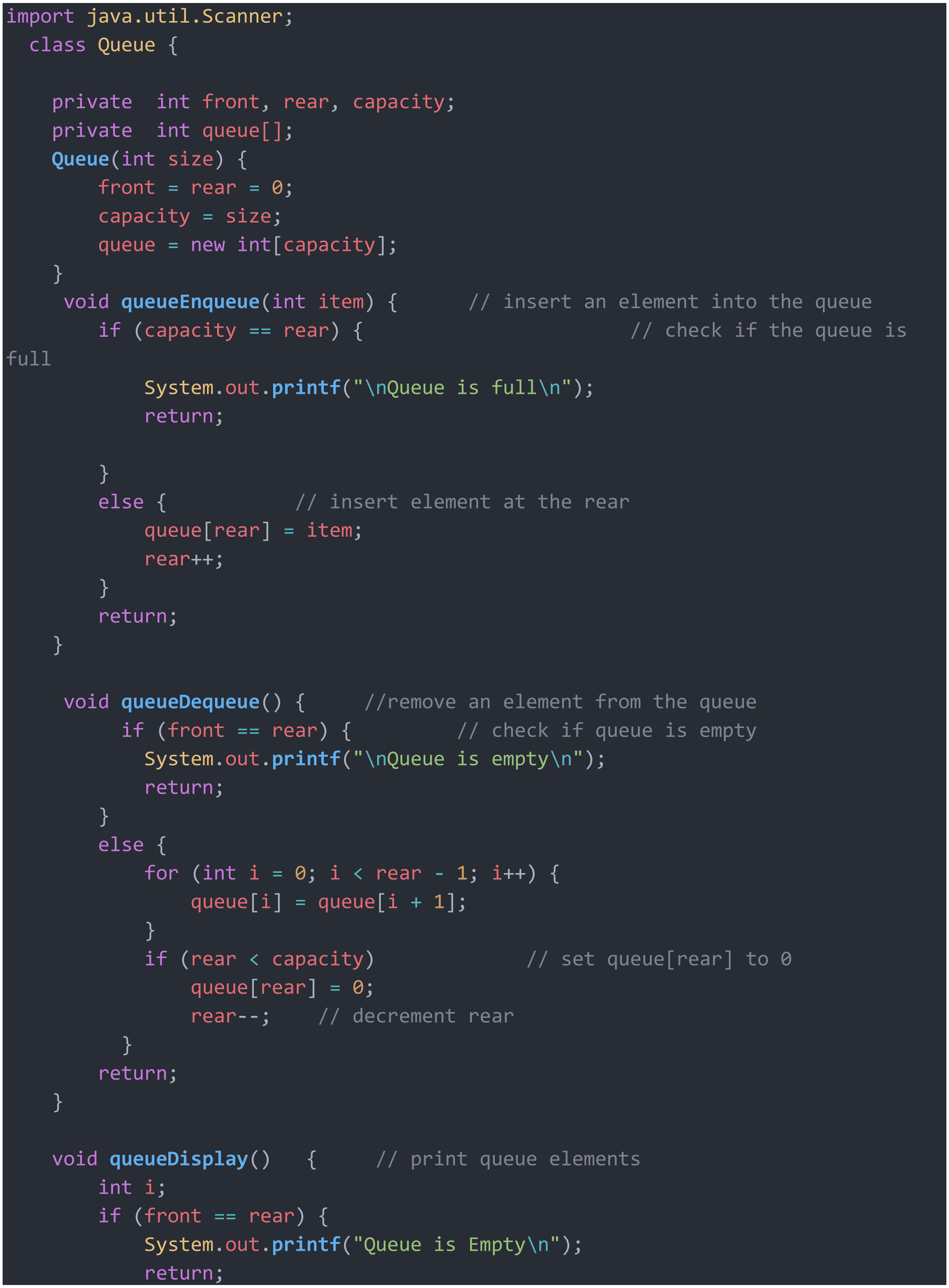
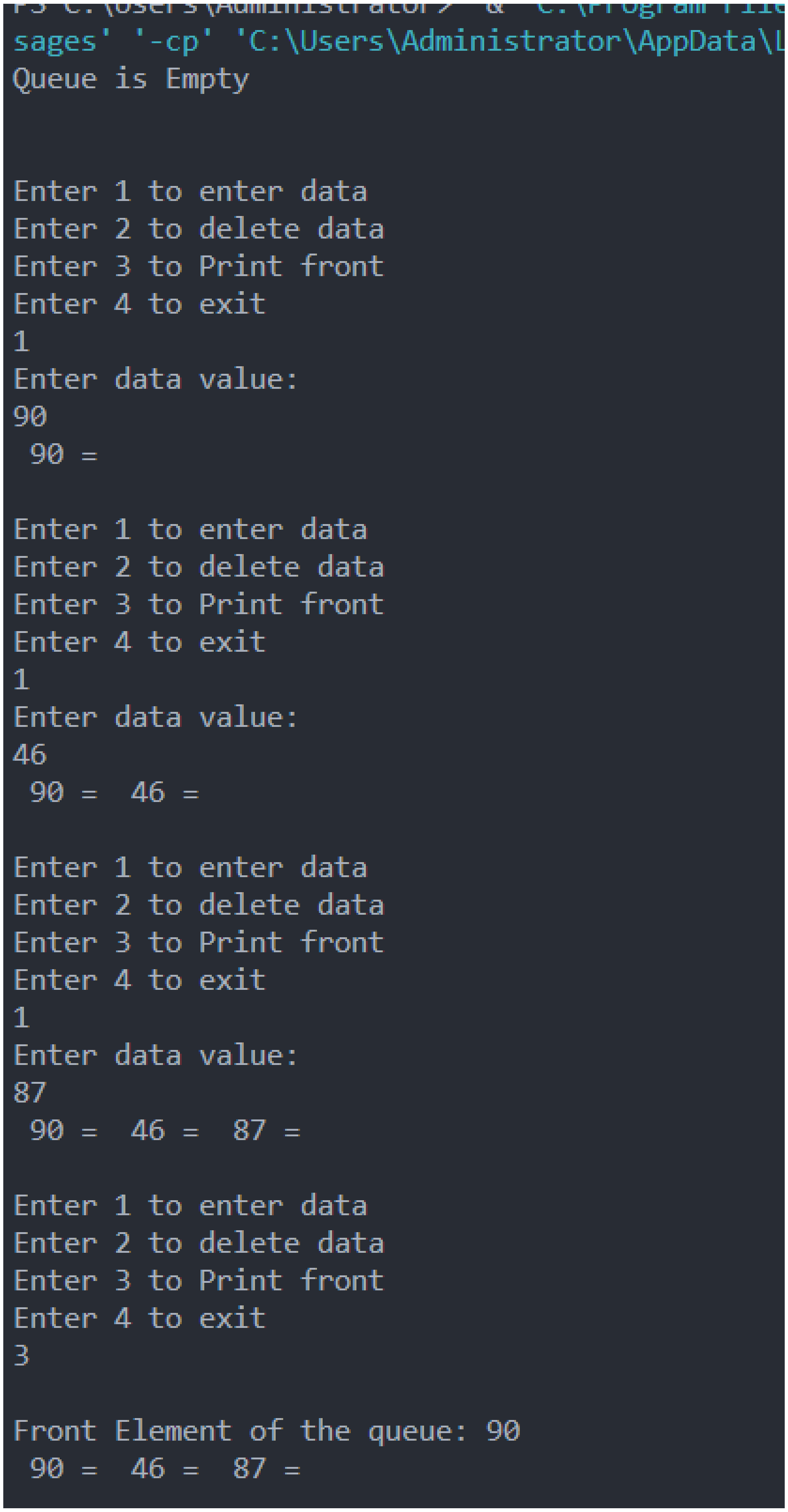
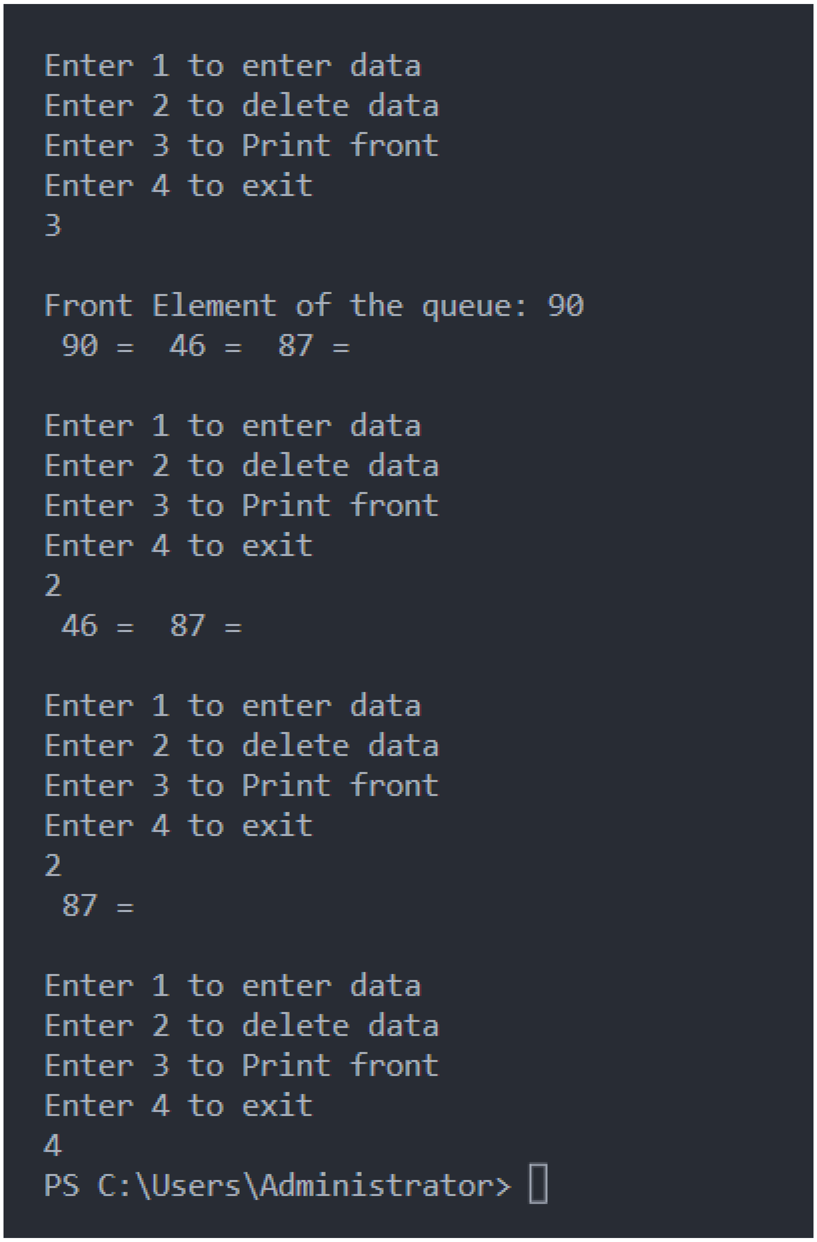


```
import java.util.Scanner;
  class Queue {

    private  int front, rear, capacity;
    private  int queue[];
    Queue(int size) {
        front = rear = 0;
        capacity = size;
        queue = new int[capacity];
    }
     void queueEnqueue(int item) {      // insert an element into the queue
        if (capacity == rear) {                       // check if the queue is
full
            System.out.printf("\nQueue is full\n");
            return;

        }
        else {           // insert element at the rear
            queue[rear] = item;
            rear++;
        }
        return;
    }

     void queueDequeue() {     //remove an element from the queue
          if (front == rear) {          // check if queue is empty
            System.out.printf("\nQueue is empty\n");
            return;
        }
        else {
            for (int i = 0; i < rear - 1; i++) {
                queue[i] = queue[i + 1];
            }
            if (rear < capacity)             // set queue[rear] to 0
                queue[rear] = 0;
                rear--;    // decrement rear
          }
        return;
    }

