Data Structures and Algorithms

Lab Assignment 3

Q1. Implement a Queue using an **array** in C/C++ that supports the following operations:

a) `enqueue(x)` – Insert an element at the rear.

b) `dequeue()` – Delete an element from the front.

c) `peek()` – Display the front element without removing it.

d) `isEmpty()` – Check if the queue is empty.

e) `isFull()` – Check if the queue is full.

f) `display()` – Print all queue elements in order.

Q2. Implement a Queue using a **linked list** in C/C++ supporting the following operations:

a) `enqueue(x)` – Insert an element at the rear node.

b) `dequeue()` – Delete an element from the front node.

c) `peek()` – Display the front element.

d) `isEmpty()` – Check if the queue is empty.

e) `display()` – Print all queue elements.

Q3. Implement a **stack using queue data structure** in C/C++ that supports the following operations:

a) `push(x)`

b) `pop()`

c) `peek()`

d) `isEmpty()`

Q4. Implement a **queue using stack data structure** in C/C++ with the following operations:

a) `enqueue(x)`

b) `dequeue()`

c) `peek()`

d) `isEmpty()`