mulpri.c

mulpri.c > pqdelete()

```c
#include<stdio.h>
#include<stdlib.h>
#define N 5
int queue[3][N];
int front[3]={0,0,0};
int rear[3]={-1,-1,-1};
int item,pr;

int 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;
 }
}
int 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 %d\n",queue[i][front[i]],i+1);
  front[i]++;
  }
 }
}
int display()
{
int i,j;
```

mulpri.c

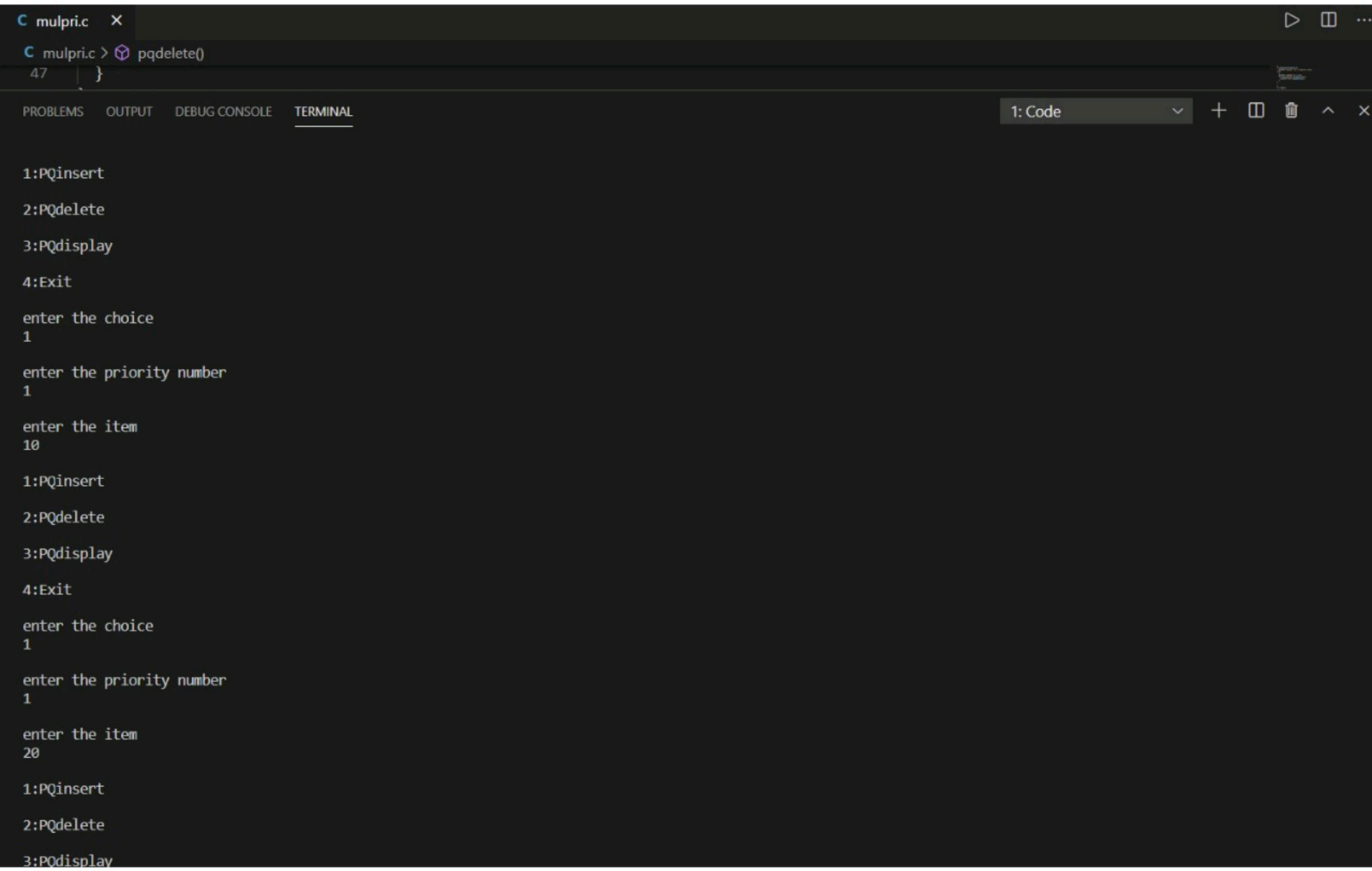
mulpri.c > pqdelete()

```
34  }
35  int display()
36  {
37  int i,j;
38  for(i=0;i<3;i++)
39  {
40  if(rear[i]==front[i]-1)
41    printf("\nqueue  %d is empty\n",i+1);
42  else
43    {
44    printf("\nQUEUE %d:",i+1);
45    for(j=front[i];j<=rear[i];j++)
46      printf("%d\t",queue[i][j]);
47    }
48  }
49
50  }
51  int main()
52  {
53  int ch;
54  while(1)
55  {
56  printf("\n1:PQinsert\n");
57  printf("\n2:PQdelete\n");
58  printf("\n3:PQdisplay\n");
59  printf("\n4:Exit\n");
60  printf("\nenter the choice\n");
61  scanf("%d",&ch);
62  switch(ch)
63  {
64   case 1:printf("\nenter the priority number\n");
65      scanf("%d",&pr);
66      if(pr>0 && pr<4)
67          pqinsert(pr-1);
68      else
69      printf("\nonly 3 priority exists 1 2 3\n");
70      break;
71  case 2: pqdelete();
```

