1_打倒怪獸 - Defeat the monster

(10分)

問題敘述

為了擊敗想要毀滅世界的怪獸 - 「WA」，身為大魔導師「AC」的徒弟的你正在幫忙他尋找可以擊敗怪獸的咒語，幸運的是，AC 對這個咒語依稀有點印象，所以他希望你從魔導書裡面找出所有和他記憶裡的咒語相似的連續片段，相似指的是兩個字串裡最多只有一個字元不同，也就是對兩個字串 s_1, s_2 ，最多只有一個 i 滿足 $s_1[i] \neq s_2[i]$

輸入格式

輸入有兩行：
第一行輸入是大魔導師記憶裡的咒語，長度為 N。
第二行輸入是魔導書的內容，長度為 M。

輸出格式

請依照字典序輸出魔法書裡所有和咒語相似的連續片段，每個相似的連續片段之間以換行做分隔，如果沒有相似的片段則輸出「Not found」（不含引號）

資料範圍

- $1 \leq NM \leq 10^6$
- $1 \leq N \leq M \leq 10^6$
- 所有的輸入都是小寫英文字母

輸入範例1

```
key
kaykeykay
```

輸出範例1

```
kay
kay
key
```

輸入範例2

```
ytp
yyppabcdefgydp
```

輸出範例2

```
ydp
ypp
yyp
```

輸入範例3

```
acac
watlerecemle
```

輸出範例3

```
Not found
```

範例說明

在範例一中，"kay" 和咒語 "key" 只有第二個字元不同、"key" 和 "key" 完全相同， "kay" 和 "key" 也只有第二個字元不同，所以結果就是這三個字串按字典序排列後輸出。

在範例二中，"yyp" 和 "ytp" 只有第二個字元、 "ypp" 跟 "ydp" 也是只有第二個字元不同，所以把他們三個排序後輸出即可。

在範例三中，"watlerecemle" 中並不存在與 "acac" 相似的字串。

1_Defeat the monster

(10 points)

Description

To defeat the monster "WA", which wants to destroy the world. You, the student of the grand wizard "AC", are helping him to find the spell to defeat the monster. Luckily, "AC" still remembers the spell roughly, so he asks you to look for all contiguous fragment with the same length which is similar to the spell in his memory in a spellbook. "Similar" means that there is at most one character different in two strings. That is, for two string s_1, s_2, there is at most one i such that $s_1[i] \neq s_2[i]$.

Input Format

The input contains two lines.
The first line contains a string, which is the spell in AC's memory. The length of it is N.
The second line contains a string, which is the content of the spellbook. The length of it is M.

Output Format

Please print all contiguous fragment in the spellbook which is similar to the spell in AC's memory in lexicographical order. Two fragments are split by newline.
If there is no fragment similar to AC's memory, just print "Not found" (without double quotation mark).

Constraints

- $1 \leq NM \leq 10^6$
- $1 \leq N \leq M \leq 10^6$
- all characters are lower case alphabet

Input Example 1

```
key
kaykeykay
```

Output Example 1

```
kay
kay
key
```

Input Example 2

```
ytp
yyppabcdefgydp
```

Output Example 2

```
ydp
ypp
yyp
```

Input Example 3

```
acac
watlerecemle
```

Output Example 3

```
Not found
```

Example Explanation

In example 1, There is only one difference between "kay" and "key".
"key" and "key" are the same.
There is only one difference between "kay" and "key".
So, just print them in lexicographical order.

In example 2, There is only one difference between "yyp" and "ytp", so "yyp" is similar to "ytp". Similarly, "ypp" and "ydp" are similar to "ytp", too.

In example 3, There is no substring in "watlerecemle" that is similar to "acac".