Expt no: 9

**PROGRAM: C Program to implement Circular Queue Linked List.**

**Input:**

#include <stdio.h>

#include <stdlib.h>

struct Node

{

int data;

struct Node\* next;

};

struct Queue {

struct Node \*front, \*rear;

};

struct Node\* createNode(int value)

{

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

newNode->data = value;

newNode->next = NULL;

return newNode;

}

struct Queue\* createQueue()

{

struct Queue\* queue = (struct Queue\*)malloc(sizeof(struct Queue));

queue->front = queue->rear = NULL;

return queue;

}

int isEmpty(struct Queue\* queue)

{

return (queue->front == NULL);

}

void enqueue(struct Queue\* queue, int value)

{

struct Node\* newNode = createNode(value);

if (isEmpty(queue))

{

queue->front = newNode;

} else {

queue->rear->next = newNode;

}

queue->rear = newNode;

queue->rear->next = queue->front;

}

int dequeue(struct Queue\* queue)

{

int value;

if (isEmpty(queue)) {

printf("Queue is empty.\n");

return -1;

}

if (queue->front == queue->rear)

{

value = queue->front->data;

free(queue->front);

queue->front = queue->rear = NULL;

} else {

struct Node\* temp = queue->front;

value = temp->data;

queue->front = queue->front->next;

queue->rear->next = queue->front;

free(temp);

}

return value;

}

void displayQueue(struct Queue\* queue)

{

struct Node\* temp = queue->front;

if (isEmpty(queue))

{

printf("Queue is empty.\n");

return;

}

printf("Elements in the circular queue are: ");

do {

printf("%d ", temp->data);

temp = temp->next;

} while (temp != queue->front);

printf("\n");

}

int main()

{

struct Queue\* queue = createQueue();

int choice, element;

do {

printf("\nCircular Queue Operations\n");

printf("1. 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 element to enqueue: ");

scanf("%d", &element);

enqueue(queue, element);

break;

case 2:

element = dequeue(queue);

if (element != -1)

{

printf("Dequeued element: %d\n", element);

}

break;

case 3:

displayQueue(queue);

break;

case 4:

printf("Exiting the program.\n");

break;

default:

printf("Invalid choice. Please enter a valid option.\n");

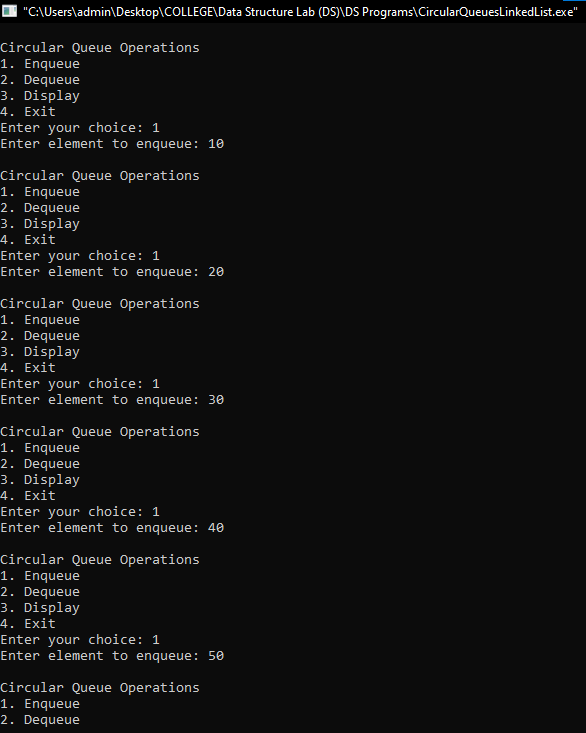
}

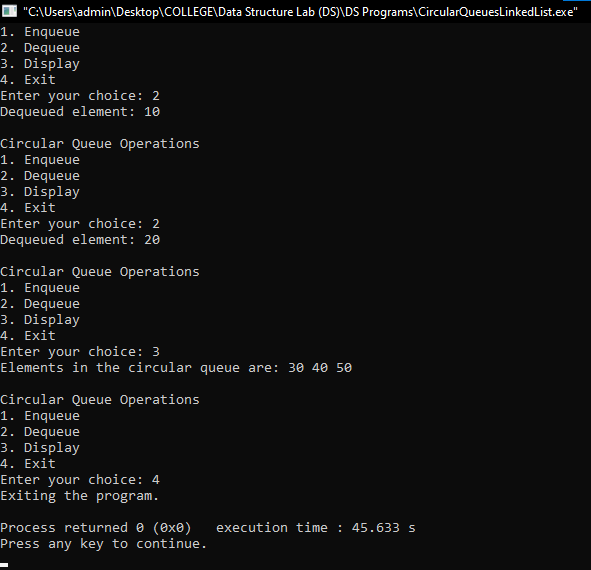
} while (choice != 4);

return 0;

}

**Output:**

****

****