3. b) WAP to simulate the working of a circular 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   // Size of circular queue

int cq[MAX];
int front = -1, rear = -1;

// Insert an element into the circular queue
void insert() {
   int value;

   // Check for overflow condition
   if ((front == 0 && rear == MAX - 1) || (front == rear + 1)) {
      printf("Queue Overflow! Cannot insert element.\n");
      return;
   }

   printf("Enter the value to insert: ");
   scanf("%d", &value);

   if (front == -1) {   // First insertion
      front = rear = 0;
```

```
    } else if (rear == MAX - 1) {
        rear = 0;
    } else {
        rear++;
    }

    cq[rear] = value;
    printf("%d inserted into circular queue.\n", value);
}

// Delete an element from the circular queue
void delete() {
    if (front == -1) {
        printf("Queue is Empty! Cannot delete element.\n");
        return;
    }

    printf("%d deleted from circular queue.\n", cq[front]);

    if (front == rear) { // Only one element
        front = rear = -1;
    } else if (front == MAX - 1) {
        front = 0;
    } else {
        front++;
    }
}
```

```
// Display the circular queue
void display() {
    if (front == -1) {
        printf("Queue is Empty.\n");
        return;
    }

    printf("Circular Queue elements are:\n");

    if (front <= rear) {
        for (int i = front; i <= rear; i++) {
            printf("%d ", cq[i]);
        }
    } else {
        for (int i = front; i < MAX; i++) {
            printf("%d ", cq[i]);
        }
        for (int i = 0; i <= rear; i++) {
            printf("%d ", cq[i]);
        }
    }
    printf("\n");
}

int main() {
    int choice;

    do {
```

```
        printf("\n--- Circular Queue 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! Try again.\n");
        }
    } while (choice != 4);

    return 0;
}
```

OUTPUT:

```
--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the value to insert: 2
2 inserted into circular queue.

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the value to insert: 3
3 inserted into circular queue.

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the value to insert: 4
4 inserted into circular queue.

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the value to insert: 5
5 inserted into circular queue.

--- Circular Queue Menu ---
1. Insert
```

```
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the value to insert: 5
5 inserted into circular queue.

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
2 deleted from circular queue.

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Circular Queue elements are:
3 4 5

--- Circular Queue Menu ---
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 4
Exiting program.
```