



Management X

Projects

View:

Search:

doubleEndedQueue.c X circular queue.c X

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<process.h>
4 #define que_size 3
5 int item,front=0,rear=-1,q[que_size],count=0;
6 void insertrear()
7 {
8     if(count==que_size)
9     {
10         printf("queue overflow");
11         return;
12     }
13     rear=(rear+1)%que_size;
14     q[rear]=item;
15     count++;
16 }
17 int deletefront()
18 {
19     if(count==0) return -1;
20     item = q[front];
21     front=(front+1)%que_size;
22     count=count-1;
23     return item;
24 }
25 void displayq()
26 {
27     int i,f;
28     if(count==0)
29     {
```

Management X doubleEndedQueue.c X circular queue.c X

Projects View: Search:

```
23     return item;
24 }
25 void displayq()
26 {
27     int i,f;
28     if(count==0)
29     {
30         printf("queue is empty");
31         return;
32     }
33     f=front;
34     printf("contents of queue \n");
35     for(i=0;i<=count;i++)
36     {
37         printf("%d\n",q[f]);
38         f=(f+1)%que_size;
39     }
40 }
41 void main()
42 {
43     int choice;
44     for(;;)
45     {
46         printf("\n1.Insert rear \n2.Delete front \n3.Display \n4.exit \n ");
47         printf("Enter the choice : ");
48         scanf("%d",&choice);
49         switch(choice)
50     {
51         case 1:printf("Enter the item to be inserted :");
52         scanf("%d",&item);
```

The screenshot shows a C programming environment with the following details:

- Toolbar:** Standard icons for file operations (New, Open, Save, Print, Find, Replace, etc.), project management, and code editing.
- Management Bar:** Shows "Management X" and two open files: "doubleEndedQueue.c X" and "circular queue.c X".
- Projects View:** A tree view showing the project structure.
- View:** Set to "Text Editor".
- Search:** A search bar.
- Code Editor:** Displays the following C code for a double-ended queue:

```
41 void main()
42 {
43     int choice;
44     for(;;)
45     {
46         printf("\n1.Insert rear \n2.Delete front \n3.Display \n4.exit \n ");
47         printf("Enter the choice : ");
48         scanf("%d",&choice);
49         switch(choice)
50         {
51             case 1:printf("Enter the item to be inserted :");
52             scanf("%d",&item);
53             insertrear();
54             break;
55             case 2:item=deletefront();
56             if(item==-1)
57                 printf("queue is empty\n");
58             else
59                 printf("item deleted is %d \n",item);
60             break;
61             case 3:displayq();
62             break;
63             default:exit(0);
64         }
65     }
66 }
67
68
69
```

The code implements a menu-driven program for a double-ended queue. It uses a switch statement to handle four choices: inserting at the rear, deleting from the front, displaying the queue, and exiting. The queue is implemented using a circular array.

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 :60

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

Enter the choice : 3

contents of queue

20

30

60

20

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 : 3

contents of queue

30

60

20

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

Enter the choice :

The screenshot shows the wxSmith IDE interface with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save, Print), search, and various tools.
- Menu Bar:** Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, Help.
- Toolbars:** Top toolbar with project-related icons like build status, file operations, and search.
- Status Bar:** Shows "goal>" and "delete_front():void".
- Project Explorer:** Shows "doubleEndedQueue.c X" and "circular queue.c X".
- Code Editor:** Displays the C code for a circular queue implementation. The code includes declarations for stdio.h, conio.h, process.h, and defines qsize as 5. It uses global variables f=0, r=-1, ch, and item, and an array q[10]. The code implements functions for checking if the queue is full or empty, inserting items at the rear, and deleting items from the front.

