Function: Strings and Numbers

Your task in this problem is to write four functions in Python to categorize a list of data elements into three groups: elements with all alphabets, elements with only digits (can be an integer, float, or a string representing integer or float), and elements with mixed alphabets and digits or any special characters.

Hint:

You can use a Python built-in function -: isinstance(a, datatype), i.e., isinstance(12, int) to check of 12 is an integer.

```
def contain_only_alphabets(s):
    # Check if an input string s contains only alphabet and space(s) if yes
return True otherwise return False



def contain_only_digits(s):
    # Check if an input string s contains only digits which could be a
string like '12', '12,0','100','12000.00' for example, which will return
True otherwise return False.


def sort_numbers_inplace(numbers):
    # Get a list of numbers which be integer, float, or a string
representing integer or float. To sort, numbers with integer and float data
types need to be sorted ascendingly and then numbers with string data types
will also be sorted ascendingly and followed numeric values.
    # *** This function will sort the numbers in place. It means the input
numbers will be modified directly… so…

def categorize(L):
    # Get a list L containing data elements with can have different data
types including string, integer, and float. This function must return three
lists:
    # first list contains only data elements from list L with alphabets only
    # second list contains only data elements from list L which are numbers
of integer, float or string data types and sorted by the sort_number_inplace
    # third list contains all remaining data elements from list L which do
not satisfy the first and second lists.


# Do not remove the following statement
exec(input().strip())
```

Example

Function call	Return value
`print(contain_only_alphabets('abced'))`	`True`
`print(contain_only_alphabets('AbCdE'))`	`True`
`print(contain_only_alphabets('Ab1CdE123'))`	`False`
`print(contain_only_digits('12'))`	`True`
`print(contain_only_digits('12.00'))`	`True`
`print(sort_numbers_inplace([1, '12', 0, '9']))`	`None`
`print(categorize(['Hello', 1, 2, '11', '111.1']))`	`(['Hello'], [1, 2, '11', '111.1'], [])`
`print(categorize(['K1', 'A1', 'Key', 18,16,14, '18','16','14']))`	`(['Key'], [14, 16, 18, '14', '16', '18'], ['K1', 'A1'])`
`print(categorize(['Hello', 1, '12', '@', '#', 0, '9', 'a', 'b']))`	`(['Hello', 'a', 'b'], [0, 1, '12', '9'], ['@', '#'])`
`print(categorize(['Hello World', 1, '12', '@', '#', 0, '9', 'a', 'b']))`	`(['Hello World', 'a', 'b'], [0, 1, '12', '9'], ['@', '#'])`

How to test sort_numbers_inplace() in your code: you can try the following code:

`D = [1, '12', 0, '9']`
`sort_numbers_inplace(D)`
`print(D)`
	`[0, 1, '12', '9']`