

or →

Linear Queue

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

```
#include <stdlib.h>
```

```
#include <process.h>
```

```
#define que-size 3
```

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

```
void insertrear(){
```

```
    if (count == que-size)
```

```
    {
```

```
        printf("Queue overflow");
```

```
        return;
```

```
    }
```

```
    rear = (rear+1) % que-size;
```

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

```
    count++;
```

```
}
```

```
int deletefront() {
```

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

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

```
    front = (front+1) % que-size;
```

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

```
    return item;
```

```
}
```

void displayq;

int i, f;

if (w == 0) {

pf("queue is empty");

return;

}

f = front;

pf("contents of queue \n");

for (i = 0; i <= w; i++) {

pf("%d \n", q[i]);

f = (f + 1) % que-size;

}

}

void main() {

int choice;

for (;;) {

pf("\n 1. Insert rear \n 2. Delete front \n 3. Display

\n 4. exit \n");

pf("Enter the choice: ");

sf("%d", &choice);

switch (choice) {

{

case 1: pf("Enter the item to be inserted: ");

sf("%d", &item);

insert rear();

break;

case 2: item = delete front();

if (item == -1)

pf("queue is empty");

else

pf("item deleted is %d \n", item);

break;

case 3: display q();

break;

default: exit(6);

}

}

}.
}

main.c

```
1- /******
2
3      Online C Compiler.
4      Code, Compile, Run and Debug C program online.
5      Write your code in this editor and press "Run" button to compile and execute it.
6
7      *****/
8
9  #include<stdio.h>
10 #include<stdlib.h>
11
12 #define que_size 3
13 int item,front=0,rear=-1,q[que_size],count=0;
14 void insertrear()
15 {
16     if(count==que_size)
17     {
18         printf("queue overflow");
19         return;
20     }
21     rear=(rear+1)%que_size;
22     q[rear]=item;
23     count++;
24 }
25 int deletefront()
26 {
27     if(count==0) return -1;
28     item = q[front];
29     front=(front+1)%que_size;
30     count=count-1;
31     return item;
32 }
```

main.c

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


main.c

```
47 }
48 }
49 void main()
50 {
51     int choice;
52     for(;;)
53     {
54         printf("\n1.Insert rear \n2.Delete front \n3.Display \n4.exit \n ");
55         printf("Enter the choice : ");
56         scanf("%d",&choice);
57         switch(choice)
58         {
59             case 1:printf("Enter the item to be inserted :");
60                     scanf("%d",&item);
61                     insertrear();
62                     break;
63             case 2:item=deletefront();
64                     if(item==-1)
65                         printf("queue is empty\n");
66                     else
67                         printf("item deleted is %d \n",item);
68                     break;
69             case 3:displayq();
70                     break;
71             default:exit(0);
72         }
73     }
74 }
75 }
76
77
```

```
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
3.Display
4.exit
Enter the choice : 3
contents of queue
```

contents of queue

10
20
30
10

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 : 1

Enter the item to be inserted :35

1.Insert rear
2.Delete front
3.Display
4.exit

Enter the choice : 3

contents of queue

20
30
35
20

1.Insert rear
2.Delete front
3.Display
4.exit


```
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
2.Delete front
3.Display
4.exit
Enter the choice : 2
item deleted is 35

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 finished with exit code 0
Press ENTER to exit console.
```