```
Linear Quelle
H include cetalions
st includez etallib-h)
H define que size 3
ind item, front=0, nen=-1, q [ que size);
void incertien () [
 if ( new = = que size - 1) {
  printf ("Queue Queflow / "));
    return: ?
    vert=1,
   q Eneral = item;
 int deletefront () {
   ? (front = rear) ?
    front =0;
    ren = -1 ! / 1 / 1 / 1 / 1 / 1 / 1 / 1
     return -1;
   : C++ Frong J provider
 void displayaes ?
  inti :
  if (front > rear)
  ¿ printf ("Queue is empty In"):
              neturn !
  print[ (" (ontate of queue \n");
     ton (i = front; i <= nen; i++) (
      : ([i] p ," Nb.1") ftwing
```

```
int main () {
 int choice: item deleted:
 ton (; ; ) {.
 printf ("In 1, insert near In 2. delete front In 3. display In 4. exiting
 printfl" Enter the choice \n");
 leanf ("-s.d" tehoice);
  ewitch (choice) of
 case 1: printy (" Enter the item to be inserted. |n');
           lean ["-1.d", & item);
           ineed sends):
           break;
  case 2: item deleted = deletefront ();
             if (item-deleted = = -1)
               Arinty ("Queve Under Jow In");
              elee.
              printf ("deleted item is 1.d", item -deleted");
                break;
   call 3: Dieplayer(1);
               break;
   core 4: exit (0);
```

```
Circular Queve
 # include Letalio.h>
 # include Letalib. h>
 # define que size 3
 int item, front =0, rear = -1, of cope size), count =0;
void incertnear () }
 if (count = = que size) f
   prints ("Queue Overflowsn");
    return; }
    near = (near+1) -1. que_size;
   of Enear ] = item;
   count + + ,
int delete front () ?
 if (count = =0)
     return -1:
  front = (front +1) 1. que_size.;
   count --;
   return item;
 void dieplay () {
 · + ; thi
  if (count =0) &
    print (" avene le empty (n");
  g return;
  c tront = +
  print( (" (ontente of Queue are \"));
  ton (i=1; il= count; i++) {
     chintlent ( and ports) their
         f= (++1) y. que-size:
```

```
I comment to
ind choice;
for (; ; )
C printp ("Int. incestment & Deteletront Ins. Dieplay Insmethy)
  aninty (" Enter the choice In");
   earl (" +d", I choice);
   ewitch (choice) [
  case 1: printf l'Enter the item to be insented (n')
           lanf (" 1.d", I also Hem);
           ineatron ();
           break:
  case 2: item = delete front (1:
           if (item == -1)
            paintf ("Queue is empty \"):
             elee.
             prints " item deleted is 1.d (n", item).
             break:
core 3: dieplay ();
            Inout:
default: exist (0);
```

```
ueue.c 🗸 multiple priorityq.c 🚜 | input restricted aq.c 🚜 | output restricted aq.c 🚜 | ascending priorityq.c 🚜 | LinearQ.c 🚜 | CircularQ.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;
        q[++rear]=item;
  Fint deletefront(){

if(front>rear)
        if(front>rear){
            front = 0;
            rear = -1;
             return -1;
        return q[front++];
  □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]);
        }
```

```
immelhie humitalie en imhar remierre adie en fonthar remierre adie en avecuantal humitalie en america en enconnece en
     }
     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: printf("Enter the item\n");
                      scanf("%d",&item);
                      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
10
20
30
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 20
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 2
item deleted is 30
1.Insert rear
Delete front
3.Display
4.exit
Enter the choice : 2
queue is empty
1.Insert rear
Delete front
3.Display
4.exit
Enter the choice : 3
queue is empty1.Insert rear
2.Delete front
Display
4.exit
Enter the choice: 4
(program exited with code: 0)
Press any key to continue \dots
```

```
Dequeue.c 🕱 multiple priorityq.c 🕱 input restricted dq.c 🕱 output restricted dq.c 🕱 ascending priorityq.c 🕱 LinearQ.c 🕱 CircularQ.c 🕱
 1
      #include<stdio.h>
 2
      #include<stdlib.h>
 3
      #define que size 3
 4
      int item, front=0, rear=-1, q[que_size], count=0;
 5
      void insertrear()
 6
     ₽{
 7
           if(count==que_size)
 8
 9
               printf("queue overflow");
10
               return;
11
           rear=(rear+1)%que_size;
12
13
           q[rear]=item;
14
           count++;
15
16
     Fint deletefront() {
17
          if(count==0) return -1;
18
           item = q[front];
19
           front=(front+1) %que_size;
20
           count=count-1;
21
           return item;
22
23
     □void displayq() {
24
           int i,f;
25
           if(count==0){
26
               printf("queue is empty");
27
               return;
28
29
           f=front;
30
           printf("contents of queue \n");
31
           for (i=1; i <= count; i++) {
32
               printf("%d\n",q[f]);
33
               f=(f+1)%que_size;
        col: 0 sel: 0 INS TAR mode: CRLE encoding: LITE-8 filetype: C scope: unknown
```

```
Dequetion of Intuitive Priority of Imput restricted upc of Output restricted upc of ascending Priority of Emerge of Circulor of the
27
              return;
28
29
          f=front:
30
         printf("contents of queue \n");
31
          for (i=1; i <= count; i++) {
32
              printf("%d\n",q[f]);
33
              f=(f+1) %que_size;
34
     t,
35
36
    □ int main(){
37
          int choice;
38
          for(;;){
39
              printf("\n1.Insert rear \n2.Delete front \n3.Display \n4.exit \n ");
40
              printf("Enter the choice : ");
              scanf("%d", &choice);
41
42
              switch(choice) {
43
                   case 1:printf("Enter the item to be inserted :");
44
                          scanf("%d",&item);
45
                          insertrear();
46
                          break;
                   case 2:item=deletefront();
47
48
                          if(item==-1)
49
                          printf("queue is empty\n");
50
                          else
51
                          printf("item deleted is %d \n", item);
52
                          break;
53
                   case 3:displayq();
54
                          break;
55
                   default:exit(0);
56
              }
57
          }
58
59
```

## C:\Windows\SYSTEM32\cmd.exe 1.Insert rear 2.Delete front

3.Display 4.exit Enter the choice : 1

Enter the item to be inserted :10

1.Insert rear

Delete front

3.Display

4.exit

Enter the choice : 1

Enter the item to be inserted :20

1.Insert rear

2.Delete front

Display

4.exit

Enter the choice : 1

Enter the item to be inserted :30

1.Insert rear

2.Delete front

Display

4.exit

Enter the choice: 1

Enter the item to be inserted :40

queue overflow

1.Insert rear

Delete front

3.Display

4.exit

Enter the choice : 3

contents of queue

10

20

30

1.Insert rear

2.Delete front

Display

4.exit

Enter the choice : 2

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

```
30
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 20
1.Insert rear
2.Delete front
Display
4.exit
Enter the choice: 2
item deleted is 30
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice :
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
Display
4.exit
Enter the choice: 4
```