mulpri.c | despri2.c

despri2.c > display()

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.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 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 larger, shift existing item to right end
            if(data > pq[i]){
               pq[i+1] = pq[i];
            }else{
               break;
            }
         }

         // insert the data
         pq[i+1] = data;
         count++;
      }

}

int removeData(){
```

mulpri.c | despri2.c

despri2.c > insert(int)

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

}

int removeData(){
   return pq[--count];
}
void display()
{int i;
if (count==0)
{
    printf("queue is empty\n");
    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\n");
        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();
```

mulpri.c | despri2.c

despri2.c > insert(int)

```c
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\n");
        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);

        }

    }

  }
```

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 :
```

Decending 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\n4exit");
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);
}
}
}
```