**Program No. 13**

Objective:-WAP for formation of Circular Queue

#include<stdio.h>

int q[10],front=0,rear=-1,max;

void Insert\_Element(int max)

{

int x;

if((front==0&&rear==max-1)||(front>0&&rear==front-1))

printf("Queue is overflow\n");

else

{

printf("\nEnter element to be insert:");

scanf("%d",&x);

if(rear==max-1&&front>0)

{

rear=0;

q[rear]=x;

}

else

{

if((front==0&&rear==-1)||(rear!=front-1))

q[++rear]=x;

}

}

}

void Delete\_Element(int max)

{

int a;

if((front==0)&&(rear==-1))

{

printf("Queue is underflow\n");

getch();

}

if(front==rear)

{

a=q[front];

rear=-1;

front=0;

}

else

if(front==max-1)

{

a=q[front];

front=0;

}

else a=q[front++];

printf("Deleted element is:%d\n",a);

}

void display()

{

int i,j;

if(front==0&&rear==-1)

{

printf("Queue is underflow\n");

getch();

}

if(front>rear)

{

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

printf("\t%d",q[i]);

for(j=front;j<=max-1;j++)

printf("\t%d",q[j]);

printf("\nRear is at %d\n",q[rear]);

printf("\nFront is at %d\n",q[front]);

}

else

{

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

{

printf("\t%d",q[i]);

}

printf("\nRear is at %d\n",q[rear]);

printf("\nFront is at %d\n",q[front]);

}

printf("\n");

}

void main()

{

int ch,n;

char choice;

printf("Enter the value between (1-2)=\n");

printf("\n\nFor Adding Element in Circular Queue Press 1:");

printf("\n\nFor Deleting Element in Circular Queue Press 2:\n");

scanf("%d",&n);

printf("\n");

switch(n)

{

case 1:

{

printf("\nEnter the size of Queue=");

scanf("%d",&max);

do

{

Insert\_Element(max);

printf("Do u want to add further element=\n");

choice=getch();

}

while(choice=='Y'||choice=='y');

printf("Your Circular Queue is in order of=\n");

display();

getch();

break;

}

case 2:

{

printf("\nEnter the size of Queue=");

scanf("%d",&max);

do

{

Insert\_Element(max);

printf("Do u want to add further element=\n");

choice=getch();

}

while(choice=='Y'||choice=='y');

printf("Your Circular Queue is in order of=\n");

display();

printf("\n\nYour Deleted Circular Queue List Is=\n");

do

{

Delete\_Element(max);

printf("\n");

display();

printf("\nDo u want to delete further node=\n");

choice=getch();

}

while(choice=='Y'||choice=='y');

printf("\n\nYour Circular Queue is in order of=\n");

display();

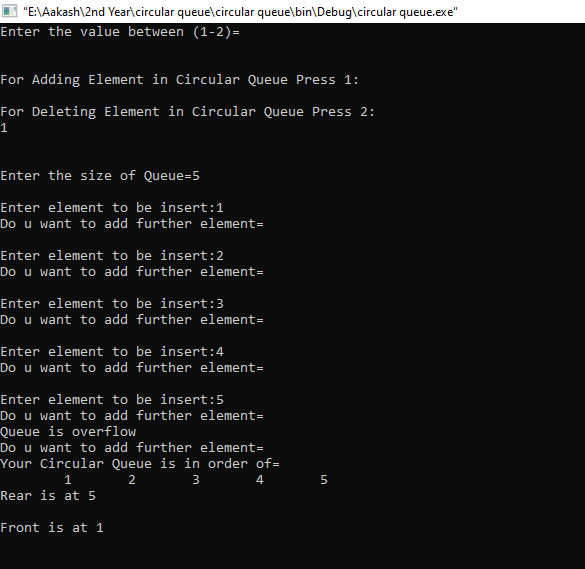
break;

}

}

}

Output:-



Operation performed on Circular Queue