mulpri.c

mulpri.c > pqdelete()

```c
    }
}

}
int main()
{
int ch;
while(1)
{
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);
}
}
}
```


mulpri.c
mulpri.c > pqdelete()
47 }
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: Code
1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit
enter the choice
1
enter the priority number
1
enter the item
10
1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit
enter the choice
1
enter the priority number
1
enter the item
20
1:PQinsert
2:PQdelete
3:PQdisplay

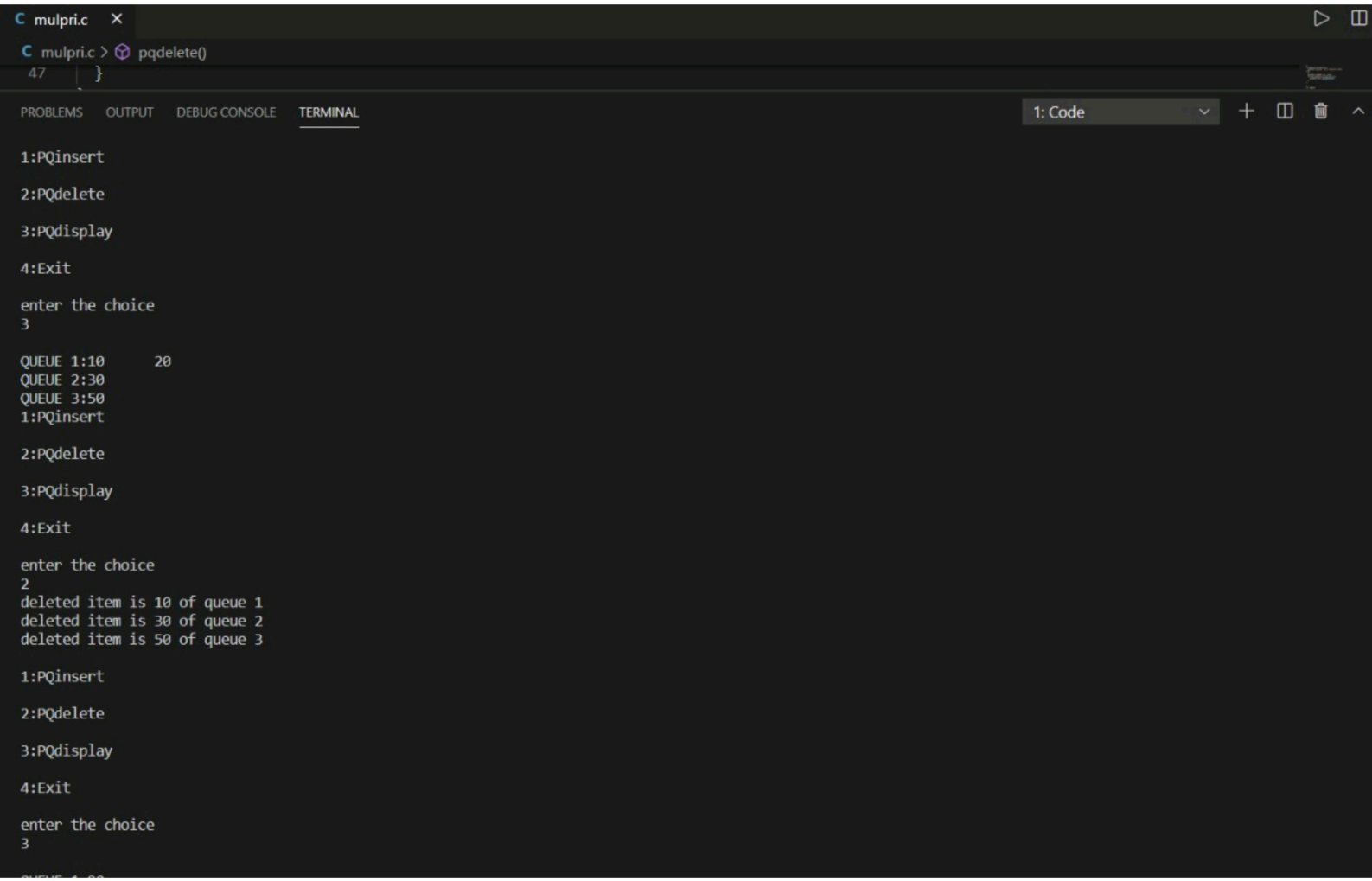
mulpri.c

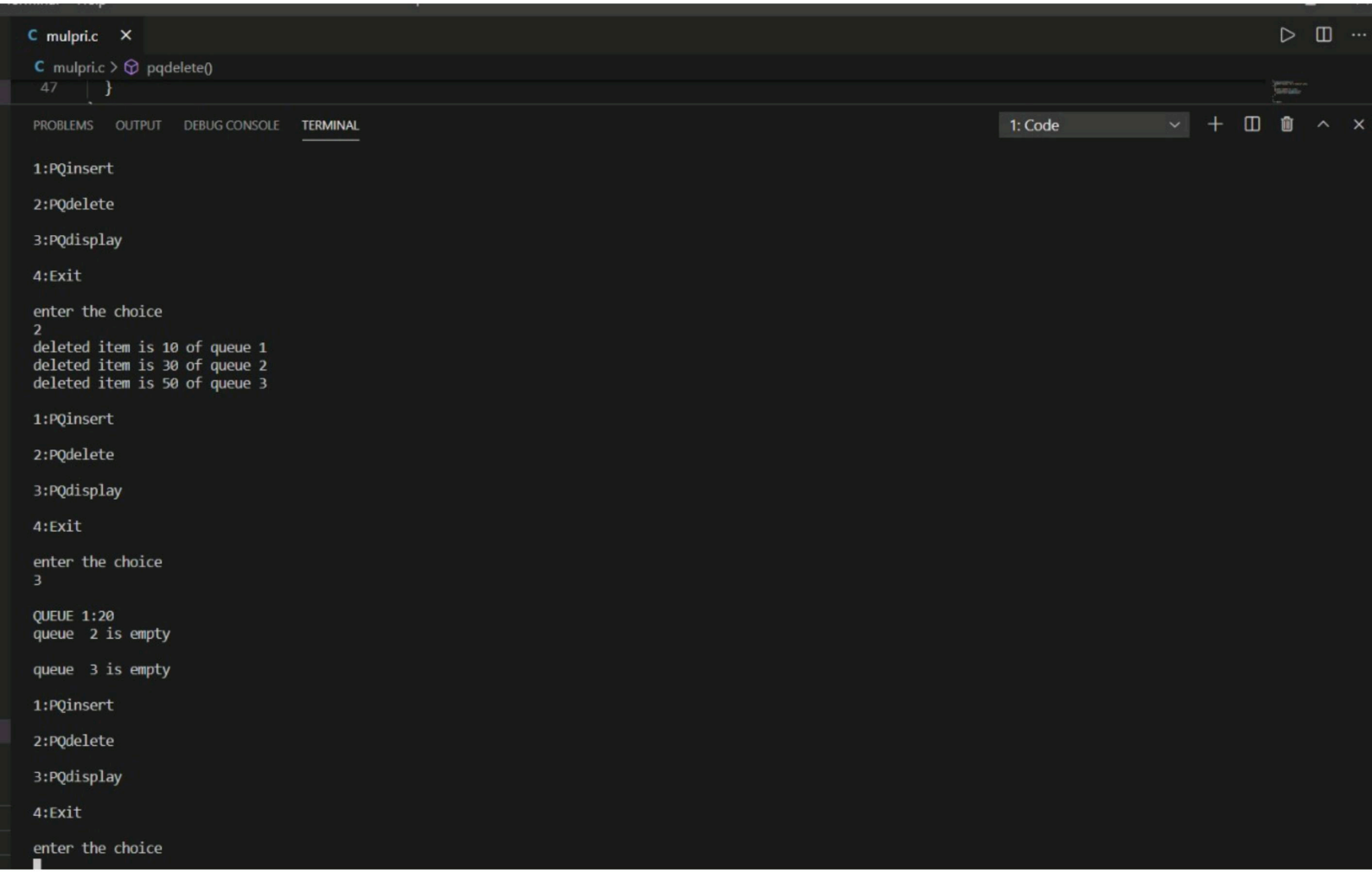
mulpri.c > pqdelete()

```
47      }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: Code

