1
2
3
4
5

Look And Say - 40 points (Coding)

Let's define a function "Look and Say" as follow: read off the digits of the input, counting the number of digits in groups of same digit. Here are some examples of this function

- `LookAndSay(1) = 11` because 1 is read off as "one 1" or 11.
- `LookAndSay(11) = 21` because 11 is read off as "two 1s" or 21.
- `LookAndSay(21) =  1211` because 21 is read off as "one 2, then one 1" or 1211.
- `LookAndSay(1211) =  111221` because 1211 is read off as "one 1, then one 2, then two 1s" or 111221.
- `LookAndSay(111221) =  312211` because 111221 is read off as "three 1s, then two 2s, then one 1" or 312211.

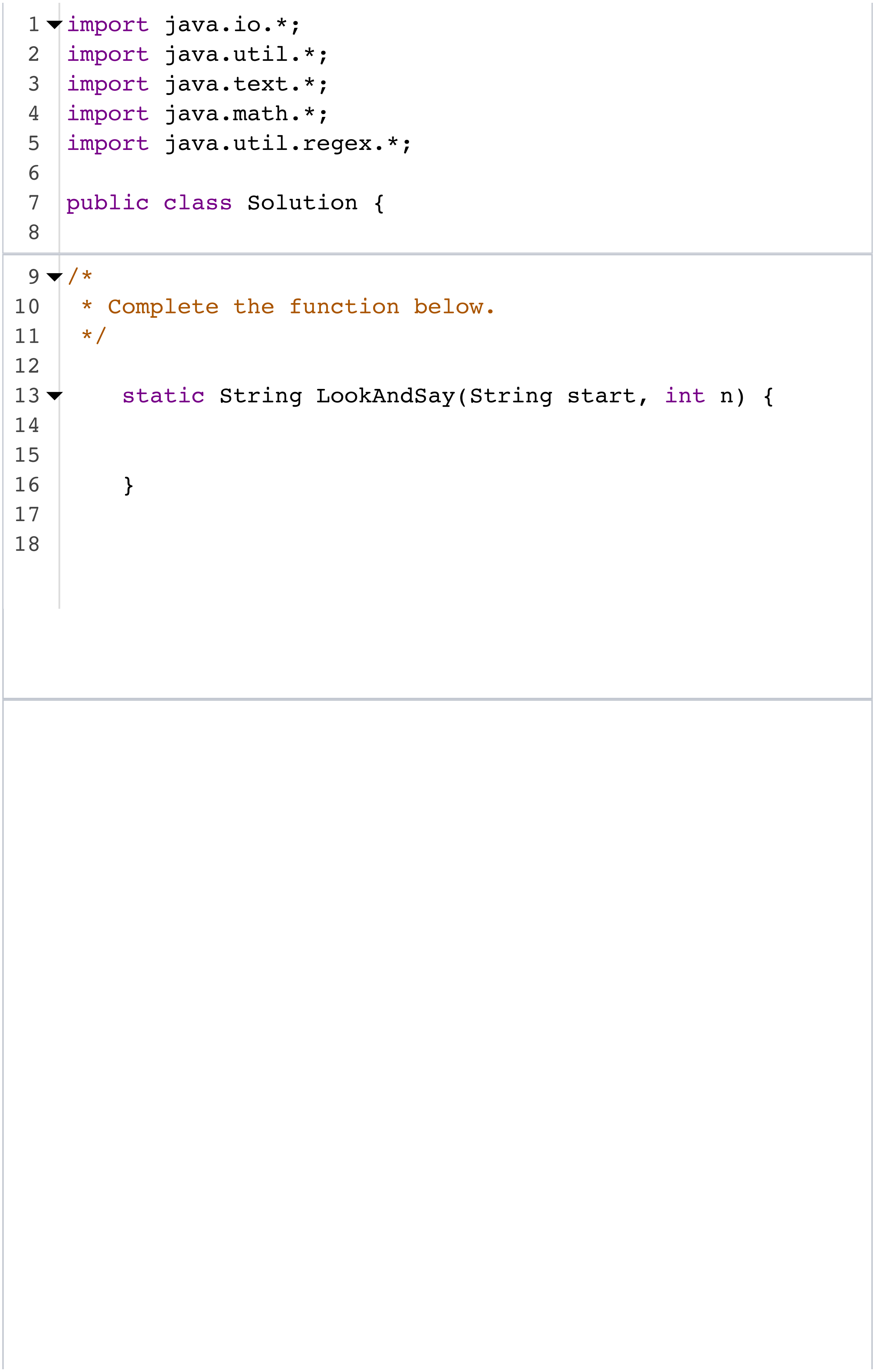
We then define a "Look and Say" sequence as repeatedly called the "Look and Say" function on its output.
Given a number `start` and a number of iteration `n`, **calculate** the `n`th number in a "Look and Say" sequence starting with `start`.

Reusing the previous example with start = 11 and n = 2, `LookAndSay(11, 2) = 1211` because `LookAndSay(LookAndSay(11)) = 1211`

Since this sequence grows quickly we will use a string representation for the number in the sequence to avoid number overflow.

YOUR ANSWER

Java



```java
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

/*
 * Complete the function below.
 */

    static String LookAndSay(String start, int n) {


    }


    public static void main(String[] args) throws IOException{
        Scanner in = new Scanner(System.in);
        final String fileName = System.getenv("OUTPUT_PATH");
        BufferedWriter bw = new BufferedWriter(new FileWriter(fileName));
        String res;
        String _start;
        try {
            _start = in.nextLine();
        } catch (Exception e) {
            _start = null;
        }

        int _n;
        _n = Integer.parseInt(in.nextLine());

        res = LookAndSay(_start, _n);
        bw.write(res);
        bw.newLine();

        bw.close();
    }
}
```

Line: 12 Col: 1

□ **Test against custom input** **Run Code** **Submit code & Continue**

Download sample testcases *The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*