**Aim:**

To write a menu driven program to insert , display & Delete elements in a circular queue.

**Source Code:**

#include<iostream.h>

#include<stdlib.h>

#include<process.h>

int Insert\_in\_CQ(int[],int) ;

void Display(int[],int,int) ;

int Del\_in\_CQ(int CQueue[]) ;

const int size = 7 ;

int CQueue[size],front=-1,rear=-1;

int main()

{

int Item,res,ch;

do

{

system("cls");

cout<<" \t\t\t Circular Queue Menu\n " ;

cout<<" \t 1. Insert\n ";

cout<<" \t 2. Delete\n ";

cout<<" \t 3. Display\n ";

cout<<" \t 4. Exit\n ";

cout<<" Enter your choice (1-4)... ";

cin>>ch ;

switch(ch)

{

case 1 :

cout<<" \n Enter ITEM for insertion : ";

cin>>Item;

res=Insert\_in\_CQ(CQueue,Item);

if(res==-1)

cout<<" OVERFLOW!!! \n ";

else

{

cout<<" \n Now the Cir\_Queue is : \n ";

Display(CQueue,front,rear);

}

system("pause");

break;

case 2 :

Item=Del\_in\_CQ(CQueue);

cout<<" Element deleted is : "<< Item << endl;

Display(CQueue,front,rear);

system("pause");

break;

case 3 :

Display(CQueue,front,rear);

system("pause");

break;

case 4 :

break;

default :

cout<<" Valid choices are 1...4 only\n ";

system("pause");

break ;

}

} while(ch!= 4);

return 0 ;

}

int Insert\_in\_CQ(int CQueue[], int ele)

{

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

return -1;

else if(rear==-1)

front=rear=0;

else if (rear==size-1)

rear=0;

else rear++;

CQueue[rear]=ele;

return 0 ;

}

void Display(int CQueue[ ], int front, int rear)

{

int i=0;

cout<<" \n Cir\_Queue is : \n";

cout<<" ( Front shown as >>>, Rear as <<< AND free space as - )\n";

if(front==-1)

return ;

if(rear>=front)

{

for(i=0;i<front;i++)

cout<<"-";

cout<<" >>> ";

for(i=front;i<rear;i++)

cout<<CQueue[i]<<" <-";

cout<<CQueue[rear]<<"<<<"<<endl;

}

else

{

for(i=0;i<rear;i++)

cout<<CQueue[i]<<" <- ";

cout<<CQueue[rear]<<" << <";

for(;i<front;i++)

cout<<" -";

cout<<">>>";

for(i=front;i<size;i++)

cout<<CQueue[i]<<" <- "=;

cout<<" \t ...wrap around... ";

}

}

int Del\_in\_CQ( int CQueue[ ] )

{

int ret;

if(front==-1)

return -1;

else

{

ret=CQueue[front] ;

if(front==rear)

front=rear=-1;

else if(front==size-1)

front =0;

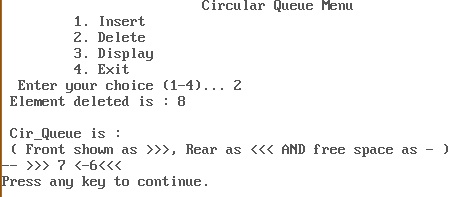
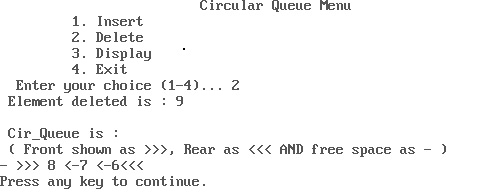
else front++ ;

}

return ret ;

}

**Output :**

****