LAB 4 - Circular Queue

Name : Abhinav Sanjay

USN : 1BM23CS009

Implement circular queue

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 3 // Define the maximum size of the circular queue

// Global variables for circular queue
int queue[MAX], front = -1,rear = -1,ch,value;

void main() {

   while(1){
      printf("\nMenu:\n");
      printf("1. Enqueue\n");
      printf("2. Dequeue\n");
      printf("3. Display\n");
      printf("4. Exit\n");
      printf("Enter your choice: ");
      scanf("%d", &ch);

      switch (ch) {
         case 1:
            printf("Enter value to enqueue: ");
            scanf("%d", &value);
            enqueue(value);
```

```
            break;
        case 2:
            value = dequeue();
            if (value != -1) {
                printf("Dequeued: %d\n", value);
            }
            break;
        case 3:
            display();
            break;
        case 4:
            printf("Exiting...\n");
            exit(0);
            break;
        default:
            printf("Invalid choice! Please try again.\n");
      }
   }

}

// Function to check if the queue is full
/*int isFull() {
   return (rear + 1) % MAX == front;
}*/

// Function to check if the queue is empty
/*int isEmpty() {
   return front == -1;
```

```
}*/

// Function to add an item to the queue
void enqueue(value) {
    if ((front==rear + 1) % MAX) {
        printf("Queue is full!\n");
    } else {
        if (front==-1) {
            front = 0; // Initialize front
        }
        rear = (rear + 1) % MAX; // Circular increment
        queue[rear] = value;
        printf("Inserted %d\n", value);
    }
}

// Function to remove an item from the queue
int dequeue() {
    if (front==-1,rear==-1) {
        printf("Queue is empty!\n");
        return -1;// Indicating the queue is empty
    } else {
        value = queue[front];
        if (front == rear) {
            // Queue has only one element, reset queue after dequeue
            front = -1;
            rear = -1;
        } else {
            front = (front + 1) % MAX; // Circular increment
```

```
        }
        return value;
    }
}

// Function to display the queue
void display() {
    if (front==-1) {
        printf("Queue is empty!\n");
    } else {
        printf("Queue elements: ");
        int i = front;
        while (1) {
            printf("%d ", queue[i]);
            if (i == rear) break;
            i = (i + 1) % MAX;
        }
        printf("\n");
    }
}
```

Output:

```
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the value to insert: 11
Inserted 11

1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the value to insert: 22
Inserted 22

1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the value to insert: 33
Inserted 33

1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the value to insert: 44
Queue is Full

1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Queue elements are: 11 22 33

1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
Deleted 11
```

```
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
Deleted 22

1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
Deleted 33

1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
Queue is Empty

1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 4

Process returned 0 (0x0)   execution time : 29.988 s
Press any key to continue.
```