Name: Mithlesh Yeole

Roll no. : B3-B3-59

Practical 3A

Code:

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 5
struct Queue {
   int front, rear;
   int *arr;
};
void initQueue(struct Queue *q) {
   q->front = -1;
   q->rear = -1;
   q->arr = (int *)malloc(MAX * sizeof(int));
   if (q->arr == NULL) {
      printf("Memory allocation failed!\n");
      exit(1);
   }
}
int isFull(struct Queue *q) {
   return ((q->rear + 1) % MAX == q->front);
}
int isEmpty(struct Queue *q) {
   return (q->front == -1);
}
void enqueue(struct Queue *q, int value) {
   if (isFull(q)) {
      printf("Queue is full", value);
   } else {
      if (q->front == -1) {
```

```
            q->front = 0;
        }
        q->rear = (q->rear + 1) % MAX;
        q->arr[q->rear] = value;
        printf("Added %d to the queue", value);
    }
}
int dequeue(struct Queue *q) {
    if (isEmpty(q)) {
        printf("Queue is empty");
        return -1;
    } else {
        int value = q->arr[q->front];
        if (q->front == q->rear) {
            q->front = q->rear = -1;
        } else {
            q->front = (q->front + 1)%MAX;
        }
        return value;
    }
}
void display(struct Queue *q) {
    if (isEmpty(q)) {
        printf("Queue is empty");
        return;
    }
    int i = q->front;
    printf("Queue: ");
    while (i != q->rear) {
        printf("%d ", q->arr[i]);
        i = (i + 1) % MAX;
    }
```

```
    printf("%d", q->arr[q->rear]);
}
void freeQueue(struct Queue *q) {
    free(q->arr);
}
int main() {
    struct Queue q;
    initQueue(&q);
    int choice, value;
    while (1) {
        printf("\n1. Enqueue\n");
        printf("2. Dequeue\n");
        printf("3. Display\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                printf("Enter value to enqueue: ");
                scanf("%d", &value);
                enqueue(&q, value);
                break;
            case 2:
                value = dequeue(&q);
                if (value != -1) {
                    printf("Dequeued: %d", value);
                }
                break;
            case 3:
                display(&q);
                break;
            case 4:
```

```
            printf("Exiting...\n");
            freeQueue(&q);
            return 0;
        default:
            printf("Invalid choice");
    }
  }
}
```

Output: