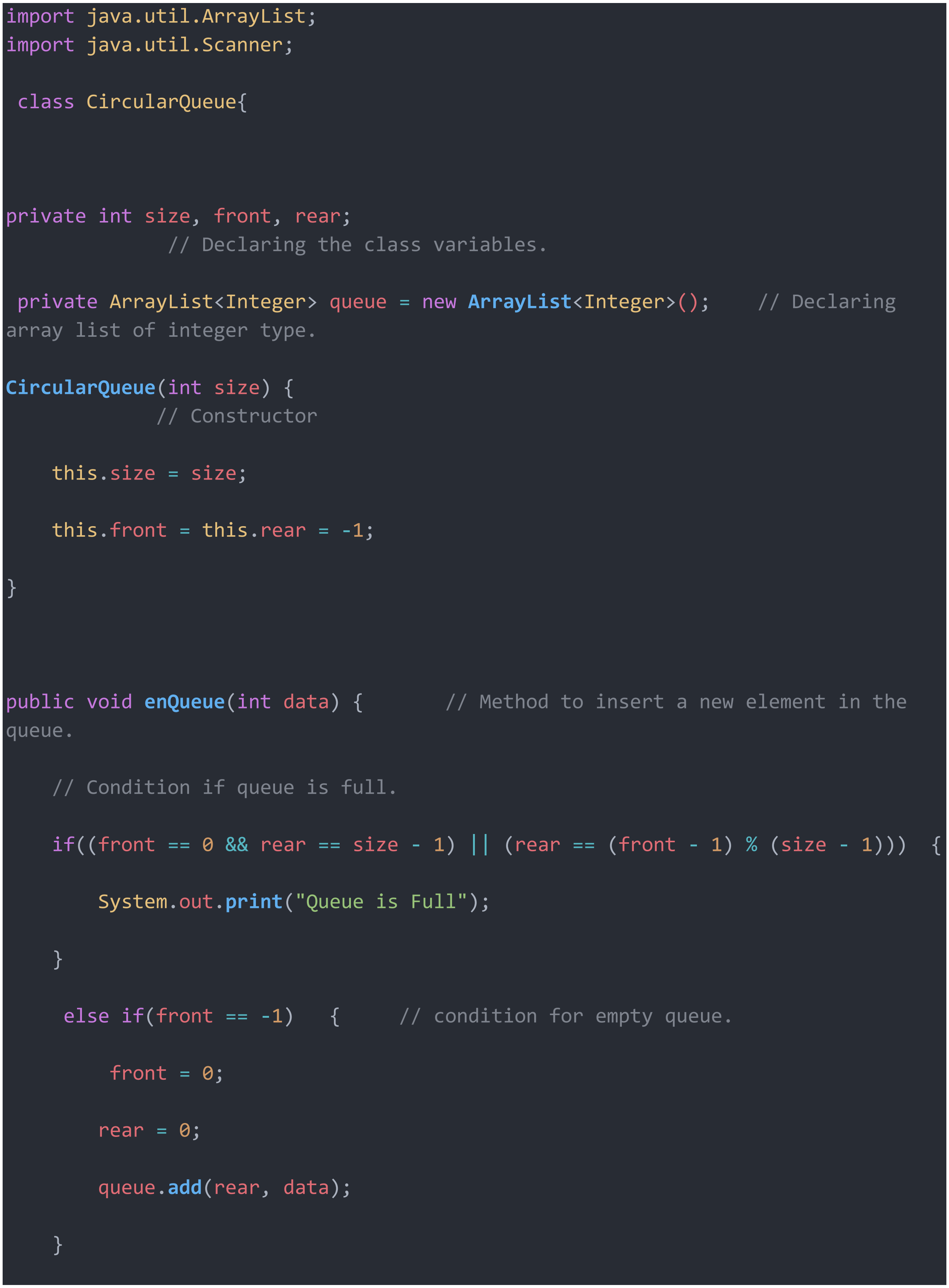
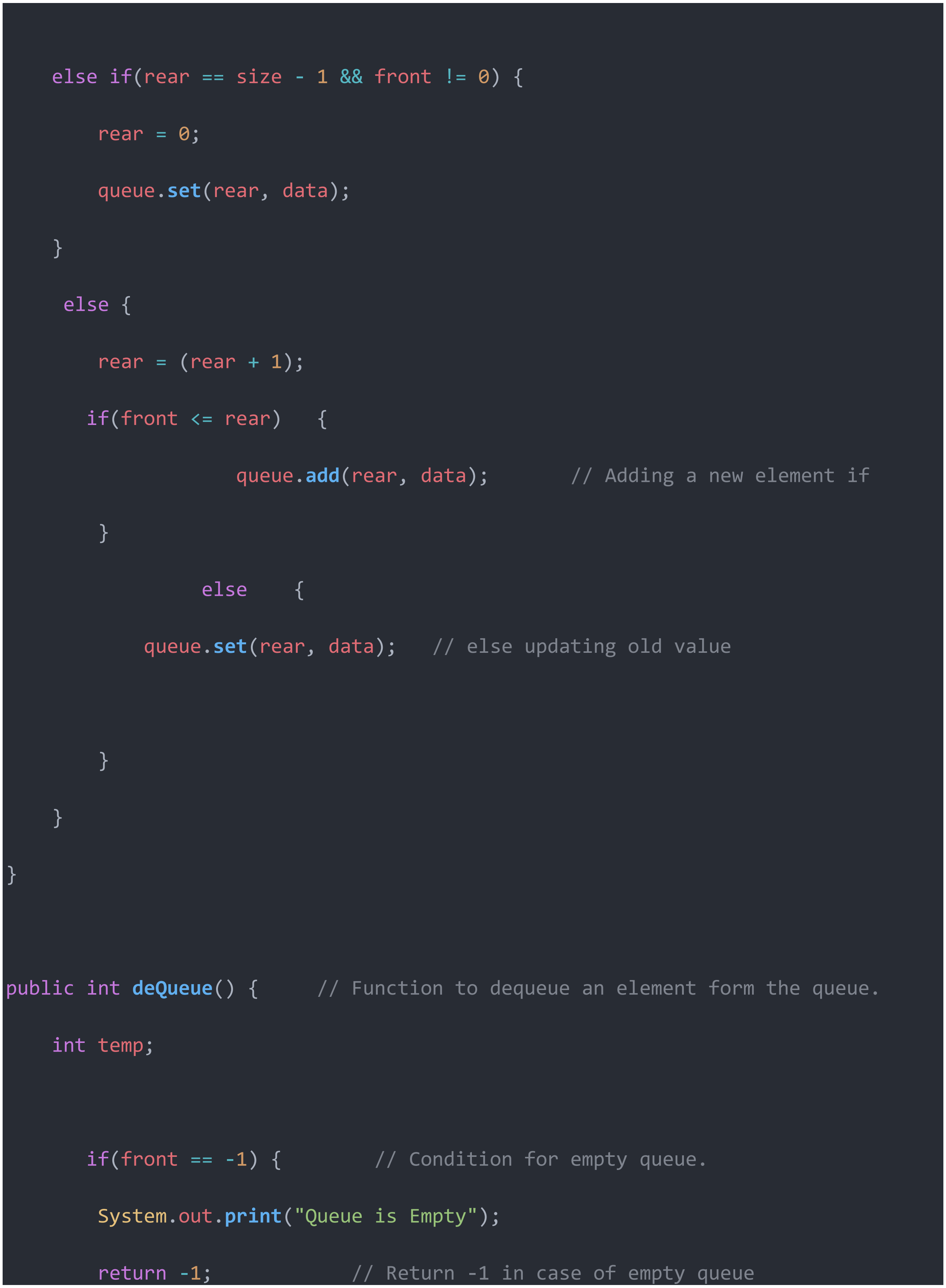
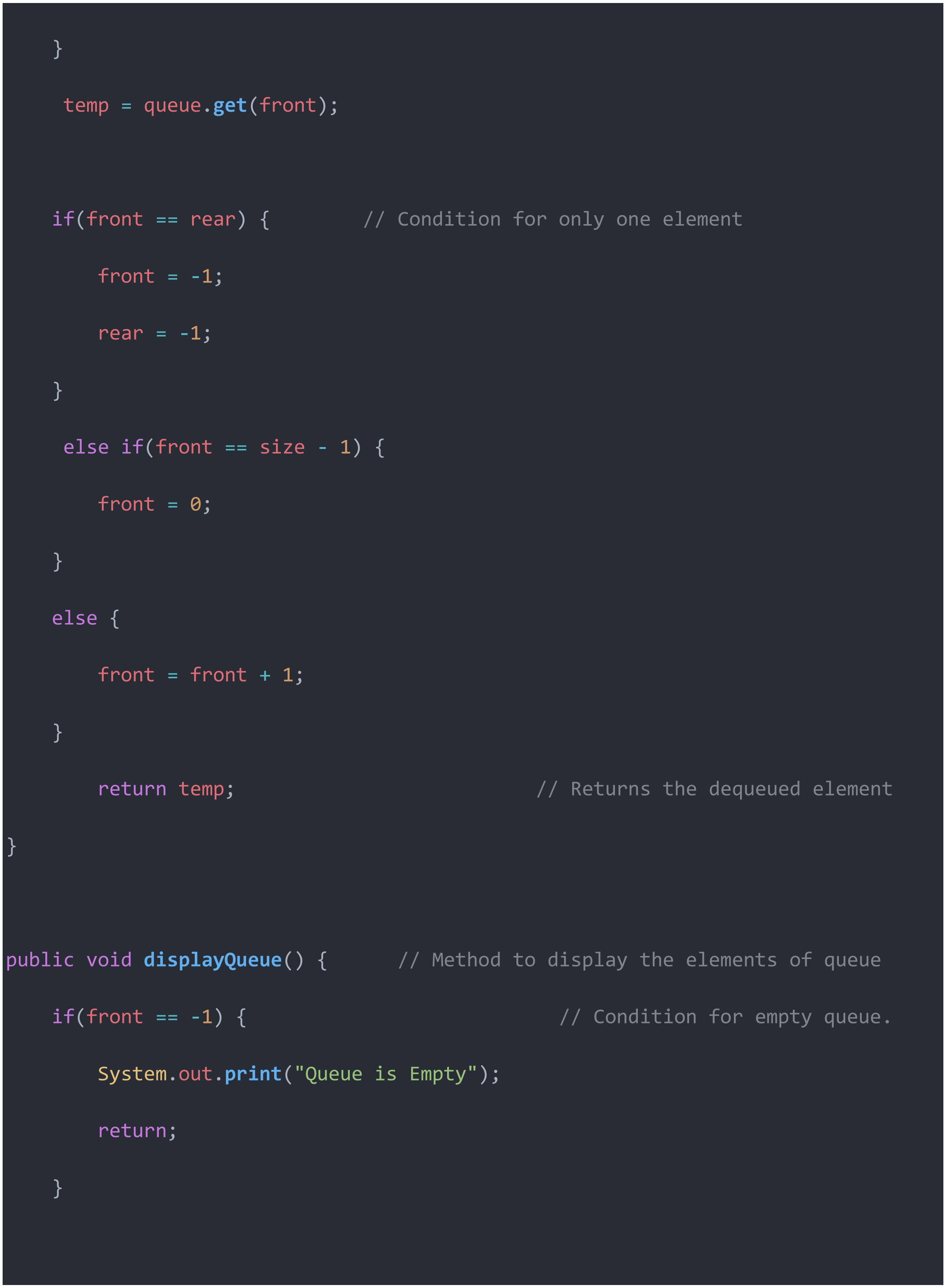


