations:
   - Insert an element (no duplicates are allowed),
   - Delete an existing element,
   - Traverse the BST (in-order, pre-order, and post-order),
   - Maximum depth, and
   - Minimum depth.
2. Implement Heapsort (Increasing/Decreasing order).
3. Implement priority queues using heaps.

Additional Questions:

- Write a non-recursive function to traverse a BST in in-order traversal using stack.
- Given preorder and inorder traversals, construct the BST.
- Given inorder and postorder traversals, construct the BST.

Lab Assignment 9
Graphs and AVL

1. Write a program to represent a graph using adjacency matrix/list and perform basic operations like degree (in/out) of a vertex, adjacent vertices, number of edges, etc.
2. Write a program to implement breadth first search algorithm.
3. Write a program to implement depth first search algorithm.
4. Write a program to implement kruskal's minimum spanning tree algorithm.
5. Write a program to implement prim's minimum spanning tree algorithm.
6. Write a program to implement Dijkstra's shortest path algorithm.
7. AVL: https://www.codechef.com/problems/UCS616A2

Additional Questions:

- Write a program to implement Floyd–Warshall's shortest path algorithm.
- https://www.hackerrank.com/challenges/kruskalmstrsub/problem.
- https://www.hackerrank.com/challenges/bfsshortreach/problem.
- https://www.codechef.com/problems/REVERSE.