

Sorting Words (`sort.c`)

Implement a sorting algorithm of your liking (do not use some already implemented sorting algorithm from some library) to sort words alphabetically (case-insensitive), then print all words sorted alphabetically.

Details

Read an arbitrary number of words (use `scanf`). Each word will consist of at most 63 characters and consists of upper-case and lower-case letters (A-Z and a-z, no other special characters). The list of words is always terminated by a `."` (which is not a valid word and thus should not be stored).

After `."` was entered, print all words that were entered in alphabetical order. For this, it should not matter whether the word starts with an upper case or lower case letter.

The word must be printed in the same way it was entered. In particular it is not allowed to convert upper-case letters to lower case letters. If the same word was entered twice, it should also be printed twice.

You can use bubble sort to sort the elements or you can read up on other sorting algorithms for practice. In particular, selection sort is easy to implement. Quicksort is more complex but it is one of the fastest sorting algorithms.

Make sure that you use a stable sorting algorithm (which means that elements that are equal do not change the position relative to each other. Selection Sort and Bubble Sort are stable).

Example

Input

```
dog
Austria
Apple
apple
.
```

Output

```
Apple
apple
Austria
dog
```

Observe that like in the input, the upper case **Apple** is before the lower case **apple** because we use a stable sorting algorithm.

If you use a non-stable sorting algorithm there will be only minor point deductions though.

Hints and Remarks

The input part is almost identical to the one of the previous exercise (even simpler because now it is not necessary to filter out duplicates), hence save time by reusing it.

String comparison

`string.h` provides some useful procedures, in particular `strcmp`. Unfortunately, it is case-sensitive, yet in our case the comparison should be case insensitive. The procedures `stricmp` (Windows) or `strcasemp` (Unix) are non-standard procedures and hence cannot be used. Therefore, you should implement your own comparison procedure. For this purpose read the documentation on `strcmp`. It might be helpful to implement a procedure that will compare single characters in a case-insensitive way and use it in your `strcmp`-procedure.