

main.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define MAX 3
5
6 void insert();
7 void delete();
8 void display();
9 int queue_array[MAX];11
10 int rear = - 1;
11 int front = - 1;
12 int main()
13 {
14     int choice;
15     while (1)
16     {
17         printf("1.Insert element to queue \n");
18         printf("2.Delete element from queue \n");
19         printf("3.Display all elements of queue \n");
20         printf("4.Quit \n");
21         printf("Enter your choice : ");
22         scanf("%d", &choice);
23         switch (choice)
24         {
25             case 1:
26                 insert();
```

main.c

```
25     case 1:  
26         insert();  
27         break;  
28     case 2:  
29         delete();  
30         break;  
31     case 3:  
32         display();  
33         break;  
34     case 4:  
35         exit(1);  
36     default:  
37         printf("Wrong choice \n");  
38     }  
39 }  
40 }  
41  
42 void insert()  
43 {  
44     int add_item;  
45     if (rear == MAX - 1)  
46         printf("Queue Overflow \n");  
47     else  
48     {  
49         if (front == - 1)
```

main.c

```
45     if (rear == MAX - 1)
46         printf("Queue Overflow \n");
47     else
48     {
49         if (front == - 1)      [
50
51             front = 0;
52             printf("Inset the element in queue : ");
53             scanf("%d", &add_item);
54             rear = rear + 1;
55             queue_array[rear] = add_item;
56     }
57 }
58
59 void delete()
60 {
61     if (front == - 1 || front > rear)
62     {
63         printf("Queue Underflow \n");
64         return ;
65     }
66     else
67     {
68         printf("Element deleted from queue is : %d\n",
69         queue_array[front]);
70         front = front + 1;
71 }
```

```
66     else
67     {
68         printf("Element deleted from queue is : %d\n",
69             queue_array[front]);
70         front = front + 1;    [ ]
71     }
72
73 void display()
74 {
75     int i;
76     if (front == - 1)
77         printf("Queue is empty \n");
78     else
79     {
80         printf("Queue is : \n");
81         for (i = front; i <= rear; i++)
82             printf("%d ", queue_array[i]);
83         printf("\n");
84     }
85 }
86
87
```

```
► clang-7 -pthread -lm -o main main.c
► ./main
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 1
Inset the element in queue : 23
1.Insert element to queue
2.Delete element from queue I
3.Display all elements of queue
4.Quit
Enter your choice : 1
Inset the element in queue : 22
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 1
Inset the element in queue : 20
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 1
Inset the element in queue : 45
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 3
Queue is :
```

4.Quit

Enter your choice : 1

Inset the element in queue : 45

1.Insert element to queue

2.Delete element from queue

3.Display all elements of queue

4.Quit

Enter your choice : 3

Queue is :

23 22 20 45

I

1.Insert element to queue

2.Delete element from queue

3.Display all elements of queue

4.Quit

Enter your choice : 2

Element deleted from queue is : 23

1.Insert element to queue

2.Delete element from queue

3.Display all elements of queue

4.Quit

Enter your choice : 2

Element deleted from queue is : 22

1.Insert element to queue

2.Delete element from queue

3.Display all elements of queue

4.Quit

Enter your choice : 2

Element deleted from queue is : 20

1.Insert element to queue

2.Delete element from queue

3.Display all elements of queue

4.Quit

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #define SIZE 3
4 int a[SIZE],t;
5 int front=-1;
6 int rear=-1;
7
8 int IsEmpty()
9 {
10     if(rear== -1 && front == -1)
11         return 1;
12     else
13         return 0;
14 }
15
16
17 int IsFull()
18 {
19     if(front == (rear+1)%SIZE)
20         return 1;
21     else
22         return 0;
23 }
24
25
26 void Enqueue(int x)
```

main.c

```
27  {
28  | if(IsFull())
29  | | printf("The queue is full\n");
30  | | else if(IsEmpty())
31  | | {
32  | | | front=0;
33  | | | rear=0;
34  | | | a[rear]=x;
35  | | }
36  | | else
37  | | {
38  | | | rear=(rear+1)%SIZE;
39  | | | a[rear]=x;
40  | | }
41  | }
42
43  int Dequeue()
44  {   int x;
45  | if(IsEmpty())
46  | | printf("The queue is empty.\n");
47  | | else if(front==rear)
48  | | {
49  | | | x=a[front];
50  | | | front=-1;
51  | | | rear=-1;
52  | | | printf("The element was removed.\n");
```

main.c

```
53     }
54     else
55     {
56         x=a[front];
57         front=(front+1)%SIZE;
58         printf("The element was removed\n");
59     }
60     return x;
61 }
62
63
64 void display()
65
66 {
67     if (front == -1)
68     {
69         printf("\nQueue is Empty");
70         return;
71     }
72     printf("\nElements in Circular Queue are:\n");
73     if (rear >= front)
74     {
75         for (int i = front; i <= rear; i++)
76             printf("%d ", a[i]);
77     }
78     else
79     {
```

