

```
m_p_queue.cpp
1  #include<stdio.h>
2  #include<conio.h>
3  #include<process.h>
4  #include<stdlib.h>
5  #define N 3
6  int queue[3][N];
7  int front[3]={0,0,0};
8  int rear[3]={-1,-1,-1};
9  int itempr;
10 int pqinsert(int);
11 int pqdelete();
12 int display();
13 int main()
14 {
15     int ch;
16     system("cls");
17     while(1)
18     {
19         printf("PRIORITY QUEUE\n");
20         printf("*****\n");
21         printf("\n\t1: PQinsert\n");
22         printf("\n\t2: PQdelete\n");
23         printf("\n\t3: PQdisplay\n");
24         printf("\n\t4: Exit\n");
25         printf("\nenter the choice\n");
26         scanf("%d",&ch);
27         switch(ch)
28         {
29             case 1: printf("\nenter the priority number\n");
30                     scanf("%d",&pr);
31                     if(pr>0 && pr<4)
32                         pqinsert(pr-1);
33                     else
34                         printf("\n only 3 priority exists 1 2 3\n");
35                     break;
36             case 2: pqdelete();
37                     break;
38             case 3: display();
```

```
(globals)
m_p_queue.cpp
36 case 2: pqdelete();
37     break;
38 case 3: display();
39     break;
40 case 4: exit(0);
41 }
42 }
43 getch();
44 return 0;
45 }
46 pqinsert(int pr)
47 {
48     if(rear[pr]==N-1)
49         printf("\n Queue overflow\n");
50     else
51     {
52         printf("\n enter the item\n");
53         scanf("%d",&item);
54         rear[pr]++;
55         queue[pr][rear[pr]]=item;
56     }
57     return 0;
58 }
59 pqdelete()
60 {
61     int i;
62     for(i=0;i<3;i++)
63     {
64         if(rear[i]==front[i]-1)
65             printf("\n queue empty\n");
66         else
67         {
68             printf("deleted item is %d of queue %d\n",queue[i][front[i]],i+1);
69             front[i]++;
70             return 0;
71         }
72     }
73 }
```

```
m_p_queue.cpp
53 scanf ("%d", &item);
54 rear[pr]++;
55 queue[pr][rear[pr]] = item;
56 }
57 return 0;
58 }
59 pdelete()
60 {
61 int i;
62 for (i=0; i<3; i++)
63 {
64 if (rear[i] == front[i] - 1)
65 printf ("\n queue empty\n");
66 else
67 {
68 printf ("deleted item is %d of queue %d\n", queue[i][front[i]], i+1);
69 front[i]++;
70 return 0;
71 }
72 }
73 }
74 display()
75 {
76 int i, j;
77 for (i=0; i<3; i++)
78 {
79 if (rear[i] == front[i] - 1)
80 printf ("\n queue empty %d\n", i+1);
81 else
82 {
83 printf ("\nQUEUE %d: ", i+1);
84 for (j=front[i]; j<=rear[i]; j++)
85 printf ("%d\t", queue[i][j]);
86 }
87 }
88 return 0;
89 }
```

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\multiple priority queue\m\_p\_queue.exe

PRIORITY QUEUE

\*\*\*\*\*

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

1

enter the priority number

32

only 3 priority exists 1 2 3

PRIORITY QUEUE

\*\*\*\*\*

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

2

queue empty

queue empty

queue empty

PRIORITY QUEUE

\*\*\*\*\*

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

1



Type here to search



ENG

16:13

07-11-2020



C:\Users\sohan\Desktop\C Programs\Data Structures Lab\multiple priority queue\m\_p\_queue.exe

enter the choice

1

enter the priority number

2

enter the item

45

PRIORITY QUEUE

\*\*\*\*\*

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

1

enter the priority number

3

enter the item

65

PRIORITY QUEUE

\*\*\*\*\*

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

2

queue empty

deleted item is 45 of queue 2

PRIORITY QUEUE

\*\*\*\*\*

1:PQinsert

2:PQdelete



Type here to search



ENG

16:13

07-11-2020



C:\Users\sohan\Desktop\C Programs\Data Structures Lab\multiple priority queue\m\_p\_queue.exe

```
4:Exit
enter the choice
2

queue empty
deleted item is 45 of queue 2
PRIORITY QUEUE
*****

1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit

enter the choice
3

queue empty 1
queue empty 2

QUEUE 3:65      PRIORITY QUEUE
*****

1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit

enter the choice
4

-----
Process exited after 37.11 seconds with return value 0
Press any key to continue . . .
```



Type here to search