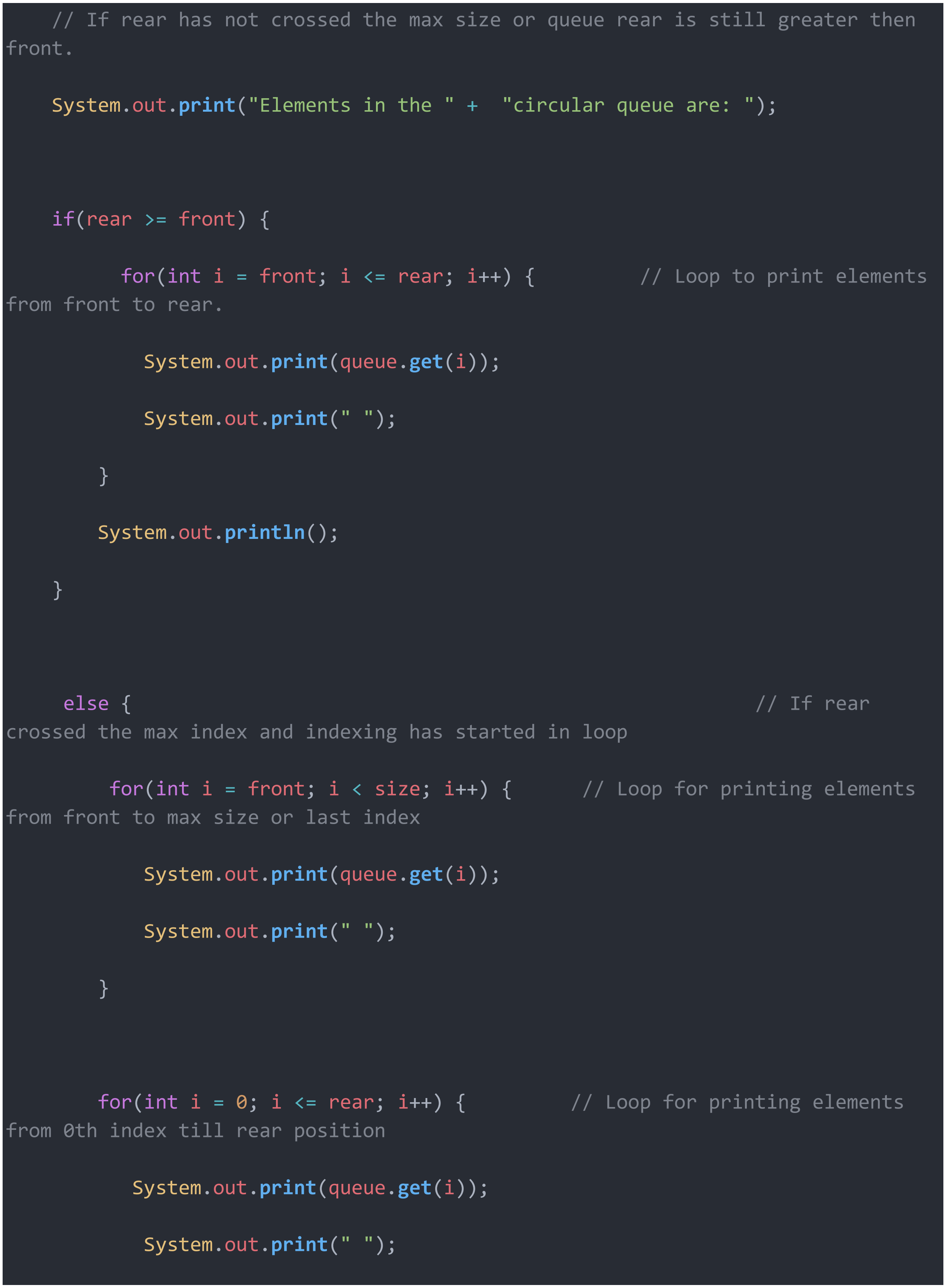
```
import java.util.ArrayList;
import java.util.Scanner;

 class CircularQueue{



private int size, front, rear;
              // Declaring the class variables.

 private ArrayList<Integer> queue = new ArrayList<Integer>();    // Declaring
array list of integer type.

CircularQueue(int size) {
             // Constructor

    this.size = size;

    this.front = this.rear = -1;

}



public void enQueue(int data) {       // Method to insert a new element in the
queue.

    // Condition if queue is full.

    if((front == 0 && rear == size - 1) || (rear == (front - 1) % (size - 1)))  {

        System.out.print("Queue is Full");

    }

     else if(front == -1)   {     // condition for empty queue.

         front = 0;

        rear = 0;

        queue.add(rear, data);

    }
```



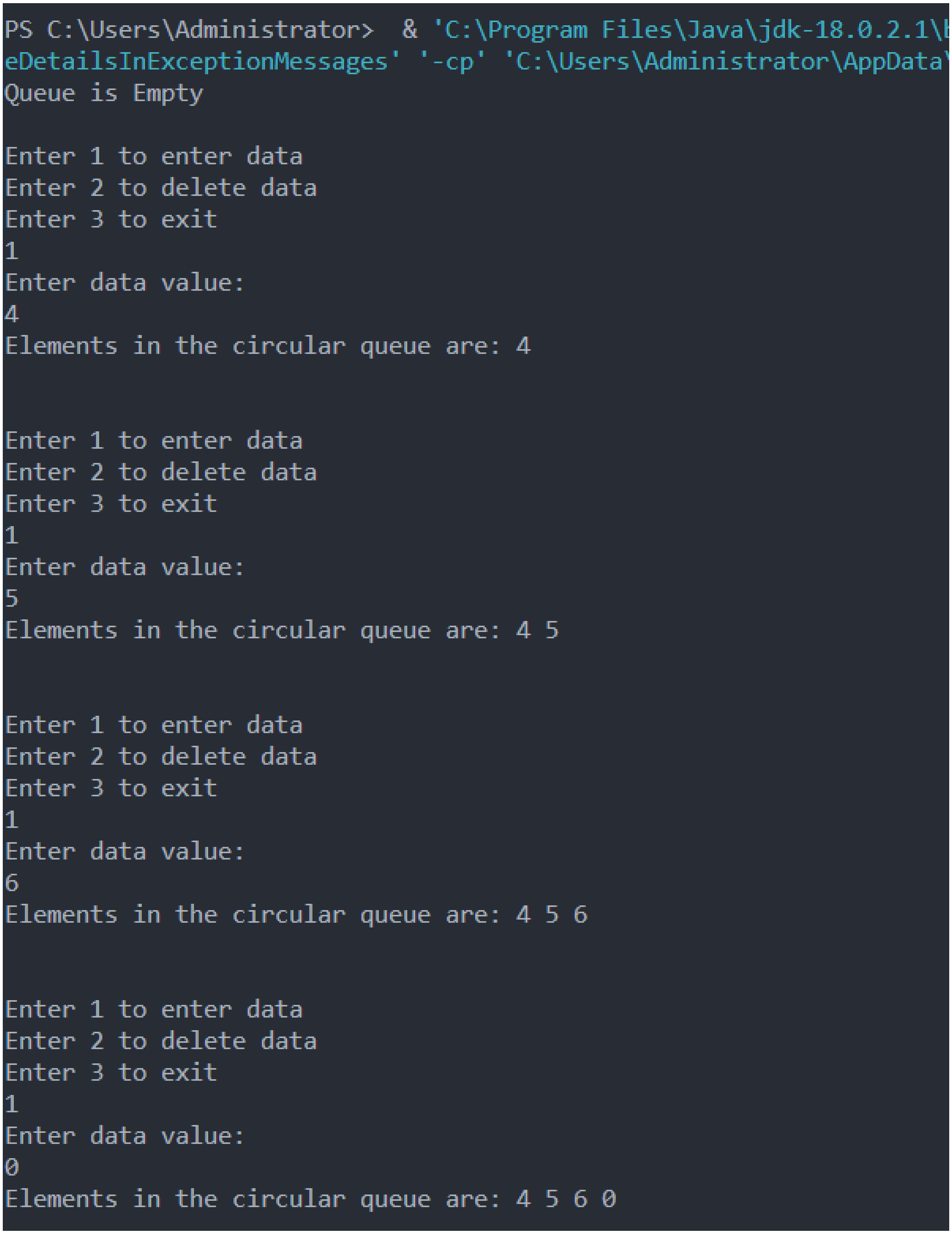
```
    else if(rear == size - 1 && front != 0) {
        rear = 0;
        queue.set(rear, data);
    }
     else {
        rear = (rear + 1);
       if(front <= rear)   {
                    queue.add(rear, data);       // Adding a new element if
        }
                 else    {
            queue.set(rear, data);   // else updating old value

        }
    }
}


public int deQueue() {     // Function to dequeue an element form the queue.
