

C mulpri.c C despri2.c X

C despri2.c > display()

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  #define MAX 6
5
6  int pq[MAX];
7  int count = 0;
8
9  void insert(int data){
10     int i = 0;
11     if(count==MAX)
12     {
13         printf("Queue overflow\n");
14         return;
15     }
16     // if queue is empty, insert the data
17     if(count == 0){
18         pq[count++] = data;
19     }else{
20         // start from the right end of the queue
21         for(i = count - 1; i >= 0; i-- ){
22             // if data is larger, shift existing item to right end
23             if(data > pq[i]){
24                 pq[i+1] = pq[i];
25             }else{
26                 break;
27             }
28         }
29
30         // insert the data
31         pq[i+1] = data;
32         count++;
33     }
34 }
35
36
37 int removeData(){
```

```
C mulpri.c C despri2.c X
```

```
C despri2.c > insert(int)
31     pq[i+1] = data,
32     count++;
33 }
34
35 }
36
37 int removeData(){
38     return pq[--count];
39 }
40 void display()
41 {int i;
42 if (count==0)
43 {
44     printf("queue is empty\n");
45     return;
46 }
47 printf("Contents of queue: ");
48 for(i=0;i<count;i++)
49 {
50     printf("%d ",pq[i]);
51 }
52 printf("\n");
53 }
54
55 int main() {
56     int choice,item;
57     for(;;)
58     {
59         printf("\n1:insert\n2:delete_largest\n3:display\n4:exit\n");
60         printf("Enter the choice :");
61         scanf("%d",&choice);
62         switch(choice)
63         {
64             case 1:printf("Enter the item to be inserted :");
65                     scanf("%d",&item);
66                     insert(item);
67                     break;
68             case 2:item=removeData();
```

```
C: \mulpri.c  C: \despri2.c x
C: \despri2.c > insert(int)
47 printf("Contents of queue: ");
48 for(i=0;i<count;i++)
49 {
50     printf("%d ",pq[i]);
51 }
52 printf("\n");
53 }
54
55 int main() {
56     int choice,item;
57     for(;;)
58     {
59         printf("\n1:insert\n2:delete_largest\n3:display\n4:exit\n");
60         printf("Enter the choice :");
61         scanf("%d",&choice);
62         switch(choice)
63         {
64             case 1:printf("Enter the item to be inserted :");
65                     scanf("%d",&item);
66                     insert(item);
67                     break;
68             case 2:item=removeData();
69                     if(item==--1)
70                         printf("Queue is empty\n");
71                     else
72                         printf("item deleted=%d\n",item);
73                     break;
74             case 3:display();
75                     break;
76             default:exit (0);
77         }
78     }
79 }
80 }
81 }
82 }
```

mulpri.c

despri2.c

despri2.c > display()

34

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

1: Code

+

□

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

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

1:insert
2:delete_largest
3:display
4:exit
Enter the choice :1
Enter the item to be inserted :5

1:insert
2:delete_largest
3:display
4:exit
Enter the choice :3
Contents of queue: 10 5 2

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

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

Deleting Priority

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

```
#include <string.h>
```

```
#define MAX 6
```

```
int pq[MAX];
```

```
int count = 0;
```

```
void insert (int data) {
```

```
    int i = 0;
```

```
    if (count == MAX)
```

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

```
        return;
```

```
    }
```

```
    if (count == 0) {
```

```
        pq[count++] = data;
```

```
    }  
    else
```

```
    { for (i = count - 1; i >= 0; i--)
```

```
        { if (data > pq[i]) {
```

```
            pq[i+1] = pq[i];
```

```
        }  
        else {
```

```
            break;
```

```
        }  
    }
```

```
    pq[i+1] = data;
```

```
    count++;
```

```
    }
```

```
}
```

```
int removeData() {
```

```
    return pq[--count];
```

```
}
```

```
void display()
```

```
{ int i;
```

```
if (count == 0)
```

```

{ printf ("queue is empty");
  return;
}
printf ("contents of queue: ");
for (i=0; i<count; i++)
{ printf ("%d", pq[i]);
  }
printf ("\n");
}

int main() {
  int choice, item;
  for (;;)
  { printf ("\n1: insert \n2: delete - largest \n3: display \n4: exit");
    printf ("enter the choice: ");
    scanf ("%d", &choice);
    switch (choice)
    {
      case 1: printf ("Enter the item to be inserted: ");
              scanf ("%d", &item);
              insert (item);
              break;
      case 2: item = removeData ();
              if (item == -1)
                printf ("Queue is empty \n");
              else
                printf ("item deleted = %d \n", item);
              break;
      case 3: display ();
              break;
      default: exit (0);
    }
  }
}

```