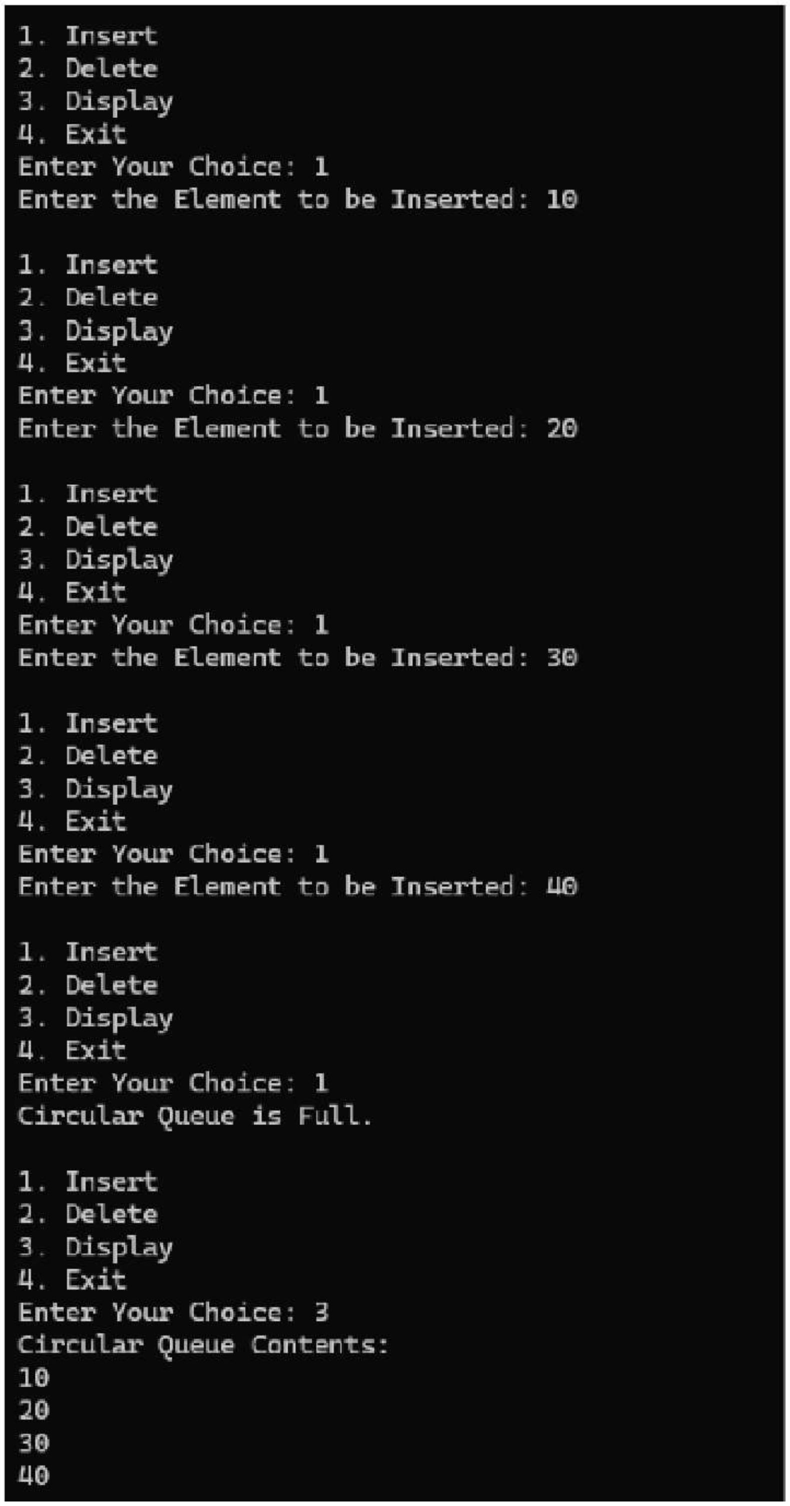
LAB PROGRAM 4:

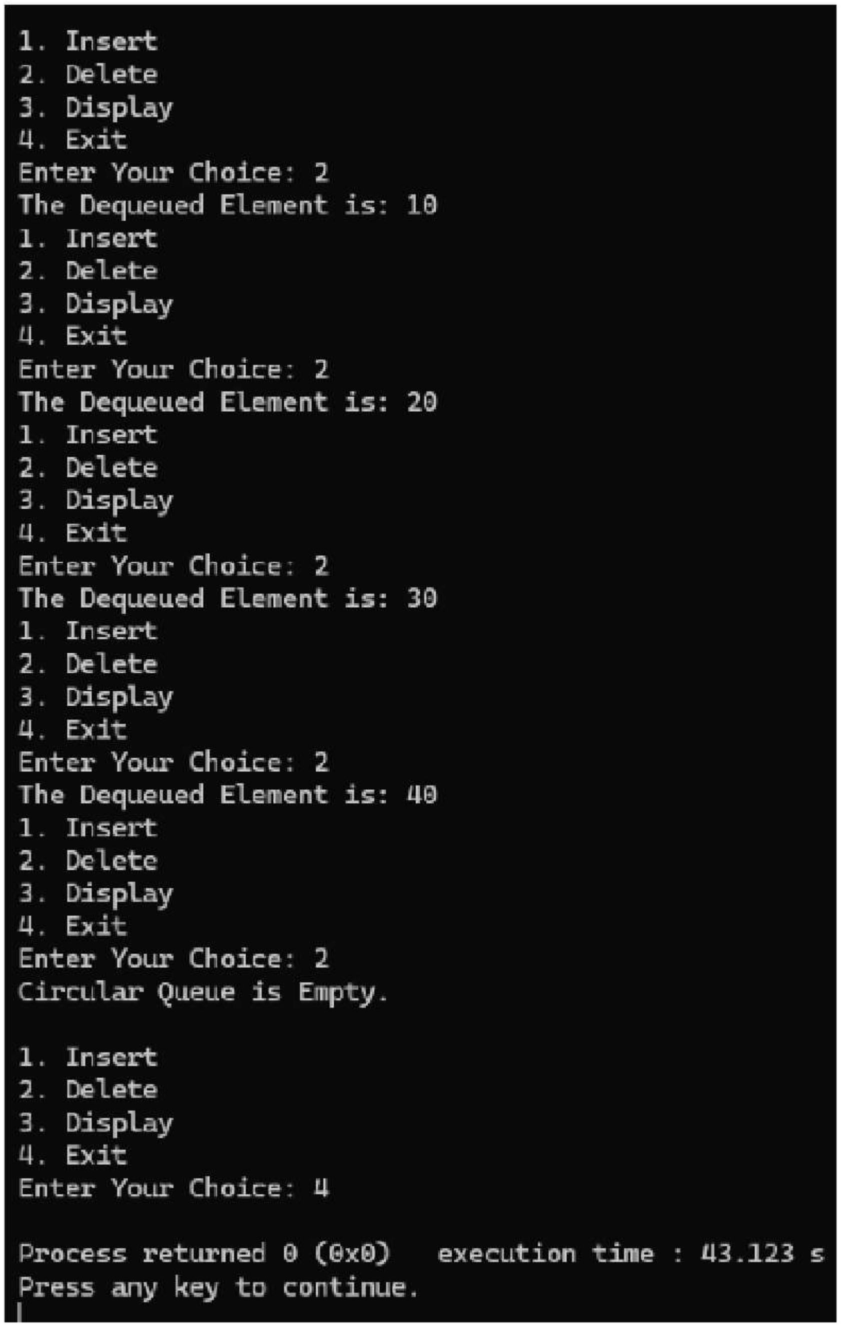
Circular Queue implementation

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX 4

void Insert();
int Delete();
void Display();

int cq[20];
int front=-1, rear=-1, item, ch, i;

void main()
{
   while(1)
   {
      printf(" \n1. Insert \n2. Delete \n3. Display \n4. Exit");
      printf("\nEnter Your Choice: ");
      scanf("%d",&ch);
      switch(ch)
      {
         case 1: Insert();
            break;
         case 2: item=Delete();
```