main.c

main.c

```
93     while(1)
94     {
95         printf("Enter the
96         operation.\n1-Insert\n2-Delete\n3-Display\n4-Exit\n");
97         scanf("%d",&n);
98         switch(n)
99         {
100             case 1: printf("Enter the element\n");
101                 scanf("%d",&a);
102                 Enqueue(a);
103                 break;
104             case 2 : Dequeue();
105                 break;
106             case 3: display();
107                 break;
108             case 4: exit(0);
109             default : printf("There is no such operation\n");
110         }
111     }
112 }
113 return 0;
114 }
```

```
* clang-7 -pthread -lm -o main main.c
* ./main
Enter the operation.
1-Insert
2>Delete
3-Display
4-Exit
1
Enter the element
15
Enter the operation.
1-Insert
2>Delete
3-Display
4-Exit
2
The element was removed
Enter the operation.
1-Insert
2>Delete
3-Display
4-Exit
1
Enter the element
23
Enter the operation.
1-Insert
2>Delete
3-Display
4-Exit
3
```

Enter the element

15

Enter the operation.

1-Insert

2-Delete

3-Display

4-Exit

2

The element was removed

Enter the operation.

1-Insert

2-Delete

3-Display

4-Exit

1

Enter the element

23

Enter the operation.

1-Insert

2-Delete

3-Display

4-Exit

3

Elements in Circular Queue are:

23

Enter the operation.

1-Insert

2-Delete

3-Display

4-Exit

□

LINER QUEUE

```
#include < stdio.h>
#include < stdlib.h>
#define MAX 3
```

```
void insert();
void delete();
void display();
int queue[ array MAX];
int rear = -1;
int front = -1;
int count;
{ int choice;
while(1)
```

```
{ printf("1. Insert element to queue\n");
printf("2. Delete element from queue\n");
printf("3. Display all elements of queue\n");
printf("4. Exit\n");
printf("Enter your choice : ");
scanf("%d", &choice);
switch(choice)
{
```

case 1:

insert();

break;

case 2:

delete();

break;

case 3:

display();

break;

case 4:

```
exit (1);
```

```
default:
```

```
    print ("Wrong choice \n");
```

```
}
```

```
else
```

```
{
```

```
    word_insert();
```

```
}
```

```
int add_item;
```

```
if (rear == MAX - 1),
```

```
    print ("Queue Overflow \n");
```

```
else
```

```
{
```

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

```
    front = 0;
```

```
    print ("Insert the element  
in queue : ");
```

```
Scony ("cd", &add_item);
```

```
rear = rear + 1;
```

```
queue_array [rear] = add_item;
```

```
}
```

```
{
```

```
Initial deletion
```

```
{
```

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

```
{
```

```
    print ("Underflow \n");
```

```
}
```

```
else
```

```
{
```

```
    print ("Element deleted from queue is : cd \n");
```

```
queue - array(front));  
front = front + 1;  
}  
}
```

```
void display()  
{
```

```
int i
```

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

```
printf("Queue is Empty \n");
```

```
else
```

```
{
```

```
printf("Queue is: \n");
```

```
for (i = front; i <= rear; i++)
```

```
{ printf("%d ", queue[i]); }  
printf("\n");
```

```
front = front + 1;
```

```
}
```

```
3
```

Circular queue

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 3
int arr[SIZE], t;
int front = -1;
int rear = -1;
```

```
int IsEmpty()
{
    if (rear == -1 && front == -1)
        return 1;
    else
        return 0;
}
```

```
int IsFull()
```

```
{
    if ((front + 1) % SIZE == rear)
        return 1;
    else
        return 0;
}
```

```
void Enqueue (int x)
```

```
{
```

```
if (IsFull())
    printf ("The queue is full\n");
else if (IsEmpty())
```

```
{
```

```
front = 0;
```

```
rear = 0;
```

```
arr[rear] = x;
```

```
}
```

else

}

$$rear = (rear + 1) \% size;$$

a[rear] = x;

}

}

int dequeue()

{ int x;

if (IsEmpty())

printf("The queue is empty.");

else if (front == rear)

{

x = a[front];

front = -1;

rear = -1;

printf("The element was removed.");

}

else

{

x = a[front];

front = (front + 1) \% size;

printf("The element was removed.");

}

return x;

}

void display()

{

if (front == -1)

{

printf("The queue is empty.");

return;

}

3
point of ("") in Elimination Circular Queue are ("n");
if (rear >= front)
{

for (int i = front; i < rear; i++)
 Priority ("./d \n", a[i]);

3

else

{

for (int i = front; i < size; i++)

 Priority ("./d \n", a[i]);

for (int i = 0; i < size; i++)

 Priority ("./d " "n", a[i]);

3

3

int main ()

{

int n, a;

while (1)

{

 Priority ("Enter the operation. \n");

1-Insert In 2-Delete In 3-Display In 4-Exit \n");

scanf ("./d ", &n);

switch (n)

{

 Case 1: Priority ("Enter the element \n");

 scanf ("./d ", &a);

 Enqueue (a);

 break;

 Case 2: Dequeue ();

 break;

Case 3: display();
Break;

Case 4: exit(0);

default: printf("There is no such operation in ");

3.

3.

return 0;

3.