```
Dequeve
# include Letdio . h>
# include Letalib. W
# define asize 5
int f=0, n=-1, ch;
int item, of [10];
int isfull () of
  return (n==qsize-1))1:0;
int ixempty () ?
    (0:10(1x1) muter
  void insert_near ()?
  if (iefund) [
   printf ("Queue Overflow 1")
      neturn!
    カナニリ
   acro = item;
 void deletefront (1 &
   if (ixempty ()) {
   arint ("Queue empty In")
        return; &
   printfl"iten deleted iz 1.d ln", a[f++]).
       $ (10 cf) fi
          fzo;
          N=-1'
```

```
void inectfront () &
   & (0 = 17) Pi
     f -- ;
     a Cf) = item;
   g return !
   elee if (+==0 & n==-1) }
     Q [++ n] = item;
      netur;
   olee.
      print[ " Invention not possible ["]"
   void deleterear() {
    if (ixempty w) [
      print (" Queue empty (n");
              return;
    printf ("item deleted is 1.d la", a, [n--]);
       3 (25 ) fi
    void display () f
        int i ;
       if (ixerpty ()) }
          printf (" greve empty (");
            return.
        2
     $ (4+i ( a= sic ; +=i) not
          (C(i)p, " nl b. 2") Itaing
```

```
int main ()?
ton (; ;) ?
print ("In1. inechear M. 2. inectfront In 3. delete real n 4. delet
                 fruntlas. dieplay la 6. exit la")
 printy (" Enter choice. In");
  early ("1.d", fch);
   ean
    ewitch (ch) &
   call! print(" Enter the item(");
          earl ("I'd" & item);
           incerties ();
           break;
   cole 2: printf ("enter the item (n");
           -ecanf ("1.d", & item);
            insent front ();
            break:
    cole. 3: delete front ():
            break;
    cole 4: deleterear();
            break;
    cose s: displayes;
    default: exit (0);
```

```
multiple priority Queve.
# include L Adio.h)
# include Letalibih)
void padelete ();
void display ():
void pained ofbi):
# define N 3
int queve [3] [N] ;
 int front [3] = [0,0,08;
 int near [2] = [-1,-1,-19;
 int-item, pri
 int main cs {
 int ch :
 while (1) {
 printf ("In PRIORITY QUEUEIn");
 :("" * * * / "");
 printy (" In Y 1: Paineet h"):
 printf ("In/ 2: Pa delete h");
 print( 1" In 1 ty: Excit In" );
 printy l'Enter the choice. la");
  ecanfluted", fch);
   ewitch (ch) (
 case 1: print (" Enter the priority number la");
        ecanf ("Y.d", & pn);
        (hrad to ora) li
          parineent (pr-1)
          elee
         print (" Only 3 priority existe (");
             econolitized break!
  case 2: pa, delete ();
             break's
```

core 3: die play ()

beneale ;

```
case 4 exit (0);
3
 void ineed Cint and I
 (1-11 = = [near Land ] fi
   printy (" Queue Overflow (n');
   elee. P
   printf (" Enter the item \n").
      ecanf ("-1.d" & item);
      C++ CrgJ rosa
     quevelon I Trear Gard = item;
  void padelete () f
  inti i
  for (i=0; i23: i++) }
  if (renti) = = front [i] -1)
    printy (" Queve empty ha"):
    elee s
   printf l'deleted item is 1.d of queue 1.d ln",
                                 queue [i] [[ist to of I [ist ]]
       tront[i] ++
 void display () ?
 int int i
 ) (++i, Esi cosi) not
  (1- Ci) that == front [i] -i)
    print (" queue empty 1.d /m", iti)
   elee s
   printly (" in greve yed: ", itil)
      ton G= front(i); jl=reaci); j++)
           [ (() (i) (i) p.i.) ] + in
133
```

```
According Priority Queve.
# include Letdians
# include Letalib.L)
# define que rize 3
int item, front =0, near -1, glave-size ]:
 void incertion () {
    if (near = = que-eize -1) {
     printly ("Queue Ovallow ("));
       return:
   int is min;
   animy lu Enter the item / ");
   ecanf ("did" & item);
   of C++ near) = item;
   2 (1=5 man) fi
    for (i=1; it= near; i++) {
           min = q [i] )
          リーニード
       glaime (ist of 0=<i) skilling
            1: Ci3P = Ci+;3p
             1--: 1
          9 [:+1] = min:
   int deletefront () (
    3 (non & trong) fi
          front = 0 !
