

The screenshot shows a web-based C debugger interface. The top navigation bar includes tabs for "DATA STRUCTURES 3B" and three "GDB online Debugger | Compile" tabs. The main window displays a code editor for "main.c" with the following content:

```
1 #include<stdio.h>
2 #include<conio.h>
3 #include<stdlib.h>
4 #define qsize 4
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
17
18 void insert_rear()
19 {
20     if(isfull())
21     {
22         printf("Queue overflow\n");
23         return;
24     }
25     r=r+1;
26     q[r]=item;
27 }
28
29
30 void delete_front()
31 {
```

The code implements a queue using arrays. It defines constants for queue size (qsize) and initializes variables f (front), r (rear), and q (queue array). It includes functions for checking if the queue is full or empty, and for inserting an item at the rear and deleting an item from the front.

DATA STRUCTURES 3B X GDB online Debugger | Compil X GDB online Debugger | Compil X GDB online Debugger | Compil X +

← → ⌂ ⌄ https://www.onlinegdb.com 110% ⌂ ⌄ ⌄ Language C ⌄ ⓘ ⌄

Run Debug Stop Share Save Beautify Ctrl+S

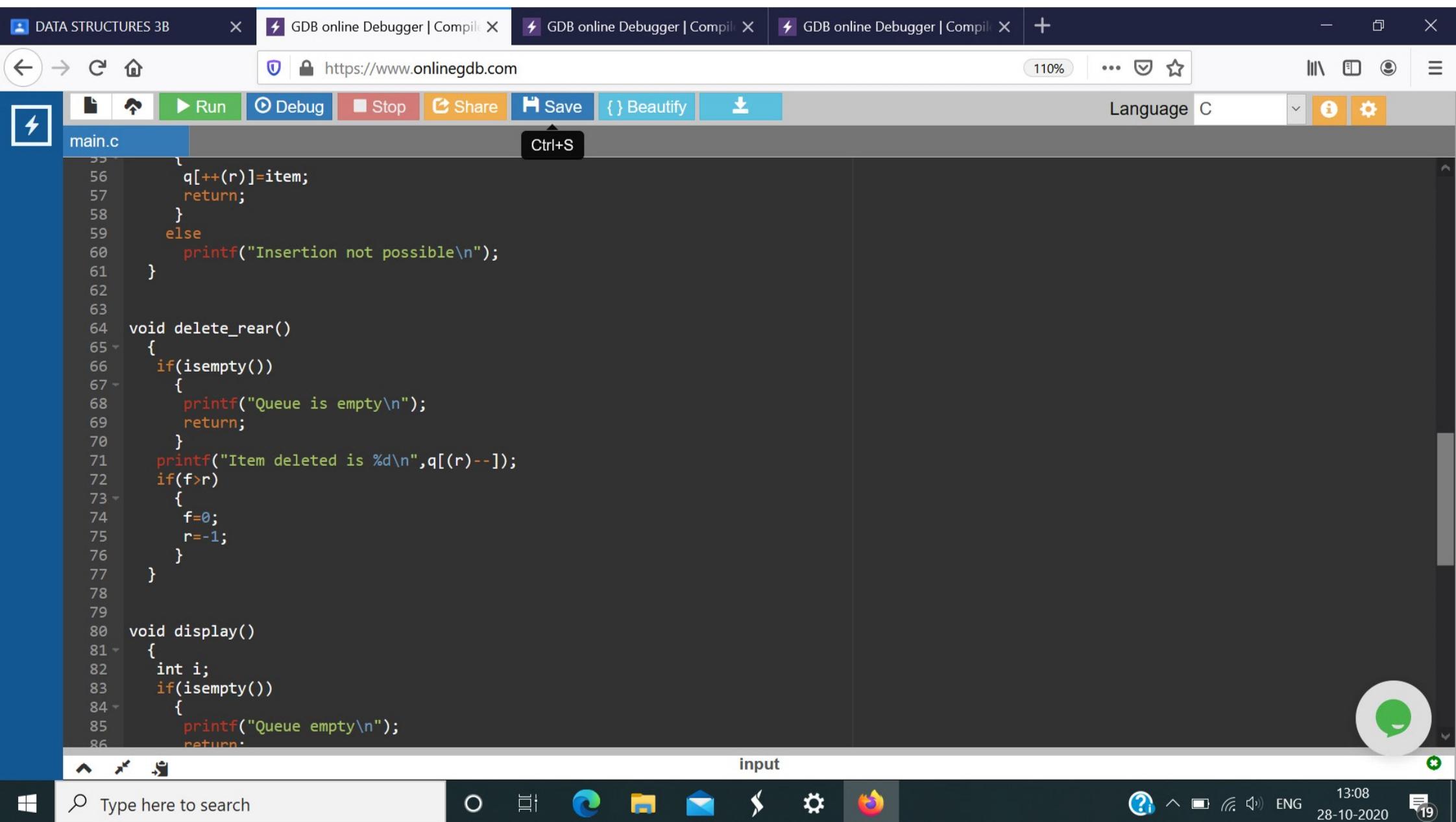
main.c

```
27 }
28
29
30 void delete_front()
31 {
32     if(isempty())
33     {
34         printf("Queue empty\n");
35         return;
36     }
37     printf("Item deleted is %d\n",q[(f)++]);
38     if(f>r)
39     {
40         f=0;
41         r=-1;
42     }
43 }
44
45
46 void insert_front()
47 {
48     if(f!=0)
49     {
50         f=f-1;
51         q[f]=item;
52         return;
53     }
54     else if((f==0)&&(r==-1))
55     {
56         q[++(r)]=item;
57         return;
58 }
```