```
                if (item!=-1)
                {
                    printf("The Dequeued Element is: %d",item);
                }
                break;
            case 3: Display();
                break;
            case 4: exit(0);
        }
    }
}

void Insert()
{
    if (front == (rear+1) % MAX)
        {
            printf("Circular Queue is Full. \n");
            return;
        }
    if (rear==-1 && front==-1)
    {
        rear=0;
        front=0;
    }
    else
        rear=(rear+1)%MAX;
    printf("Enter the Element to be Inserted: ");
    scanf("%d",&item);
    cq[rear]=item;
    return;
}

int Delete()
{
    if(front==-1 && rear==-1)
    {
        printf("Circular Queue is Empty. \n");
        return (-1);
    }
    item=cq[front];
    if(front==rear)
    {
        front=-1;
        rear=-1;
    }
    else
        front=(front+1)%MAX;
    return item;
}

void Display()
{
    if(front==-1 && rear==-1)
    {
        printf("Circular Queue is Empty. \n");
        return;
    }

    printf("Circular Queue Contents: \n");
    if (front<=rear)
    {
        for (int i=front;i<=rear;i++)
        {
            printf("%d\n",cq[i]);
        }
    }

   else
   {
        for(int i=front;i<=MAX-1;i++)
        {
            printf("%d\n",cq[i]);
        }
        for (int i=0;i<=rear;i++)
        {
            printf("%d\n",cq[i]);
        }
   }
   return;
}
```



```
1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice: 1
Enter the Element to be Inserted: 10

1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice: 1
Enter the Element to be Inserted: 20

1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice: 1
Enter the Element to be Inserted: 30

1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice: 1
Enter the Element to be Inserted: 40

1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice: 1
Circular Queue is Full.

1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice: 3
Circular Queue Contents:
10
20
30
40
```


```
1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice: 2
The Dequeued Element is: 10
1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice: 2
The Dequeued Element is: 20
1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice: 2
The Dequeued Element is: 30
1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice: 2
The Dequeued Element is: 40
1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice: 2
Circular Queue is Empty.

1. Insert
2. Delete
3. Display
4. Exit
Enter Your Choice: 4

Process returned 0 (0x0)   execution time : 43.123 s
Press any key to continue.
```

LAB 4 LEET CODE:

For a stream of integers, implement a data structure that checks if the last k integers parsed in the stream are equal to value.

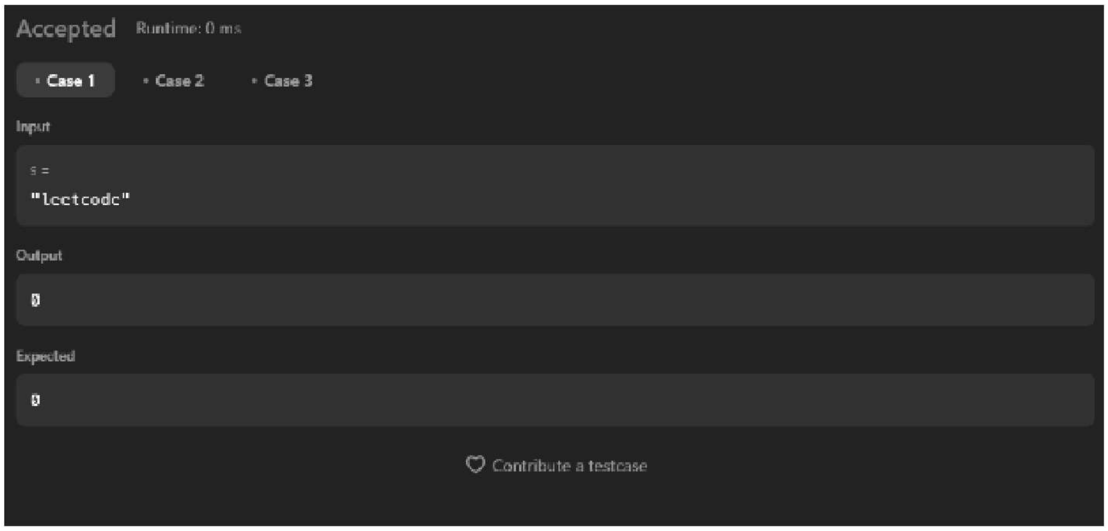
Implement the DataStream class:

DataStream(int value, int k) Initializes the object with an empty integer stream and the two integers value and k.

boolean consec(int num) Adds num to the stream of integers. Returns true if the last k integers are equal to value, and false otherwise. If there are less than k integers, the condition does not hold true, so returns false.