    int temp;

       if(front == -1) {        // Condition for empty queue.
        System.out.print("Queue is Empty");
        return -1;            // Return -1 in case of empty queue
```



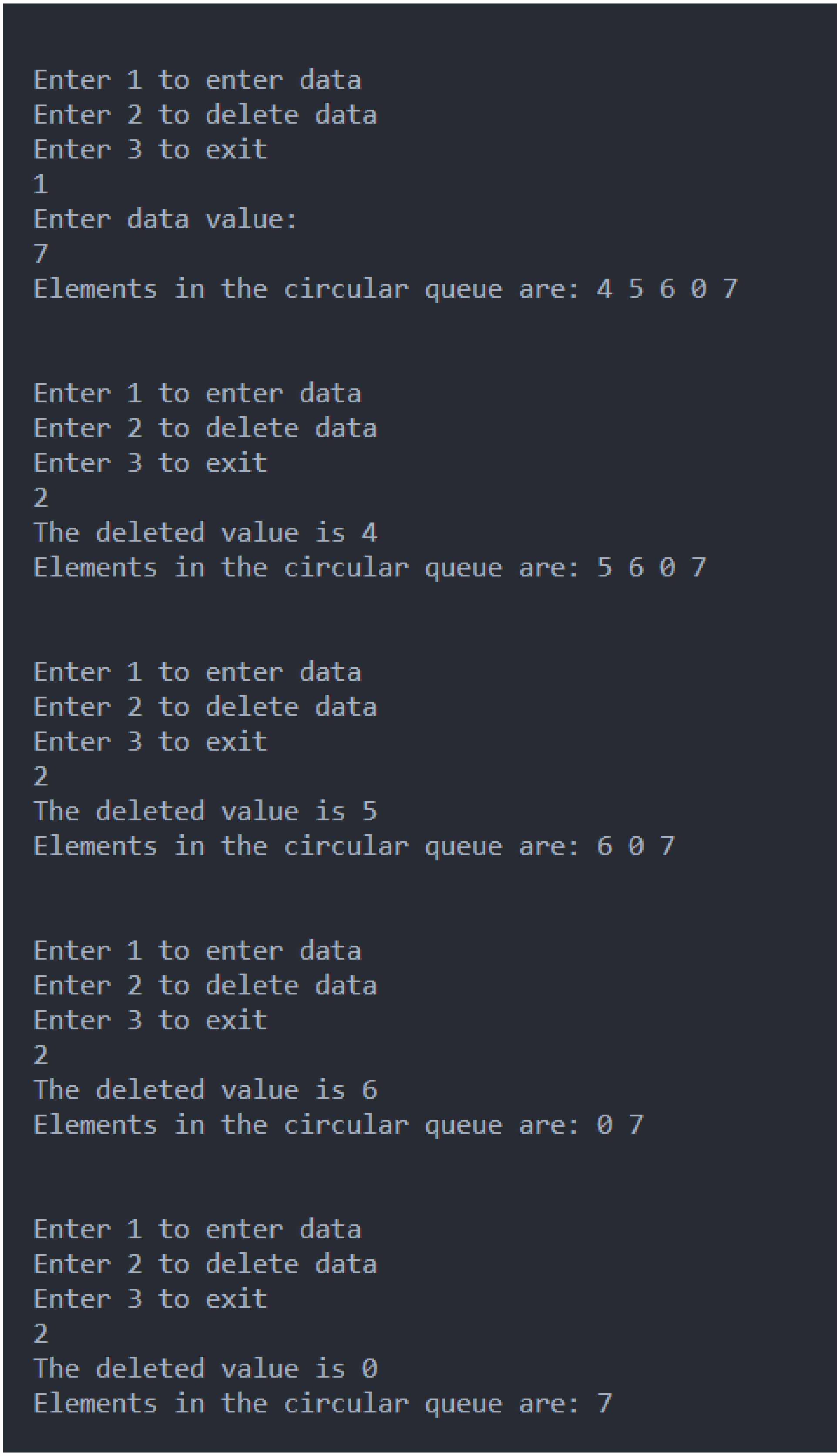
```
    }
     temp = queue.get(front);

    if(front == rear) {        // Condition for only one element
        front = -1;
        rear = -1;
    }
     else if(front == size - 1) {
        front = 0;
    }
    else {
