18-10-2025

3a) a) WAP to simulate the working of a queue of integers using an array. Provide the following operations: Insert, Delete, Display The programshould print appropriate messagesfor queue empty and queue overflow conditions

```
#include <stdio.h>
#define MAX 5   // maximum size of queue
int queue[MAX];
int front = -1, rear = -1;
/* Function to insert an element into the queue */
void insert() {
    int item;
    if (rear == MAX - 1) {
        printf("Queue Overflow! Cannot insert.\n");
        return;
    }
    printf("Enter element to insert: ");
    scanf("%d", &item);
    if (front == -1)
        front = 0;
    queue[++rear] = item;
    printf("Element inserted successfully.\n");
}
/* Function to delete an element from the queue */
void delete() {
    if (front == -1 || front > rear) {
        printf("Queue Empty! Cannot delete.\n");
        return;
    }
    printf("Deleted element: %d\n", queue[front++]);

    if (front > rear) {
        front = rear = -1;
    }
}
```

```
/* Function to display queue elements */
void display() {
    int i;
    if (front == -1) {
        printf("Queue Empty! Nothing to display.\n");
        return;
    }

    printf("Queue elements: ");
    for (i = front; i <= rear; i++) {
        printf("%d ", queue[i]);
    }
    printf("\n");
}
int main() {
    int choice;
    do {
        printf("\n--- Queue Operations Menu ---\n");
        printf("1. Insert\n");
        printf("2. Delete\n");
        printf("3. Display\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                insert();
                break;
            case 2:
                delete();
                break;
            case 3:
                display();
                break;
```

```
            case 4:
                printf("Exiting program.\n");
                break;
            default:
                printf("Invalid choice!\n");
        }
    } while (choice != 4);
    return 0;
}
```