

## LAB PROGRA-3

```
#include <stdio.h>
#define MAX 50
void insert();
void delete();
void display();
int queue_array[MAX];
int rear = -1;
int front = -1;
main()
{
    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. Quit\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;
```

```
default: printf("Wrong choice\n");
```

```
{
```

```
}
```

```
}
```

```
void insert()
```

```
{
```

```
    int add-item;
```

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

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

```
    else
```

```
    {
```

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

```
            front = 0;
```

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

```
            scanf("%d", &add-item);
```

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

```
            queue[rear] = add-item;
```

```
    }
```

```
}
```



```

void delete()
{
    if (front == -1 || front > rear)
    {
        printf("Queue Underflow \n");
        return;
    }
    else
    {
        printf("Element deleted from queue is: %d\n",
            queue-array[front]);
        front = front + 1;
    }
}

```

```

void display()
{
    int i;
    if (front == -1)
        printf("Queue is empty \n");
    else
    {
        printf("Queue is: \n");
    }
}

```

Date \_\_\_\_\_  
Page \_\_\_\_\_

```
for (i = front ; i <= rear ; i++)  
    printf("%d", queue-array[i]);  
    printf("\n");  
}  
}
```

## LAB PROGRAM-4

```
# include <stdio.h>
# include <stdlib.h>
# define SIZE 6
int a[SIZE], t;
int front = -1;
int rear = -1;
```

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

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



```
void Enqueue ( int x )
{
    if ( isFull ( ) )
        printf ( "The queue is full \n" );
    else if ( isEmpty ( ) )
    {
        front = 0;
        rear = 0;
        a[rear] = x;
    }
    else
    {
        rear = ( rear + 1 ) % SIZE;
        a[rear] = x;
    }
}
```

```
int Dequeue ( )
{
    int x;
    if ( isEmpty ( ) )
        printf ( "The queue is empty. \n" );
    else if ( front == rear )
    {
        x = a[front];
    }
}
```

```

        front = -1;
        rear = -1;
        printf("The element was removed\n");
    }
    else
    {
        x = a[front];
        front = (front + 1) % SIZE;
        printf("The element was removed\n");
    }
    return x;
}

```

```

void display()
{

```

```

    if (front == -1)
    {

```

```

        printf("\n Queue is Empty");
        return;
    }

```

```

    printf("\n Element in Circular Queue are: \n");

```

```

    if (rear >= front)
    {

```

```

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

```

```

            printf("%d\n", a[i]);
        }
    }

```



```
else
```

```
{
```

```
for (int i = front; i < SIZE; i++)  
    printf("%d\n", a[i]);
```

```
for (int i = 0; i <= rear; i++)  
    printf("%d\n", a[i]);
```

```
}
```

```
}
```

```
int main()
```

```
{
```

```
int n, a;
```

```
while (1)
```

```
{
```

```
printf("Enter the operation. \n 1-Insert \n  
2-delete \n 3-Display \n 4-Exit \n");
```

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

```
switch(n)
```

```
{
```

```
case 1: printf("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\n");  
}  
}  
return 0;

Replit - C Online Compiler and IDE, Editor, Interpreter and REPL

Code, collaborate, compile, run, share, and deploy C online from your browser

Save Stop

Sign up

Files

main.c

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

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 : 2

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 : 2

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 : 2

1.Insert element to queue

2.Delete element from queue

3.Display all elements of queue

4.Quit

Enter your choice : 1

Queue Overflow

1.Insert element to queue

2.Delete element from queue

3.Display all elements of queue

clang-7 -pthread -lm -o main main.c

SVA\_DQA (1).txt

SVA\_DQA.txt

Show all

02:49 PM 19-10-2020



Replit - C Online Compiler and REPL

replit.it/languages/c

Apps Gmail YouTube Maps

C Online Compiler, IDE, Editor, Interpreter and REPL  
Code, collaborate, compile, run, share, and deploy C online from your browser

Save Stop

Talk Sign up

Files

main.c

```
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
```

```
switch (choice)
{
    case 1:
        insert();
        break;
    case 2:
        delete();
        break;
    case 3:
        display();
        break;
    case 4:
        exit(1);
    default:
        printf("Wrong choice \n");
}

void insert()
{
    int add_item;
    if (rear == MAX - 1)
        printf("Queue Overflow \n");
    else
```

Queue Overflow