input

Type here to search

13:08 28-10-2020 19



DATA STRUCTURES 3B X GDB online Debugger | Compil X GDB online Debugger | Compil X GDB online Debugger | Compil X +

https://www.onlinegdb.com 110% ... 🌐 ⚡

Run Debug Stop Share Save { } Beautify Ctrl+S Language C

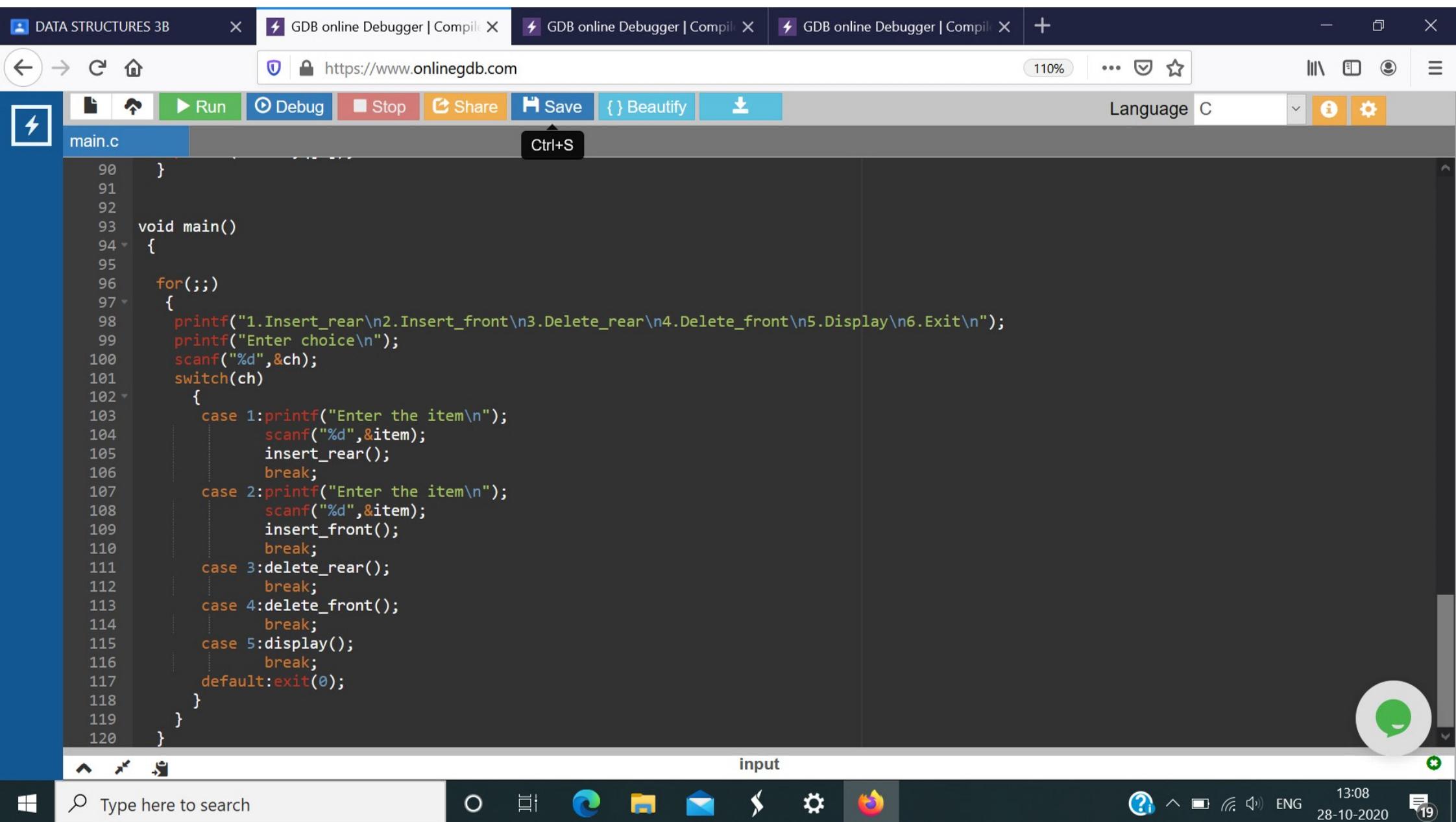
main.c

```
84     {
85         printf("Queue empty\n");
86         return;
87     }
88     for(i=f;i<=r;i++)
89         printf("%d\n",q[i]);
90     }
91
92
93 void main()
94 {
95
96     for(;;)
97     {
98         printf("1.Insert_rear\n2.Insert_front\n3.Delete_rear\n4.Delete_front\n5.Display\n6.Exit\n");
99         printf("Enter choice\n");
100        scanf("%d",&ch);
101        switch(ch)
102        {
103            case 1:printf("Enter the item\n");
104                scanf("%d",&item);
105                insert_rear();
106                break;
107            case 2:printf("Enter the item\n");
108                scanf("%d",&item);
109                insert_front();
110                break;
111            case 3:delete_rear();
112                break;
113            case 4:delete_front();
114                break;
115        }
116    }
117 }
```

