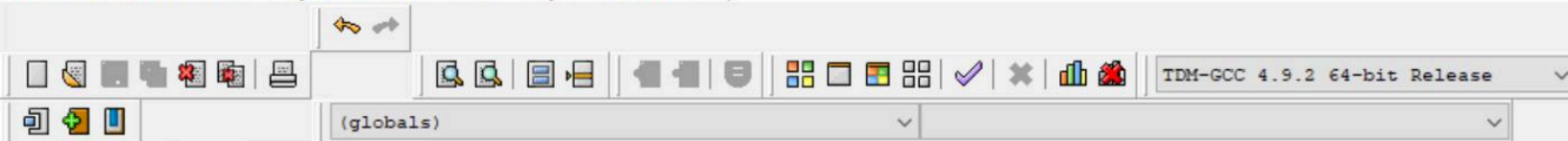


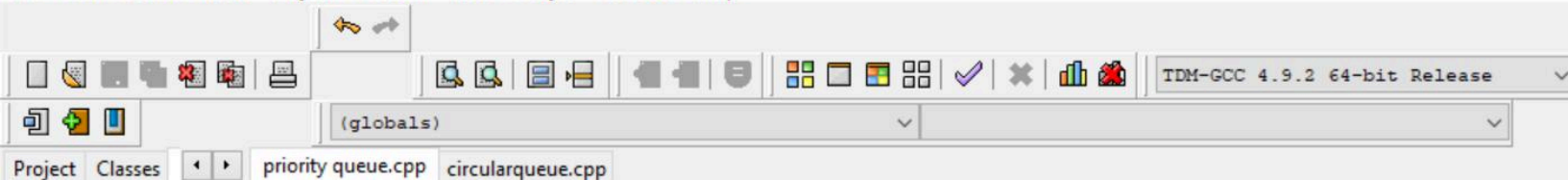
Project Classes priority queue.cpp circularqueue.cpp

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  #define qsize 5
4  int f=0,r=-1,ch;
5  int item,q[10];
6
7  int isfull()
8  {
9      return(r==qsize-1)?1:0;
10 }
11 int isempty()
12 {
13     return(f>r)?1:0;
14 }
15 void insert_rear()
16 {
17     if(isfull())
18     {
19         printf("queue overflow\n");
20         return;
21     }
22     r=r+1;
23     q[r]=item;
24 }
25 void delete_front()
26 {
27     if(isempty())
28     {
29         printf("queue empty\n");
30         return;
31     }
32     printf("item deleted is %d\n",q[(f++)]);
33     if(f>r)
34     {
35         f=0;
36         r=-1;
37     }
38 }
39 void insert_front()
40 {
```



Project Classes priority queue.cpp circularqueue.cpp

```
38 }
39 void insert_front()
40 {
41     if(f!=0)
42     {
43         f=f-1;
44         q[f]=item;
45         return;
46     }
47     else if((f==0)&&(r==1))
48     {
49         q[++(r)]=item;
50         return;
51     }
52     else
53         printf("insertion not possible\n");
54 }
55 void delete_rear()
56 {
57     if(isempty())
58     {
59         printf("queue is empty\n");
60         return;
61     }
62     printf("item deleted is %d\n",q[(r)--]);
63     if(f>r)
64     {
65         f=0;
66         r=-1;
67     }
68 }
69 void display()
70 {
71     int i;
72     if(isempty())
73     {
74         printf("queue empty\n");
75         return;
76     }
77     for(i=f;i<=r;i++)
```



```
69 void display()
70 {
71     int i;
72     if(isempty())
73     {
74         printf("queue empty\n");
75         return;
76     }
77     for(i=f; i<=r; i++)
78         printf("%d\n", q[i]);
79 }
80 int main()
81 {
82
83
84     for(;;)
85     {
86         printf("1.insert_rear\n2.insert_front\n3.delete_rear\n4.delete_front\n5.display\n6.exit\n");
87         printf("enter choice\n");
88         scanf("%d", &ch);
89         switch(ch)
90         {
91             case 1: printf("enter the item\n");
92                     scanf("%d", &item);
93                     insert_rear();
94                     break;
95             case 2: printf("enter the item\n");
96                     scanf("%d", &item);
97                     insert_front();
98                     break;
99             case 3: delete_rear();
100                    break;
101             case 4: delete_front();
102                    break;
103             case 5: display();
104                    break;
105             default: exit(0);
106         }
107     }
108 }
```

C:\Users\Nikita\Desktop\programs\priority queue.exe

```
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
10
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
20
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
30
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
40
1.insert_rear
```

C:\Users\Nikita\Desktop\programs\priority queue.exe

```
enter choice
1
enter the item
40
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
50
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
60
queue overflow
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
3
item deleted is 50
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
```

C:\Users\Nikita\Desktop\programs\priority queue.exe

```
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
3
item deleted is 50
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 10
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
20
30
40
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
```