        front = front + 1;
    }
        return temp;                          // Returns the dequeued element
}

public void displayQueue() {      // Method to display the elements of queue
    if(front == -1) {                           // Condition for empty queue.
        System.out.print("Queue is Empty");
        return;
    }
```



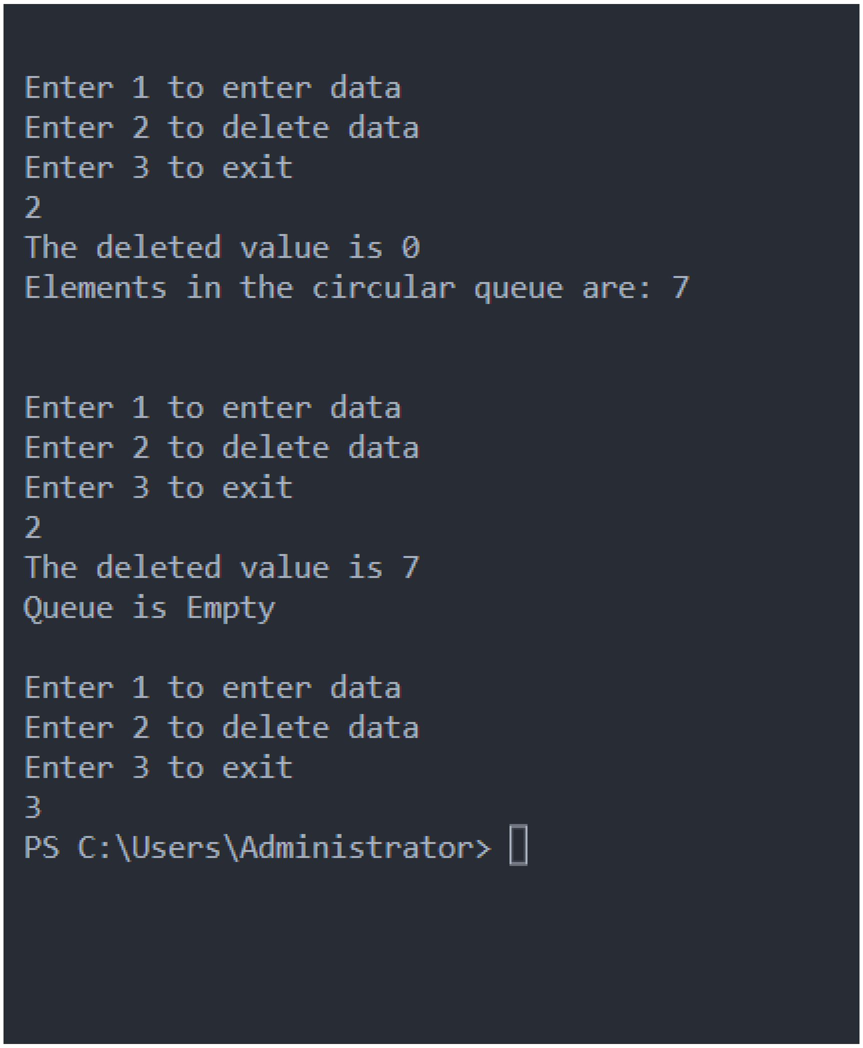
```
    // If rear has not crossed the max size or queue rear is still greater then
front.