input

Type here to search

13:08 28-10-2020 19



DATA STRUCTURES 3B X GDB online Debugger | Compil X GDB online Debugger | Compil X GDB online Debugger | Compil X +

← → ⌂ ⌄ ⌁ https://www.onlinegdb.com 110% ⌂ ⌄ ⌁

input

```
1.Insert_rear
2.Insert_front
3.Delete_rear
4.Delete_front
5.Display
6.Exit
Enter choice
1
Enter the item
20
1.Insert_rear
2.Insert_front
3.Delete_rear
4.Delete_front
5.Display
6.Exit
Enter choice
1
Enter the item
30
1.Insert_rear
2.Insert_front
3.Delete_rear
4.Delete_front
5.Display
6.Exit
```

13:08 28-10-2020 ENG 19

DATA STRUCTURES 3B X GDB online Debugger | Compil X GDB online Debugger | Compil X GDB online Debugger | Compil X +

← → ⌂ ⌄ ⌁ https://www.onlinegdb.com 110% ⌂ ⌄ ⌁

input

```
5.Display
6.Exit
Enter choice
5
30
1.Insert_rear
2.Insert_front
3.Delete_rear
4.Delete_front
5.Display
6.Exit
Enter choice
3
Item deleted is 30
1.Insert_rear
2.Insert_front
3.Delete_rear
4.Delete_front
5.Display
6.Exit
Enter choice
3
Queue is empty
1.Insert_rear
2.Insert_front
3.Delete_rear
```

13:09 28-10-2020 ENG 19

DATA STRUCTURES 3B X GDB online Debugger | Compil X GDB online Debugger | Compil X GDB online Debugger | Compil X +

← → ⌂ ⌄ ⌁ https://www.onlinegdb.com 110% ⌂ ⌄ ⌁

input

```
5.Display
6.Exit
Enter choice
3
Item deleted is 30
1.Insert_rear
2.Insert_front
3.Delete_rear
4.Delete_front
5.Display
6.Exit
Enter choice
3
Queue is empty
1.Insert_rear
2.Insert_front
3.Delete_rear
4.Delete_front
5.Display
6.Exit
Enter choice
6

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

13:09 28-10-2020 19

⚡ input_dequeue - GDB online De X +

Back Forward Home https://www.onlinegdb.com/edit/By5j0KUOD 110% ⋮ 🔍 ⚡

Run Debug Stop Share Save { } Beautify Language C ⚡ ⓘ ⚡

main.c

