

Multiple priority queue:

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<conio.h>
4  #define N 3
5  void pqinsert(int);
6  void pqdelete();
7  void display();
8  int queue[3][N];
9  int front[3]={0,0,0};
10 int rear[3]={-1,-1,-1};
11 int item,pr;
12 int main()
13 {
14     int ch;
15     while(1)
16     {
17         printf("\nPRIORITY QUEUE\n");
18         printf("*****\n");
19         printf("\n\t1:PQinsert\n");
20         printf("\n\t2:PQdelete\n");
21         printf("\n\t3:PQdisplay\n");
22         printf("\n\t4:Exit\n");
23         printf("\nenter the choice\n");
24         scanf("%d",&ch);
25         switch(ch)
26         {
27             case 1:printf("\n enter the priority number\n");
28                     scanf("%d",&pr);
29                     if(pr>0 && pr<4)
30                         pqinsert(pr-1);
31
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();
39                     break;
40             case 4:exit(0);
41         }
42         getch();
43     }
44     void pqinsert(int pr)
45     {
46         if(rear[pr]==N-1)
47             printf("\n Queue overflow\n");
48         else
49         {
50             printf("\n enter the item\n");
51             scanf("%d",&item);
52             rear[pr]++;
53             queue[pr][rear[pr]]=item;
54         }
55     }
56     void pqdelete()
57     {
58         int i;
59         for(i=0;i<3;i++)
```

```

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

```

```

PRIORITY QUEUE
*****

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

enter the choice
1

enter the priority number
2

enter the item
6

PRIORITY QUEUE
*****

1:PQinsert
2:PQdelete
3:PQdisplay

```

```
3:PQdisplay

4:Exit

enter the choice
1

enter the priority number
3

enter the item
8

PRIORITY QUEUE
*****

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice
1

enter the priority number
1
```

```
enter the priority number
1

enter the item
4

PRIORITY QUEUE
*****

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice
3

QUEUE 1:4
QUEUE 2:6
QUEUE 3:8
PRIORITY QUEUE
*****

1:PQinsert

2:PQdelete
```

```
2:PQdelete

3:PQdisplay

4:Exit

enter the choice
2
deleted item is 4 of queue 1
deleted item is 6 of queue 2
deleted item is 8 of queue 3

PRIORITY QUEUE
*****

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice
3

queue empty 1

queue empty 2
```

```
queue empty 2

queue empty 3

PRIORITY QUEUE
*****

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice
4

...Program finished with exit code 0
Press ENTER to exit console.
```

PRIORITY QUEUE:

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  struct node
4  {
5      int priority;
6      int info;
7      struct node *link;
8  }*front=NULL;
9  void insert(int item, int item_priority);
10 int del();
11 void display();
12 int isEmpty();
13 int main()
14 {
15     int choice,item,item_priority;
16     while(1)
17     {
18         printf("\n1.Insert\n");
19         printf("2.Delete\n");
20         printf("3.Display\n");
21         printf("4.Quit\n");
22         printf("\nEnter your choice : ");
23         scanf("%d", &choice);
24         switch(choice)
25         {
26             case 1:
27                 printf("\nInput the item to be added in the queue : ");
28                 scanf("%d",&item);
29                 printf("\nEnter its priority : ");
30                 scanf("%d",&item_priority);
31                 insert(item, item_priority);

```

```

32                 break;
33             case 2:
34                 printf("\nDeleted item is %d\n",del());
35                 break;
36             case 3:
37                 display();
38                 break;
39             case 4:
40                 exit(1);
41             default :
42                 printf("\nWrong choice\n");
43         }
44     }
45     return 0;
46 }
47 void insert(int item,int item_priority)
48 {
49     struct node *tmp,*p;
50
51     tmp=(struct node *)malloc(sizeof(struct node));
52     if(tmp==NULL)
53     {
54         printf("\nMemory not available\n");
55         return;
56     }
57     tmp->info=item;
58     tmp->priority=item_priority;
59
60     if( isEmpty() || item_priority < front->priority )

```

```

61     if( isEmpty() || item_priority < front->priority )
62     {
63         tmp->link=front;
64         front=tmp;
65     }
66     else
67     {
68         p = front;
69         while( p->link!=NULL && p->link->priority<=item_priority )
70             p=p->link;
71         tmp->link=p->link;
72         p->link=tmp;
73     }
74 }
75 int del()
76 {
77     struct node *tmp;
78     int item;
79     if( isEmpty() )
80     {
81         printf("\nQueue Underflow\n");
82         exit(1);
83     }
84     else
85     {
86         tmp=front;
87         item=tmp->info;
88         front=front->link;
89         free(tmp);
90     }
91     return item;
92 }

```

```

93 int isEmpty()
94 {
95     if( front == NULL )
96         return 1;
97     else
98         return 0;
99 }
100 void display()
101 {
102     struct node *ptr;
103     ptr=front;
104     if( isEmpty() )
105         printf("\nQueue is empty\n");
106     else
107     {
108         printf("\nQueue is :\n");
109         printf("\nPriority      Item\n");
110         while(ptr!=NULL)
111         {
112             printf("%5d      %5d\n",ptr->priority,ptr->info);
113             ptr=ptr->link;
114         }
115     }
116 }
117

```

```
1.Insert
2.Delete
3.Display
4.Quit

Enter your choice : 1

Input the item to be added in the queue : 2

Enter its priority : 3

1.Insert
2.Delete
3.Display
4.Quit

Enter your choice : 1

Input the item to be added in the queue : 5

Enter its priority : 2

1.Insert
2.Delete
3.Display
4.Quit
```

```
4.Quit

Enter your choice : 1

Input the item to be added in the queue : 4

Enter its priority : 2

1.Insert
2.Delete
3.Display
4.Quit

Enter your choice : 1

Input the item to be added in the queue : 3

Enter its priority : 3

1.Insert
2.Delete
3.Display
4.Quit

Enter your choice : 3

Queue is :
```

Queue is :

Priority	Item
2	5
2	4
3	2
3	3

1.Insert
2.Delete
3.Display
4.Quit

Enter your choice : 2

Deleted item is 5

1.Insert
2.Delete
3.Display
4.Quit

Enter your choice : 1

Input the item to be added in the queue : 6

Enter its priority : 1

Enter its priority : 1

1.Insert
2.Delete
3.Display
4.Quit

Enter your choice : 3

Queue is :

Priority	Item
1	6
2	4
3	2
3	3

1.Insert
2.Delete
3.Display
4.Quit

Enter your choice : 2

Deleted item is 6

1.Insert
2.Delete

3.Display
4.Quit

Enter your choice : 2

Deleted item is 4

1.Insert
2.Delete
3.Display
4.Quit

Enter your choice : 3


```

Queue is :

Priority      Item
   3           2
   3           3

1.Insert
2.Delete
3.Display
4.Quit

Enter your choice : 4

...Program finished with exit code 1
Press ENTER to exit console.

```

Deque:

```

1  #include<stdio.h>
2  #include<conio.h>
3  #include<stdlib.h>
4  #define qsize 5
5  int f=0,r=-1,ch;
6  int item,q[10];
7
8
9  int isfull()
10 {
11     return(r==qsize-1)?1:0;
12 }
13 int isempty()
14 {
15     return(f>r)?1:0;
16 }
17 void insert_rear()
18 {
19     if(isfull())
20     {
21         printf("queue overflow\n");
22         return;
23     }
24     r=r+1;
25     q[r]=item;
26 }
27 void delete_front()
28 {
29     if(isempty())
30     {
31         printf("queue empty\n");
32         return;
33     }
34     printf("item deleted is %d\n",q[(f)++]);
35     if(f>r)
36     {
37         f=0;
38         r=-1;
39     }
40 }
41 void insert_front()
42 {
43     if(f!=0)
44     {
45         f=f-1;
46         q[f]=item;
47         return;
48     }
49     else if((f==0)&&(r==1))
50     {
51         q[++(r)]=item;
52         return;
53     }
54     else
55         printf("insertion not possible\n");
56 }
57 void delete_rear()
58 {
59     if(isempty())
60     {
61         printf("queue is empty\n");
62         return;
63     }
64     r--;
65 }

```

```

61     printf("queue is empty\n");
62     return;
63 }
64     printf("item deleted is %d\n",q[(r)--]);
65     if(f>r)
66     {
67         f=0;
68         r=-1;
69     }
70 }
71 void display()
72 {
73     int i;
74     if(isempty())
75     {
76         printf("queue empty\n");
77         return;
78     }
79     for(i=f;i<=r;i++)
80     printf("%d\n",q[i]);
81 }
82 void main()
83 {for(;;)
84 {
85     printf("1.insert_rear\n2.insert_front\n3.delete_rear\n4.delete_front\n5.display\n6.exit\n");
86     printf("enter choice\n");
87     scanf("%d",&ch);
88     switch(ch)
89     {
90         case 1:printf("enter the item\n");
91         scanf("%d",&item);

```

```

90         case 1:printf("enter the item\n");
91         scanf("%d",&item);
92         insert_rear();
93         break;
94         case 2:printf("enter the item\n");
95         scanf("%d",&item);
96         insert_front();
97         break;
98         case 3:delete_rear();
99         break;
100        case 4:delete_front();
101        break;
102        case 5:display();
103        break;
104        default:exit(0);
105    }}}
106

```

```
enter the item
6
insertion not possible
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
9
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
2
9
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
```

```
enter choice
4
item deleted is 2
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 9
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
```

```
enter choice
5
queue empty
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
6

...Program finished with exit code 0
Press ENTER to exit console.
```