```
//descending priority queue
using array
#include<stdio.h>
#include<conio.h>
int q_full(void);
int q_empty(void);
void priority(void);
struct pqueue
{
int a[5];
int front,rear;
}dpq;
void main()
{
int choice,item,temp;
clrscr();
dpq.front=-1;
dpq.rear=-1;
```

```
do
{
printf("\n1.insertion\n2.deletion\n3.display\n4.exit");
printf("\nenter the choice");
scanf("%d",&choice);
switch(choice)
{
case 1:
{
if(q_full()==1)
printf("\nqueue is full");
else
{
printf("\nenter the item");
scanf("%d",&item);
if(dpq.rear==-1)
{
```

```
dpq.front=0;
dpq.rear=0;
dpq.a[dpq.rear]=item;
}
else
dpq.a[++dpq.rear]=item;
}
priority();
break;
}
case 2:
{
if(q_empty()==1)
printf("\nqueue empty");
else
{
if((dpq.rear==dpq.front)&&(dpq
.rear!=-1))
```

```
{
dpq.rear=-1;
dpq.front=-1;
}
else
dpq.front++;
}
break;
}
case 3:
{
if(q_empty()==1)
printf("\nqueue empty");
else
{
//priority();
temp=dpq.front;
while(temp<=dpq.rear)
```

```
{
printf("%d\t",dpq.a[temp]);
temp++;
}
}
}
break;
}}while(choice<4);
getch();
}
int q_full()
{
if(dpq.rear==4)
return 1;
else
return 0;
}
int q_empty()
```

```
{
if(dpq.rear==-1)
return 1;
else
return 0;
}
void priority()
{
int i,j,k,temp;
for(i=1;i<5;i++)//number of
iterations
for(j=0;j<i;j++)//number of
comparisons
{
if(dpq.a[j]<dpq.a[i])
{
temp=dpq.a[j];
dpq.a[j]=dpq.a[i];
for(k=i;k>j;k--)//right shift
dpq.a[k]=dpq.a[k-1];
dpq.a[k+1]=temp;
}}}
```