**PROGRAM: Implementation of Circular Queue Linked Lists**

**Input:**

#include<stdio.h>

#include<stdlib.h>

struct Node{

    int data;

    struct Node \*link;

};

struct Node \*front=NULL;

struct Node \*rear=NULL;

void enqueue(){

    int x;

    printf("Enter element: ");

    scanf("%d", &x);

    struct Node \*tmp;

    tmp=(struct Node\*)malloc(sizeof(struct Node));

    if(tmp==NULL){

        printf("Dynamic Memory Allocation not possible!\n");

    }

    else{

        tmp->data=x;

        tmp->link=front;

        if(front==NULL){

            front=tmp;

        }

        else{

            rear->link=tmp;

        }

        rear=tmp;

        printf("Element inserted successfully\n");

    }

}

void dequeue(){

    struct Node \*p;

    if(front==NULL){

        printf("Queue Underflow\n");

    }

    else if(rear==front){

        p =front;

        free(p);

Expt no: 9

        front=rear=NULL;

    }

    else{

        p=front;

        front=front->link;

        rear->link=front;

        free(p);

        printf("Element deleted successfully\n");

    }

}

void peek(){

    if(front==NULL){

        printf("Queue Underflow\n");

    }

    else{

        printf("The topmost element is %d\n", front->data);

    }

}

void display(){

    struct Node \*p;

    if(front==NULL){

        printf("Queue Empty\n");

    }

    else{

        p=front;

        printf("\nQueue:");

        do{

            printf("%d ", p->data);

            p=p->link;

        }while(p!=rear->link);

        printf("\n");

    }

}

int main(){

    int a;

    printf("1.Enqueue\n2.Dequeue\n3.Peek\n4.Display\n5.Exit\n");

    while(1)

    {

        printf("\nEnter choice:");

        scanf("%d",&a);

        switch(a)

        {

            case 1:

                enqueue();

                display();

                break;

            case 2:

                dequeue();

                display();

                break;

            case 3:

                peek();

                break;

            case 4:

                display();

                break;

            case 5:

                exit(1);

            default:

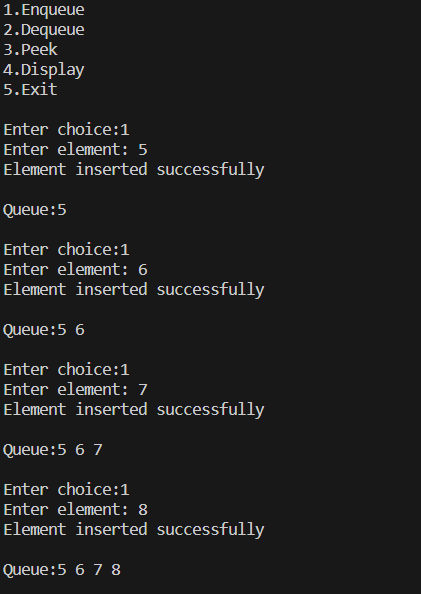
                printf("Invalid Choice\n");

                break;}

}}

**Output:**

1. Enqueue:

****

Expt no: 9

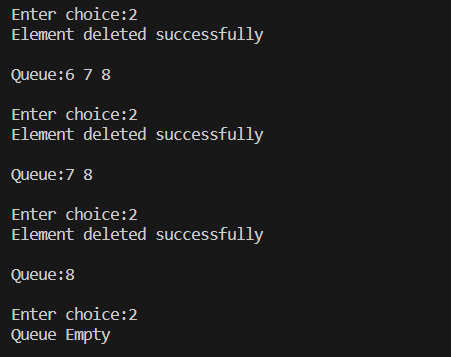
1. Peek:

****

1. Display:



1. Dequeue:



1. Display (empty queue):

