10. Write a c program for implementing of stack and it's operation.

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

#define SIZE 5

int stack[SIZE];
int top = -1;

void push(int val)
{
    if (top == SIZE - 1)
    {
        printf("Stack is Full..\n");
    }
    else
    {
        stack[top] = val;
        printf("Element Pushed : %d \n", val);
    }
}

void pop()
{
    if (top == -1)
    {
        printf("Stack is Empty..\n");
    }
    else
    {
        printf("Element Popped : %d \n", stack[top]);
        top--;
    }
}

int length()
{
    int i = 0;
    if (top == -1)
    {
        printf("Stack is Empty..\n");
    }
    else
    {
        while (i <= top)
        {
            i++;
        }
    }
    return i;
}

void display()
{
    int i = 0;
    if (top == -1)
    {
        printf("Stack is Empty..\n");
    }
    else
    {
        while (i <= top)
        {
            printf("%d \n", stack[i]);
            i++;
        }
    }
}

void peek()
{
    if (top == -1)
        printf("Stack is Empty..\n");

    else
        printf("Top Item is : %d \n", stack[top]);
}

void main()
{
    int choice;
    int item, len;
    while (1)
    {
        printf("1. Push. \n");
        printf("2. Pop. \n");
        printf("3. Length \n");
        printf("4. Display. \n");
        printf("5. Peek. \n");
        printf("0. Exit. \n");
        printf("Enter Your Choice : ");
        scanf("%d", &choice);

        switch (choice)
        {
        case 1:
            printf("Enter Element to Push : ");
            scanf("%d", &item);
            push(item);
            break;
        case 2:
            pop();
            break;
        case 3:
            len = length();
            printf("Length : %d \n", len);
            break;
        case 4:
            display();
            break;
        case 5:
            peek();
            break;
        case 0:
            printf("Bye Bye \n \n");
            exit(1);

        default:
            printf("Invalid Choice. \n\n");
        }
    }
}
```

Output :

```
1. Push.
2. Pop.
3. Length
4. Display.
5. Peek.
0. Exit.
Enter Your Choice : 1
Enter Element to Push : 10
Element Pushed : 10
```

```
1. Push.
2. Pop.
3. Length
4. Display.
5. Peek.
0. Exit.
Enter Your Choice : 1
Enter Element to Push : 60
Stack is Full..
```

```
1. Push.
2. Pop.
3. Length
4. Display.
5. Peek.
0. Exit.
Enter Your Choice : 2
Element Popped : 50
```

```
1. Push.
2. Pop.
3. Length
4. Display.
5. Peek.
0. Exit.
Enter Your Choice : 3
Length : 4
```

```
1. Push.
2. Pop.
3. Length
4. Display.
5. Peek.
0. Exit.
Enter Your Choice : 4
10
20
30
40
```

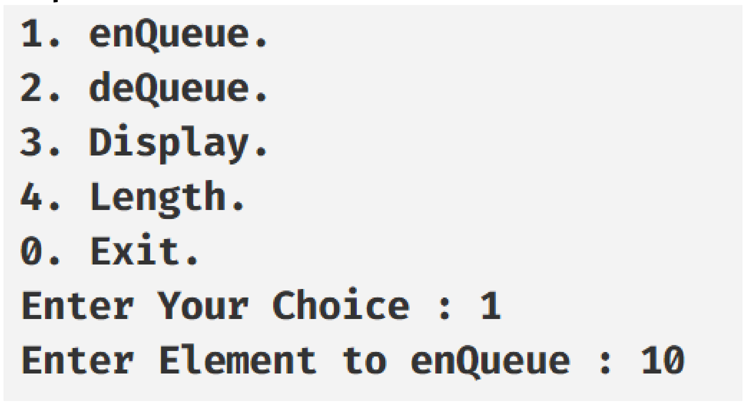
```
1. Push.
2. Pop.
3. Length
4. Display.
5. Peek.
0. Exit.
Enter Your Choice : 5
Top Item is : 40
```

13. Write a c program for implementing a simple queue and its operation.

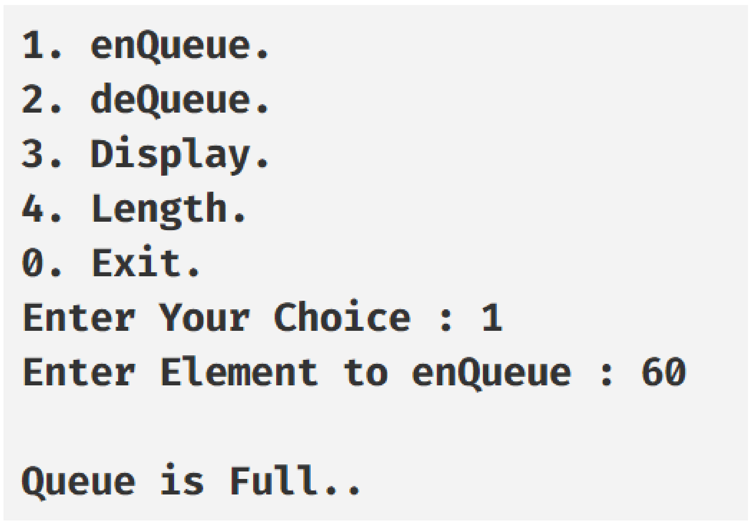
```
#include <stdio.h>
#define SIZE 5

