Java Week 8: Q5

Due on 2020-11-12, 23:59 IST

Write a program to ***display any digit(n) from 0-9 represented as a "7 segment display"***.

For example:

input : 5

output :

```
 _
|_
 _|
```

input : 4

output :

```
|_|
  |
```

Private Test cases used for evaluation	Input	Expected Output	Actual Output	Status
Test Case 1	3	_ \n _|\n _|\n	_ \n _|\n _|\n	Passed

The due date for submitting this assignment has passed.
1 out of 1 tests passed.
You scored 100.0/100.

Assignment submitted on 2020-11-06, 00:30 IST

Your last recorded submission was :

```
import java.util.*;

public class Pattern5 {

    private static final Map<Integer, Integer> encodings =
        new HashMap<Integer, Integer>();

    static {
        encodings.put(0, 0x7E);
        encodings.put(1, 0x30);
        encodings.put(2, 0x6D);
        encodings.put(3, 0x79);
        encodings.put(4, 0x33);
        encodings.put(5, 0x5B);
        encodings.put(6, 0x5F);
        encodings.put(7, 0x70);
        encodings.put(8, 0x7F);
        encodings.put(9, 0x7B);
    }

    public static void printDigit(int digit) {
        int code = encode(digit);
        char[] bits =
            String.format("%7s", Integer.toBinaryString(code))
                .replace(' ', '0').toCharArray();

        lightSegment(bits[0] == '1', "  _ \n", "    \n");
        lightSegment(bits[5] == '1', "|", " ");
        lightSegment(bits[6] == '1', "_", " ");
        lightSegment(bits[1] == '1', "|\n", " \n");
        lightSegment(bits[4] == '1', "|", " ");
        lightSegment(bits[3] == '1', "_", " ");
        lightSegment(bits[2] == '1', "|\n", " \n");
    }

    private static void lightSegment(boolean on, String onValue, String offValue) {
        System.out.print(on ? onValue : offValue);
        try {
            Thread.sleep(0);
        }
        catch (InterruptedException e) {
            e.printStackTrace();
        }
    }

    private static int encode(int digit) {
        return encodings.containsKey(digit) ? encodings.get(digit) : 0x00;
    }
   public static void main(String[] args) throws Exception {
            Scanner inr = new Scanner(System.in);
       int n = inr.nextInt();
                printDigit(n);

    }
}
```

Sample solutions (Provided by instructor)

```
import java.util.*;
public class Pattern5 {
    private static final Map<Integer, Integer> encodings =
        new HashMap<Integer, Integer>();
    static {
        encodings.put(0, 0x7E);
        encodings.put(1, 0x30);
        encodings.put(2, 0x6D);
        encodings.put(3, 0x79);
        encodings.put(4, 0x33);
```

```
        encodings.put(5, 0x5B);
        encodings.put(6, 0x5F);
        encodings.put(7, 0x70);
        encodings.put(8, 0x7F);
        encodings.put(9, 0x7B);
    }
    public static void printDigit(int digit) {
        int code = encode(digit);
        char[] bits =
            String.format("%7s", Integer.toBinaryString(code))
                .replace(' ', '0').toCharArray();

        lightSegment(bits[0] == '1', "  \n", "   \n");
        lightSegment(bits[5] == '1', "|", " ");
        lightSegment(bits[6] == '1', "_", " ");
        lightSegment(bits[1] == '1', "_\n", " \n");
        lightSegment(bits[4] == '1', "|", " ");
        lightSegment(bits[3] == '1', "_", " ");
        lightSegment(bits[2] == '1', "_\n", " \n");
    }

    private static void lightSegment(boolean on, String onValue, String offValue) {
        System.out.print(on ? onValue : offValue);
        try {
            Thread.sleep(0);
        }
        catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
    private static int encode(int digit) {
       return encodings.containsKey(digit) ? encodings.get(digit) : 0x00;
    }
   public static void main(String[] args) throws Exception {
          Scanner inr = new Scanner(System.in);
         int n = inr.nextInt();
          printDigit(n);
    }
}

```