NAME : Ramya Ramesh

USN : 1BM19CH038

**Lab 5: Circular Queue**

QUESTION :

WAP to simulate the working of a circular queue of integers using an array. Provide the following operations. a) Insert b) Delete c) Display The program should print appropriate messages for queue empty and queue overflow conditions

CODE :

#include <stdio.h>

#include <stdlib.h>

#define SIZE 4

int items[SIZE];

int front = -1, rear = -1;

int isFull() {

if ((front == rear + 1) || (front == 0 && rear == SIZE - 1)) return 1;

return 0;

}

int isEmpty() {

if (front == -1) return 1;

return 0;

}

void enQueue(int element) {

if (isFull())

printf("\n Queue overflow \n");

else {

if (front == -1) front = 0;

rear = (rear + 1) % SIZE;

items[rear] = element;

printf("\n Inserted element = %d", element);

}

}

int deQueue() {

int element;

if (isEmpty()) {

printf("\n Queue underflow \n");

return (-1);

} else {

element = items[front];

if (front == rear) {

front = -1;

rear = -1;

}

else{

front = (front + 1) % SIZE;

}

printf("\n Deleted element = %d \n", element);

return (element);

}

}

void display() {

int i;

if (isEmpty())

printf(" \n Empty Queue\n");

else {

printf("\n Front = %d ", front);

printf("\n Items are: ");

for (i = front; i != rear; i = (i + 1) % SIZE) {

printf("%d ", items[i]);

}

printf("%d ", items[i]);

printf("\n Rear = %d \n", rear);

}

}

void main()

{

int choice,x,b;

while(1)

{

printf("\n\*\*\*MENU\*\*\*\n");

printf("\n 1.Enqueue\n 2.Dequeue\n 3.Display\n 4.Exit\n");

printf("Enter your choice\n");

scanf("%d",&choice);

switch(choice)

{

case 1:printf("Enter the number to be inserted into the queue\n");

scanf("%d",&x);

enQueue(x);

break;

case 2:b=deQueue();

printf("%d was removed from the queue\n",b);

break;

case 3:display();

break;

case 4:exit(1);

default:printf("Invalid input\n");

}

}

}

OUTPUT :