```
1 #include<stdio.h>
2 #include<conio.h>
3 #include<stdlib.h>
4 #define qsize 4
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
17
18 void insert_rear()
19 {
20     if(isfull())
21     {
22         printf("Queue overflow\n");
23         return;
24     }
25     r=r+1;
26     q[r]=item;
27 }
28
29
30 void delete_front()
31 {
```

input

Type here to search

19:17 28-10-2020

The screenshot shows a web-based C debugger interface from [onlinegdb.com](https://www.onlinegdb.com). The code in the editor is for a queue implementation, specifically a circular queue. It includes functions for inserting and deleting elements at the front of the queue. The code uses global variables `f` and `r` to track the front and rear indices of the queue array `q`.

```
29 void delete_front()
30 {
31     if(isempty())
32     {
33         printf("Queue empty\n");
34         return;
35     }
36     printf("Item deleted is %d\n", q[(f++)]);
37     if(f>r)
38     {
39         f=0;
40         r=-1;
41     }
42 }
43
44
45
46 /*void insert_front()
47 {
48     if(f!=0)
49     {
50         f=f-1;
51         q[f]=item;
52         return;
53     }
54     else if((f==0)&&(r==-1))
55     {
56         q[++(r)]=item;
57         return;
58     }
59 }
```

⚡ input_dequeue - GDB online De X +

https://www.onlinegdb.com/edit/By5j0KUOD 110% ⋮ 🔍 ⚡

Run Debug Stop Share Save Beautify Language C ⚡ 🔍 ⚡

main.c