int myqueue[SIZE];
int front = -1, rear = -1;

void enQueue(int value)
{
    if (rear == SIZE - 1)
        printf("\nQueue is Full.. \n\n");
    else
    {
        rear++;
        myqueue[rear] = value;
        printf("\nInserted : %d \n\n", value);
        if (front == -1)
        {
            front++;
        }
    }
}

void deQueue()
{
    if (front == -1)
        printf("\nQueue is Empty!!\n");
    else
    {
        printf("\nItem Deleted : %d \n", myqueue[front]);
        front++;

        if (front > rear)
        {
            front = -1;
            rear = -1;
        }
    }
}

void display()
{
    if (front == -1)
        printf("\nQueue is Empty!!!");
    else
    {
        int i = front;
        printf("\nQueue elements are:\n");

        while (i <= rear)
        {
            printf("%d  ", myqueue[i]);
            i++;
        }
    }
    printf("\n");
}

void length()
{
    int count = 0;
    int i = front;
    if (front == -1)
    {
        printf("Queue is Empty..\n");
    }
    else
    {
        while (i <= rear)
        {
            count++;
            i++;
        }
        printf("Length is : %d \n", count);
    }
}

void main()
{
    int choice;
    int item, len;
    while (1)
    {
        printf("1. enQueue. \n");
        printf("2. deQueue. \n");
        printf("3. Display. \n");
        printf("4. Length. \n");
        printf("5. Delete all. \n");
        printf("0. Exit. \n");

        printf("Enter Your Choice : ");
        scanf("%d", &choice);

        switch (choice)
        {
        case 1:
            printf("Enter Element to enQueue : ");
            scanf("%d", &item);
            enQueue(item);
            break;
        case 2:
            deQueue();
            break;
        case 3:
            display();
            break;
        case 4:
            length();
```

```
                break;
            case 0:
                exit(0);
            default:
                printf("Invalid Choice. \n\n");
            }
        }
    }
```

Output :



```
1. enQueue.
2. deQueue.
3. Display.
4. Length.
0. Exit.
Enter Your Choice : 1
Enter Element to enQueue : 10
```



```
1. enQueue.
2. deQueue.
3. Display.
4. Length.
0. Exit.
Enter Your Choice : 1
Enter Element to enQueue : 60

Queue is Full..
```

```
1. enQueue.
2. deQueue.
3. Display.
4. Length.
0. Exit.
Enter Your Choice : 2

Item Deleted : 10
```

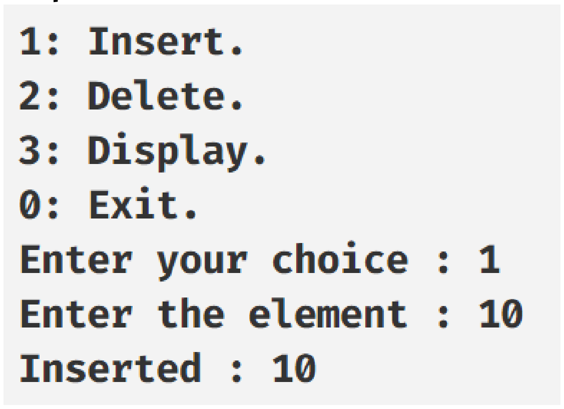
```
1. enQueue.
2. deQueue.
3. Display.
4. Length.
0. Exit.
Enter Your Choice : 3

