



Run



Debug



Stop



main.c

```

8
9  #include<stdio.h>
10 #include<stdlib.h>
11 struct node
12 {
13     int data;
14     struct node *next;
15 };
16 struct node *front;
17 struct node *rear;
18 void insert();
19 void delete();
20 void display();
21 void main ()
22 {
23     int choice;
24     while(choice != 4)
25     {
26         printf("\nMain Menu
27         printf("\n=====
28         printf("\n1.insert
29         printf("\nEnter yo
30         scanf("%d",& choice

```



input

4 5

Main Menu

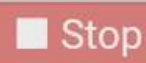


```

=====
=====

```





main.c

```
31         switch(choice)
32     {
33         case 1:
34             insert();
35             break;
36         case 2:
37             delete();
38             break;
39         case 3:
40             display();
41             break;
42         case 4:
43             exit(0);
44             break;
45         default:
46             printf("\nEnter
47     }
48 }
49 }
50 void insert()
51 {
52     struct node *ptr;
53     int item;
```



input

45

Main Menu

```
=====
=====
```



main.c

```

53     int item;
54
55     ptr = (struct node *)
56     if(ptr == NULL)
57     {
58         printf("\nOVERFLOW
59         return;
60     }
61     else
62     {
63         printf("\nEnter va
64         scanf("%d",&item);
65         ptr -> data = item
66         if(front == NULL)
67         {
68             front = ptr;
69             rear = ptr;
70             front -> next =
71             rear -> next =
72         }
73         else
74         {
75             rear -> next =

```



input

45

Main Menu

=====

=====





Run



Debug



Stop



main.c

```
76         rear = ptr;
77         rear->next =
78     }
79 }
80 }
81 void delete ()
82 {
83     struct node *ptr;
84     if(front == NULL)
85     {
86         printf("\nUNDERFLOW\n");
87         return;
88     }
89     else
90     {
91         ptr = front;
92         front = front -> next;
93         free(ptr);
94     }
95 }
96 void display()
97 {
98     struct node *ptr;
```



input

45

Main Menu

```
=====
=====
```





main.c

```

91         ptr = front;
92         front = front -> n
93         free(ptr);
94     }
95 }
96 void display()
97 {
98     struct node *ptr;
99     ptr = front;
100    if(front == NULL)
101    {
102        printf("\nEmpty q
103    }
104    else
105    {
106        printf("\ndisplay
107        while(ptr != NULL
108        {
109            printf("\n%d\n
110            ptr = ptr -> n
111        }
112    }
113

```



input

45

Main Menu

=====





```
91 ptr = front;  
92 front = front -> next;  
93 free(ptr);  
94 }
```



input

Main Menu

=====

- 1.insert an element
- 2.Delete an element
- 3.Display the queue
- 4.Exit

Enter your choice :1

Enter value:

45

Main Menu

=====

- 1.insert an element
- 2.Delete an element
- 3.Display the queue





```
91 ptr = front;  
92 front = front -> next;  
93 free(ptr);  
94 }
```



input

4.Exit

Enter your choice :2

Main Menu

```
=====
```

```
1.insert an element  
2.Delete an element  
3.Display the queue  
4.Exit
```

Enter your choice :3

Empty queue

Main Menu

```
=====
```

```
1.insert an element
```



Enter your choice :3

Empty queue

Main Menu

=====

- 1.insert an element
- 2.Delete an element
- 3.Display the queue
- 4.Exit

Enter your choice :4

...Program finished with exit code

0

Press ENTER to exit console.

Q. Program to implement queue using
Linked List

Program

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

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

```
struct node
```

```
{  
    int data;
```

```
    struct node *next;
```

```
};
```

```
struct node *front;
```

```
struct node *rear;
```

```
void insert();
```

```
void delete();
```

```
void display();
```

```
void main()
```

```
{
```

```
    int choice;  
    while (choice != 4)
```

```
    {  
        printf("\n Main Menu");
```

```
        printf("\n ===== \n");
```

```
        printf("\n 1. Insert an element in 2. Delete an  
        element in 3. Display the element queue in 4.  
        Exit \n");
```

```
        printf("\n Enter your choice: ");
```

```
        scanf("%d", &choice);
```

```
        switch (choice)
```

```
        {
```

```
            case 1:
```

```
                insert();
```

```
                break;
```

```
            case 2:
```

```
                delete();
```

```
                break;
```

```

Case 3:
display();
break;
Case 4:
eat(0);
break;
default:
printf("Enter valid choice: \n");
}
}

void insert()
{
    struct node * ptr;
    int item;
    ptr = (struct node *) malloc (sizeof (struct
    node));
    if (ptr == NULL)

```

```

{
    printf("OVERFLOW \n");
    return;
}
else
{
    printf("Enter value: \n");
    scanf ("%d", &item);
    ptr->data = item;
    if (front == NULL)
    {
        front = ptr;
        rear = ptr;
        front->next = NULL;
        rear->next = NULL;
    }
    else
    {
        rear->next = ptr;
    }
}

```

```

    Tnode = ptr;
    Tnode->next = NULL;
}
}

void delete()
{
    struct tnode * ptr;
    if (front == NULL)
    {
        printf("No element in queue\n");
        return;
    }
    else
    {
        ptr = front;
        front = front->next;
        free(ptr);
    }
}

```

```

}
}

void display()
{
    struct tnode * ptr;
    ptr = front;
    if (front == NULL)
    {
        printf("No element in queue\n");
    }
    else
    {
        printf("Displaying values...\n");
        while (ptr != NULL)
        {
            printf("%d ", ptr->data);
            ptr = ptr->next;
        }
    }
}

```

Output

Main menu

=====

1. Insert the element
2. Delete an element
3. Display the queue
4. Exit

Enter your choice : 1

Enter value:

45

Main menu

=====

1. Insert an element
2. Delete an element
3. Display the queue
4. Exit

Enter your choice : 1

Main Menu

=====

1. insert an element
2. delete an element
3. Display the queue
4. exit

Enter your choice: 3

Empty queue

Main Menu

=====

1. insert an element
2. delete an element
3. display the queue
4. Exit

Enter your choice: 4

... program finished with exit code 0