```
58     }
59     else
60         printf("Insertion not possible\n");
61 } */
62
63
64 void delete_rear()
65 {
66     if(isempty())
67     {
68         printf("Queue is empty\n");
69         return;
70     }
71     printf("Item deleted is %d\n",q[(r)-]);
72     if(f>r)
73     {
74         f=0;
75         r=-1;
76     }
77 }
78
79
80 void display()
81 {
82     int i;
83     if(isempty())
84     {
85         printf("Queue empty\n");
86         return;
87     }
88     for(i=f;i<=r;i++)
89         printf("%d\n", q[i]).
```

input

Type here to search

19:17 28-10-2020 ENG 19

⚡ input_dequeue - GDB online De X +

https://www.onlinegdb.com/edit/By5j0KUOD 90% ⋮ 🔍 ⚡

Run Debug Stop Share Save { } Beautify Language C ⚡

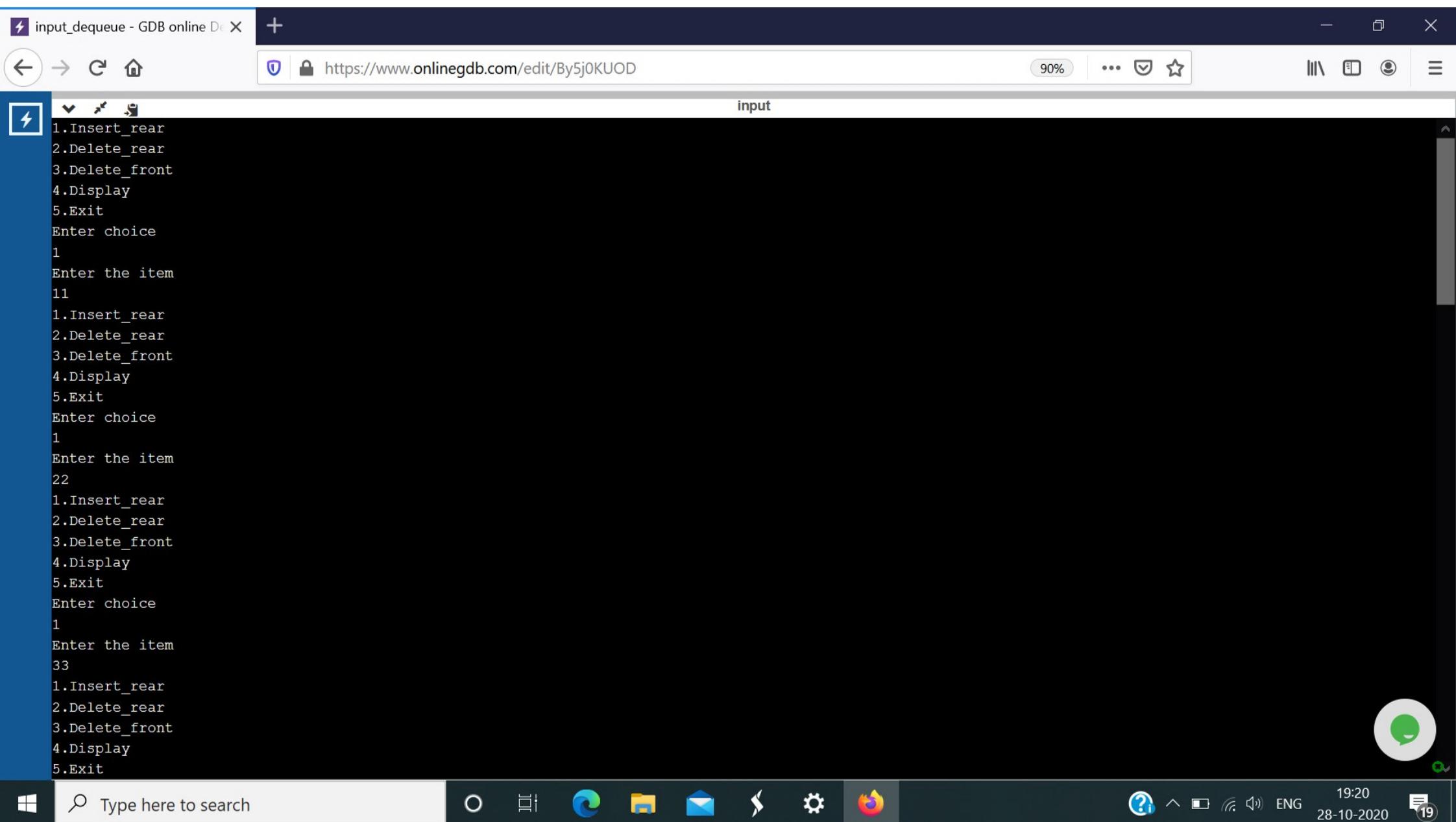
main.c

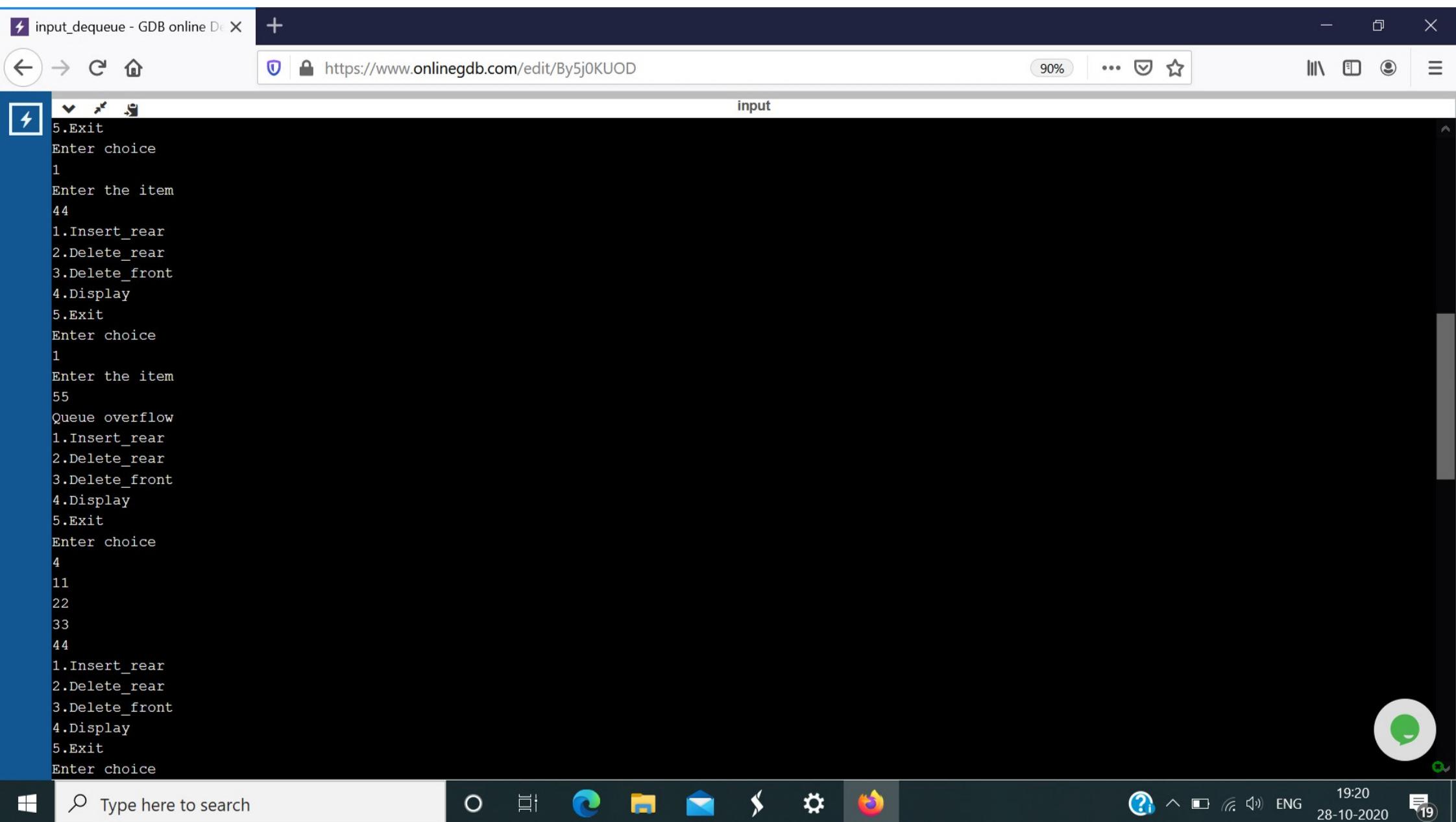
```
82     int i;
83     if(isempty())
84     {
85         printf("Queue empty\n");
86         return;
87     }
88     for(i=f;i<=r;i++)
89         printf("%d\n",q[i]);
90     }
91
92
93 void main()
94 {
95
96     for(;;)
97     {
98         printf("1.Insert_rear\n2.Delete_rear\n3.Delete_front\n4.Display\n5.Exit\n");
99         printf("Enter choice\n");
100        scanf("%d",&ch);
101        switch(ch)
102        {
103            case 1:printf("Enter the item\n");
104                scanf("%d",&item);
105                insert_rear();
106                break;
107            /*case 2:printf("Enter the item\n");