```
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
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 : 1
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 : 2
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 : 3
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 1
Queue Overflow
```

SVA\_DQA (1).txt SVA\_DQA.txt

Show all

02:50 PM 19-10-2020

Replit - C Online Compiler and IDE, Editor, Interpreter and REPL

Code, collaborate, compile, run, share, and deploy C online from your browser

Save Stop

Talk Sign up

Files

main.c

```
40
41 void insert()
42 {
43     int add_item;
44     if (rear == MAX - 1)
45         printf("Queue Overflow \n");
46     else
47     {
48         if (front == - 1)
49             front = 0;
50         printf("Inset the element in queue : ");
51         scanf("%d", &add_item);
52         rear = rear + 1;
53         queue_array[rear] = add_item;
54     }
55 }
56
57 void delete()
58 {
59     if (front == - 1 || front > rear)
60     {
61         printf("Queue Underflow \n");
62         return ;
63     }
64 }
```

1.Insert element to queue  
2.Delete element from queue  
3.Display all elements of queue  
4.Quit  
Enter your choice : 1  
Queue Overflow  
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 : 2  
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 : 1  
1.Insert element to queue  
2.Delete element from queue  
3.Display all elements of queue  
4.Quit

SVA\_DQA (1).txt SVA\_DQA.txt

Show all

02:50 PM 19-10-2020



Replit - C Online Compiler and IDE, Editor, Interpreter and REPL

Code, collaborate, compile, run, share, and deploy C online from your browser

Save Stop

Talk Sign up

Files

main.c

```
main.c
56 }
57
58 void delete()
59 {
60     if (front == - 1 || front > rear)
61     {
62         printf("Queue Underflow \n");
63         return ;
64     }
65     else
66     {
67         printf("Element deleted from queue is : %d\n",
68             queue_array[front]);
69         front = front + 1;
70     }
71 }
72 void display()
73 {
74     int i;
75     if (front == - 1)
76         printf("Queue is empty \n");
77     else
78     {
79         printf("Queue is : \n");
```

```
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 1
Queue Overflow
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 : 2
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 : 1
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 : 2
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
```

SVA\_DQA (1).txt SVA\_DQA.txt

02:50 PM 19-10-2020

Replit - C Online Compiler and IDE, Editor, Interpreter and REPL

Code, collaborate, compile, run, share, and deploy C online from your browser

Save Stop

Talk Sign up

Files

main.c

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

1 2 3  
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 : 1  
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 : 2  
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 : 3  
1.Insert element to queue  
2.Delete element from queue  
3.Display all elements of queue  
4.Quit  
Enter your choice : 2  
Queue Underflow  
1.Insert element to queue  
2.Delete element from queue  
3.Display all elements of queue  
4.Quit  
Enter your choice : 0

SVA\_DQA (1).txt SVA\_DQA.txt

02:51 PM 19-10-2020



Replit - C Online Compiler and IDE, Editor, Interpreter and REPL

Code, collaborate, compile, run, share, and deploy C online from your browser

Save Stop

Files

main.c

```
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 int IsFull()
17 {
18     if(front==(rear+1)%SIZE)
19         return 1;
20     else
21         return 0;
22 }
23
24
25
```

```
clang-7 -pthread -lm -o main main.c
./main
Enter the operation.
1-Insert
2-Delete
3-Display
4-Exit
1
Enter the element
1
Enter the operation.
clang-7 -pthread -lm -o main main.c
./main
Enter the operation.
1-Insert
2-Delete
3-Display
4-Exit
1
Enter the element
1
Enter the operation.
1-Insert
2-Delete
3-Display
4-Exit
1
Enter the element
2
Enter the operation.
```

SVA\_DQA (1).txt SVA\_DQA.txt

02:51 PM 19-10-2020

Replit - C Online Compiler and IDE, Editor, Interpreter and REPL

Code, collaborate, compile, run, share, and deploy C online from your browser

Save Stop

Talk Sign up

Files

main.c

