

Linear Queue: #include <stdio.h> #include <stdlib.h>

8/01/2024

#define size 5;

int front = -1, rear = -1;

char queue[size];

void enqueue(char a)

{

if (rear == size - 1)

printf("Overflow");

else

{

if (front == -1)

front = 0;

queue[rear] = a;

}

}

char dequeue()

{

if (front == -1 || front > rear)

printf("underflow");

else

{

char s = queue[front];

if (front == rear)

front = rear = -1;

else

{

~~front~~ front

}

return s;

}

void display()

{

if (front == -1 || front > rear)

printf("underflow");

else

{

for (i = rear; i >= 0; i--)

{

printf("%d", queue[i]);

}

}

output:

~~input~~

----- Menu -----

1. enqueue

2. dequeue

3. display

4. exit

enter your choice: 1

enter a value: 32

enter your choice: 1

enter a value: 45

enter your choice: 32
45

enter your choice: 2

enter your choice: 3

45

enter your choice: 4

exit the queue.

Circular queue

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

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

```
#define size 5
```

```
int items[size], rear = -1, front = -1;
```

```
int isfull()
```

```
{
```

```
    if ((front == rear - 1) || (front == 0 && rear == size - 1))  
        return 1;
```

```
    return 0;
```

```
}
```

```
int isempty()
```

```
{
```

```
    if (front == -1)
```

```
        return 1;
```

```
    return 0;
```

```
}
```

```
void enqueue (int element)
```

```
{
```

```
    if (isfull())
```

```
    {
```

```
        printf("queue is full");
```

```
    }
```

```
    else
```

```
    {
```

```
        if (front == -1)
```

```
            front = 0;
```

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

```
        items[rear] = element;
```

```
        printf("%d is inserted", element);
```

```
    }
```

```
}
```

```
int dequeue()
```

```
{
```

```
    int value;
```

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

```
    {  
        printf("queue is empty") return -1;  
    }
```

✓
Su
8/1/24

```

    if (front == -1)
        front = 0;

    rear =
    value = items[front];

    if (front == rear)
    {
        front = -1;
        rear = -1;
    }
    else
    {
        front = (front + 1) % size;
    }

    return value;
}

```

```

void display()
{
    int i;
    if (isEmpty())
        printf("queue is empty");

    else
    {
        printf("front position = %d\n", front);
        for (i = front; i != rear; i = (i + 1) % size)
        {
            printf("%d ", items[i]);
        }
        printf("%d", items[i]);
    }
}
}

```

output:

-- Menu --

1. enqueue
2. dequeue
3. display
4. exit

enter your choice: 1

enter your choice : 3

3/

enter your choice : 2

enter your choice : 4.



3rd_sem > C learning > C queue1.c > enqueue()

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  # define size 5
4  int front=-1,rear=-1,queue[size];
5
6  void enqueue()
7  {
8      int a;
9      if (rear==size-1)
10     {
11         printf("overflow\n");
12     }
13     else
14     {
15         if(front==-1)
16             front=0;
17         printf("enter a element:");
18         scanf("%d",&a);
19         queue[++rear]=a;
20     }
21 }
22
23 int dequeue()
24 {
25     if(front==-1||front>rear)
26     {
27         printf("underflow\n");
28     }
29     else
30     {
31         int s = queue[front];
32         ++front;
33         if (front==rear)
34         {
35             front=-1;
36             rear=-1;
37         }
38     }
```


37 } enqueue() {

38 return s;

39 }

40 }

41

42 void display()

43 {

44 if(front==-1)

45 {

46 printf("overflow\n");

47 }

48 else

49 {

50 for (int i=front;i<=rear;i++)

51 {

52 printf("%d\n",queue[i]);

53 }

54 }

55 }

56

57 int main()

58 {

59 int choice;

60

61 while(1)

62 {

63 printf("-----MENU-----\n");

64 printf("1.enqueue\n 2.dequeue\n 3.display\n 4.exit\n");

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

66

67 switch(choice)

68 {

69 case 1:enqueue();

70 break;

71 case 2:dequeue();

72 break;

73 case 3:display();

74 break;

