# EXPERIMENT-4

## Program :

```
#include <stdio.h>
int FRONT = 0,REAR=-1;
void Enqueue(int Q[],int value,int MAX){
    if(REAR==MAX-1){
        printf("Queue Overflow!\n");
    }
    else{
        Q[++REAR]=value;
        printf("Enqueued %d!\n",value);
        printf("QUEUE Elements: ");
        for(int i=FRONT;i<=REAR;i++){
      printf("%d\t",Q[i]);
      }
      printf("\n");
    }
}
void Dequeue(int Q[]){
    if(FRONT>REAR){
        printf("Queue Underflow!\n");
    }else{
        printf("Dequeued %d!\n",Q[FRONT]);
        FRONT++;
        printf("QUEUE Elements: ");
        for(int i=FRONT;i<=REAR;i++){
```

```
      printf("%d\t",Q[i]);
      }
        printf("\n");
    }
}
int main(){
    int MAX;
    printf("Enter the size of queue: ");
    scanf("%d",&MAX);
    int queue[MAX];
    while(1){
        int choice,value;
        printf("\n---------!Select queue option!---------\n");
        printf("1. Enqueue\t");
        printf("2. Dequeue\t");
        printf("3. Exit\n");
        printf("Choice: ");
        scanf("%d", &choice);
        switch(choice){
            case 1: if(REAR==MAX-1){
            printf("Queue is FULL cannot enqueue.");
            break;
            }else{
            printf("Enter the element to enqueue: ");
            scanf("%d",&value);
            Enqueue(queue,value,MAX);
            }
```

```
            break;
            case 2:printf("Dequeuing........done!\n");
            Dequeue(queue);
            break;
            case 3: return 0;
            default:printf("Invalid choice!");
        }
    }
}
```

# Output :

```
student@sjcet-ulin24: ~/Alwin$ gcc deque.c
student@sjcet-ulin24: ~/Alwin$ ./a.out
Enter the size of the Deque:3

1. Insert Front
2. Insert Rear
3. Delete Front
4. Delete Rear
5. Exit
Enter choice: 1
Enter value to insert at front: 2
2 inserted at front
Deque: 2

1. Insert Front
2. Insert Rear
3. Delete Front
4. Delete Rear
5. Exit
Enter choice: 1
Enter value to insert at front: 3
3 inserted at front
Deque: 3 2

1. Insert Front
2. Insert Rear
3. Delete Front
4. Delete Rear
5. Exit
Enter choice: 2
Enter value to insert at rear: 4
4 inserted at rear
Deque: 3 2 4
```

```
Deque: 3 2 4

1. Insert Front
2. Insert Rear
3. Delete Front
4. Delete Rear
5. Exit
Enter choice: 2
Enter value to insert at rear: 5
Deque is Full
Deque: 3 2 4

1. Insert Front
2. Insert Rear
3. Delete Front
4. Delete Rear
5. Exit
Enter choice: 3
3 deleted from front
Deque: 2 4

1. Insert Front
2. Insert Rear
3. Delete Front
4. Delete Rear
5. Exit
Enter choice: 4
4 deleted from rear
Deque: 2

1. Insert Front
2. Insert Rear
3. Delete Front
4. Delete Rear
5. Exit
Enter choice: 4
2 deleted from rear
```

```
3 deleted from front
Deque: 2 4

1. Insert Front
2. Insert Rear
3. Delete Front
4. Delete Rear
5. Exit
Enter choice: 4
4 deleted from rear
Deque: 2

1. Insert Front
2. Insert Rear
3. Delete Front
4. Delete Rear
5. Exit
Enter choice: 4
2 deleted from rear
Deque is Empty

1. Insert Front
2. Insert Rear
3. Delete Front
4. Delete Rear
5. Exit
Enter choice: 3
Deque is Empty
Deque is Empty

1. Insert Front
2. Insert Rear
3. Delete Front
4. Delete Rear
5. Exit
Enter choice: 5
student@sjcet-ulin24: ~/Alwin$
```