```
main.c
25
26 void Enqueue(int x)
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 {
45     int x;
46     if(IsEmpty())
47         printf("The queue is empty.\n");
48     else if(front==rear)
49     {
50         front=rear;
51     }
52     else
53     {
54         front=(front+1)%SIZE;
55     }
56     return a[front];
57 }
58
59 int main()
60 {
61     int n;
62     printf("Enter the number of elements: ");
63     scanf("%d", &n);
64     for(int i=0; i<n; i++)
65     {
66         int x;
67         printf("Enter element %d: ", i+1);
68         scanf("%d", &x);
69         Enqueue(x);
70     }
71     printf("Elements in Circular Queue are: ");
72     for(int i=0; i<n; i++)
73     {
74         printf("%d ", Dequeue());
75     }
76     printf("\n");
77     return 0;
78 }
```

4-Exit  
3

Elements in Circular Queue are:  
1  
2  
3

Enter the operation.  
1-Insert  
2-Delete  
3-Display  
4-Exit  
3  
clang-7 -pthread -lm -o main main.c  
./main

Enter the operation.  
1-Insert  
2-Delete  
3-Display  
4-Exit  
1  
Enter the element  
1

Enter the operation.  
1-Insert  
2-Delete  
3-Display  
4-Exit  
1  
clang-7 -pthread -lm -o main main.c  
./main

Enter the operation.  
1-Insert  
2-Delete  
3-Display  
4-Exit  
1

SVA\_DQA (1).txt SVA\_DQA.txt

02:51 PM 19-10-2020

Replit - C Online Compiler and IDE, Editor, Interpreter and REPL

Code, collaborate, compile, run, share, and deploy C online from your browser

Save Stop

Talk Sign up

Files

main.c

```
main.c
09 printf("Queue 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\n", a[i]);
77 }
78 else
79 {
80     for (int i = front; i < SIZE; i++)
81         printf("%d\n", a[i]);
82     for (int i = 0; i <= rear; i++)
83         printf("%d\n", a[i]);
84 }
85 }
86 }
87
88
89
90 int main()
91 {
92     int n,a;
93     while(1)
```

```
2-Delete
3-Display
4-Exit
1
Enter the element
4
The queue is full
Enter the operation.
1-Insert
2-Delete
3-Display
4-Exit
3
Elements in Circular Queue are:
1
2
3
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
```

SVA\_DQA (1).txt SVA\_DQA.txt

02:51 PM 19-10-2020



Replit - C Online Compiler and IDE, Editor, Interpreter and REPL

Code, collaborate, compile, run, share, and deploy C online from your browser

Save Stop

Talk Sign up

Files

main.c

```
51     rear=-1;
52     printf("The element was removed\n");
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 void display()
64 {
65     if (front == -1)
66     {
67         printf("\nQueue is Empty");
68         return;
69     }
70     printf("\nElements in Circular Queue are:\n");
71     if (rear >= front)
72     {
73         for (int i = front; i <= rear; i++)
```

4-Exit  
1  
Enter the element  
3  
Enter the operation.  
1-Insert  
2-Delete  
3-Display  
4-Exit  
1  
Enter the element  
4  
The queue is full  
Enter the operation.  
1-Insert  
2-Delete  
3-Display  
4-Exit  
3  
Elements in Circular Queue are:  
1  
2  
3  
Enter the operation.  
1-Insert  
2-Delete  
3-Display  
4-Exit  
2

SVA\_DQA (1).txt SVA\_DQA.txt

Show all

02:51 PM  
19-10-2020

Replit - C Online Compiler and IDE, Editor, Interpreter and REPL

Code, collaborate, compile, run, share, and deploy C online from your browser

Save Stop

Talk Sign up

Files

main.c

```
88
89
90 int main()
91 {
92     int n,a;
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     }
```

4-Exit  
2  
The element was removed  
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  
2  
The queue is empty.  
Enter the operation.  
1-Insert  
2-Delete  
3-Display  
4-Exit  
[]

SVA\_DQA (1).txt SVA\_DQA.txt

02:51 PM 19-10-2020

Replit - C Online Compiler and REPL

replit.it/languages/c

Apps Gmail YouTube Maps

C Online Compiler, IDE, Editor, Interpreter and REPL  
Code, collaborate, compile, run, share, and deploy C online from your browser

Save Stop

Talk Sign up

Files

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 return 0;
113 }
114 }
```

4-Exit  
2  
The element was removed  
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  
2  
The element was removed  
Enter the operation.  
1-Insert  
2-Delete  
3-Display  
4-Exit  
2  
The queue is empty.  
Enter the operation.  
1-Insert  
2-Delete  
3-Display  
4-Exit  
[]

SVA\_DQA (1).txt SVA\_DQA.txt

Show all

02:51 PM  
19-10-2020