Full Name:	Qifeng Zhou
Email:	zhou.qif@husky.neu.edu
Test Name:	**INFO6205_Final_Exam_Section_5**
Taken On:	23 Apr 2019 13:49:40 EDT
Time Taken:	22 min 3 sec/ 60 min
Work Experience:	< 1 years
Invited by:	Robin
Invited on:	23 Apr 2019 13:30:26 EDT
Tags Score:	

100%

50/50

scored in **INFO6205_Final_Exam_Section_5** in 22 min 3 sec on 23 Apr 2019 13:49:40 EDT

Recruiter/Team Comments:

No Comments.

	Question Description	Time Taken	Score	Status
Q1	Count the runs > **Coding**	20 min 53 sec	50/ 50	✓

QUESTION 1

Correct Answer

Score 50

Count the runs > Coding

QUESTION DESCRIPTION

This problem involves counting the length of *runs* in a String. A run is a sequence of the same character. Its properties are its length and the character.

In this exercise, you are to derive the n^{th} term by counting the runs in the $(n-1)^{th}$ term. Sound like a recursion? It sure is. Your code will look something like this:

```
public static String numberAndOccurences (int n) {
   if (n>1) return countTheRuns(numberAndOccurences(n-1));
   else return "1";
}
private static String countTheRuns(String s) {
   // TODO
}
```

I suggest that, rather than deal with *Char*, you should just convert everything to a *String*. This might not be best for performance, but we're not concerned with that here.

Since you need to return two different pieces of information from your *getRun* method (or whatever you call it), you might also want to create a class for a run like this:

```
  private static class Run {
  private final int n;
  private final String x;

  public Run(int n, String x) {
    this.n = n;
    this.x = x;
  }

  @Override
  public String toString() {
    return n+x;
  }
}
```

So, it turns out that the sequence of terms starts like this:
1, 11, 21, 1211, 111221, 312211, 13112221, 1113213211, …

The first five terms are:

```
1.     1
2.     11
3.     21
4.     1211
5.     111221
```

How is above sequence generated?
n'th term in generated by reading (n-1)'th term.

```
The first term is "1"

Second term is "11", generated by reading first term as "One 1"
(There is one 1 in previous term)

Third term is "21", generated by reading second term as "Two 1"

Fourth term is "1211", generated by reading third term as "One 2 One 1"

Fifth term is "111221", generated by reading fourth term as "One 1 One 2
Two 1"

and so on
```

Example:

```
Input: n = 3
Output: 21

Input: n = 5
Output: 111221
```

CANDIDATE ANSWER

Language used: **Java 7**

```
1 class Result {
```

```java
    /*
     * Complete the 'numberAndOccurences' function below.
     *
     * The function is expected to return a STRING.
     */

    private static class Run {
        private final int n;
        private final String x;

        public Run(int n, String x) {
            this.n = n;
            this.x = x;
        }
        public Run(int n, char x) {
            this.n = n;
            this.x = x +"";
        }

        @Override
        public String toString() {
            return n+x;
        }
    }

    public static String numberAndOccurences (int n) {

        if (n>1) return countTheRuns(numberAndOccurences(n-1));

        else return "1";

    }

    private static String countTheRuns(String s) {

        int count = 0;
        StringBuilder sb = new StringBuilder();
        char start = s.charAt(0);
        int tmp = -1;
        for(int i=0;i<s.length();i++){
            if(start-s.charAt(i)==0) {
                count++;
            }
            else{
                start = s.charAt(i);
                sb.append(count+"");
                sb.append(s.charAt(i-1));
                count = 1;
            }
            tmp = i;
        }
        if(count!=0)sb.append(count+""+s.charAt(tmp));
        return sb.toString();

    }
}
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED

Testcase 0	Easy	Sample case	Success	8	0.1 sec	23.4 MB
Testcase 1	Easy	Sample case	Success	8	0.101 sec	23.2 MB
Testcase 2	Easy	Sample case	Success	8	0.0938 sec	23.4 MB
Testcase 3	Easy	Sample case	Success	8	0.101 sec	23.1 MB
Testcase 4	Easy	Sample case	Success	8	0.0926 sec	22.8 MB
Testcase 5	Medium	Sample case	Success	10	0.0898 sec	23 MB

No Comments