L0038

September 19, 2022

```
# Solution.py
     # Author: Jagadeesh Vasudevamurthy
     # Copyright: Jagadeesh Vasudevamurthy 2020
     class Solution:
      def __init__(self):
        pass
      #THIS IS LEETCODE
      def countAndSay(self,n:'int')->'string':
        ## NOTHING CAN BE CHANGED HERE
        return self._alg_integer(n)
      def countAndSayString(self,n:'string')->'string':
        #Nothing can be changed here
        return self._alg_string(n)
      ### YOU CAN HAVE ANY NUMBER OF FUNCTIONS AND CLASSES BELOW
      ## n is str here
      def _alg_string(self,n:'string')->'string':
       #print("WRITE CODE BELOW")
        s=""
        count=1
        for i in range(0,len(n)):
           if(i < len(n)-1):
               if(n[i]==n[i+1]):
                  count=count+1
               else:
                  s= s+"%s"%count+n[i]
                  count=1;
            else:
               s=s+"%s"%count+n[i];
        return s
```

```
## n is int here
def _alg_integer(self,n:'int')->'string':
    if(n==1):
        return "1"
    result_previous="1"
    result_new=""
    for i in range(2,n+1):
        result_new=self._alg_string(result_previous)
        result_previous=result_new
    return result_new
```

1 YOU CANNOT CHANGE ANYTHING BELOW

```
# LookAndSay.py
   # Author: Jagadeesh Vasudevamurthy
   # Copyright: Jagadeesh Vasudevamurthy 2020
   # NOTHING CAN BE CHANGED IN THIS FILE
   # All imports
   class LookAndSay():
    def __init__(self):
     self._testBench()
    def _look_and_say_integer(self,n:'int')->'string':
     s = Solution()
     return s.countAndSay(n) #implement in solution
    def _look_and_say_string(self,n:'string')->'string':
     s = Solution()
     return s.countAndSayString(n) #implement in solution
    #private function
    def _testBench(self):
     self. tests()
     self._testn()
```

```
print("ALL TESTS PASSED")
#Private sunction
def _tests(self):
 n = "123"
 e = "111213"
  s = self._look_and_say_string(n)
  assert(s == e);
 n = "9999999999"
 e = "109"
  s = self._look_and_say_string(n)
  assert(s == e);
 n = "9876543210"
  e = "19181716151413121110"
  s = self._look_and_say_string(n)
  assert(s == e);
#Private sunction
def _testn(self):
 n = 1
 e = "1"
  s = self._look_and_say_integer(n)
 assert(s == e);
 n = 2
  e = "11"
  s = self._look_and_say_integer(n)
  assert(s == e);
 n = 3
  e = "21"
  s = self._look_and_say_integer(n)
  assert(s == e);
 n = 4
  e = "1211"
 s = self._look_and_say_integer(n)
  assert(s == e);
 n = 5
 e = "111221"
  s = self._look_and_say_integer(n)
  assert(s == e);
```

```
for n in range (1,32):
    s = self._look_and_say_integer(n)
    print("n = ",n, " Length = ", len(s))
    print("n = ",n," ",s)
# YOU CANNOT CHANGE ANYTHING BELOW
def main():
 t = LookAndSay()
# start up
if (__name__ == '__main__'):
 main()
n = 1 Length = 1
n = 1 1
n = 2 Length = 2
n = 2 11
n = 3 Length = 2
     21
n = 3
n = 4 Length = 4
n = 4 	 1211
n = 5 Length = 6
n = 5 111221
n = 6 Length = 6
n = 6 312211
n = 7 Length = 8
n = 7 	 13112221
n = 8 Length = 10
n = 8 1113213211
n = 9 Length = 14
n = 9 \quad 31131211131221
n = 10 Length = 20
n = 10 \quad 13211311123113112211
n = 11 Length = 26
n = 11 11131221133112132113212221
n = 12 Length = 34
n = 12 3113112221232112111312211312113211
n = 13 Length = 46
n = 13 1321132132111213122112311311222113111221131221
n = 14 Length = 62
n = 15 Length = 78
n = 15
```

- n = 16 Length = 102
- n = 17 Length = 134
- n = 18 Length = 176
- n = 19 Length = 226
- n = 20 Length = 302
- n = 21 Length = 408
- n = 22 Length = 528
- n = 23 Length = 678
- n = 24 Length = 904

n = 25 Length = 1182

n = 26 Length = 1540

n = 26121122132112311321322112111312211312111322211213111213122112132113121113222112132113213221133112132123222112311311222113111231132231121113112221121321133112132122132113213221133112132123222112311311222113111231132231121113112221121321133112 131122212322211331222113112211

n = 27 Length = 2012

23211231131122211211131221131112311332211213211321223112111311222112132113212221 21121113122113111231133221121321132132211331222113321112131122211332113221122112 1122132112311321322112111312211312111322211213111213122112132113121113222112132113213221133112132123222112311311222113111231132231121113112221121321133112132112 211211131221131211132221232112111312111213322112131112131221121321131211132221121332212311322113212221

n = 28 Length = 2606

21121113122113121113222123211211131221132211131221121321131211132221123113122111331121321232221123113112221131112311322311211131122211213211331121321122112132

n = 29 Length = 3410

12112213211231132132211211131221131211132221121311121312211213211312111322211213211321322113311213212322211231131122211311123113223112111311222112132113311213212221121113122113121113222123211211131211121332211213111213122112132113121132221112221121113122113111231133221121321132122311211131122211213211321222113222122212113122122132113213221133112132123123112111311222112132113311213211221121332211211232221121321132132211331121321231231121113112221121321133112132112211213322112

n = 30 Length = 4462

n = 301213211321223112111311222112132113213221133122211311221122111312211312111322212312132123222112311311222113111231132231121113112221121321133112132112211213322112

n = 31 Length = 5808

31 n = 21133112132112312321123113112221121113122113111231133221121321132122311211131122 1332211322132113213221133112132123123112111311222112132113311213211221121332211213211331121321123123211231131122211211131221131112311332211213211321223112111311 3221123113221113122112132113213211121332212311322113212221

ALL TESTS PASSED