**Program:**

#include<stdio.h>

#include<math.h>

#define Max 5

int cqueue[Max];

int front=-1;

int rear=-1;

void cenqueue(int x)//Related to Rear

{

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

        printf("Overflow \n");

        else if (front==-1 && rear==-1)//initial element

        front=rear=0;

        else if(rear==Max-1 && front!=0)//Circular special

        rear=0;

        else

        rear++;

        cqueue[rear]=x;//Actually addding elem

        printf("\n front is %d",front);

        printf("\n rear is %d \n",rear);

}

void cdequeue() //Related to Front

{

    if(front==-1 && rear==-1)

        printf("Underflow \n");

    printf("Deleted %d from the Queue \n",cqueue[front]);

    if (front==rear)//Delete last

        front=rear=-1;//reset to empty

    else

    {  if(front==Max-1)

            front=0;

        else

            front++;

    }

}

void cgetFront()

{

    if (front==-1 && rear==-1)

    {

        printf("Queue is empty \n");

    }

    else

        printf("The element in front is %d \n",(cqueue[front]));

}

void cgetRear()

{

    if((front==-1 && rear==-1) )

    {

        printf("Queue is empty \n");

    }

    else

       printf("The element in rear is %d  \n ",(cqueue[rear]));

}

void cisFull()//Related to Rear

{

        if(front==0 && rear==Max-1)

        printf("Queue is Full \n");

        else

        printf("Queue is Not Full \n");

}

void cisEmpty()

{

    if((front==-1 && rear==-1) )

        printf("Queue is empty \n");

    else

        printf("Queue is Not Empty \n");

}

void cdisplay()

{

    int i;

    if((front==-1 && rear==-1) )

        printf("Queue is empty \n");

    else

    {

        if(front<rear)

        {

            for(i=front;i<=rear;i++)

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

        }

        else

        {

            for(i=front;i<Max;i++)

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

            for(i=0;i<=rear;i++)

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

        }

        printf("\n");

    }

}

void csize()

{

    int i,count=0;

    if((front==-1 && rear==-1)  )

        printf("Queue is empty \n");

    else

    {

        if(front<rear)

        {

            for(i=front;i<=rear;i++)

               count++;

        }

         else

        {

            for(i=front;i<Max;i++)

                count++;

            for(i=0;i<=rear;i++)

                count++;

        }

    }

     printf("\n There are %d elements \n",count);

}

int main()

{

    int choice,ans;

    printf("1 To insert an element \n");

    printf("2 To delete an  element \n");

    printf("3 To check  element in front \n");

    printf("4 To check  element in rear \n");

    printf("5 To check if Queue is full \n");

    printf("6 To check if Queue is empty \n");

    printf("7 To count number of elements \n");

    printf("8 To Display the queue \n");

    do

    {

        printf("Enter your choice \n");

        scanf("%d",&choice);

    switch (choice)

    {

    case 1:

        int num,x;

        do

        {

            printf("Enter number to be inserted \n");

            scanf("%d",&num);

            cenqueue(num);

            printf("Press 1 to add more numbers \n");

            scanf("%d",&x);

        }while(x==1);

        break;

    case 2:

        cdequeue();

        break;

    case 3:

        cgetFront();

        break;

    case 4:

        cgetRear();

        break;

    case 5:

        cisFull();

        break;

    case 6:

        cisEmpty();

        break;

    case 7:

        csize();

        break;

    case 8:

        cdisplay();

        break;

    default:

        printf("Invalid Choice \n ");

        break;

    }

    printf("Press 1 to choose another option \n");

    scanf("%d",&ans);

    }while (ans==1);

    return 0;

}

**Output:**

1 To insert an element

2 To delete an element

3 To check element in front

4 To check element in rear

5 To check if Queue is full

6 To check if Queue is empty

7 To count number of elements

8 To Display the queue

Enter your choice

1

Enter number to be inserted

10

front is 0

rear is 0

Press 1 to add more numbers

1

Enter number to be inserted

20

front is 0

rear is 1

Press 1 to add more numbers

1

Enter number to be inserted

30

front is 0

rear is 2

Press 1 to add more numbers

1

Enter number to be inserted

40

front is 0

rear is 3

Press 1 to add more numbers

0

Press 1 to choose another option

1

Enter your choice

2

Deleted 10 from the Queue

Press 1 to choose another option

1

Enter your choice

1

Enter number to be inserted

60

front is 1

rear is 4

Press 1 to add more numbers

1

Enter number to be inserted

70

front is 1

rear is 0

Press 1 to add more numbers

0

Press 1 to choose another option

1

Enter your choice

3

The element in front is 20

Press 1 to choose another option

1

Enter your choice

4

The element in rear is 70

Press 1 to choose another option

1

Enter your choice

8

20 30 40 60 70

Press 1 to choose another option

1

Enter your choice

7

There are 5 elements

Press 1 to choose another option

1

Enter your choice

5

Queue is Not Full

Press 1 to choose another option

1

Enter your choice

6

Queue is Not Empty

Press 1 to choose another option

0