**Assignment 7**

**Q1 :- Write a program implementing insert, delete and display operation of Circular Queue.**

**Answer :-**

A circular queue in C stores the data in a very practical manner. It is a linear data structure. It is very similar to the queue. The only difference is that the last node is connected back to the first node. Thus it is called a circular queue.

A circular queue solved the limitations of the normal queue. Thus making it a better pick than the normal queue. It also follows the first come first serve algorithm. Circular Queue is also called ring Buffer.

**Operations On A Circular Queue**

1. Enqueue- adding an element in the queue if there is space in the queue.
2. Dequeue- Removing elements from a queue if there are any elements in the queue
3. Front- get the first item from the queue.
4. Rear- get the last item from the queue.
5. isEmpty/isFull- checks if the queue is empty or full.

Program:-

#include<stdio.h>

# define MAX 5

int cqueue\_arr[MAX];

int front = -1;

int rear = -1;

void insert(int item)

{

if((front == 0 && rear == MAX-1) || (front == rear+1))

{

printf("Queue Overflow n");

return;

}

if(front == -1)

{

front = 0;

rear = 0;

}

else

{

if(rear == MAX-1)

rear = 0;

else

rear = rear+1;

}

cqueue\_arr[rear] = item ;

}

void deletion()

{

if(front == -1)

{

printf("Queue Underflown");

return ;

}

printf("Element deleted from queue is : %dn",cqueue\_arr[front]);

if(front == rear)

{

front = -1;

rear=-1;

}

else

{

if(front == MAX-1)

front = 0;

else

front = front+1;

}

}

void display()

{

int front\_pos = front,rear\_pos = rear;

if(front == -1)

{

printf("Queue is empty \n");

return;

}

printf("Queue elements :\n");

if( front\_pos <= rear\_pos )

while(front\_pos <= rear\_pos)

{

printf("%d ",cqueue\_arr[front\_pos]);

front\_pos++;

}

else

{

while(front\_pos <= MAX-1)

{

printf("%d ",cqueue\_arr[front\_pos]);

front\_pos++;

}

front\_pos = 0;

while(front\_pos <= rear\_pos)

{

printf("%d ",cqueue\_arr[front\_pos]);

front\_pos++;

}

}

printf("\n");

}

int main()

{

int choice,item;

do

{

printf("1.Insert\t");

printf("2.Delete\t");

printf("3.Display\t");

printf("4.Quit\t");

printf("Enter your choice : ");

scanf("%d",&choice);

switch(choice)

{

case 1 :

printf("Input the element for insertion in queue : ");

scanf("%d", &item);

insert(item);

break;

case 2 :

deletion();

break;

case 3:

display();

break;

case 4:

break;

default:

printf("Wrong choicen");

}

}while(choice!=4);

return 0;

}

Output:--