    void queueDisplay()   {     // print queue elements
        int i;
        if (front == rear) {
            System.out.printf("Queue is Empty\n");
            return;
        }
        for (i = front; i < rear; i++) {           // traverse front to rear and print elements
            System.out.printf(" %d = ", queue[i]);
        }
        return;

    }



  void queueFront()    {     // print front of queue

           if (front == rear) {
            System.out.printf("Queue is Empty\n");
            return;
        }
        System.out.printf("\nFront Element of the queue: %d%n", queue[front]);
        return;

    }

}



public class Main {

    public static void main(String[] args) {

        Queue queue = new Queue(4);
        Scanner scanner = new Scanner(System.in);
        int userChoice;
        do{
            queue.queueDisplay();
            System.out.println("\n\nEnter 1 to enter data\nEnter 2 to delete data\nEnter 3 to Print front\nEnter 4 to exit");
            userChoice = scanner.nextInt();
            switch(userChoice){
                case 1:
                    System.out.println("Enter data value: ");
                    int dataVal = scanner.nextInt();
                    queue.queueEnqueue(dataVal);
                    break;
                case 2:
                    queue.queueDequeue();
                    break;
                case 3:
                    queue.queueFront();

            }
        }while(userChoice != 4);

    }

}
```



```
PS C:\Users\Administrator> & 'C:\Program File
sages' '-cp' 'C:\Users\Administrator\AppData\L
Queue is Empty


Enter 1 to enter data
Enter 2 to delete data
Enter 3 to Print front
Enter 4 to exit
1
Enter data value:
90
 90 =

Enter 1 to enter data
Enter 2 to delete data
Enter 3 to Print front
Enter 4 to exit
1
Enter data value:
46
 90 =  46 =

Enter 1 to enter data
Enter 2 to delete data
Enter 3 to Print front
Enter 4 to exit
1
Enter data value:
87
 90 =  46 =  87 =

Enter 1 to enter data
Enter 2 to delete data
Enter 3 to Print front
Enter 4 to exit
3

Front Element of the queue: 90
 90 =  46 =  87 =
```



```
Enter 1 to enter data
Enter 2 to delete data
Enter 3 to Print front
Enter 4 to exit
3

Front Element of the queue: 90
 90 =  46 =  87 =

Enter 1 to enter data
Enter 2 to delete data
Enter 3 to Print front
Enter 4 to exit
2
 46 =  87 =

Enter 1 to enter data
Enter 2 to delete data
Enter 3 to Print front
Enter 4 to exit
2
 87 =

Enter 1 to enter data
Enter 2 to delete data
Enter 3 to Print front
Enter 4 to exit
4
PS C:\Users\Administrator>
```