

The screenshot shows a web browser window with multiple tabs open. The active tab is 'Online C Compiler - online editor'. The code editor displays the following C program:

```
main.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 }
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--;
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=0;i<count;i++)
35     {
36         printf("%d\n",q[i]);
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 : ");
47         scanf("%d",&choice);
48         switch(choice)
49         {
50             case 1:printf("Enter the item to be inserted :");
51                 scanf("%d",&item);
52                 insertrear();
```

The screenshot shows the same web browser window with the 'Online C Compiler - online editor' tab active. The code editor now contains the complete C program for a queue implementation with a menu:

```
main.c
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=0;i<count;i++)
35     {
36         printf("%d\n",q[i]);
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 : ");
47         scanf("%d",&choice);
48         switch(choice)
49         {
50             case 1:printf("Enter the item to be inserted :");
51                 scanf("%d",&item);
52                 insertrear();
```

The screenshot shows a web browser window with multiple tabs open. The active tab is "Online C Compiler - online editor". The code in the editor is:

```
main.c
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 : ");
47         scanf("%d",&choice);
48         switch(choice)
49         {
50             case 1:printf("Enter the item to be inserted :");
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     getch();
66 }
```

The status bar at the bottom right shows the time as 2:47 PM and the date as 10/19/2020.

The screenshot shows the same web browser window after running the code. The output pane displays the program's interaction with the user:

```
input
main.c:65:2: warning: implicit declaration of function 'getch' [-Wimplicit-function-declaration]

1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 1
Enter the item to be inserted :89

1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 2
item deleted is 89

1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 1
Enter the item to be inserted :89
```

The status bar at the bottom right shows the time as 2:48 PM and the date as 10/19/2020.

(no subject) - pranavkumar.cs19@... | Meet - 3C DS lab Monday 2... | Online C Compiler - online editor | Online C Compiler - online editor | +

Incognito (3) ::

onlinegdb.com/online_c_compiler

Gmail YouTube Maps Netflix Online Shopping sit... Login + Instagram Login on Twitter / T... Facebook - log in o... Welcome to Prime... Physics Online Quiz...

input

```
3.Display
4.exit
Enter the choice : 1
Enter the item to be inserted :89

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 :55

1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 1
Enter the item to be inserted :63
queue overflow
1.Insert rear
2.Delete front
3.Display
```

2:48 PM 10/19/2020

(no subject) - pranavkumar.cs19@... | Meet - 3C DS lab Monday 2... | Online C Compiler - online editor | Online C Compiler - online editor | +

Incognito (3) ::

onlinegdb.com/online_c_compiler

Gmail YouTube Maps Netflix Online Shopping sit... Login + Instagram Login on Twitter / T... Facebook - log in o... Welcome to Prime... Physics Online Quiz...

input

```
55
89

1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 1
Enter the item to be inserted :45

1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 3
contents of queue
20
55
45
20

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

...Program finished with exit code 9
Press ENTER to exit console.
```