108                scanf("%d",&item);
109                insert_front();
110                break;*/
111            case 2:delete_rear();
112                break;
113            case 3:delete_front();
114                break;
115            case 4:display();
116                break;
117            default:exit(0);
118        }
119    }
120
121 }
```

input

Type here to search

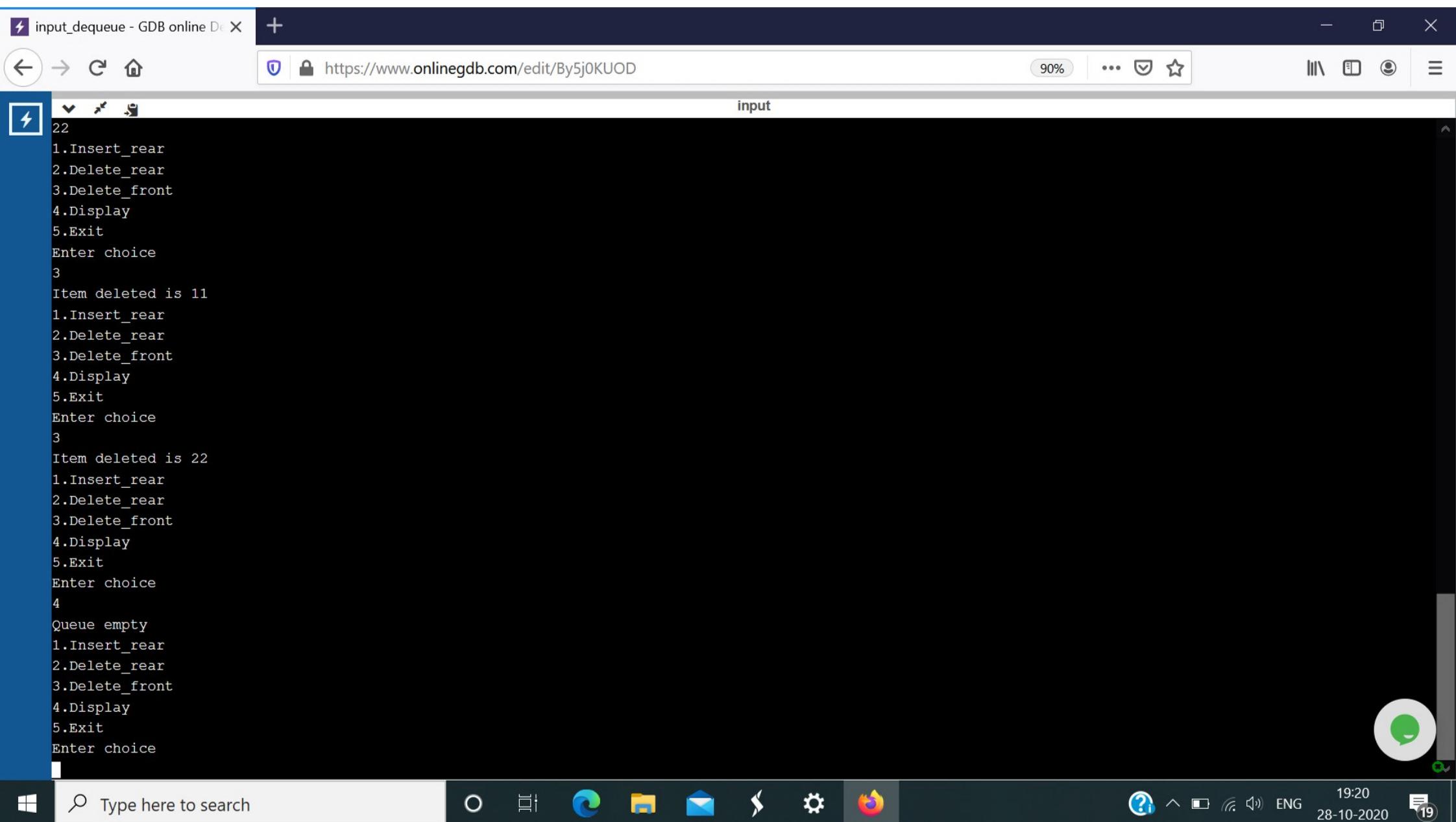
19:18 28-10-2020 ENG 19





The screenshot shows a browser window with the URL <https://www.onlinegdb.com/edit/By5j0KUOD>. The page title is "input_dequeue - GDB online D". The main content area displays a GDB session for a C program. The session starts with a menu of options: 1.Insert_rear, 2.Delete_rear, 3.Delete_front, 4.Display, and 5.Exit. The user enters choice 2, which results in the deletion of item 44. The session then loops back to the menu. The user enters choice 2 again, resulting in the deletion of item 33. The session loops back to the menu. The user enters choice 4, displaying the list [11, 22]. The session loops back to the menu. The user enters choice 3, resulting in the deletion of item 11. The session loops back to the menu. The user enters choice 1, inserting item 11 at the rear. The session loops back to the menu. The user enters choice 3, resulting in the deletion of item 11. The session loops back to the menu. The user enters choice 5, exiting the program.

