

## 761 Transform those strings

Given below is a set of rules that can be used to transform one string into another:

- $'t' \rightarrow 'i'$  (*change 't' to 'i'*)
- $\{'a' - 'z'\} \rightarrow 'm'$  (*change 'a', or 'b', or 'c', ... , or 'z' to 'm'*)
- $'s' \rightarrow 'i'$
- $'a' \rightarrow 'aa'$
- $\{'a' - 'z'\} \rightarrow ''$  (*delete 'a', or 'b', or 'c', ... , or 'z'*)
- $'t' \rightarrow 'r'$
- $\{'a' - 'g'\} \rightarrow 'h'$
- $'o' \rightarrow 'a'$
- $'t' \rightarrow 's'$
- $\{'t' - 'z'\} \rightarrow 'u'$

Sets are always given as a range, i.e.  $\{'a' - 'c'\}$  is the set with the characters  $\{'a', 'b', 'c'\}$

### Input and Output

Write a program that reads two strings, and prints the steps required to transform the first string into the second. The steps in the transformation must be printed in the form:

$$rule : origString \rightarrow newString$$

Note that several transformations may exist, your program only needs to find one. It is also possible that the first string cannot be transformed into the second. In this case, your program should simply output the word “no”

If there are more than one test cases, it must be a blank line between two consecutive, both in input and output files.

### Sample input

```
thinner
rih
```

```
a
x
```

### Sample output

```
6: thinner -> rhinner
5: rhinner -> rinner
7: rinner -> rinnhr
5: rinnhr -> rinhr
5: rinhr -> rihr
5: rihr -> rih
```

no