doubleEndedQueue.c | *priorityQue.c | mpq.c | ascendingPQ.c | *descendingPQ.c

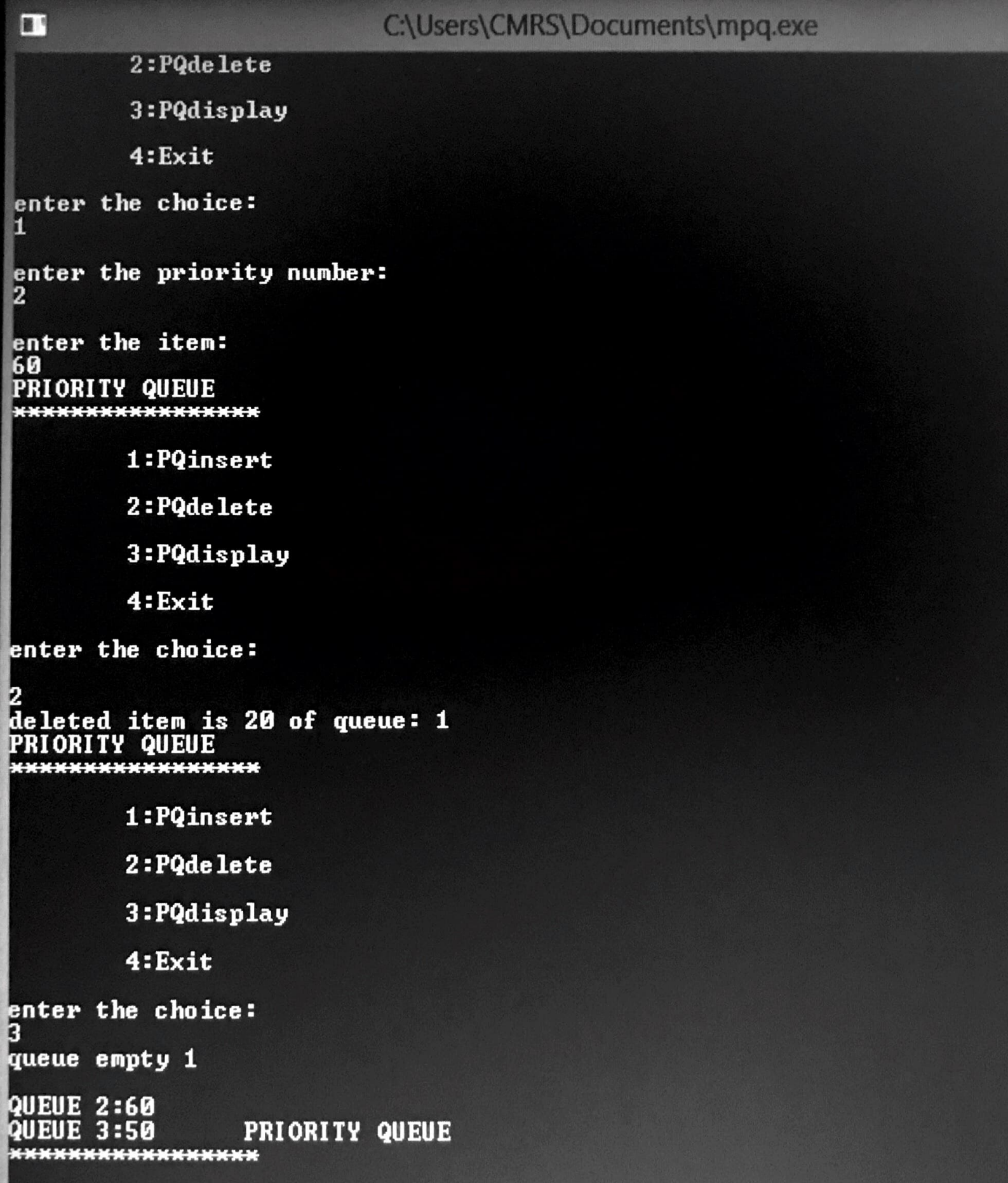
```c
#include<stdio.h>
#include<conio.h>
#define N 3
int queue[3][N];
int front[3]={0,0,0};
int rear[3]={-1,-1,-1};
int item,pr;
void main()
{
int ch;
while(1)
{
printf("PRIORITY QUEUE\n");
printf("******************\n");
printf("\n\t1:PQinsert\n");
printf("\n\t2:PQdelete\n");
printf("\n\t3:PQdisplay\n");
printf("\n\t4: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("\only 3 priority exists 1 2 3\n");
        break;
case 2:pqdelete();
       break;
case 3:display();
       break;
case 4:exit(0);
}
}
```

```
26              pqinsert(pr-1);
27              else
28              printf("\only 3 priority exists 1 2 3\n");
29              break;
30      case 2:pqdelete();
31             break;
32      case 3:display();
33             break;
34      case 4:exit(0);
35      }
36      }
37      }
38      pqinsert(int pr)
39      {
40       if(rear[pr]==N-1)
41       printf("\n Queue overflow:\n");
42       else
43       {
44       printf("\nenter the item:\n");
45       scanf("%d",&item);
46       rear[pr]++;
47       queue[pr][rear[pr]]=item;
48       }
49       return;
50      }
51      pqdelete()
52      {
53      int i;
54      for(i=0;i<3;i++)
55      {
56        if(rear[i]==front[i]-1)
57        printf("\queue empty\n");
58        else
59        {
60        printf("deleted item is %d of queue: %d\n",queue[i][front[i]],i+1);
61        front[i]++;
```

```
48      }
49      return;
50    }
51    pqdelete()
52    {
53    int i;
54    for(i=0;i<3;i++)
55    {
56      if(rear[i]==front[i]-1)
57      printf("\queue empty\n");
58      else
59      {
60      printf("deleted item is %d of queue: %d\n",queue[i][front[i]],i+1);
61      front[i]++;
62      return;
63      }
64    }
65    }
66    display()
67    {
68    int i,j;
69    for(i=0;i<3;i++)
70    {
71    if(rear[i]==front[i]-1)
72      printf("\queue empty %d\n",i+1);
73    else
74      {
75      printf("\nQUEUE %d:",i+1);
76      for(j=front[i];j<=rear[i];j++)
77        printf("%d\t",queue[i][j]);
78      }
79    }
80      return;
81    }
82
```

