main.c

```c
/*ASCENDING PRIORITY QUEUE*/
#include<stdio.h>
#include<stdlib.h>
#define QUE_SIZE 5
int item,rear=-1, q[QUE_SIZE],count=0;
void insertrear()
{
    if(rear== QUE_SIZE-1)
    {
        printf("Queue overflow \n");
        return ;
    }

    rear=rear+1;
    q[rear]= item;
    count++;

}

int deleteasc()
{
    int small=99;
    int spos=-1;
    if(count==0)
        {return -1;
    }
    for(int i=0;i<QUE_SIZE;i++)
    {
        if(q[i]<small)
        {
          small=q[i];
          spos=i;
        }

    }

    q[spos]=99;
    count=count-1;
    return small;
```

main.c

```
32          }
33
34      }
35
36      q[spos]=99;
37      count=count-1;
38      return small;
39  }
40
41  void display()
42  {
43      int i;
44      if(count==0)
45      {
46          printf("Queue empty \n");
47          return;
48      }
49
50
51      printf("Contents of queue \n");
52      for(i=0; i<QUE_SIZE; i++)
53      {
54         if(q[i]==99)
55          continue;
56         else
57          printf("%d ",q[i]);
58      }
59  }
60
61  void main()
62  {
63      int choice ;
64      for(;;)
65      {
66          printf("\nENTER 1.insert rear 2.delete  3.display \n");
67          scanf("%d", &choice);
68          switch(choice)
69          {
70              case 1: printf("Enter the item\n");
71                      scanf("%d", &item);
```

main.c

```
47              return;
48          }
49
50
51          printf("Contents of queue \n");
52          for(i=0; i<QUE_SIZE; i++)
53          {
54              if(q[i]==99)
55               continue;
56              else
57               printf("%d ",q[i]);
58          }
59      }
60
61      void main()
62      {
63          int choice ;
64          for(;;)
65          {
66              printf("\nENTER 1.insert rear 2.delete  3.display \n");
67              scanf("%d", &choice);
68              switch(choice)
69              {
70                  case 1: printf("Enter the item\n");
71                          scanf("%d", &item);
72                          insertrear();
73                          break;
74                  case 2: item= deleteasc();
75                          if(item == -1)
76                              printf("Queue empty \n");
77                          else
78                              printf("Item deleted is  %d", item);
79                          break;
80                  case 3: display();
81                          break;
82                  default: exit(0);
83              }
84          }
85      }
86
```

```
ENTER 1.insert rear 2.delete  3.display
1
Enter the item
11

ENTER 1.insert rear 2.delete  3.display
1
Enter the item
30

ENTER 1.insert rear 2.delete  3.display
1
Enter the item
55

ENTER 1.insert rear 2.delete  3.display
1
Enter the item
10

ENTER 1.insert rear 2.delete  3.display
1
Enter the item
90

ENTER 1.insert rear 2.delete  3.display
3
Contents of queue
11 30 55 10 90
ENTER 1.insert rear 2.delete  3.display
2
```

```
1
Enter the item
90

ENTER 1.insert rear 2.delete  3.display
3
Contents of queue
11 30 55 10 90
ENTER 1.insert rear 2.delete  3.display
2
Item deleted is  10
ENTER 1.insert rear 2.delete  3.display
2
Item deleted is  11
ENTER 1.insert rear 2.delete  3.display
2
Item deleted is  30
ENTER 1.insert rear 2.delete  3.display
2
Item deleted is  55
ENTER 1.insert rear 2.delete  3.display
2
Item deleted is  90
ENTER 1.insert rear 2.delete  3.display
2
Queue empty

ENTER 1.insert rear 2.delete  3.display
^C

...Program finished with exit code 130
Press ENTER to exit console.
```

main.c

```c
/*DESCENDING PRIORITY*/
#include<stdio.h>
#include<stdlib.h>
#define QUE_SIZE 3
int item,rear=-1, q[QUE_SIZE],count=0;
void insertrear()
{
    if(rear== QUE_SIZE-1)
    {
        printf("Queue overflow \n");
        return ;
    }

    rear=rear+1;
    q[rear]= item;
    count++;

}

int deletedesc()
{
    int largest=0;
    int spos=-1;
    if(count==0)
        {return -1;
    }
    for(int i=0;i<QUE_SIZE;i++)
    {
        if(q[i]>largest)
        {
          largest=q[i];
          spos=i;
        }

    }

    q[spos]=0;
    count=count-1;
    return largest;
```

```
            spos=i;
        }

    }

    q[spos]=0;
    count=count-1;
    return largest;
}

void display()
{
    int i;
    if(count==0)
    {
        printf("Queue empty \n");
        return;
    }


    printf("Contents of queue \n");
    for(i=0; i<QUE_SIZE; i++)
    {
       if(q[i]==0)
        continue;
       else
        printf("%d ",q[i]);
    }
}

void main()
{
    int choice ;
    for(;;)
    {
        printf("\nENTER 1.insert rear 2.delete  3.display\n");
        scanf("%d", &choice);
        switch(choice)
        {
            case 1: printf("Enter the item \n");
```

```
47          return;
48      }
49
50
51      printf("Contents of queue \n");
52      for(i=0; i<QUE_SIZE; i++)
53      {
54          if(q[i]==0)
55           continue;
56          else
57           printf("%d ",q[i]);
58      }
59  }
60
61  void main()
62  {
63      int choice ;
64      for(;;)
65      {
66          printf("\nENTER 1.insert rear 2.delete  3.display\n");
67          scanf("%d", &choice);
68          switch(choice)
69          {
70              case 1: printf("Enter the item \n");
71                      scanf("%d", &item);
72                      insertrear();
73                      break;
74              case 2: item= deletedesc();
75                      if(item == -1)
76                          printf("Queue empty \n");
77                      else
78                          printf("Item deleted is %d", item);
79                      break;
80              case 3: display();
81                      break;
82              default: exit(0);
83          }
84      }
85  }
86
```

```
ENTER 1.insert rear 2.delete  3.display
1
Enter the item
30

ENTER 1.insert rear 2.delete  3.display
1
Enter the item
14

ENTER 1.insert rear 2.delete  3.display
1
Enter the item
80

ENTER 1.insert rear 2.delete  3.display
3
Contents of queue
30 14 80
ENTER 1.insert rear 2.delete  3.display
2
Item deleted is 80
ENTER 1.insert rear 2.delete  3.display
2
Item deleted is 30
ENTER 1.insert rear 2.delete  3.display
2
Item deleted is 14
ENTER 1.insert rear 2.delete  3.display
2
Queue empty
```

```c
/*MULTIPLE PRIORITY QUEUE*/
#include<stdio.h>
#include<stdlib.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("PRIORITY QUEUE:");
printf("\n1:PQinsert");
printf("\n2:PQdelete");
printf("\n3:PQdisplay");
printf("\n4:Exit");
printf("\nenter the choice\n");
scanf("%d",&ch);
switch(ch)
{
 case 1:printf("enter the priority number\n");
        scanf("%d",&pr);
        if(pr>0 && pr<4)
        pqinsert(pr-1);
        else
        printf("only 3 priority exists 1 2 3\n");
        break;
case 2:pqdelete();
       break;
case 3:display();
       break;
case 4:exit(0);
}
}
}
pqinsert(int pr)
{
```

```
}
pqinsert(int pr)
{
 if(rear[pr]==N-1)
 printf("Queue overflow\n");
 else
 {
 printf("enter 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("Queue empty\n");
  else
  {
  printf("deleted item is %d of queue %d\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("Queue%d empty\n",i+1);
else
  {
  printf("\nQUEUE %d:",i+1);
```

```
 printf("enter 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("Queue empty\n");
  else
  {
  printf("deleted item is %d of queue %d\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("Queue%d empty\n",i+1);
else
  {
  printf("\nQUEUE %d:",i+1);
  for(j=front[i];j<=rear[i];j++)
    printf("%d\t",queue[i][j]);
  }
}
  return;
}
```

```
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
1
enter the priority number
1
enter the item
20
PRIORITY QUEUE:
1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit
enter the choice
1
enter the priority number
1
enter the item
30
PRIORITY QUEUE:
1:PQinsert
2:PQdelete
3:PQdisplay
```

```
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
1
enter the priority number
2
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
1
enter the priority number
2
```

```
1
enter the priority number
2
enter the item
60
PRIORITY QUEUE:
1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit
enter the choice
1
enter the priority number
2
enter the item
80
PRIORITY QUEUE:
1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit
enter the choice
3

QUEUE 1:20      30
QUEUE 2:50      60      80      Queue3 empty
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.
```