Queue elements are:
20  30  40  50
```

```
1. enQueue.
2. deQueue.
3. Display.
4. Length.
0. Exit.
Enter Your Choice : 4
Length is : 4
```

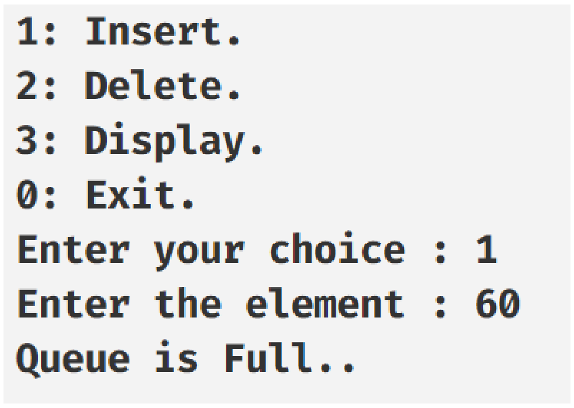
15. Write a c program for implementing a circular Queue and its operation.

```
#include <stdio.h>

#define SIZE 5

int CQueue[SIZE];
int front = -1;
int rear = -1;

void enQueue(int value)
{
    if (front == rear + 1 || front == 0 && rear == SIZE - 1)
    {
        printf("Queue is Full..\n");
    }
    else
    {
        rear = (rear + 1) % SIZE;
        CQueue[rear] = value;

        printf("Inserted : %d \n", value);

        if (front == -1)
        {
            front = 0;
        }
    }
}

int deQueue()
{
    if ((front == -1) && (rear == -1)) // check CQueue is empty
    {
        printf("\nQueue is Empty..");
    }
    else if (front == rear)
    {
        printf("\nThe dequeued element is %d", CQueue[front]);
        front = -1;
        rear = -1;
    }
    else
    {
        printf("\nThe dequeued element is %d", CQueue[front]);
        front = (front + 1) % SIZE;
    }
}

void display()
{
    int i = front;
    if (front == -1 && rear == -1)
    {
        printf("\n Queue is empty..");
    }
    else
    {
        printf("\nElements in a Queue are :");
        while (i != rear)
        {
            printf("%d ", CQueue[i]);
            i = (i + 1) % SIZE;
        }
        printf("%d ", CQueue[i]);
    }
}

void main()
{
    int choice = 1, x;

    while (1)
    {
        printf("\n1: Insert.");
        printf("\n2: Delete.");
        printf("\n3: Display.");
        printf("\n0: Exit.");
        printf("\nEnter your choice : ");
        scanf("%d", &choice);

        switch (choice)
        {
        case 1:

            printf("Enter the element : ");
            scanf("%d", &x);
            enQueue(x);
            break;
        case 2:
            deQueue();
            break;
        case 3:
            display();
            break;
        case 0:
            exit(0);
            break;
        default:
            printf("Enter a Valid Choice.\n");
        }
    }
}
```

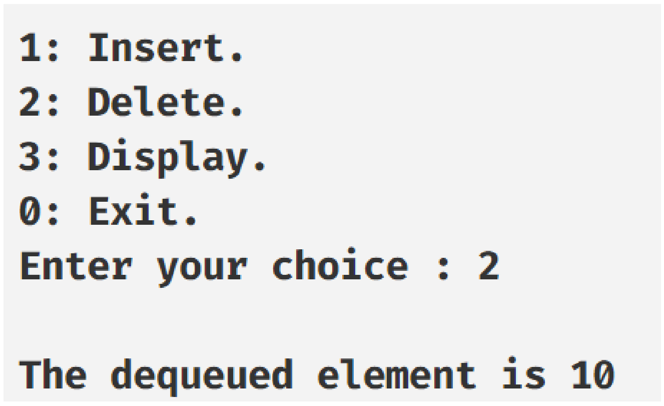
Output :



```
1: Insert.
2: Delete.
3: Display.
0: Exit.
Enter your choice : 1
Enter the element : 10
Inserted : 10
```



```
1: Insert.
2: Delete.
3: Display.
0: Exit.
Enter your choice : 1
Enter the element : 60
Queue is Full..
```



```
1: Insert.
2: Delete.
3: Display.
0: Exit.
Enter your choice : 2

The dequeued element is 10
```

```
1: Insert.
2: Delete.
3: Display.
0: Exit.
Enter your choice : 3

Elements in a Queue are :20 30 40 50
```