Description :

* The numbers in the string are sorted according to their natural order (5>10).
* The alphabets in the string are sorted according to their ASCII values.
* A non alphanumeric character is considered smaller than a alpha-numeric character and sorting within the non alphanumeric characters is based on their ASCII values.
* The comparision between the strings is considered case insensitive.
* Time complexity : O(min(string1.length(),string2.length())).
* Space compexity : O(1).

Corner cases:

* There are many cases where the ordering done by the end users is different from the actual ordering done in the computers.
* There are also many cases where ordering done by some end users differ from ordering done by other end users.
* The above mentioned differences are due to certain ambiguities like

1. The file extensions should be considered or not if the file extensions are different.
2. Case sensitivity, whitespaces, trailing zeroes, brackets, special characters should be considered or not.

Possible modifications:

* If provided more time, I will collect the results of ordering done by different end users and analyze them and try to produce a sorting which is more intuitive to maximum number of end users.
* If provided more time, I will try to produce a sorting which can take decisions in certain cases to ignore articles like a, an, the etc.
* Though the above mentioned features may not be suitable for all type of applications, they can be included in certain special type of special applications.