```
input
5.Exit
Enter choice
2
Item deleted is 44
1.Insert_rear
2.Delete_rear
3.Delete_front
4.Display
5.Exit
Enter choice
2
Item deleted is 33
1.Insert_rear
2.Delete_rear
3.Delete_front
4.Display
5.Exit
Enter choice
4
11
22
1.Insert_rear
2.Delete_rear
3.Delete_front
4.Display
5.Exit
Enter choice
3
Item deleted is 11
1.Insert_rear
2.Delete_rear
3.Delete_front
```



The screenshot shows a web-based C debugger interface. The top bar includes a title 'output_dequeue - GDB online X', a search bar with the URL 'https://www.onlinegdb.com/edit/B1jdqqlOP', and various browser controls. Below the header is a toolbar with icons for Run, Debug, Stop, Share, Save, Beautify, and a download arrow. The main area displays the 'main.c' file content:

```
1 #include<stdio.h>
2 #include<conio.h>
3 #include<stdlib.h>
4 #define qsize 4
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
17
18 void insert_rear()
19 {
20     if(isfull())
21     {
22         printf("Queue overflow\n");
23         return;
24     }
25     r=r+1;
26     q[r]=item;
27 }
28
29
30 void delete_front()
31 {
32     if(isempty())
33     {
34         printf("Queue empty\n");
35         return;
36     }
37     printf("Item deleted is %d\n",q[(f)+]);
38     if(f>r)
39     {
40         f=0;
41     }
42 }
```

The bottom of the screen features a taskbar with the word 'input' and a system tray with icons for battery, signal, volume, and date/time (19:22, 28-10-2020).

The screenshot shows a browser window titled "output_dequeue - GDB online" with the URL <https://www.onlinegdb.com/edit/B1jdqqlOP>. The page displays a C program named "main.c" containing code for a queue. The code includes functions for enqueueing, dequeuing, and inserting at the front. The browser interface includes standard navigation buttons, a search bar at the bottom, and a status bar at the bottom right indicating the date and time.

