

Problem 5.3 See No Evil, Speak No Evil

(page 1 of 1)

Overview: Compute the n -th look-say-reverse number.

Description: Great news! The jRobot Corporation, widely celebrated for its self-vacuuming Zoomba robot, is on the verge of completing its prototype LISA (Lactose Intolerant Super Automaton) robot, a friendly, self-learning machine that is sure to win even the most skeptical of technophobes over with its human-like charms.

But the research team seems to have run into a slight snag: while the rate of LISA's emotional development is off the charts, she has formed an odd affinity for look-say numbers, and the rest of her mathematical processes have shut down. LISA is well aware of this problem and is dangerously close to self-destructing in desperation. To prevent this catastrophe from occurring, you have been hired to repair LISA's glitch.

Clearly, the best course of action is to reverse LISA's fondness for look-say numbers by having her develop an attachment for "look-say-reverse" numbers as well and having the attachments cancel each other out. To introduce these numbers to her, you have been asked to generate the n -th look-say-reverse number. Are you up to the task?

The first look-say-reverse numbers are 1, 11, 12, 2111, 1321, 11213111, and 1331112112. If we know a look-say-reverse number, we can get the next one by splitting the number into consecutive runs of identical digits (so 11213111 becomes "11 2 1 3 111"), saying how many of each digit there are (e.g., "two 1, one 2, one 1, one 3, three 1"), writing this down as a new number ("2112111331"), and reversing it (to get 1331112112).

Time Allocation: 1 second

Input: The input consists of a single integer n .

Output: The output should consist of a string s on one line, representing the n -th look-say-reverse number.

The output is to be formatted exactly like the sample output given below.

Assumptions: n will be an integer between 1 and 49, inclusive.
 s will contain no more than 1,000,000 digits.
All input will be valid.

Sample Input #1: 4

Sample Output #1: 2111

Sample Input #2: 7

Sample Output #2: 1331112112