      : C++ trunt Ip nowter
  void displayer ()
      int i :
      of Churt 2 rear &
        printy ("queve in empty bi")
               return: 8
```

```
printf (" Contente of queue in");
 for (i= front : ic= nen; i++) }
      printy ("I d la", a [i])
int main () &
  int choice :
  ton (;;) 6
 printf ("In1. Ingest new In2. Delete front In 3. Display In 4. exit ("1)
        paintf (" Enter the choice : ");
        ecarf - (" +.d", & choice)",
        ewitch (choice.) ?
       case 1: i'reat rear ();
                breaki
        caee2: item = delete front ();
                 if (item = = -1)
                  printf (4 Queve il empty ["];
                    clee.
                    print[(" item deleted is 1-d \n", item]
                   breaki
        care 3: dieplayqui
                   break:
       default
                  exit (0):
```

```
Decending Priority Queue.
At include addio. hs
at include coldlibility
It define quesize ?
 int item, front =0, non=-1, of tome-eized;
  void inerdnan () ?
   it (rear = = que size -i) {
     prof ("Queue Ovallow In")
        noturn;
   ind i.i. this
   Printf (" Enter the item In");
   econfluted", & item);
   althrong = item:
    3 (nears = 1) {
    3(++in; non=si; 1=i) not
         max = q. Cij;
          リニューリン
     while (12=0 4f of C) 72 max) {
              ( [ [ ] p = [ [ + i] p
              crim = CI+iJp
      int deldefront ()?
        I (non = trong) fi
          front =0;
          sear = -1?
         returns -1.
        Ethnort p nuter
    void displaya co &
       it this
     } ( front > nen) }
        runt (" Queue ie emply ("))
               return ? }
        print (" contente of queue are (""))
          Son Ciefront; ice near; i++)
           Paint-1(44, d) 11
```

int main () { ind choice; ton (; ;) { print ("h1. Ineal non In 2. delete front In3. display In4. exit [") printy ("Enter the choice(n"); land (" Yed", & choice); lwitch (choice) { (see 1: inects (); break; call 2: item = delete front (); if (item ==-1) printy ("Queue empty h"); paintf (" Deleted item is 1.d \n", item); break; case 3: displaya (); break; exit (0); defaultoner. back ?

```
C:\Windows\SYSTEM32\cmd.exe
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 1
Enter the item
10
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 1
Enter the item
80
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 1
Enter the item
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 1
queue overflow
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 3 contents of queue
80
10
1.Insert rear
2.Delete front
3.Display
```

```
C:\Windows\SYSTEM32\cmd.exe
4.exit
Enter the choice : 3
contents of queue
80
10
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 2
item deleted is 80
1. Insert rear
2.Delete front
3.Display
4.exit
Enter the choice: 2
item deleted is 10
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice: 2
item deleted is 5
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 2
queue is empty
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 4
(program exited with code: 0)
```

```
ueue.c 🗶 multiple priorityq.c 🗶 ascending priorityq.c 🗶 descending pq.c 🗶 multiple pq.c 🗶
   #include<stdio.h>
   #include<stdlib.h>
   #define que_size 3
   int item, front=0, rear=-1, q[que size];
  □void insertrear(){
        if(rear==que_size-1){
           printf("queue overflow");
            return;
        int i, j, min;
       printf("Enter the item\n");
        scanf("%d",&item);
        q[++rear]=item;
        if(rear>=1){
            for(i=1;i<=rear;i++) {
                min=q[i];
                 j=i-1;
                 while(j>=0 && q[j]>min){
                     q[j+1]=q[j];
                     j--;
                 q[j+1]=min;
  L<sub>3 } }</sub>
  Fint deletefront(){
Fig. (front>rear)
        if(front>rear){
            front = 0;
            rear = -1;
             return -1;
        return q[front++];
  □void displayq() {
        int i;
1/63 col: 0 cel: 0 INS TAR mode: CRLE encoding: LITE-8
```

```
Dequeue.c 🗶 multiple priorityq.c 🗶 ascending priorityq.c 🗶 descending pq.c 🗶 multiple pq.c 🗶
     L}
31
32
     □void displayq() {
33
          int i;
           if(front>rear){
34
35
               printf("queue is empty");
36
               return; }
37
          printf("contents of queue \n");
38
          for(i=front;i<=rear;i++){
39
               printf("%d\n",q[i]);
40
41
42
     □int main() {
43
          int choice;
44
           for(;;){
45
               printf("\n1.Insert rear \n2.Delete front \n3.Display \n4.exit \n ");
46
               printf("Enter the choice : ");
47
               scanf ("%d", &choice);
     占
48
               switch (choice) {
49
                   case 1:insertrear();
50
                           break;
51
                   case 2:item=deletefront();
52
                           if(item==-1)
53
                           printf("queue is empty\n");
54
                           else
55
                           printf("item deleted is %d \n",item);
56
                           break;
                   case 3:displayq();
57
58
                           break;
59
                   default:exit(0);
60
              }
           }
61
62
63
                     INS TAR mode CRLE encoding UTE-8 filetype: C scope unkno
```

```
C:\Windows\SYSTEM32\cmd.exe
3.Display
4.exit
Enter the choice : 2
item deleted is 5
1.Insert rear
2.Delete front
Display
4.exit
Enter the choice : 2
item deleted is 10
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 2
item deleted is 20
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 2
queue is empty
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 3
queue is empty
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 4
(program exited with code: 0)
Press any key to continue . . .
