

Write a program that will compare two texts. The two texts are assumed to be different versions of the same text. Three different things can happen to a text between two versions:

- Text could be deleted
- Text could be inserted
- Text could be changed

Any combination of the three can occur at once. A move, for example will look as a deletion at one place and an insert at another. You should list all changes that happened from the old to the new version.

Input

The input file contains several datasets, each containing two lines with *text1* and *text2* strings. They contain the two versions of the text to compare. String *text2* should be regarded as the older version, and string *text1* as the newer.

Output

For each test case, the outputfile contain all deletions, insertions and changes that happened to the text between the two versions in the following format:

- deletion: Position of the first character deleted, the word 'deleted' and the number of characters deleted and the text that has been deleted.
- insertion: Position of the first character of the text inserted, the word 'inserted' since the last version, the number of characters inserted and the text inserted.
- change: Position of the first character changed, the word 'changed', the number of characters changed, the original text and what it has been changed to.

The position of the first character always refers to the position in the *older* version (i.e. the position within *text2*). The first character in a file is numbered zero (0). The program should **not** be case-sensitive.

See the sample below for details. Print a blank line after the output of each test case.

Note: The numbers for position characters and the headings of lines are not included in the input file.

Sample Input

```
0          1          2          3          4          5          6
0123456789012345678901234567890123456789012345678901234567890
text1: This is a joke. This is not life. Don't consider it anyway...
text2: This is not a joke. This is life. Consider it thoroughly...
```

Sample Output

```
pos 8 deleted 4 chars "not "
pos 28 inserted 4 chars "not "
pos 34 inserted 6 chars "don't "
pos 46 changed 10 chars from "thoroughly" to "anyway"
```