Project Euler #79: Passcode derivation

This problem is a programming version of Problem 79 from projecteuler.net

A common security method used for online banking is to ask the user for three random characters from a passcode. For example, if the passcode was 531278, they may ask for the 2^{nd}, 3^{rd}, and 5^{th} characters; the expected reply would be: 317.

Given that the three characters are always asked for in order, analyse the file so as to determine the shortest possible secret passcode of unknown length.

Assume all characters in your password are different. You're given T successful login attempts each containing 3 characters with *ASCII* codes from 33 to 126 both inclusive. You need to recover the shortest original password possible. If there are multiple choices, select the lexicographically smallest one.

If something went wrong and the original password cannot be recovered,
output `SMTH WRONG`.

Constraints
$1 \leq T \leq 3000$

Sample Input 1

```
5
SMH
TON
RNG
WRO
THG
```

Sample Output 1

```
SMTHWRONG
```

Sample Input 2

```
3
an0
n/.
.#a
```

Sample Output 2

```
SMTH WRONG
```

Explanation

As you can see, there is a unique shortest password in the first test case, namely, "SMTHWRONG" (why not?).
Let's look at the second case. We see that 'a' comes before 'n' (first attempt), then 'n' comes before '.' (second attempt) and '.' comes before 'a' (third attempt). But 'a' cannot come before 'a' as all characters

must be different.