2:49 PM 10/19/2020

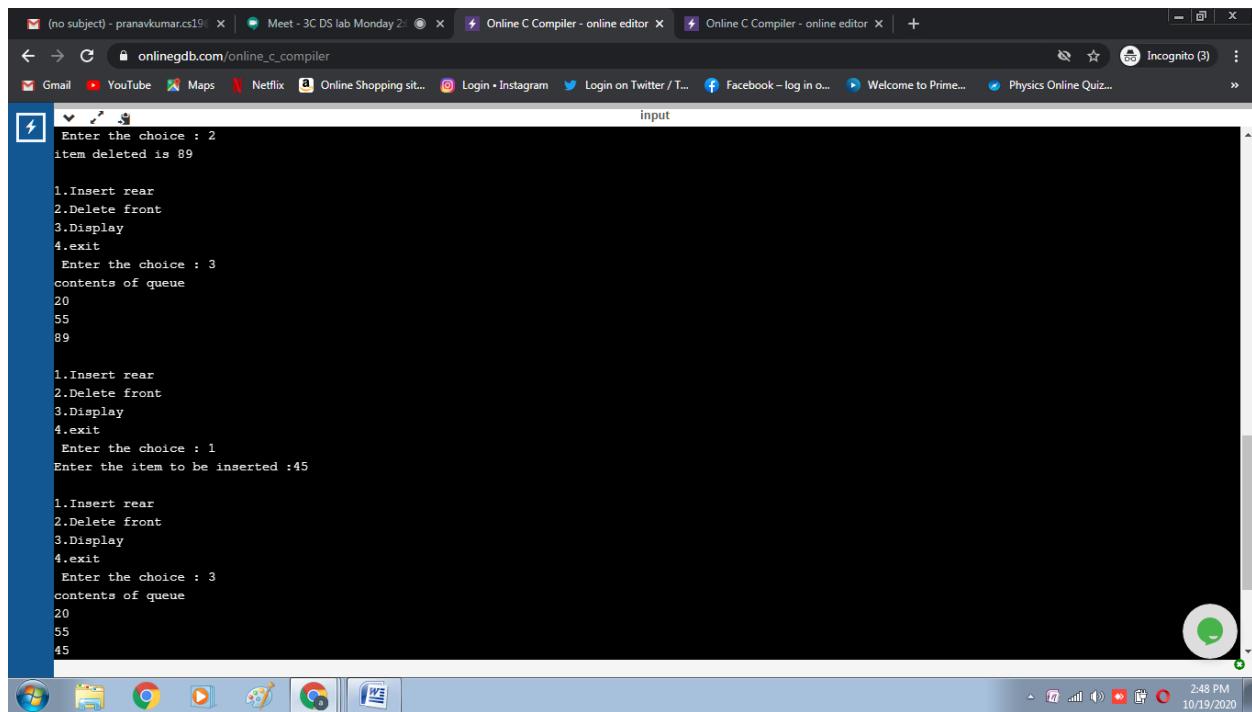
Circular queue

DATE _____
PAGE 77

```
#include <stdio.h>
#include <stdlib.h>
#define que_size 3
Pnt item, front = 0, rear = -1, q[que_size], count = 0;
void insertrear()
{
    if (count == que_size)
        printf("queue overflow");
    else
        q[rear] = item;
    rear = (rear + 1) % que_size
    count++;
}
int deletefront()
{
    if (count == 0) return -1;
    item = q[front];
    front = (front + 1) % que_size;
    count = count - 1;
    return item;
}
void displayq()
{
    Pnt i;
    if (count == 0)
        printf("queue is empty");
    else
        for (i = front; i != front; i++)
            printf("%d", q[i]);
}
```



```
    printf ("Contents of queue \n");
    for (i=0, i<= count, i++)
    {
        printf ("%d \n", q[i]);
        i = i + 1;
    }
    void main()
    {
        int choice;
        for (j);
        {
            printf ("1. Insert rear \n 2. delete front
                    \n 3. Display \n 4. exit \n");
            printf ("Enter the choice : ");
            scanf ("%d", &choice);
            switch (choice)
            {
                case 1: printf ("Enter the item to be inserted : ");
                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: display();
                break;
                default: exit(0);
            }
        }
        getch();
    }
```

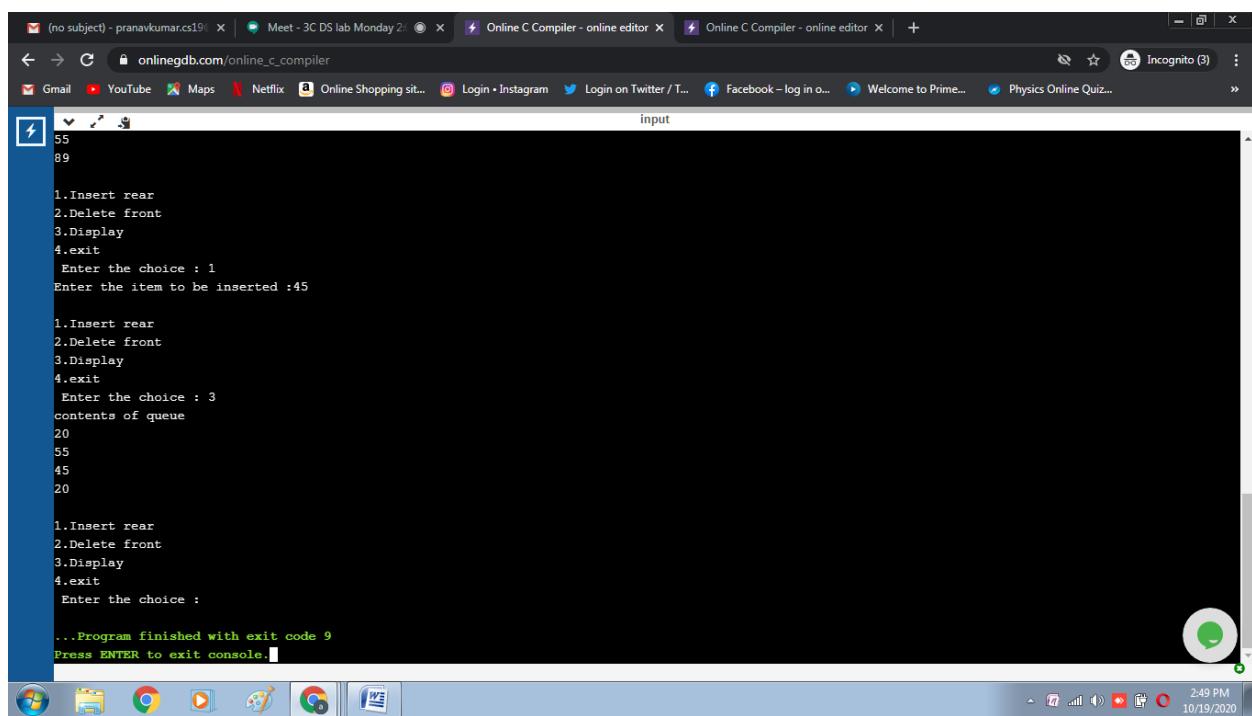


```
Enter the choice : 2
item deleted is 89

1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 3
contents of queue
20
55
89

1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 1
Enter the item to be inserted :45

1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 3
contents of queue
20
55
45
```



```
55
89

1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 1
Enter the item to be inserted :45

1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 3
contents of queue
20
55
45
20

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

...Program finished with exit code 9
Press ENTER to exit console.
```

