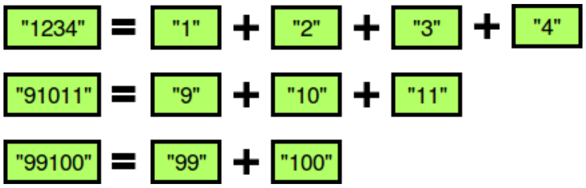
Separate the Numbers

by DmitriyH

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 $1 < i \le n$ (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., $1 - 3 \ne 1$).

The diagram below depicts some beautiful strings:



You must perform q queries, where each query consists of some string s. For each query, print whether or not the string is beautiful on a new line. If it's 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` instead.

Input Format

The first line contains an integer denoting q (the number of strings to evaluate).
Each of the q subsequent lines contains some string s for a query.

Constraints

- $1 \le q \le 10$
- $1 \le |s| \le 32$
- Each character in s is a decimal digit from 0 to 9 (inclusive).

Output Format

For each query, print its answer on a new line (i.e., either `YES x` where x is the smallest first number of the increasing sequence, or `NO`).

Sample Input 0

```
7
1234
91011
99100
101103
010203
13
1
```

Sample Output 0

```
YES 1
YES 9
YES 99
NO
NO
NO
NO
```

Explanation 0

The first three numbers are beautiful (see the diagram above). The remaining numbers are not beautiful:

- For $s = 101103$, all possible splits violate the first and/or second conditions.
- For $s = 010203$, it starts with a zero so all possible splits violate the second condition.
- For $s = 13$, the only possible split is $\{1, 3\}$, which violates the first condition.
- For $s = 1$, there are no possible splits because s only has one digit.

Submissions: 186
Max Score: 20
Difficulty: Easy

Rate This Challenge:
★★★★★

More

Current Buffer (saved locally, editable)

C#

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Numerics;

class Solution {

    static BigInteger Proximo(string s, BigInteger anterior, int desde)
        {


            if (s[desde] == '0')
            {
                return -1;
            }
            string concat = "";
            for (int i = desde; i < s.Length; i++)
            {
                concat += s[i].ToString();

                if (BigInteger.Parse(concat) == anterior + 1)
                {
                    return BigInteger.Parse(concat);
                }
                else if (BigInteger.Parse(concat) > anterior + 1)
                {
                    return -1;
                }
            }
            return -1;
        }

    static void Main(String[] args) {
        int q = Convert.ToInt32(Console.ReadLine());
        for(int a0 = 0; a0 < q; a0++){
            string s = Console.ReadLine();
            // your code goes here
            if (s.Length == 1)
            {
                Console.WriteLine("NO");
            }
            else
            {

                BigInteger primero = BigInteger.Parse(s[0].ToString());

                BigInteger x = primero;

                int len_primero = 1;

                int desde = 1;


                string ans = "YES";


                while (len_primero < s.Length)
                {
                    ans = "YES";
                    while (desde < s.Length)
                    {
                        // Console.WriteLine(primero);
                        BigInteger prox = Proximo(s, primero, desde);

                        if (prox == -1)
                        {
                            ans = "NO";
                            break;
                        }

                        primero = prox;
                        desde += primero.ToString().Length;
                        if (desde == s.Length)
                        {
                            ans = "YES";
                            break;
                        }
                    }
                    if (ans == "YES")
                    {
                        break;
                    }

                    len_primero++;
                    primero = BigInteger.Parse(s.Substring(0, len_primero));
                    x = primero;
                    desde = len_primero;

                }

                if (ans == "YES")
                {
                    Console.WriteLine(ans + " " + x);
                }
                else
                {
                    Console.WriteLine("NO");
                }

            }


        }
    }
}

```

Line: 6 Col: 1

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

Congrats, you solved this challenge!

- ✓ Test Case #0
- ✓ Test Case #1
- ✓ Test Case #2
- ✓ Test Case #3
- ✓ Test Case #4
- ✓ Test Case #5
- ✓ Test Case #6
- ✓ Test Case #7
- ✓ Test Case #8
- ✓ Test Case #9
- ✓ Test Case #10
- ✓ Test Case #11
- ✓ Test Case #12
- ✓ Test Case #13
- ✓ Test Case #14
- ✓ Test Case #15
- ✓ Test Case #16
- ✓ Test Case #17
- ✓ Test Case #18
- ✓ Test Case #19
- ✓ Test Case #20

Next Challenge

Copyright © 2017 HackerRank. All Rights Reserved

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature