

## **PRACTICE PROGRAM MULTIPLE PRIORITY QUEUE(Ascending and Descending)**

### **1.MULTIPLE PQ**

```
#include<stdio.h>

#define N 3

int queue[3][N];

int front[3]={0,0,0};

int rear[3]={-1,-1,-1};

int item,pr;

void main()

{

int ch;

while(1)

{

printf("\nPRIORITY QUEUE\n");

printf("*****\n");

printf("\n1:PQinsert\n");

printf("\n2:PQdelete\n");

printf("\n3:PQdisplay\n");
```

```
printf("\n4:Exit\n");  
printf("\nenter the choice\n");  
scanf("%d",&ch);  
switch(ch)  
{  
    case 1:printf("\nenter the priority number\n");  
            scanf("%d",&pr);  
            if(pr>0 && pr<4)  
                pqinsert(pr-1);  
            else  
                printf("\nonly 3 priority exists 1 2 3\n");  
            break;  
    case 2:pqdelete();  
            break;  
    case 3:display();  
            break;  
    case 4:exit(0);  
}
```

```
}  
  
}  
  
pqinsert(int pr)  
{  
    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;  
}  
  
pqdelete()  
{  
    int i;
```

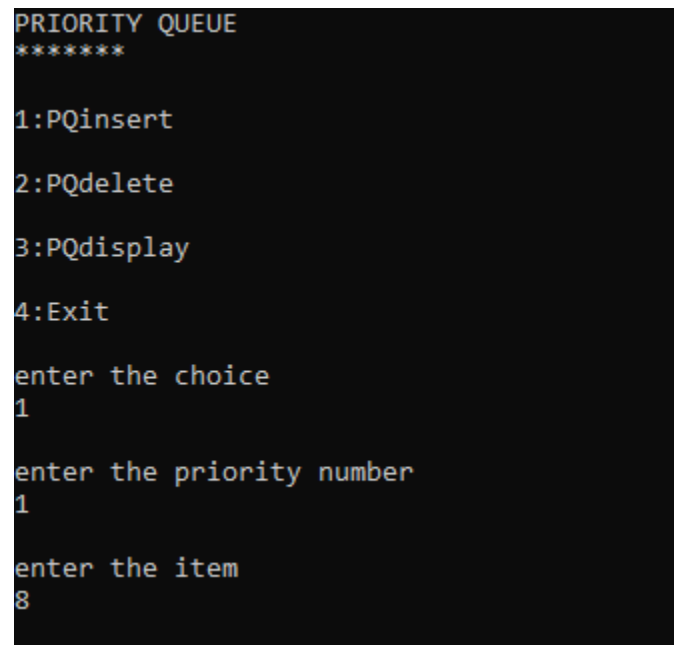
```
for(i=0;i<3;i++)
{
    if(rear[i]==front[i]-1)
        printf("\nqueue empty\n");
    else
    {
        printf("deleted item is %d of queue\n",queue[i][front[i]],i+1);
        front[i]++;
        return;
    }
}

display()
{
    int i,j;
    for(i=0;i<3;i++)
    {
        if(rear[i]==front[i]-1)
```

```
    printf("\nqueue empty %d\n",i+1);
else
{
    printf("\nQUEUE %d:",i+1);
    for(j=front[i];j<=rear[i];j++)
        printf("%d\t",queue[i][j]);
}
}

return;
}
```

## OUTPUT:



```
PRIORITY QUEUE
*****
1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit
enter the choice
1
enter the priority number
1
enter the item
8
```

PRIORITY QUEUE  
\*\*\*\*\*

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

1

enter the priority number

2

enter the item

4

PRIORITY QUEUE  
\*\*\*\*\*

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

1

enter the priority number

3

enter the item

7

PRIORITY QUEUE  
\*\*\*\*\*

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

3

QUEUE 1:8

QUEUE 2:4

QUEUE 3:7

PRIORITY QUEUE  
\*\*\*\*\*

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice

2

deleted item is 8 of queue 1

## 2.ASCENDING PQ

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#define PS 5
```

```
int f=0,r=-1,item,q[10];
```

```
//ascending
```

```
void insert()
```

```
{
```

```
    if(r == PS - 1)
```

```
        printf("Queue Overflow!\n");
```

```
    q[++r] = item;
```

```
    sort();
```

```
}
```

```
void sort()
```

```
{
```



```
int i,key,j;
for (i=1;i<=r;i++)
{
    key = q[i];
    j = i-1;
    while (j >= 0 && q[j]>key)
    {
        q[j + 1] = q[j];
        j = j - 1;
    }
    q[j + 1] = key;
}

}

int delete()
```

```
{  
    if(r == -1)  
        printf("Queue Underflow\n");  
    else  
        return q[f++];  
  
}  
  
void display()  
{  
    int i;  
    if(r == -1)  
        printf("\n QUEUE UNDERFLOW!!\n");  
    printf("\nThe elements of the queue are:\n");  
    for(i=f;i<=r;i++)  
    {
```

```
        printf("%d\n",q[i]);
    }
}
void main()
{
    int ch;
    for(;;)
    {
        printf("Enter
\n1.insertion\n2.deletion\n3.display\n4.exit\n"
);
        scanf("%d",&ch);
        switch(ch)
        {
```

```
case 1: printf("Enter the item to be  
inserted\n");
```

```
scanf("%d",&item);
```

```
insert(item);
```

```
break;
```

```
case 2: item = delete();
```

```
if(item == -1)
```

```
printf("Queue underflow\n");
```

```
else
```

```
printf("\nItem popped =  
%d\n",item);
```

```
break;
```

```
case 3: display();
```

```
break;
```

```
default: exit(0);
```

}

}

}

## OUTPUT:

```
Enter
1.insertion
2.deletion
3.display
4.exit
1
Enter the item to be inserted
1
Enter
1.insertion
2.deletion
3.display
4.exit
1
Enter the item to be inserted
44
Enter
1.insertion
2.deletion
3.display
4.exit
1
Enter the item to be inserted
20
Enter
1.insertion
2.deletion
3.display
4.exit
1
Enter the item to be inserted
70
Enter
1.insertion
2.deletion
3.display
4.exit
3

The elements of the queue are:
1
20
44
70
```

The elements of the queue are:

1

20

44

70

Enter

1.insertion

2.deletion

3.display

4.exit

2

Item popped = 1

Enter

1.insertion

2.deletion

3.display

4.exit

2

Item popped = 20

Enter

1.insertion

2.deletion

3.display

4.exit

2

Item popped = 44

### 3.DESCENDING PQ

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#define PS 5
```

```
int f=0,r=-1,item,q[10];
```

```
//descending
```

```
void insert()
```

```
{
```

```
    if(r == PS - 1)
```

```
        printf("Queue Overflow!\n");
```

```
    q[++r] = item;
```

```
    sort();
```

```
}
```

```
void sort()
```

```
{
```



```
int i,key,j;
for (i=1;i<=r;i++)
{
    key = q[i];
    j = i-1;
    while (j >= 0 && q[j]<key)
    {
        q[j + 1] = q[j];
        j = j - 1;
    }
    q[j + 1] = key;
}

}

int delete()
```

```
{  
    if(r == -1)  
        printf("Queue Underflow\n");  
    else  
        return q[f++];  
  
}  
  
void display()  
{  
    int i;  
    if(r == -1)  
        printf("\nQUEUE UNDERFLOW!!\n");  
    printf("\nThe elements of the queue are:\n");  
    for(i=f;i<=r;i++)  
    {
```

```
        printf("%d\n",q[i]);
    }
}
void main()
{
    int ch;
    for(;;)
    {
        printf("Enter
\n1.insertion\n2.deletion\n3.display\n4.exit\n"
);
        scanf("%d",&ch);
        switch(ch)
        {
```

```
case 1: printf("Enter the item to be  
inserted\n");
```

```
scanf("%d",&item);
```

```
insert(item);
```

```
break;
```

```
case 2: item = delete();
```

```
if(item == -1)
```

```
printf("Queue underflow\n");
```

```
else
```

```
printf("\nItem popped =  
%d\n",item);
```

```
break;
```

```
case 3: display();
```

```
break;
```

```
default: exit(0);
```

}

}

}

## OUTPUT:

```
Enter
1.insertion
2.deletion
3.display
4.exit
1
Enter the item to be inserted
20
Enter
1.insertion
2.deletion
3.display
4.exit
1
Enter the item to be inserted
10
Enter
1.insertion
2.deletion
3.display
4.exit
1
Enter the item to be inserted
100
Enter
1.insertion
2.deletion
3.display
4.exit
1
Enter the item to be inserted
3
Enter
1.insertion
2.deletion
3.display
4.exit
3

The elements of the queue are:
100
20
10
3
```

```
Enter
1.insertion
2.deletion
3.display
4.exit
2

Item popped = 100
Enter
1.insertion
2.deletion
3.display
4.exit
2

Item popped = 20
Enter
1.insertion
2.deletion
3.display
4.exit
2

Item popped = 10
```