The screenshot shows a web browser window with multiple tabs open. The active tab is "Online C Compiler - online editor". The code editor displays the following C code for a queue implementation:

```
main.c
1 #include<stdio.h>
2 #include<stdlib.h>
3 #define qsize 5
4 int f=0,r=-1,ch;
5 int item,q[10];
6
7 int isfull()
8 {
9     return(r==qsize-1)?1:0;
10 }
11 int isempty()
12 {
13     return(f>r)?1:0;
14 }
15 void insert_rear()
16 {
17     if(isfull())
18     {
19         printf("queue overflow\n");
20         return;
21     }
22     r=r+1;
23     q[r]=item;
24 }
25 void delete_front()
26 {
27     if(isempty())
28     {
29         printf("queue empty\n");
30         return;
31 }
```

The screenshot shows the same web browser window with the code editor displaying the continuation of the C code:

```
main.c
23     q[r]=item;
24 }
25 void delete_front()
26 {
27     if(isempty())
28     {
29         printf("queue empty\n");
30         return;
31     }
32     printf("item deleted is %d\n",q[(f++)]);
33     if(f>r)
34     {
35         f=0;
36         r=-1;
37     }
38 }
39 void insert_front()
40 {
41     if(f!=0)
42     {
43         f=f-1;
44         q[f]=item;
45         return;
46     }
47     else if((f==0)&&(r==1))
48     {
49         q[++(r)]=item;
50     }
51 }
52 }
```

The screenshot shows a web browser window with multiple tabs open. The active tab is 'Online C Compiler - online editor'. The code in the editor is as follows:

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

The screenshot shows the same web browser window with the code now fully implemented. The code in the editor is as follows:

