**Program to implement Circular Queues using Linked lists.**

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

void enqueue();

int isEmpty();

void dequeue();

void peek();

void display();

struct node{

int data;

struct node \*next;

};

struct node \*front = NULL;

struct node \*rear = NULL;

int main(){

int choice = 0, element = 0;

printf("\n ===== Menu =====");

printf("\n1. ENQUEUE");

printf("\n2. DEQUEUE");

printf("\n3. PEEK");

printf("\n4. DISPLAY");

printf("\n5. EXIT\n");

while (1){

printf("\nEnter your choice: ");

scanf("%d", &choice);

switch (choice){

case 1:

enqueue();

break;

case 2:

dequeue();

break;

case 3:

peek();

break;

case 4:

display();

break;

case 5: return 0;

default: printf("\nEnter correct choice");

}

}

\_getch();

}

int isEmpty(){

if (front == NULL)

return 1;

return 0;

}

void enqueue(){

struct node \*newNode, \*temp;

int element;

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

printf("\nEnter element : ");

scanf("%d", &element);

newNode->data = element;

newNode->next = NULL;

if (front == NULL){

front = newNode;

newNode->next = front;

rear = newNode;

}else{

newNode->next = front;

rear->next = newNode;

rear = newNode;

}

printf("Element Inserted\n");

}

void dequeue(){

struct node \*temp;

if (isEmpty()){

printf("Queue Empty\n");

return;

}

else if (front == rear){

temp = front;

front = rear = NULL;

printf("Element(%d) Deleted\n",temp -> data);

free(temp);

} else {

temp = front;

front = front->next;

rear->next = front;

printf("Element(%d) Deleted\n",temp -> data);

free(temp);

}

}

void peek(){

if (isEmpty()){

printf("Queue Empty\n");

return;

}

printf("Front element : %d\n", front->data);

}

void display(){

struct node \*ptr;

if (front == NULL){

printf("Queue Empty\n");

return;

}

ptr = front;

printf("Queue :");

printf("%d\t", ptr->data);

ptr = ptr->next;

while (ptr != front){

printf("%d\t", ptr->data);

ptr = ptr->next;

}

printf("\n");}

**Output:**