    System.out.print("Elements in the " +  "circular queue are: ");



    if(rear >= front) {

          for(int i = front; i <= rear; i++) {         // Loop to print elements
from front to rear.

            System.out.print(queue.get(i));

            System.out.print(" ");

        }

        System.out.println();

    }



     else {                                                     // If rear
crossed the max index and indexing has started in loop

         for(int i = front; i < size; i++) {      // Loop for printing elements
from front to max size or last index

            System.out.print(queue.get(i));

            System.out.print(" ");

        }



        for(int i = 0; i <= rear; i++) {         // Loop for printing elements
from 0th index till rear position

           System.out.print(queue.get(i));

            System.out.print(" ");
        }

        System.out.println();

    }

}



public static void main(String[] args) {                                     //
Driver code


    CircularQueue cQueue = new CircularQueue(5);
    Scanner scanner = new Scanner(System.in);
    int userChoice;
    do{
        cQueue.displayQueue();
        System.out.println("\n\nEnter 1 to enter data\nEnter 2 to delete
data\nEnter 3 to exit");
        userChoice = scanner.nextInt();
        switch(userChoice){
            case 1:
                System.out.println("Enter data value: ");
                int dataVal = scanner.nextInt();
                cQueue.enQueue(dataVal);
                break;
            case 2:
                int dequeueVal = cQueue.deQueue();
                if(dequeueVal != -1){
                    System.out.println("The deleted value is " + dequeueVal);

                }else{
                    System.out.println("List is empty");
                }
        }
    }while(userChoice != 3);

}

}
```



```
PS C:\Users\Administrator>  & 'C:\Program Files\Java\jdk-18.0.2.1\
eDetailsInExceptionMessages' '-cp' 'C:\Users\Administrator\AppData
Queue is Empty

Enter 1 to enter data
Enter 2 to delete data
Enter 3 to exit
1
Enter data value:
4
Elements in the circular queue are: 4


Enter 1 to enter data
Enter 2 to delete data
Enter 3 to exit
1
Enter data value:
5
Elements in the circular queue are: 4 5


Enter 1 to enter data
Enter 2 to delete data
Enter 3 to exit
1
Enter data value:
6
Elements in the circular queue are: 4 5 6


Enter 1 to enter data
Enter 2 to delete data
Enter 3 to exit
1
Enter data value:
0
Elements in the circular queue are: 4 5 6 0
```



```
Enter 1 to enter data
Enter 2 to delete data
Enter 3 to exit
1
Enter data value:
7
Elements in the circular queue are: 4 5 6 0 7


Enter 1 to enter data
Enter 2 to delete data
Enter 3 to exit
2
The deleted value is 4
Elements in the circular queue are: 5 6 0 7


Enter 1 to enter data
Enter 2 to delete data
Enter 3 to exit
2
The deleted value is 5
Elements in the circular queue are: 6 0 7


Enter 1 to enter data
Enter 2 to delete data
Enter 3 to exit
2
The deleted value is 6
Elements in the circular queue are: 0 7


Enter 1 to enter data
Enter 2 to delete data
Enter 3 to exit
2
The deleted value is 0
Elements in the circular queue are: 7
```



```
Enter 1 to enter data
Enter 2 to delete data
Enter 3 to exit
2
The deleted value is 0
Elements in the circular queue are: 7


Enter 1 to enter data
Enter 2 to delete data
Enter 3 to exit
2
The deleted value is 7
Queue is Empty

Enter 1 to enter data
Enter 2 to delete data
Enter 3 to exit
3
PS C:\Users\Administrator>
```