

```
Enter 1:insert 2:delete 3:display 4:exit      1
Enter value3
3 inserted

Enter 1:insert 2:delete 3:display 4:exit      1
Enter value7
7 inserted

Enter 1:insert 2:delete 3:display 4:exit      3
3
7

Enter 1:insert 2:delete 3:display 4:exit      2
the value 3 is deleted

Enter 1:insert 2:delete 3:display 4:exit      3
7

Enter 1:insert 2:delete 3:display 4:exit      4
Process returned 0 (0x0)  execution time : 20.046 s
Press any key to continue.
```

```
Enter 1:insert 2:delete 3:display 4:exit      1
```

```
Enter value3
```

```
3 inserted
```

```
Enter 1:insert 2:delete 3:display 4:exit      1
```

```
Enter value7
```

```
7 inserted
```

```
Enter 1:insert 2:delete 3:display 4:exit      3
```

```
3
```

```
7
```

```
Enter 1:insert 2:delete 3:display 4:exit      2
```

```
the value 3 is deleted
```

```
Enter 1:insert 2:delete 3:display 4:exit      3
```

```
7
```

```
Enter 1:insert 2:delete 3:display 4:exit      4
```

```
Process returned 0 (0x0)  execution time : 20.046 s
```

```
Press any key to continue.
```

## DS LAB -3

### Queue

```
# include <Studio.h>
# define MAX 5
int front = -1; rear = -1, q[MAX];
Void enqueue (int value) {
    If (front == -1 && rear == -1) {
        front = rear = 0;
        q[rear] = value;
    }
    else if (rear == MAX - 1)
        Printf ("Overflow");
    else
    {
        q[++rear] = value;
    }
}
void dequeue () {
    If (front == -1)
        Printf ("Underflow");
    else
    {
        If (front > rear)
            front = -1;
        else
    }
}
```

```
    printf ("%d", q[front]);  
    front--;
```

{

{

```
void display()
```

{

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

```
    printf ("underflow");
```

{

```
else
```

{

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

```
    printf ("%d", q[i]);
```

{

{

{

```
int min ()
```

```
int boolean = 1, choice, value;
```

```
while (boolean)
```

{

```
    printf ("1.Enqueue\n 2.Dequeue\n 3.Display  
 4.Exit\n");
```

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

```
    switch (choice)
```

{

Case 1: printf ("Enter a value");  
scanf ("%d", value);  
enqueue (value);  
break;

case 2: dequeue ();  
break;

case 3: display;  
break;

case 4: boolean = 0; break;  
default: printf ("Invalid Input");  
break;  
}  
return 0;

11

## circular queue

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

```
#define N 5
```

```
int front = -1, rear = -1, q[MAX];
```

```
void enqueue (int value)
```

```
{
```

```
If ( front == -1 && rear == -1 ) {
```

```
    front = 0;
```

```
    rear = 0;
```

```
    q[rear] = value;
```

```
}
```

```
else if ((rear+1)%N == front)
```

```
{
```

```
    printf ("Overflow");
```

```
}
```

```
else
```

```
{
```

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

```
    q[rear] = value;
```

~~void dequeue () {~~~~If (front == -1) {~~ ~~printf ("Underflow");~~~~If ( front == rear )~~~~{~~ ~~front = rear = -1;~~~~}~~

else  
{  
printf ("%d", q[front]);  
front = (front + 1) % N;  
}  
}

void display()  
{

if (front == -1)  
{  
}

printf (" Underflow");  
}  
else {

int i = front;

while (i != rear) {

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

i = (i + 1) % N;

}

}

int main() {

int boolean1, mainValue;

while (Boolean)

{

printf (" 1. enqueue\n 2. Dequeue\n")

3. display\n

4. exit();

Scanf ("%d", &choice);  
switch (choice) {

case 1: printf ("Enter the value");

Scanf ("%d", &value);  
enqueue (value);

break;

case 2: dequeue(); break;

case 3: display(); break;

case 4: boolean = 0; break;

default: printf ("Invalid"); break;

}

{

{

linear  
Output

→ Enter 1. insert 2. delete 3. display 4. exit  
2

underflow

enter value:

3

Value inserted

Enter 1. insert 2. delete 3. display 4. exit 1

enter value:

4

Value inserted

Enter 1. insert 2. delete 3. display 4. exit 1

enter value:

5

value inserted

Enter 1. insert 2. delete 3. display 4. exit 1

enter value:

6

Overflow

Enter 1. insert 2. delete 3. display 4. exit 3

3

4

5

Enter 1. Insert 2. delete 3. display 4. exit 4

Bkt  
Overflow

YBLC  
6/1/21

circular

→ enter 1. insert 2. delete 3. display 4. exit 2  
underflow

enter 1. insert 2. delete 3. display 4. exit 1

Enter value: 4

value inserted

enter 1. insert 2. delete 3. display 4. exit 1  
value is enter value 5

value inserted

enter 1. insert 2. delete 3. display 4. exit 1

Enter value: 6

value inserted

enter 1. insert 2. delete 3. display 4. exit 1

Enter value: 7

Overflow

enter 1. insert 2. delete 3. display 4. exit 3

4 5 6

enter 1. insert 2. delete 3. display 4. exit 1

1. 2. 3. 4. 5. 6. 7. 8. 9.

1. 2. 3. 4. 5. 6. 7. 8. 9.

1. 2. 3. 4. 5. 6. 7. 8. 9.

1. 2. 3. 4. 5. 6. 7. 8. 9.