

```
1  /*****
2
3      Online C Compiler.
4      Code, Compile, Run and Debug C program online.
5      Write your code in this editor and press "Run" button to compile and e
6
```

```
7  *****/
```

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



ain.c

```

46 }
47 }
48 }
49 void pqinsert(int pr)
50 {
51     if(rear[pr]==N-1)
52         printf("\n Queue overflow\n");
53     else
54     {
55         printf("\nenter the item\n");
56         scanf("%d",&item);
57         rear[pr]++;
58         queue[pr][rear[pr]]=item;
59     }
60     return;
61 }
62 void pqdelete()
63 {
64     int i;
65     for(i=0;i<3;i++)
66     {
67         if(rear[i]==front[i]-1)
68             printf("\nqueue empty\n");
69         else
70         {
71             printf("deleted item is %d of queue %d\n",queue[i][front[i]],i+1);
72             front[i]++;
73             return;
74         }
75     }
76 }
77 void display()
78 {
79     int i,j;
80     for(i=0;i<3;i++)
81     {
82         if(rear[i]==front[i]-1)
83             printf("\nqueue empty %d\n",i+1);
84         else
85         {
86             printf("\nQUEUE %d:",i+1);
87             for(j=front[i];j<=rear[i];j++)
88                 printf("%d\t",queue[i][j]);
89         }
90     }

```



```

49 void pqinsert(int pr)
50 {
51     if(rear[pr]==N-1)
52         printf("\n Queue overflow\n");
53     else
54     {
55         printf("\n enter the item\n");
56         scanf("%d",&item);
57         rear[pr]++;
58         queue[pr][rear[pr]]=item;
59     }
60     return;
61 }
62 void pqdelete()
63 {
64     int i;
65     for(i=0;i<3;i++)
66     {
67         if(rear[i]==front[i]-1)
68             printf("\nqueue empty\n");
69         else
70         {
71             printf("deleted item is %d of queue %d\n",queue[i][front[i]],i+1);
72             front[i]++;
73             return;
74         }
75     }
76 }
77 void display()
78 {
79     int i,j;
80     for(i=0;i<3;i++)
81     {
82         if(rear[i]==front[i]-1)
83             printf("\nqueue empty %d\n",i+1);
84         else
85         {
86             printf("\nQUEUE %d:",i+1);
87             for(j=front[i];j<=rear[i];j++)
88                 printf("%d\t",queue[i][j]);
89         }
90     }
91     return;
92 }
93

```

PRIORITY QUEUE

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

1

enter the priority number

1

enter the item

10

PRIORITY QUEUE

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

2

deleted item is 10 of queue 1

PRIORITY QUEUE

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

2



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

enter the priority number

1

enter the item

10

PRIORITY QUEUE

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

□



2:PQdelete

3:PQdisplay

4:Exit

enter the choice

1

enter the priority number

2

enter the item

20

PRIORITY QUEUE

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

1

enter the priority number

3

enter the item

30

PRIORITY QUEUE

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

□



1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

3

QUEUE 1:10

QUEUE 2:20

QUEUE 3:30 PRIORITY QUEUE

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

1

enter the priority number

1

enter the item

40

PRIORITY QUEUE

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

□



enter the item

40

PRIORITY QUEUE

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

1

enter the priority number

1

Queue overflow

PRIORITY QUEUE

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

50

PRIORITY QUEUE

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

□



2:PQdelete

3:PQdisplay

4:Exit

enter the choice

1

enter the priority number

3

enter the item

50

PRIORITY QUEUE

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

1

enter the priority number

2

enter the item

60

PRIORITY QUEUE

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice



PRIORITY QUEUE

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

3

QUEUE 1:10 40

QUEUE 2:20 60 70

QUEUE 3:30 50 PRIORITY QUEUE

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

1

enter the priority number

2

Queue overflow

PRIORITY QUEUE

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

□



1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

1

enter the priority number

3

enter the item

90

PRIORITY QUEUE

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

3

QUEUE 1:10 40

QUEUE 2:20 60 70

QUEUE 3:30 50 90 PRIORITY QUEUE

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

□





Multiple priority queue

```
#include <stdio.h>
#include <stdlib.h>
#define N 3
int queue[N][N];
int front[N] = {0, 0, 0};
int rear[N] = {-1, -1, -1};
void pqInsert(int pr);
void pqdelete();
void display();
int ele, pr;
void main()
{
    int ch;
    while (1)
    {
        printf("Priority Queue |n");
        printf("*****|n");
        printf("|n|t 1: pqInsert |n");
        printf("|n|t 2: pqdelete |n");
        printf("|n|t 3: pqdisplay |n");
        printf("|n|t 4: Exit |n");
        printf("|n enter the choice |n");
        scanf("%d", &ch);
        switch (ch)
        {
            case 1: printf("|n enter the priority number |n");
                    scanf("%d", &pr);
                    if (pr > 0 && pr < 4)
                        pqInsert(pr-1);
                    else
                        printf("|n only 3 priority exists 1 2 3 |n");
                        break;
        }
    }
}
```




```
case 2: pq delete();  
    break;  
case 3: display();  
    break;  
case 4: exit(0).  
}  
}  
}  
void pqinsert(int px)  
{  
    if (rear[pr] == N-1)  
        printf("\n Queue overflow\n");  
    else  
    {  
        printf("\nEnter the item\n");  
        scanf("%d", &item);  
        rear[pr]++;  
        queue[pr][rear[pr]] = item;  
    }  
    return;  
}  
void pqdelete()  
{  
    int i;  
    for (i=0; i<3; i++)  
    {  
        if (rear[i] == front[i]-1)  
            printf("\n queue empty\n");  
        else  
        {  
            printf("deleted item is %d of queue t.d\n",  
                queue[i][front[i]], i+1);  
            front[i]++;  
        }  
    }  
    return;  
}
```




```
void display()
{
    int i, j;
    for (i = 0; i < 3; i++)
    {
        if (rear[i] == front[i] - 1)
            printf("\n queue empty %d \n", i + 1);
        else
        {
            printf("\n Queue %d :", i + 1);
            for (j = front[i]; j <= rear[i]; j++)
                printf("%d\t", queue[i][j]);
        }
    }
    return;
}
```