```
1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice
1

enter the priority number
2

enter the item
30

1:PQinsert

2:PQdelete

3:PQdisplay

4:Exit

enter the choice
1

enter the priority number
3

enter the item
50

1:PQinsert

2:PQdelete

3:PQdisplay
```


mulpri.c
mulpri.c > pqdelete()
47 }
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: Code
1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit
enter the choice
3
QUEUE 1:10 20
QUEUE 2:30
QUEUE 3:50
1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit
enter the choice
2
deleted item is 10 of queue 1
deleted item is 30 of queue 2
deleted item is 50 of queue 3
1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit
enter the choice
3


mulpri.c
mulpri.c > pqdelete()
47 }
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: Code
1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit
enter the choice
2
deleted item is 10 of queue 1
deleted item is 30 of queue 2
deleted item is 50 of queue 3
1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit
enter the choice
3
QUEUE 1:20
queue 2 is empty
queue 3 is empty
1:PQinsert
2:PQdelete
3:PQdisplay
4:Exit
enter the choice

Multiple priority queue

```
#include <stdio.h>
#include <conio.h>
# define N 5

int queue[3][N];
int front[3]={0,0,0};
int rear[3]={-1,-1,-1};
int item, pr;

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

```
int display()
{
int i,j;
for (i=0; i<3; i++)
{
if (rear[i] == front[i]-1)
printf ("\n queue %d is empty \n", i+1);
else
{
printf ("\n Queue %d :", i+1);
for(j=front[i] ; j<=rear[i] ;j++)
printf ("%d ", queue[i][j]);
}
}
int main()
{
int ch;
while
printf (" 1. PQinsert \n2. PQdelete \n3. PQdisplay \n4. Exit \n");
printf ("enter the choice \n");
scanf ("%d", &ch);
switch(ch)
{
case1: printf ("enter the priority number \n");
scanf ("%d ,&pr);
if (pr>0 && pr<4)
pqinsert(pr-1);
else
printf ("only 3 queue exist \n");
break;
case2: pqdelete();
break;
case3: display();
break;
case4 : exit(0);
}
}
}
```