DMOPC '14 Contest 4 P4 - Sketchley Park

Time limit: 0.9s **Memory limit:** 128M

At Sketchley Park, you're part of the effort to break enemy communications encrypted with the Paradox machine. Breaking the encryption would shorten the war by 5 years and save millions of lives.

Thankfully, Intelligence has managed to obtain a functional Paradox machine for you to reverse engineer. You've discovered that the machine operates ***only* with uppercase alphabetical characters**, encrypting and decrypting based off a letter map cipher. Every letter is mapped to a different letter in the alphabet, i.e. not necessarily shifted by a constant value.

Through arduous manual labour, you've isolated N $(1 \le N \le 500\,000)$ possible letter mappings, and you'd like to find which one correctly decrypts an enemy message. You also know that the enemy is very zealous, such that many decrypted messages contain the substring `HAILHYDRA` in them, **not necessarily at the end**.

To illustrate, with the letter map `HALOVJEGBRMCZFSUYQKIDNTXWP`,

- `HELLOWORLDANDHAILHYDRA` → `GVCCSTSQCOHFOGHBCGWOQH`
- `NICEWEATHERHAILHYDRA` → `FBLVTVHIGVQGHBCGWOQH`
- `HAILHYDRAHAIL` → `GHBCGWOQHGHBC`

Given an encrypted message and a set of N possible keys, identify the correct letter map and print the decrypted message. You can assume a letter map is **correct** if the decrypted message contains `HAILHYDRA`. It is also possible, however, that the message itself contains no `HAILHYDRA`, in which case we're out of luck. If no letter map is found, print `KeyNotFoundError`.

Input Specification

The first line will contain the encrypted message (no longer than $100\,000$ characters long). The second line will contain the single integer N, and the following N lines will each contain a possible decryption key (each 26 characters long).

It is guaranteed that the input is randomly generated. If there is a valid key, the string `HAILHYDRA` (after encryption) will be inserted into the random string at a random position.

Output Specification

The decrypted message, on one line. If multiple input keys are valid, use the first key in the order of input.

Sample Input 1

```
GVCCSTSQCOHFOGHBCGWOQH
5
ZTKJFCMLBGVPORSIEUQYHDANXW
QURVLGJKWAYHBONPSXFETMIDZC
IANUJTHMVXBLCDGQZKFYOWPSER
UWBPCEKGFZLMJDYSOAQHRIXVNT
HALOVJEGBRMCZFSUYQKIDNTXWP
```

Sample Output 1

```
HELLOWORLDANDHAILHYDRA
```

Explanation for Sample Output 1

`GVCCSTSQCOHFOGHBCGWOQH` decrypted with the 5th key in the input, `HALOVJEGBRMCZFSUYQKIDNTXWP`, results in `HELLOWORLDANDHAILHYDRA`. Since the substring `HAILHYDRA` is recognizable, we can assume we have found the correct key.

Sample Input 2

```
GVCCSTSQCOHFOGHBCGWOQH
5
ZTKJFCMLBGVPORSIEUQYHDANXW
IANUJTHMVXBLCDGQZKFYOWPSER
VMOAZUIFNELRCPJGKSBTDHWXYQ
UWBPCEKGFZLMJDYSOAQHRIXVNT
HNXKGSLBEZCYMTUDVPIOFWAJQR
```

Sample Output 2

```
KeyNotFoundError
```

Explanation for Sample Output 2

None of the 5 keys produce a decrypted string containing `HAILHYDRA`, so the correct key is not present.

Everything appearing in this problem is fictitious. Any resemblance to real locations, countries, organizations (?), or machines is *not* coincidental.