

LAB PROGRAM-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 - array [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");
    }
}
```

```
for (p = front ; p <= 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) % SITE;
    printf ("The element was removed \n");
}
return x;
}

void display()
{
if (front == -1)
    printf ("Queue is Empty");
    return;
}
printf ("Element in Circular Queue : ");
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. 1-Insert\n2-Delete\n3-Display\n4-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: Dispaly();
break;

case 4: exit(0);

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

The screenshot shows a web-based C compiler interface on Repl.it. On the left, the code for `main.c` is displayed:

```
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();
```

On the right, the terminal window shows the execution of the program and its output:

```
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
```

The taskbar at the bottom shows other open files: `SVA_DQA (1).txt` and `SVA_DQA.txt`. The system tray indicates the date as 19-10-2020 and the time as 02:49 PM.

Repl.it - C Online Compiler and | x Repl.it - C Online Compiler and | x Repl.it - C Online Compiler and | x +

repl.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

Files main.c

```
main.c
22     switch (choice)
23     {
24         case 1:
25             insert();
26             break;
27         case 2:
28             delete();
29             break;
30         case 3:
31             display();
32             break;
33         case 4:
34             exit(1);
35         default:
36             printf("Wrong choice \n");
37     }
38 }
39 }
40 void insert()
41 {
42     int add_item;
43     if (rear == MAX - 1)
44         printf("Queue Overflow \n");
45     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

Repl.it - C Online Compiler and | x Repl.it - C Online Compiler and | x Repl.it - C Online Compiler and | x +

← → ⌂ repl.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
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
50             front = 0;
51         printf("Inset the element in queue : ");
52         scanf("%d", &add_item);
53         rear = rear + 1;
54         queue_array[rear] = add_item;
55     }
56 }
57
58 void delete()
59 {
60     if (front == - 1 || front > rear)
61     {
62         printf("Queue Underflow \n");
63         return ;
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
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 Show all

Repl.it - C Online Compiler and | x Repl.it - C Online Compiler and | x Repl.it - C Online Compiler and | x +

← → ⌂ repl.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
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",
queue_array[front]);
68         front = front + 1;
69     }
70 }
71 void display()
72 {
73     int i;
74     if (front == - 1)
75         printf("Queue is empty \n");
76     else
77     {
78         printf("Queue is : \n");
79     }
```

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 -lpthread -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 Show all

Repl.it - C Online Compiler and | x Repl.it - C Online Compiler and | x Repl.it - C Online Compiler and | x +

repl.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

Files main.c

```
main.c
67     printf("Element deleted from queue is : %d\n",
68     queue_array[front]);
69 }
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 Show all

The screenshot shows a Repl.it interface for C programming. On the left, the code editor displays a file named 'main.c' containing the following C code:

```
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
```

On the right, the terminal window shows the output of the program. It starts with compilation commands:

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

Then it prompts for an operation:

```
Enter the operation.
```

The user enters '1' for Insert:

```
1-Insert
```

It then asks for an element to insert:

```
Enter the element
```

The user enters '1' again:

```
1
```

It then prompts for another operation:

```
Enter the operation.
```

The user enters '1' again:

```
1-Insert
```

It then asks for an element to insert:

```
Enter the element
```

The user enters '2' again:

```
2
```

It then prompts for another operation:

```
Enter the operation.
```

Repl.it - C Online Compiler and IDE | Repl.it - C Online Compiler and IDE | Repl.it - C Online Compiler and IDE | +

repl.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

Files main.c

```
main.c
1
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     {
```

4-Exit
3
Elements in Circular Queue are:
1
2
3
Enter the operation.
1-Insert
2-Delete
3-Display
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

SVA_DQA (1).txt SVA_DQA.txt Show all

Repl.it - C Online Compiler and IDE | Repl.it - C Online Compiler and IDE | Repl.it - C Online Compiler and IDE | +

repl.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

Files main.c

```
main.c
1   // Code in main.c
2
3   #include <stdio.h>
4   #include <stdlib.h>
5
6   #define SIZE 10
7
8   struct Queue
9   {
10    int front;
11    int rear;
12    int queue[SIZE];
13 };
14
15 void enqueue(struct Queue *q, int value)
16 {
17    if (q->rear == SIZE - 1)
18    {
19      printf("Queue is full\n");
20      return;
21    }
22    q->queue[q->rear] = value;
23    q->rear++;
24 }
25
26 int dequeue(struct Queue *q)
27 {
28    if (q->front == q->rear)
29    {
30      printf("Queue is empty\n");
31      return -1;
32    }
33    int value = q->queue[q->front];
34    q->front++;
35    return value;
36 }
37
38 void display(struct Queue *q)
39 {
40    if (q->front == q->rear)
41    {
42      printf("\nElements in Circular Queue are:\n");
43      return;
44    }
45    if (q->front >= q->rear)
46    {
47      for (int i = q->front; i <= q->rear; i++)
48        printf("%d\n", q->queue[i]);
49    }
50    else
51    {
52      for (int i = q->front; i < SIZE; i++)
53        printf("%d\n", q->queue[i]);
54
55      for (int i = 0; i <= q->rear; i++)
56        printf("%d\n", q->queue[i]);
57    }
58 }
59
60 int main()
61 {
62    int n,a;
63    while(1)
64    {
65      printf("2-Delete\n");
66      printf("3-Display\n");
67      printf("4-Exit\n");
68      printf("1\n");
69      printf("Enter the element\n");
70      printf("4\n");
71      printf("The queue is full\n");
72      printf("Enter the operation.\n");
73      printf("1-Insert\n");
74      printf("2-Delete\n");
75      printf("3-Display\n");
76      printf("4-Exit\n");
77      printf("3\n");
78
79      Elements in Circular Queue are:
80      1
81      2
82      3
83      Enter the operation.
84      1-Insert
85      2>Delete
86      3-Display
87      4-Exit
88
89      2
90      The element was removed
91      Enter the operation.
92      1-Insert
93      2>Delete
94      3-Display
95      4-Exit
96    }
97 }
```

SVA_DQA (1).txt SVA_DQA.txt Show all

Repl.it - C Online Compiler and IDE | Repl.it - C Online Compiler and IDE | Repl.it - C Online Compiler and IDE | +

repl.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

Files main.c

```
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
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++)
```

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

The screenshot shows a browser window with three tabs, all titled "Repl.it - C Online Compiler and...". The active tab displays a C program named "main.c" and its execution output.

Code (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             case 3: display();
106                 break;
107             case 4: exit(0);
108             default : printf("There is no such operation\n");
110         }
111     }
112 }
```

Output:

```
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
[]
```

The code implements a simple queue using enqueue and dequeue operations. The user can choose to insert an element, delete an element, display the queue, or exit. The output shows the execution of these operations, including cases where the queue is empty or an invalid operation is chosen.

Repl.it - C Online Compiler and IDE | Repl.it - C Online Compiler and IDE | Repl.it - C Online Compiler and IDE | +

repl.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

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 | }
```

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