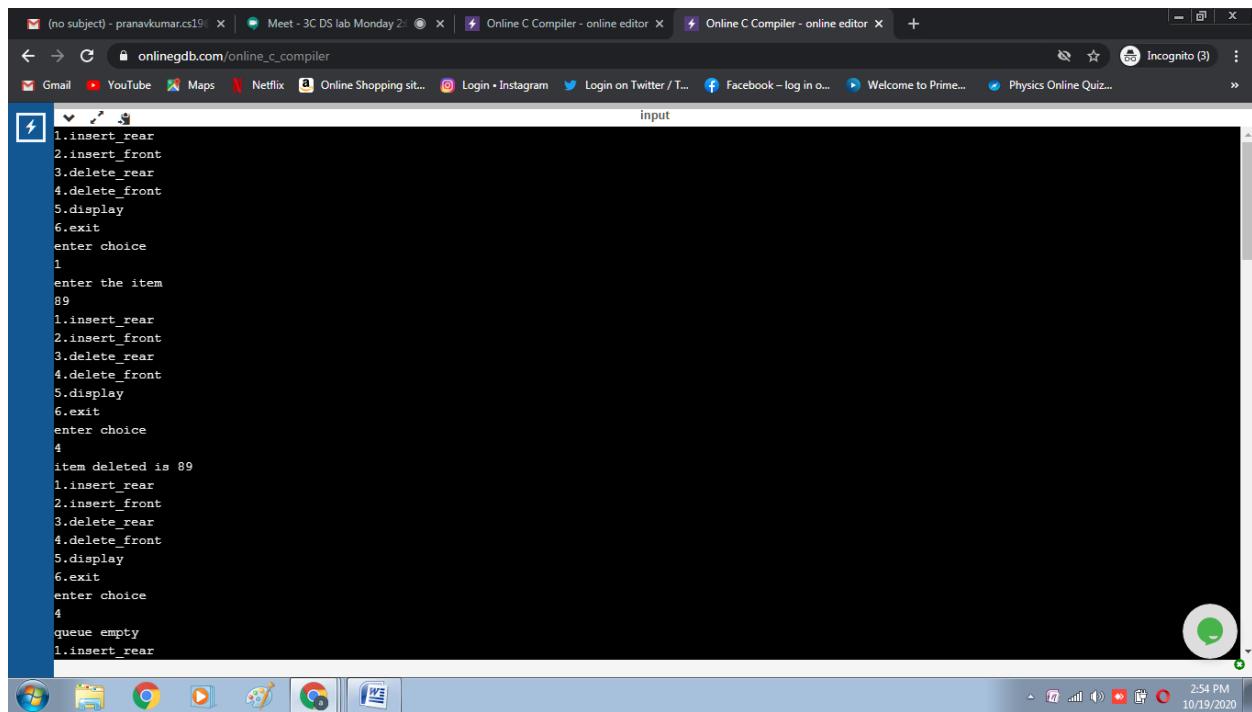
```
main.c
67     }
68 }
69 void display()
70 {
71     int i;
72     if(isempty())
73     {
74         printf("queue empty\n");
75         return;
76     }
77     for(i=f;i<=r;i++)
78     printf("%d\n",q[i]);
79 }
80 int main()
81 {
82     for(;;)
83     {
84         printf("1.insert_rear\n2.insert_front\n3.delete_rear\n4.delete_front\n5.display\n6.exit\n");
85         print("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;
```

The screenshot shows a web browser window with multiple tabs open. The active tab is "Online C Compiler - online editor". The page displays a C program named "main.c" which implements a circular queue using arrays and switch statements. The code includes functions for insertion at rear, deletion from front, and displaying the queue. The browser's toolbar includes "Run", "Debug", "Stop", "Save", and "Beautify" buttons. The status bar at the bottom right shows the date and time: 10/19/2020, 2:53 PM.

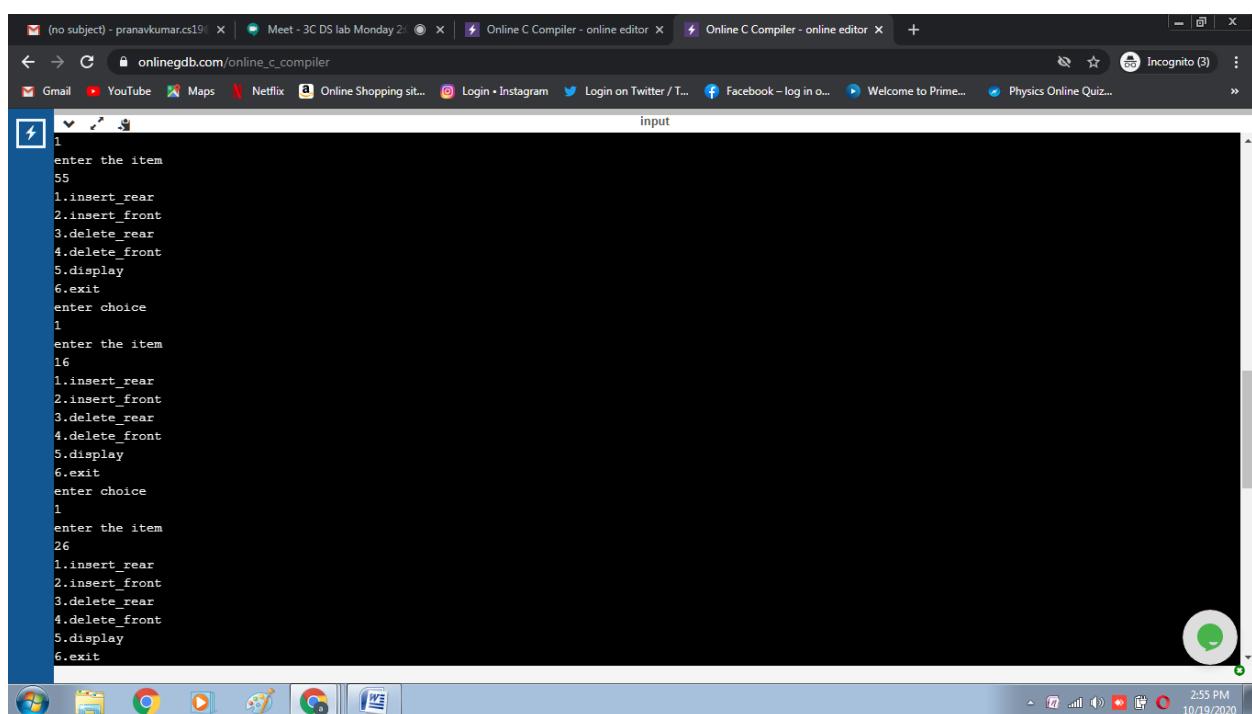
```
main.c
1 // for(i=f;i<=r;i++)
2 printf("%d\n",q[i]);
3
4 int main()
5 {
6     for(;;)
7     {
8         printf("1.insert_rear\n2.insert_front\n3.delete_rear\n4.delete_front\n5.display\n6.exit\n");
9         printf("enter choice\n");
10        scanf("%d",&ch);
11        switch(ch)
12        {
13            case 1:printf("enter the item\n");
14            scanf("%d",&item);
15            insert_rear();
16            break;
17            case 2:printf("enter the item\n");
18            scanf("%d",&item);
19            insert_front();
20            break;
21            case 3:delete_rear();
22            break;
23            case 4:delete_front();
24            break;
25            case 5:display();
26            break;
27            default:exit(0);
28        }
29    }
30 }
```

```
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
89
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 89
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
queue empty
1.insert_rear
```

```
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
89
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 89
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
queue empty
1.insert_rear
```



```
input
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
89
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 89
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
queue empty
1.insert_rear
```



```
input
1
enter the item
55
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
16
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
26
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
```

