Separate the Numbers



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., $1-3 \neq 1$).

The diagram below depicts some beautiful strings:

You must 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'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.

Function Description

Complete the separateNumbers function in the editor below. It should print a string as described above.

separateNumbers has the following parameter:

• s: an integer value represented as a string

Input Format

The first line contains an integer q, the number of strings to evaluate. Each of the next q lines contains an integer string s to query.

Constraints

- $1 \le q \le 10$
- $1 \le |s| \le 32$
- $s[i] \in [0-9]$

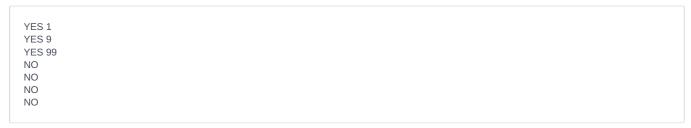
Output Format

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



Explanation 0

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

- ullet For ${\it s}=101103$, 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.
- ullet 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.

Sample Input 1

```
4
99910001001
7891011
9899100
999100010001
```

Sample Output 1

```
YES 999
YES 7
YES 98
NO
```