


C linearqueues.c >  displayq()

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  #define que_size 3
4  int item,front=0,rear=-1,q[que_size];
5  void insertrear(){
6      if(rear==que_size-1)
7      {
8          printf("queue overflow");
9          return;
10     }
11
12     q[++rear]=item;
13 }
14 int deletefront(){
15     if(front>rear){
16         front = 0;
17         rear = -1;
18         return -1;
19     }
20     return q[front++];
21 }
22 void displayq(){
23     int i;
24     if(front>rear)
25     {
26         printf("queue is empty");
27         return;
28     }
29     printf("contents of queue \n");
30     for(i=front;i<=rear;i++)
31     {
32         printf("%d\n",q[i]);
33     }
34 }
35 int main(){
36     int choice;
37     for(;;)
38     {
```

C linearqueues.c > displayq()

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

C linearqueues.c > displayq()

27 return;

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: Code

```
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
20
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 1
Enter the item
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 : 3
contents of queue
20
30
1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : █
```

Simple Queue

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

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

```
#define QUEUE_SIZE 5
```

```
int item, front = 0, rear = -1, q[10];
```

```
void insert_rear()
```

```
{ if (rear == QUEUE_SIZE - 1)
```

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

```
return;
```

```
}
```

```
rear = rear + 1;
```

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

```
}
```

```
int delete_front()
```

```
{
```

```
if (front > rear)
```

```
{ front = 0;
```

```
rear = -1;
```

```
return = 1;
```

```
}
```

```
return q[front++];
```

```
}
```

```
void display_Q()
```

```
{ int i;
```

```
if (front > rear)
```

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

```
return;
```

```
}
```

```
printf("contents of queue\n");
```

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

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

```
}
```

```
}
```

void main()

{ int choice;

for (;;))

{ printf (" 1: insert 2: delete 3: display 4: 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);

insert rear();

break;

case 2: item = delete front();

if (item == -1)

printf ("queue is empty\n");

else

printf ("item deleted = %d\n", item);

break;

case 3: display -> @();

break;

case 4: exit;

}

}