Problem E: Cousins

Two strings *a* and *b* are defined to be *first cousins* if they can be made equal by removing no more than half the characters from each. For example "abcdef" and "axcyd" are first cousins because we can remove 3 of the 6 characters (b,e,f) from the first string and 2 of the 5 characters in the second string (x,y) resulting in "acd". Two strings *c* and *d* are said to be n+1*st cousins* if there exists a string *e* that is a first cousin of *c* and is an n*th* cousin of *d*.

Given two strings *x* and *y*, determine the smallest $n \geq 1$ such that *x* is an n*th* cousin of *y*.

Input consists of several test cases. Each test case consists of two lines representing *x* and *y*. *x* and *y* each consist of at least 1 and at most 100 lower case letters. Two lines containing 0 follow the last test case. For each test case, output a line containing *n* or **not related** if *x* and *y* are not n*th* cousins for any *n*.

Sample Input

```
a
b
abb
baa
abcdef
axcyd
0
0
```

Output for Sample Input

```
2
2
1
```

Gordon V. Cormack