```
1 #include<stdio.h>
2 #include<conio.h>
3 #include<process.h>
4 #define qsize 5
5 int f=0,r=-1,ch;
6 int item,q[10];
7
8 int isfull()
9 {
10     return(r==qsize-1)?1:0;
11 }
12 int isempty()
13 {
14     return(f>r)?1:0;
15 }
16 void insert_rear()
17 {
18     if(isfull())
19     {
20         printf("queue overflow\n");
21         return;
22     }
23     r=r+1;
24     q[r]=item;
25 }
26 void delete_front()
27 {
28     if(isempty())
29     {
```

management X

doubleEndedQueue.c X circular queue.c X

Projects

View:



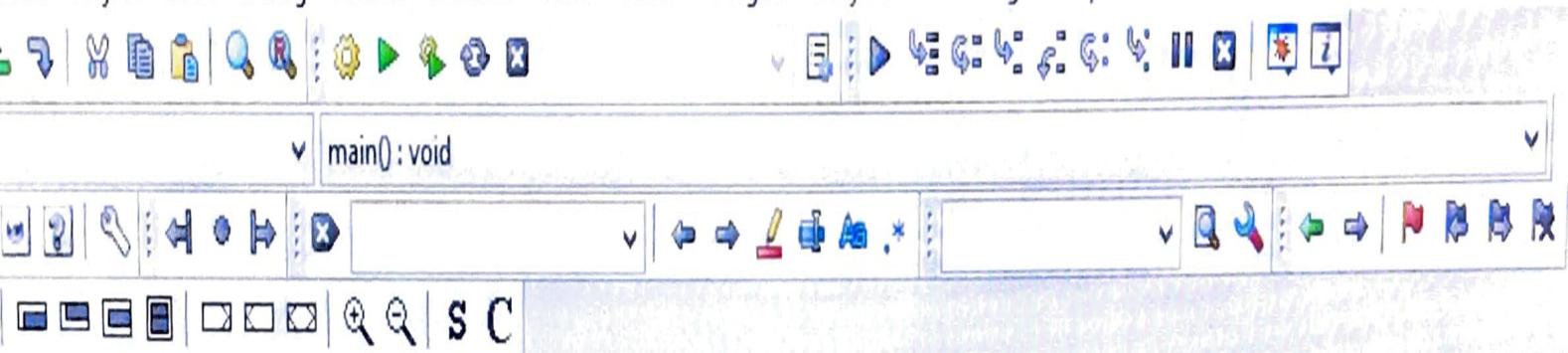
Search:



```
29     {
30         printf("queue empty\n");
31         return;
32     }
33     printf("item deleted is %d\n",q[(f)++]);
34     if(f>r)
35     {
36         f=0;
37         r=-1;
38     }
39 }
40 void insert_front()
41 {
42     if(f!=0)
43     {
44         f=f-1;
45         q[f]=item;
46         return;
47     }
48     else if((f==0)&&(r==-1))
49     {
50         q[++(r)]=item;
51         return;
52     }
53     else
54         printf("insertion not possible\n");
55 }
56 void delete_rear()
57 {
```

The screenshot shows a software development environment with a toolbar at the top and a sidebar on the left. The main workspace displays two tabs: 'doubleEndedQueue.c' and 'circular queue.c'. The code for 'doubleEndedQueue.c' is visible:

```
47     }
48     else if( (f==0) && (r==-1) )
49     {
50         q[ ++(r) ]=item;
51         return;
52     }
53     else
54         printf("insertion not possible\n");
55     }
56     void delete_rear()
57     {
58         if(isempty())
59         {
60             printf("queue is empty\n");
61             return;
62         }
63         printf("item deleted is %d\n",q[ (r)-- ]);
64         if(f>r)
65         {
66             f=0;
67             r=-1;
68         }
69     }
70     void display()
71     {
72         int i;
73         if(isempty())
74         {
75             printf("queue empty\n");
76         }
77     }
78 }
```



doubleEndedQueue.c X circular queue.c X

```
76     return;
77 }
78 for(i=f;i<=r;i++)
79     printf("%d\n",q[i]);
80 }
81 void main()
82 {for(;;)
83 {
84     printf("1.insert_rear\n2.insert_front\n3.delete_rear\n4.delete_front\n5.display\n6.exit\n");
85     printf("enter choice\n");
86     scanf("%d",&ch);
87     switch(ch)
88     {
89         case 1:printf("enter the item\n");
90                 scanf("%d",&item);
91                 insert_rear();
92                 break;
93         case 2:printf("enter the item\n");
94                 scanf("%d",&item);
95                 insert_front();
96                 break;
97         case 3:delete_rear();
98                 break;
99         case 4:delete_front();
100                break;
101        case 5:display();
102                break;
103        default:exit(0);
104    }}}}
```

```
1.insert_rear  
2.insert_front  
3.delete_rear  
4.delete_front  
5.display  
6.exit  
enter choice  
1  
enter the item  
25  
1.insert_rear  
2.insert_front  
3.delete_rear  
4.delete_front  
5.display  
6.exit  
enter choice  
1  
enter the item  
45  
1.insert_rear  
2.insert_front  
3.delete_rear  
4.delete_front  
5.display  
6.exit  
enter choice  
2  
enter the item  
60  
insertion not possible  
1.insert_rear  
2.insert_front  
3.delete_rear  
4.delete_front  
5.display  
6.exit  
enter choice  
1  
enter the item  
58  
1.insert_rear  
2.insert_front  
3.delete_rear  
4.delete_front  
5.display  
6.exit  
enter choice  
2  
enter the item  
30  
insertion not possible  
1.insert_rear  
2.insert_front  
3.delete_rear  
4.delete_front  
5.display
```

```
5.display
6.exit
enter choice
1
enter the item
58
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
2
enter the item
30
insertion not possible
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
25
45
58
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 25
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
```