```
class DataStream {

public:

   int val;

   int k;

   int cnt=0;


   DataStream(int value, int K) {

      val=value;

      k=K;

   }


   bool consec(int num) {

      if(num==val)

      {

         cnt++;

         if(cnt>=k)

            return true;

         return false;
```

```
        }
        cnt=0;
        return false;
    }
};
```



Accepted Runtime: 0 ms

Input

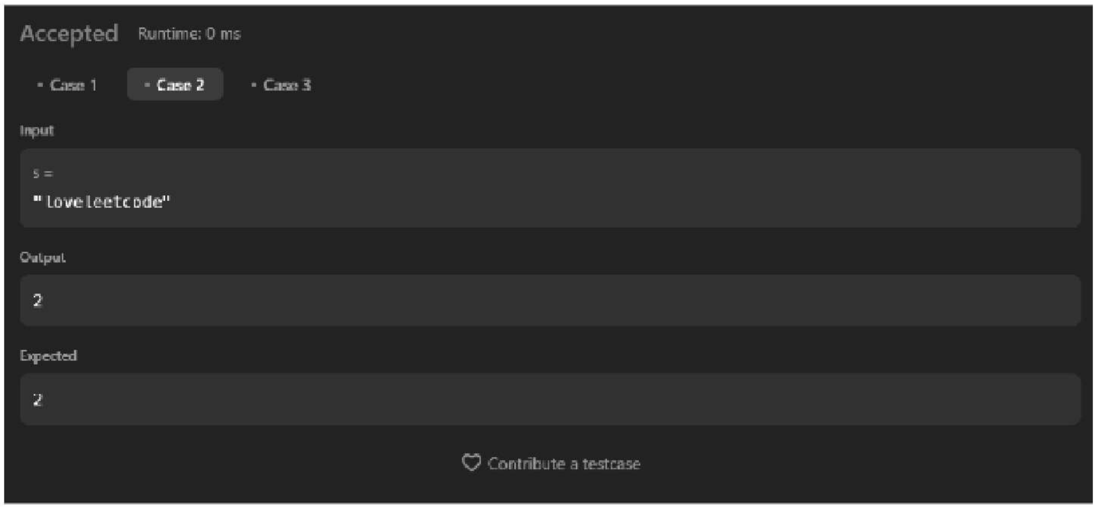
```
s =
"leetcode"
```

Output

```
0
```

Expected

```
0
```



Accepted Runtime: 0 ms

Input

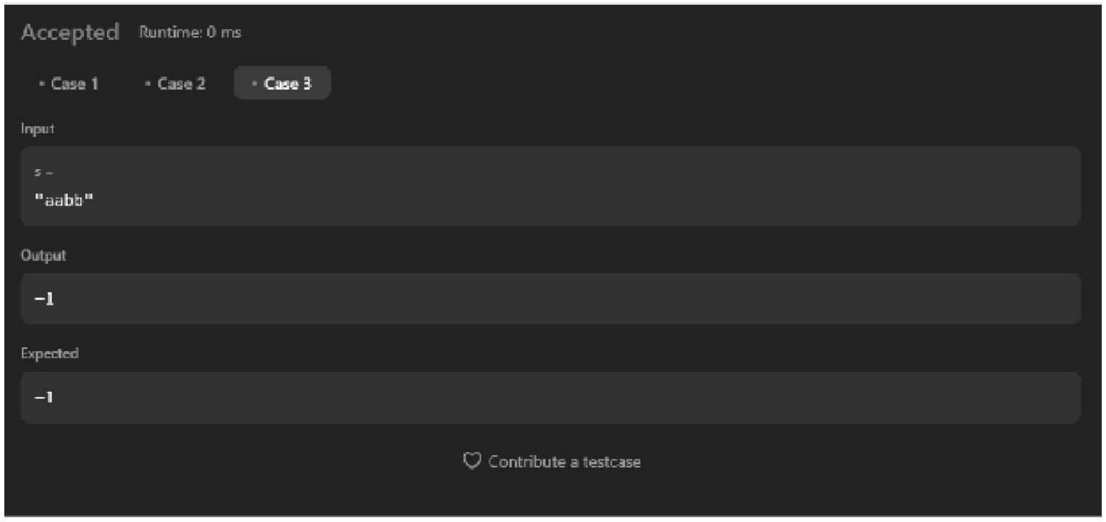
```
s =
"loveleetcode"
```

Output

```
2
```

Expected

```
2
```


Accepted Runtime: 0 ms

Input

```
s =
"aabb"
```

Output

```
-1
```

Expected

```
-1
```