C:\Users\CMRS\Documents\mpq.exe

```
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:
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
```


C:\Users\CMRS\Documents\mpq.exe
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:
2
deleted item is 20 of queue: 1
PRIORITY QUEUE
1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit
enter the choice:
3
queue empty 1
QUEUE 2:60
QUEUE 3:50
PRIORITY QUEUE

doubleEndedQueue.c | *priorityQue.c | mpq.c | ascendingPQ.c | *descendingPQ.c

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define MAX 4

int pq[MAX];
int count = 0;
int d = 0;

void insert(int data){
 int i = 0;
 if(count==MAX)
 {
 printf("Queue overflow\n");
 return;
 }
 // if queue is empty, insert the data
 if(count == 0){
 pq[count++] = data;
 }else{
 // start from the right end of the queue
 for(i = count - 1; i >= 0; i-- ){
 //if data is smaller shift right
 if(data<pq[i]){
 pq[i+1] = pq[i];
 }else{
 break;
 }
 }

 pq[i+1] = data;
 count++;
 }

 }
```

doubleEndedQueue.c | *priorityQue.c | mpq.c | ascendingPQ.c | *descendingPQ.c

```
30
31      // insert the data
32      pq[i+1] = data;
33      count++;
34      }
35
36      }
37
38  int removeData(){
39
40  return pq[d++];
41  }
42  void display()
43  {int i;
44  if (count==0)
45  {
46  printf("queue is empty\n");
47  return;
48  }
49  printf("Contents of queue: ");
50  for(i=d;i<count;i++)
51  {
52  printf("%d ",pq[i]);
53  }
54  printf("\n");
55  }
56
57  int main() {
58  int choice,item;
59  for(;;)
60  {
61  printf("\n1:insert 2:delete_smallest 3:display 4:exit\n");
62  printf("Enter the choice :");
63  scanf("%d",&choice);
64  switch(choice)
65  {
```

```
51  {
52  printf("%d ",pq[i]);
53  }
54  printf("\n");
55  }
56
57  int main() {
58  int choice,item;
59  for(;;)
60  {
61  printf("\n1:insert 2:delete smallest 3:display 4:exit\n");
62  printf("Enter the choice :");
63  scanf("%d",&choice);
64  switch(choice)
65  {
66  case 1:printf("Enter the item to be inserted :");
67  scanf("%d",&item);
68  insert(item);
69  break;
70  case 2:item=removeData();
71  if(item==-1)
72  printf("Queue is empty\n");
73  else
74  printf("item deleted=%d\n",item);
75  break;
76  case 3:display();
77  break;
78  default:exit (0);
79
80  }
81
82  }
83
84  }
85
```

C:\Users\CMRS\Documents\ascendingPQ.exe

```
1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :1
Enter the item to be inserted :10

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :1
Enter the item to be inserted :3

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :1
Enter the item to be inserted :7

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :1
Enter the item to be inserted :9

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :2
item deleted=3

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :3
Contents of queue: 7 9 10

1:insert 2:delete_smallest 3:display 4:exit
Enter the choice :
```