```

```
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 1
Enter the item
10
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 1
Enter the item
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 1
Enter the item
20
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 3
contents of queue
10
20
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 2
item deleted is 5
1.Insert rear
2.Delete front
```

```
ieue.c 🗶 multiple priorityq.c 🗶 ascending priorityq.c 🗶 descending pq.c 🗶 multiple pq.c 🗶
   #include<stdio.h>
   #include<stdlib.h>
   #define que_size 3
   int item, front=0, rear=-1, q[que_size];
 □void insertrear(){
       if(rear==que_size-1){
           printf("queue overflow");
           return;
       int i, j, max;
       printf("Enter the item\n");
       scanf ("%d", &item);
       q[++rear]=item;
       if(rear>=1){
           for (i=1; i <= rear; i++) {
               max=q[i];
                j=i-1;
                while(j \ge 0 && q[j] < max) {
                    q[j+1]=q[j];
                    j--;
                q[j+1]=max;
     }
 Pint deletefront() {
       if(front>rear){
           front = 0;
           rear = -1;
             return -1;
       return q[front++];

pvoid displayq() {

       int i;
     cal- 0 sel- 0 INS TAR mode: CRLE encoding: UTE-8 filetype: C scope: unknown
```

```
Upen
             Save
                   Save All | Kevert
                                   Close | Back Forward | Compile Build
                                                                            execute | Color Chooser |
ieue.c 🗶 multiple priorityq.c 🗶 ascending priorityq.c 🗶 descending pq.c 🗶 multiple pq.c 🗶
 L
 □void displayq() {
       int i;
       if(front>rear){
           printf("queue is empty");
           return; }
       printf("contents of queue \n");
       for(i=front;i<=rear;i++){
           printf("%d\n",q[i]);
 □int main(){
       int choice;
       for(;;) {
           printf("\n1.Insert rear \n2.Delete front \n3.Display \n4.exit \n ");
           printf("Enter the choice : ");
           scanf ("%d", &choice);
           switch (choice) {
                case 1:insertrear();
                       break;
                case 2:item=deletefront();
                       if(item==-1)
                       printf("queue is empty\n");
                       else
                       printf("item deleted is %d \n", item);
                       break;
                case 3:displayq();
                       break;
               default:exit(0);
           }
       }
```

```
C:\Windows\SYSTEM32\cmd.exe
enter the item
60
PRIORITY QUEUE
       1:PQinsert
       2:PQdelete
        3:PQdisplay
        4:Exit
enter the choice
enter the priority number
enter the item
80
PRIORITY QUEUE
       1:PQinsert
        2:PQdelete
        3:PQdisplay
        4:Exit
enter the choice
enter the priority number
enter the item
90
```

```
C:\Windows\SYSTEM32\cmd.exe
       3:PQdisplay
       4:Exit
enter the choice
enter the priority number
 Queue overflow
PRIORITY QUEUE
       1:PQinsert
       2:PQdelete
       3:PQdisplay
       4:Exit
enter the choice
QUEUE 1:20 queue empty 2
                       90
QUEUE 3:60
              80
PRIORITY QUEUE
       1:PQinsert
       2:PQdelete
       3:PQdisplay
       4:Exit
enter the choice
```

```
C:\Windows\SYSTEM32\cmd.exe
PRIORITY QUEUE
       1:PQinsert
       2:PQdelete
       3:PQdisplay
       4:Exit
enter the choice
enter the priority number
enter the item
30
PRIORITY QUEUE
*********
       1:PQinsert
       2:PQdelete
       3:PQdisplay
       4:Exit
enter the choice
deleted item is 10 of queue 1
deleted item is 30 of queue 2
queue empty
PRIORITY QUEUE
       1:PQinsert
        2:PQdelete
```

```
PRIORITY QUEUE
        1:PQinsert
        2:PQdelete
        3:PQdisplay
        4:Exit
enter the choice
enter the priority number
enter the item
10
PRIORITY QUEUE
        1:PQinsert
        2:PQdelete
        3:PQdisplay
        4:Exit
enter the choice
enter the priority number
enter the item