```
main.c
1  #include <stdio.h>
2
3  #define MAX 10
4
5  int q[MAX];
6  int f=-1, r=-1;
7
8  void enqueue(int item)
9  {
10     if(r==MAX-1)
11     {
12         printf("Queue is full\n");
13         return;
14     }
15     else
16     {
17         q[++r]=item;
18     }
19 }
20
21 void dequeue()
22 {
23     if(f>r)
24     {
25         printf("Queue is empty\n");
26         return;
27     }
28     else
29     {
30         printf("Item deleted is %d\n",q[(f)+]);
31         if(f>r)
32         {
33             f=0;
34             r=-1;
35         }
36     }
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         return;
51     }
52     else
53         printf("Insertion not possible\n");
54 }
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     }
68 }
```

⚡ output_dequeue - GDB online X +

https://www.onlinegdb.com/edit/B1jdqgLOP 70% ⋮ 🔍 ⚡

Run Debug Stop Save {Beautify} Language C ⚡

main.c

```
71     printf("item deleted is %d\n",q[f--]);
72     if(f>r)
73     {
74         f=0;
75         r=-1;
76     }
77 }/*
78
79
80 void display()
81 {
82     int i;
83     if(isempty())
84     {
85         printf("Queue empty\n");
86         return;
87     }
88     for(i=f;i<=r;i++)
89         printf("%d\n",q[i]);
90 }
91
92
93 void main()
94 {
95
96     for(;;)
97     {
98         printf("1.Insert_rear\n2.Insert_front\n3.Delete_front\n4.Display\n5.Exit\n");
99         printf("Enter choice\n");
100        scanf("%d",&ch);
101        switch(ch)
102        {
103            case 1:printf("Enter the item\n");
104                scanf("%d",&item);
105                insert_rear();
106                break;
107            case 2:printf("Enter the item\n");
108                scanf("%d",&item);
109                insert_front();
110                break;
111            /*case 3:delete_rear();
112            break;*/
113            case 3:delete_front();
114            break;
115            case 4:display();
116            break;
117            default:exit(0);
118        }
119    }
120 }
```

input

Command line arguments:

Type here to search

O ⏺ 📂 📎 🔍 ⚡

19:23 28-10-2020 ENG

⚡ output_dequeue - GDB online X +

← → ⌂ ⌂ https://www.onlinegdb.com/edit/B1jdqqlOP 80% ⌂ ⌂ ⌂

input

```
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
1
Enter the item
90
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
1
Enter the item
80
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
1
Enter the item
65
Insertion not possible
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
1
Enter the item
```

Type here to search 19:26 19 ENG 28-10-2020

⚡ output_dequeue - GDB online X +

← → ⌂ ⌄ ⌁ https://www.onlinegdb.com/edit/B1jdqgLOP 80% ⌋ ⌃ ⌅ ⌆

⚡ Enter the item
70
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
4
90
80
70
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
3
Item deleted is 90
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
3
Item deleted is 80
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
4
70

input

Type here to search

19:26 28-10-2020 ENG 19

The screenshot shows a browser window with the URL <https://www.onlinegdb.com/edit/B1jdqqlOP>. The page displays a C program for a queue. The program starts with a choice of 70, followed by a loop of choices 1 through 5. The user enters choice 3 (Delete_front) twice, resulting in the deletion of items 90 and 80. The user then enters choice 4 (Display) twice, showing the queue containing 70. Finally, the user exits the program with choice 5. The browser interface includes a back/forward button, search bar, and zoom controls.

```
70
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
3
Item deleted is 90
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
3
Item deleted is 80
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
4
70
1.Insert_rear
2.Insert_front
3.Delete_front
4.Display
5.Exit
Enter choice
5

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