

Analytics Infrastructure Exercise - Sorting Strings

Task

Implement a function **sortStrings()** that sorts a list of strings in a human-friendly way.

Input

Your function should take exactly one argument: **strings**, which is a list (array) of strings. Each of the strings contains only the following characters:

- Digits: 0 - 9
- Letters: a - z, A - Z
- Special characters: space (" "), dot ((".")), dash/negative sign ("-"), and slash ("/")

You do not need to verify the validity of the input.

Sorting Requirements

There are three basic types of strings you need to handle:

- Numbers: you only need to consider integers ("-2") and decimal numbers ("2.5"). Sort them by value.
- Dates: formatted as YYYY-mm-dd or YYYY/mm/dd. Sort them by the date.
- Alphabetic strings: Sort them alphabetically while ignoring case.

Here are some sorted lists containing strings of one of the basic types:

- ["-2.4", "-1", ".2", "2", "10"]
- ["2016/10/10", "2016-10-12", "2017-01-01"]
- ["Apple", "bacon", "Watermelon"]

Of course, a string might be a concatenation of multiple basic types, *optionally* separated by the special characters, for example 'apple2' is a valid input containing both digits and alphabetic characters. When sorting such compound strings, resolve the types in the order they appear.

Examples of sorted lists:

- ["abc45", "abc123", "def45"]
- ["Ended on 2016-01-02", "ended on 2017-01-05", "ended ON 2017-02-05", "started on 2016-01-02"]

When strings of different basic types appear, you must separate the basic types. The order between the types is up to you: both ["a1", "a3", "a2016-01-01"] and ["a2016-01-01", "a1", "a3"] are accepted, but ["a1", "a2016-01-01", "a3"] is not.

Instructions

We've provided you with a link to a Coderpad in the language of your choice (Python, JavaScript, Ruby, or PHP). There should be a function called **sortStrings()**.

- Complete the implementation of **sortStrings()** as described above.
- While you're free to research around the subject on the Internet, all the code you write must be your own.
- If there is an edge case not explicitly specified above, use your best judgment.
- If there is an edge case you didn't get to solve, note them at the top of the file as comments so we know you're aware of them.
- Once you complete the assignment, respond to this email to let us know. Please do not make more edits to this file once you've submitted.
- We will evaluate your assignment based on its correctness and readability.

Hints

- A dash ("-") can be a negative sign, part of a date string, or a separator. Similarly, a slash ("/") can be part of a date string or a separator. Can you find a relatively simple algorithm that will cover all these cases?
- We expect you to spend roughly 2 - 3 hours on this assignment. As such, we do not expect you to solve all possible corner cases. Instead, you should focus on the easier inputs before working on the more complicated ones.

Good luck!