20
PRIORITY QUEUE
 **********
```

```
enter the choice
deleted item is 10 of queue 1
deleted item is 30 of queue 2
queue empty
PRIORITY QUEUE
        1:PQinsert
        2:PQdelete
       3:PQdisplay
       4:Exit
enter the choice
40
PRIORITY QUEUE
*********
       1:PQinsert
       2:PQdelete
       3:PQdisplay
       4:Exit
enter the choice
enter the priority number
enter the item
60
PRIORITY QUEUE
 ***********
```

```
jueue.c 💥 multiple priorityq.c 💥 ascending priorityq.c 💥 descending pq.c 💥 multiple pq.c 💥
    #include<stdio.h>
    #include<stdlib.h>
    void pqdelete();
    void display();
    void pginsert(int pr);
    #define N 3
    int queue[3][N];
    int front[3]={0,0,0};
    int rear[3]={-1,-1,-1};
    int item, pr;
    int main()
  ∃{
    int ch:
    while (1)
  □{
    printf("\nPRIORITY 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)
            pginsert (pr-1);
            else
            printf("only 3 priority exists 1 2 3\n");
            break:
    case 2:pqdelete();
```

1 / 02 cole 0 cole 0 INS TAR moder CRIE encoding LITE 9 filetype C scope upknown

```
Save Save All Revert Close Back Forward Compile Build Execute
   Open
ieue.c 💥 multiple priorityq.c 💥 ascending priorityq.c 💥 descending pq.c 💥 multiple pq.c 💥
   case 2:pqdelete();
          break:
   case 3:display();
          break;
   case 4:exit(0);
   void 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;
   void pqdelete()
 ₽{
   int i:
   for(i=0;i<3;i++)
     if(rear[i] == front[i]-1)
     printf("queue empty\n");
     else
     printf("deleted item is %d of queue %d\n", queue[i][front[i]],i+1);
     front[i]++;
```

```
Open
                  Save All | Revert Close | Back Forward | Compile Build
                                                                              Execute | Color Chooser | -
ieue.c 🗶 multiple priorityq.c 🗶 ascending priorityq.c 🗶 descending pq.c 🗶 multiple pq.c 🗶
   queue[pr][rear[pr]]=item;
 }
   void pqdelete()
 ₽{
  int i;
  for(i=0;i<3;i++)
 申{
     if(rear[i] == front[i]-1)
    printf("queue empty\n");
     else
    printf("deleted item is %d of queue %d\n", queue[i][front[i]], i+1);
    front[i]++;
   void display()
 ₽{
  int i,j;
  for(i=0;i<3;i++)
 中{
  if(rear[i] == front[i]-1)
    printf("queue empty %d\n",i+1);
   else
    printf("\nQUEUE %d:",i+1);
     for(j=front[i];j<=rear[i];j++)</pre>
       printf("%d\t", queue[i][j]);
         }
```

```
oute outernit thereis close to buck formula i complie build
    open
                                                                            EXCEUTE | COIOI CHOOSEI |
jueue.c 🗶 multiple priorityq.c 🗶 ascending priorityq.c 💥 descending pq.c 💥
   #include<stdio.h>
   #include<stdlib.h>
   #define qsize 5
   int f=0, r=-1, ch;
   int item, q[10];
  Dint isfull() {
      return(r==qsize-1)?1:0;
     }
  Fint isempty() {
     return (f>r) ?1:0;
  □void insert_rear(){
      if(isfull())
         printf("queue overflow\n");
         return;
       r=r+1;
       q[r]=item;
  □void delete front(){
      if(isempty()){
         printf("queue empty\n");
         return;
      printf("item deleted is %d\n",q[(f)++]);
      if(f>r){
         f=0;
         r=-1;
         }
  □void insert_front(){
      if(f!=0)
1/80 col·3 cel·0 INS TAR MOD mode CRLE encoding UTE-8 filetype: C scope delete front
```

```
Save Save All | Revert Close | Back Forward | Compile Build | Execute | Color Chooser |
   Open
eue.c 🗶 multiple priorityq.c 🗶 ascending priorityq.c 🗶 descending pq.c 🗶
      if(f!=0)
       {
        f=f-1;
        q[f]=item;
        return;
       else if((f==0)&&(r==-1))
        q[++(r)]=item;
        return;
       }
       else
        printf("insertion not possible\n");
  void delete_rear()
 早
     if(isempty()){
        printf("queue is empty\n");
        return;
     printf("item deleted is %d\n",q[(r)--]);
      if(f>r){
        f=0;
        r=-1;
    }
 □void display() {
     int i;
      if(isempty()){
         printf("queue empty\n");
         return;
      for (i=f;i<=r;i++)
```

```
₽ .