```
57 int main()
58 {
59     int choice;
60
61     while(1)
62     {
63         printf("-----MENU-----\n");
64         printf("1.enqueue\n 2.dequeue\n 3.display\n 4.exit\n");
65         scanf("%d",&choice);
66
67         switch(choice)
68         {
69             case 1:enqueue();
70                 break;
71             case 2:dequeue();
72                 break;
73             case 3:display();
74                 break;
75             case 4: exit(0);
76                 break;
77             default:printf("wrong input");
78         }
79     }
80 }
81
```


-----MENU-----

- 1.enqueue
- 2.dequeue
- 3.display
- 4.exit

1
enter a element:65

-----MENU-----

- 1.enqueue
- 2.dequeue
- 3.display
- 4.exit

1
enter a element:98

-----MENU-----

- 1.enqueue
- 2.dequeue
- 3.display
- 4.exit

1
enter a element:68

-----MENU-----

- 1.enqueue
- 2.dequeue
- 3.display
- 4.exit

3
65
98
68

-----MENU-----

- 1.enqueue
- 2.dequeue
- 3.display
- 4.exit

2
-----MENU-----

- 1.enqueue
- 2.dequeue
- 3.display
- 4.exit

3
98
68

-----MENU-----

- 1.enqueue
- 2.dequeue
- 3.display
- 4.exit

d sem > C learning > C cqueue.c > enqueue()

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  # define size 5
4  int front=-1,rear=-1,queue[size];
5
6  void enqueue()
7  {
8      int a;
9      if (front==rear+1||front==0 && rear==size-1)
10     {
11         printf("overflow\n");
12     }
13     else
14     {
15         if(front==-1)
16             front=0;
17         printf("enter a element:");
18         scanf("%d",&a);
19         rear=(rear+1)%size;
20         queue[rear]=a;
21     }
22 }
23
24 int dequeue()
25 {
26     if(front==-1)
27     {
28         printf("underflow\n");
29     }
30     else
31     {
32
33         int s = queue[front];
34         if (front==rear)
35         {
36             front=-1;
37             rear=-1;
```



```

24 int dequeue()
25 {
26     if(front==-1)
27     {
28         printf("underflow\n");
29     }
30     else
31     {
32
33         int s = queue[front];
34         if (front==rear)
35         {
36             front=-1;
37             rear=-1;
38         }
39         else{
40             front=(front+1)%size;
41         }
42         printf("deleted element:%d",s);
43         return s;
44     }
45 }
46 void display()
47 {
48     int i;
49     if(front==-1)
50     {
51         printf("overflow\n");
52     }
53     else
54     {
55         for (i=front;i!=rear;i=(i+1)%size)
56         {
57             printf("%d\n",queue[i]);
58         }
59         printf("%d\n",queue[i]);
60     }

```

```
53 int main()
54 {
55     int choice;
56
57     while(1)
58     {
59         printf("-----MENU-----\n");
60         printf("1.enqueue\n 2.dequeue\n 3.display\n 4.exit\n");
61         scanf("%d",&choice);
62
63         switch(choice)
64         {
65             case 1:enqueue();
66                 break;
67             case 2:dequeue();
68                 break;
69             case 3:display();
70                 break;
71             case 4: exit(0);
72                 break;
73             default:printf("wrong input");
74         }
75     }
76 }
77
78
79
80
81
82
83
84
85
86
87
```


-----MENU-----

- 1.enqueue
- 2.dequeue
- 3.display
- 4.exit

1

enter a element:32

-----MENU-----

- 1.enqueue
- 2.dequeue
- 3.display
- 4.exit

1

enter a element:65

-----MENU-----

- 1.enqueue
- 2.dequeue
- 3.display
- 4.exit

3

32

65

-----MENU-----

- 1.enqueue
- 2.dequeue
- 3.display
- 4.exit

1

enter a element:98

-----MENU-----

- 1.enqueue
- 2.dequeue
- 3.display
- 4.exit

1

enter a element:74

-----MENU-----

- 1.enqueue
- 2.dequeue
- 3.display
- 4.exit

3.display

4.exit

1

enter a element:54

-----MENU-----

1.enqueue

2.dequeue

3.display

4.exit

1

overflow

-----MENU-----

1.enqueue

2.dequeue

3.display

4.exit

88

wrong input-----MENU-----

1.enqueue

2.dequeue

3.display

4.exit

1

overflow

-----MENU-----

1.enqueue

2.dequeue

3.display

4.exit

2

deleted element:32-----MENU-----

1.enqueue

2.dequeue

3.display

4.exit

1

enter a element:88

-----MENU-----

1.enqueue

2.dequeue

3.display

enter a element:88

-----MENU-----

- 1.enqueue
- 2.dequeue
- 3.display
- 4.exit

3

65

98

74

54

88

-----MENU-----

- 1.enqueue
- 2.dequeue
- 3.display
- 4.exit