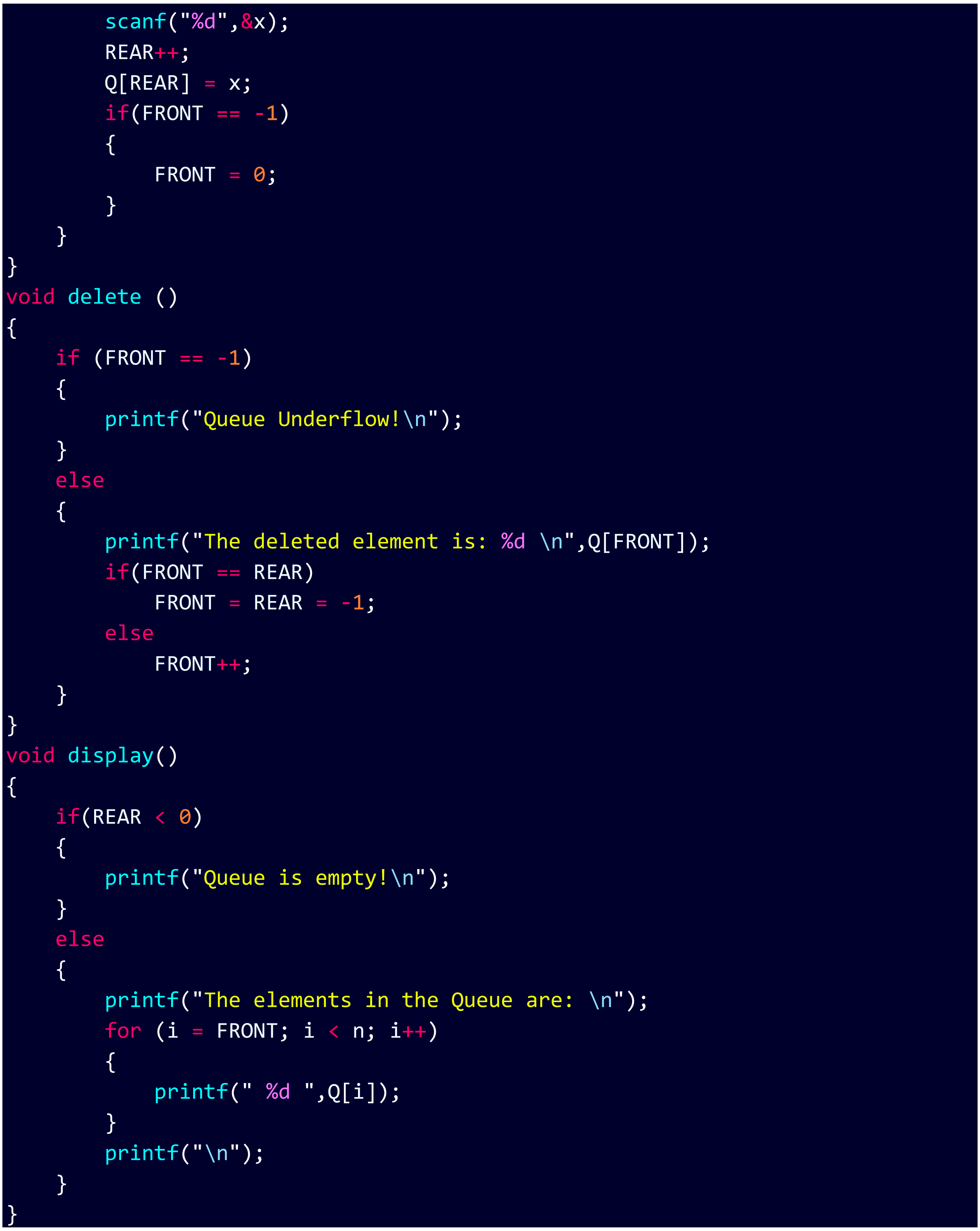
Experiment 2:

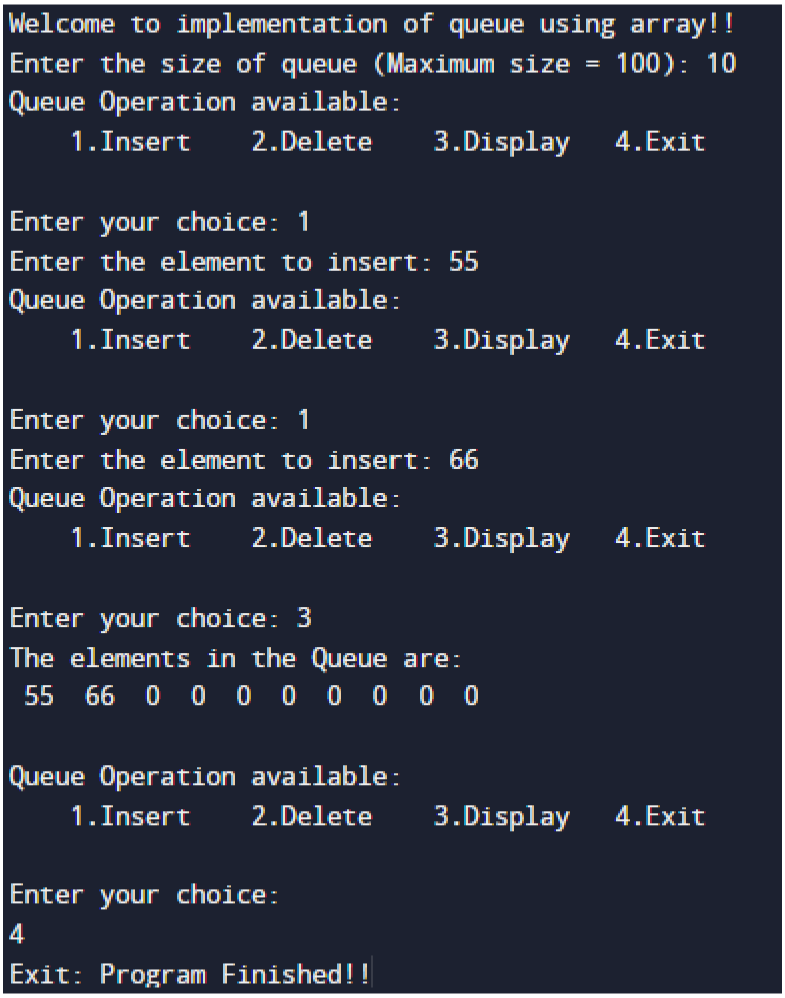
Code:

```
#include <stdio.h>
int Q[100], FRONT = -1, REAR = -1, i, n, x, choice;
void insert();
void delete ();
void display();
void main()
{
    printf("\t Welcome to implementation of queue using array!!\n");
    printf("Enter the size of queue (Maximum size = 100): ");
    scanf("%d", &n);
    do
    {
        printf("\nQueue Operation available: \n");
        printf("\t1.Insert \t2.Delete \t3.Display \t4.Exit \n");
        printf("\nEnter your choice: ");
        scanf("%d",&choice);
        switch (choice)
        {
        case 1:
            insert();
            break;
        case 2:
            delete ();
            break;
        case 3:
            display();
            break;
        case 4:
            printf("Exit: Program Finished!!");
            break;
        default:
            printf("Please enter a valid choice 1, 2, 3, 4 \n");
            break;
        }
    } while (choice != 4);
}
void insert()
{
    if (REAR >= n - 1)
    {
        printf("Queue Overflow!\n");
    }
    else
    {
        printf("Enter the element to insert: ");
```



```
        scanf("%d",&x);
        REAR++;
        Q[REAR] = x;
