

C cir.c > que\_size

```
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     }
12     rear=(rear+1)%que_size;
13     q[rear]=item;
14     count++;
15 }
16 int deletefront()
17 {
18     if(count==0) return -1;
19     item = q[front];
20     front=(front+1)%que_size;
21     count=count-1;
22     return item;
23 }
24 void displayq()
25 {
26     int i,f;
27     if(count==0)
28     {
29         printf("queue is empty");
30         return;
31     }
32     f=front;
33     printf("contents of queue \n");
34     for(i=1;i<=count;i++)
35     {
36         printf("%d\n",q[f]);
37         f=(f+1)%que_size;
38     }
```

```

C dequeue.c  C cir.c  X  C mulpri.c  C despri2.c
C cir.c > que_size
31     }
32     f=front;
33     printf("contents of queue \n");
34     for(i=1;i<=count;i++)
35     {
36         printf("%d\n",q[f]);
37         f=(f+1)%que_size;
38     }
39 }
40 void main()
41 {
42     int choice;
43     for(;;)
44     {
45         printf("\n1.Insert rear \n2.Delete front \n3.Display \n4.exit \n ");
46         printf("Enter the choice :\n ");
47         scanf("%d",&choice);
48         switch(choice)
49         {
50             case 1:printf("Enter the item to be inserted :\n");
51                     scanf("%d",&item);
52                     insertrear();
53                     break;
54             case 2:item=deletefront();
55                     if(item== -1)
56                         printf("queue is empty\n");
57                     else
58                         printf("item deleted is %d \n",item);
59                     break;
60             case 3:displayq();
61                     break;
62             default:exit(0);
63         }
64     }
65     return ;
66 }
67 }

```

```
C cir.c > que_size
1  #include<stdio.h>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: Code

```
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice :
1
Enter the item to be inserted :
10
```

```
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice :
1
Enter the item to be inserted :
20
```

```
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice :
1
Enter the item to be inserted :
30
```

```
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice :
1
Enter the item to be inserted :
40
```

```
queue overflow
1.Insert rear
2.Delete front
```

```
cir.c > que_size
1 #include<stdio.h>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: Code + [ ] [X] ^

```
4.exit
Enter the choice :
1
Enter the item to be inserted :
40
queue overflow
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 :
1
Enter the item to be inserted :
50

1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice :
1
Enter the item to be inserted :
60
```

```
C cir.c > que_size
1  #include<stdio.h>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: Code

```
4.exit
Enter the choice :
2
item deleted is 20

1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice :
1
Enter the item to be inserted :
50

1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice :
1
Enter the item to be inserted :
60

1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice :
3
contents of queue
30
50
60

1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice :
```

## CIRCULAR QUEUE

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

```
#include <conio.h>
```

```
#define Qs 5
```

```
int item, count=0, front=0, rear=-1, q[Qs];
```

```
void insertrear()
```

```
{  
    if (count == Qs)
```

```
{  
    printf("queue overflow");
```

```
    return;
```

```
}
```

```
    rear = (rear + 1) % Qs;
```

```
    q[rear] = item;
```

```
    count++;
```

```
}
```

```
int deletefront()
```

```
{
```

```
    if (count == 0) return -1;
```

```
    item = q[front];
```

```
    front = (front + 1) % Qs;
```

```
    count = count - 1;
```

```
    return item;
```

```
}
```

```
void displayQ()
```

```
{  
    int i, f;
```

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

```
{  
        printf("the queue is empty");
```

```
        return;
```

```
}
```

```
    f = front;
```

```
    printf("the contents of the queue are:");
```

```
    for (i = front; i <= rear; i++)
```

```
{ printf ("%d\n", q[i]);
```

```
    f = (f + 1) % Q;
```

```
}
```

```
{
```

```
void main()
```

```
{ int choice;
```

```
for(;;)
```

```
{ printf ("1: insert from rear\n2: delete from front\n3: display\n4: exit\n");
```

```
printf ("Enter the choice\n");
```

```
scanf ("%d", &choice);
```

```
switch (choice)
```

```
{
```

```
case 1: printf ("Enter the item to be inserted\n");
```

```
scanf ("%d", &item);
```

```
insertrear();
```

```
break;
```

```
case 2: item = deletefront();
```

```
if (item == -1)
```

```
printf ("queue is empty\n");
```

```
else
```

```
printf ("item deleted = %d", item);
```

```
break;
```

```
case 3: display@();
```

```
break;
```

```
default: exit(0);
```

```
}
```

```
}
```

```
}
```