hackerrank.com/challenges/separate-the-numbers/problem?isFullScreen=true

HackerRank Prepare > Algorithms > Strings > Separate the Numbers

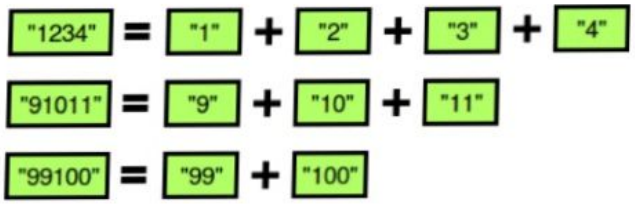
Exit Full Screen View

Problem | Submissions | Leaderboard | Discussions

A numeric string, s, is beautiful if it can be split into a sequence of two or more positive integers, $a[1], a[2], \ldots, a[n]$, satisfying the following conditions:

1. $a[i] - a[i-1] = 1$ for any $1 < i \leq n$ (i.e., each element in the sequence is 1 more than the previous element).
2. No $a[i]$ contains a leading zero. For example, we can split $s = 10203$ into the sequence $\{1, 02, 03\}$, but it is not beautiful because 02 and 03 have leading zeroes.
3. The contents of the sequence cannot be rearranged. For example, we can split $s = 312$ into the sequence $\{3, 1, 2\}$, but it is not beautiful because it breaks our first constraint (i.e., $1 - 3 \neq 1$).

The diagram below depicts some beautiful strings:



Perform q queries where each query consists of some integer string s. For each query, print whether or not the string is beautiful on a new line. If it is beautiful, print `YES x`, where x is the first number of the increasing sequence. If there are multiple such values of x, choose the smallest. Otherwise, print `NO`.

Function Description

```
        int q = sc.nextInt();
        for (int tc = 0; tc < q; tc++) {
            String s = sc.next();

            long result = solve(s);
            System.out.println(result > 0 ? "YES " + result : "NO");
        }

        sc.close();
    }

    static long solve(String s) {
        if (s.charAt(0) == '0') {
            return -1;
        }

        for (int length = 1; length * 2 <= s.length(); length++) {
            long firstNumber = Long.parseLong(s.substring(0, length));

            StringBuilder sequence = new StringBuilder();
```

Line: 39 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

30°C Mostly cloudy | Search | ENG IN | 11:51 11-07-2023

hackerrank.com/challenges/utopian-tree/problem?isFullScreen=true

HackerRank Prepare > Algorithms > Implementation > Utopian Tree

Exit Full Screen View

Problem | Submissions | Leaderboard | Discussions

The Utopian Tree goes through 2 cycles of growth every year. Each spring, it doubles in height. Each summer, its height increases by 1 meter.

A Utopian Tree sapling with a height of 1 meter is planted at the onset of spring. How tall will the tree be after n growth cycles?

For example, if the number of growth cycles is $n = 5$, the calculations are as follows:

Period	Height
0	1
1	2
2	3
3	6
4	7
5	14

Function Description

Complete the utopianTree function in the editor below.

utopianTree has the following parameter(s):

- int n: the number of growth cycles to simulate

Returns

- int: the height of the tree after the given number of cycles

Input Format

The first line contains an integer, t, the number of test cases.

t subsequent lines each contain an integer, n, the number of cycles for that test case.

Constraints

```
16              case MONSOON:
17                  treeHeight = treeHeight * 2;
18                  cycleType = SUMMER;
19                  break;
20              case SUMMER:
21                  treeHeight += 1;
22                  cycleType = MONSOON;
23                  break;
24              }
25          }
26          System.out.println(treeHeight);
```

Line: 30 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0

Test case 1

Test case 2

Compiler Message

Success

Input (stdin) Download

30°C Mostly cloudy | Search | ENG IN | 11:47 11-07-2023