

Name: Mithlesh Yeole

Roll no. : B3-B3-59

Practical 3B

Code:

```
#include <stdio.h>

#define SIZE 5

typedef struct {
    int items[SIZE];
    int front, rear;
} CircularQueue;

void initialize(CircularQueue *q) {
    q->front = -1;
    q->rear = -1;
}

int isEmpty(CircularQueue *q) {
    return (q->front == -1);
}

int isFull(CircularQueue *q) {
    return ((q->rear + 1) % SIZE == q->front);
}

void enqueue(CircularQueue *q, int value) {
    if (isFull(q)) {
        printf("Queue is full %d\n", value);
        return;
    }
    if (isEmpty(q)) {
        q->front = 0;
        q->rear = 0;
    } else {
        q->rear = (q->rear + 1) % SIZE;
    }
}
```

```

    q->items[q->rear] = value;
    printf("Added %d\n", value);
}

void dequeue(CircularQueue *q) {
    if (isEmpty(q)) {
        printf("Queue is empty\n");
        return;
    }
    printf("Removed %d from the queue\n", q->items[q->front]);
    if (q->front == q->rear) {
        q->front = -1;
        q->rear = -1;
    } else {
        q->front = (q->front + 1) % SIZE;
    }
}

void display(CircularQueue *q) {
    if (isEmpty(q)) {
        printf("Queue is empty\n");
        return;
    }
    printf("Queue elements: ");
    int i = q->front;
    while (1) {
        printf("%d ", q->items[i]);
        if (i == q->rear) break;
        i = (i + 1) % SIZE;
    }
    printf("\n");
}

int main() {
    CircularQueue q;

```

```

initialize(&q);

enqueue(&q, 1);
enqueue(&q, 2);
enqueue(&q, 3);
enqueue(&q, 4);
enqueue(&q, 5);
enqueue(&q, 6);

display(&q);

dequeue(&q);
dequeue(&q);

display(&q);

enqueue(&q, 10);
enqueue(&q, 20);

display(&q);

return 0;
}

```

Output:

```

main.c
25      return;
26    }
27    if (isEmpty(q)) {
28      q->front = 0;
29      q->rear = 0;
30    } else {
31      q->rear = (q->rear + 1) % SIZE;
32    }
33    q->items[q->rear] = value;
34    printf("Added %d\n", value);
35  }
36
37  void dequeue(CircularQueue *q) {
38    if (isEmpty(q)) {
39      printf("Queue is empty\n");
40    }
41  }
42
43  int main() {
44    CircularQueue q;
45    initialize(&q);
46
47    enqueue(&q, 1);
48    enqueue(&q, 2);
49    enqueue(&q, 3);
50    enqueue(&q, 4);
51    enqueue(&q, 5);
52    enqueue(&q, 6);
53
54    display(&q);
55
56    dequeue(&q);
57    dequeue(&q);
58
59    display(&q);
60
61    enqueue(&q, 10);
62    enqueue(&q, 20);
63
64    display(&q);
65
66    return 0;
67  }

```

Input

```

Added 1
Added 2
Added 3
Added 4
Added 5
Queue is full 6
Queue elements: 1 2 3 4 5
Removed 1 from the queue
Removed 2 from the queue
Queue elements: 3 4 5
Added 10
Added 20
Queue elements: 3 4 5 10 20
...Program finished with exit code 0
Press ENTER to exit console.

```