        if(FRONT == -1)
        {
            FRONT = 0;
        }
    }
}
void delete ()
{
    if (FRONT == -1)
    {
        printf("Queue Underflow!\n");
    }
    else
    {
        printf("The deleted element is: %d \n",Q[FRONT]);
        if(FRONT == REAR)
            FRONT = REAR = -1;
        else
            FRONT++;
    }
}
void display()
{
    if(REAR < 0)
    {
        printf("Queue is empty!\n");
    }
    else
    {
        printf("The elements in the Queue are: \n");
        for (i = FRONT; i < n; i++)
        {
            printf(" %d ",Q[i]);
        }
        printf("\n");
    }
}
```

Output:



```
Welcome to implementation of queue using array!!
Enter the size of queue (Maximum size = 100): 10
Queue Operation available:
    1.Insert    2.Delete    3.Display   4.Exit

Enter your choice: 1
Enter the element to insert: 55
Queue Operation available:
    1.Insert    2.Delete    3.Display   4.Exit

Enter your choice: 1
Enter the element to insert: 66
Queue Operation available:
    1.Insert    2.Delete    3.Display   4.Exit

Enter your choice: 3
The elements in the Queue are:
 55  66  0  0  0  0  0  0  0  0

Queue Operation available:
    1.Insert    2.Delete    3.Display   4.Exit

Enter your choice:
4
Exit: Program Finished!!
```