                                              400
                                                     9
                                                              4
              0.
                     ₹.
                              ×
                                             Back Forward Compile Build Execute Color Chooser
                   Save All
             Save
                             Revert
                                     Close
ueue.c 🗶 multiple priorityq.c 🗶 ascending priorityq.c 🗶 descending pq.c 🗶
  t
    }
 □void display() {
      int i;
      if(isempty()){
        printf("queue empty\n");
         return;
      for(i=f;i<=r;i++)
      printf("%d\n",q[i]);
 Pint main() {
    for(;;) {
       printf("1.insert_rear\n2.insert_front\n3.delete_rear\n4.delete_front\n5.display\n6.exit\n");
       printf("enter choice\n");
       scanf ("%d", &ch);
       switch(ch){
          case 1:printf("enter the item\n");
                 scanf("%d", &item);
                 insert_rear();
                 break;
          case 2:printf("enter the item\n");
                 scanf("%d", &item);
                 insert_front();
                 break;
          case 3:delete rear();
                 break;
          case 4:delete_front();
                 break;
          case 5:display();
                 break;
          default:exit(0);
         }}}
```

```
C:\Windows\SYSTEM32\cmd.exe
```

```
1.insert rear
2.insert_front
3.delete_rear
4.delete front
5.display
6.exit
enter choice
enter the item
10
1.insert_rear
2.insert_front
3.delete_rear
4.delete front
5.display
6.exit
enter choice
enter the item
20
1.insert_rear
2.insert_front
3.delete rear
4.delete_front
5.display
6.exit
enter choice
enter the item
30

    insert rear

2.insert_front
3.delete_rear
4.delete front
5.display
6.exit
enter choice
10
20
30
1.insert_rear
2.insert_front
3.delete rear
```

```
C:\Windows\SYSTEM32\cmd.exe
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
 6.exit
enter choice
item deleted is 10
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
 6.exit
 enter choice
item deleted is 20
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
 5.display
 6.exit
 enter choice
 enter the item
50
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
 5.display
 6.exit
 enter choice
 enter the item
60
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
```

```
C:\Windows\SYSTEM32\cmd.exe
2.insert front
3.delete_rear
4.delete front
5.display
6.exit
enter choice
enter the item
50
1.insert_rear
2.insert_front
3.delete rear
4.delete_front
5.display
6.exit
enter choice
enter the item
60
1.insert rear
2.insert_front
3.delete_rear
4.delete front
5.display
6.exit
enter choice
60
50
30
1.insert_rear
2.insert front
3.delete rear
4.delete_front
5.display
6.exit
enter choice
6
(program exited with code: 0)
Press any key to continue
```

```
1.Insert rear
Delete front
Display
4.exit
Enter the choice : 1
Enter the item
10
1.Insert rear
Delete front
3.Display
4.exit
Enter the choice: 1
Enter the item
20
1. Insert rear
Delete front
Display
4.exit
Enter the choice: 1
Enter the item
30
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 1
Enter the item
40
queue overflow1.Insert rear
Delete front
Display
4.exit
Enter the choice: 3
contents of queue
10
20
30
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice: 2
item deleted is 10
1.Insert rear
```