dequeue

```
#include <stdio.h>
#include <stdlib.h>
#define qsize 5.
int f=0, r=-1, ch;
int c[5], q[10];
```

```
int isfull()
{ return (r==q[qsize-1]) ? 1:0
}
```

```
int isempty()
{ return (f>r) ? 1:0
}
```

```
void insert_rear()
{ if (isfull())
    { printf("queue overflow\n");
    return;
}
```

```
    r=r+1;
    q[r]=c[r];
}
```

```
void delete_front()
{ if (isempty())
    { printf("queue empty\n");
    return;
}
```

```
printf("element deleted is %d\n", q[f++]);
if (f>r)
{ f=0;
    r=r-1;
}
```

```
void insert_front ()  
{ if (f == 0)  
    { r = f - 1;  
     q[f] = item;  
     return;  
    }
```

```
else if ((f == 0) && (r == -1))  
    { q[++(r)] = item;  
     return;  
    }
```

else

```
printf ("Insertion not possible\n");  
}
```

```
void delete_rear ()
```

```
{ if (isempty ())  
    printf ("queue is empty\n");  
    return;  
}
```

```
printf ("element deleted is %d\n", q[r-1]);  
if (f > r)
```

```
{ f = 0;  
r = -1;
```

}

}

```
void display()
```

```
{ int i;
```

```
if (isempty ())
```

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

}

```
for (i=f, i<=n; i++)  
    printf ("%d\n", q[i]);
```

One main()

{ for (j; j)

{ printf ("1. insert-rear\n 2. insert-front\n 3. delete-rear\n 4. delete-front\n 5. display\n 6. exit\n");

scanf ("%d", &ch);

switch (ch)

{ case 1: printf ("enter the item\n");

scanf ("%d", &item);

insert_rear();

break;

case 2: printf ("enter the item\n");

scanf ("%d", &item);

insert_front();

break;

case 3: delete_rear();

break;

case 4: delete_front();

break;

case 5: display();

break;

default: exit(0);

3

3

3

The screenshot shows a web browser window with an online C compiler. The code input field contains the following C program:

```
2.insert_front  
3.delete_rear  
4.delete_front  
5.display  
6.exit  
enter choice  
5  
89  
20  
55  
16  
26  
1.insert_rear  
2.insert_front  
3.delete_rear  
4.delete_front  
5.display  
6.exit  
enter choice  
2  
enter the item  
52  
insertion not possible  
1.insert_rear  
2.insert_front  
3.delete_rear  
4.delete_front  
5.display  
6.exit
```

The output window displays the execution results of the program. The system prompt "input" is visible at the top of the output area.