Computing Basic

B

School of Computer Science
Universidad de Oviedo

1 The file

There is a file `.ini` in Campus Virtual. It is a text file for setting environment variables.

The structure of this file is very simple. Each line starts in the first column and can be one of the following types: section lines, blank lines, assignment lines or comment lines. No other line is possible.

Section lines Section lines contain a section name in square brackets. The section name can contain just alphabetic characters.

Assignment lines Assignment lines are `key = value` pairs with optional spaces around the = sign. The `key` can contain only letters. The `value` can contain letters, digits and whitespaces.

Blank and Comment lines In addition, there can be empty lines (containing just `\t`, whitespaces and new lines) and lines starting with #, which are comments and can contain any characters.

There are no repeated names for sections or variables.
For instance, these are the lines of `inifileB.ini`:

```
# inifileB.ini

# comment
[alpha]

base  = moon
ship= alpha 3

[earth]
# ?
base=London
ship=  x-wing

[beta]
letter=b?
variable =
vari able = valor
```

2 Get the content of the file (1p)

Write a function `format_file(F)` that receives one parameters (string) corresponding to a filename. This function returns a list containing formatted lines in `F` in the way:

- Don't return blank lines
- Don't return comment lines
- Get rid of newlines at the end of the lines

Example: `format_file('inifileB.ini')` will generate a list (items are in different lines because of space limitations):

```
['[alpha]','base  = moon', 'ship= alpha 3',
'[earth]', 'base=London', 'ship=  x-wing'],
'[beta]', 'letter=b?','variable =','vari able = valor']
```

When formatted, the final list will contain just sections and assignments.

3 Check wrong lines (3p)

Write a function `find_errors(L)` that receives a list `L` generated by `format_file` and returns a list containing the numbers of lines containing errors. A line contains an error if there is a `=` and either the string before doesn't follow the rules for key name or the string after the `=` doesn't follow the rules for key value. You can count lines starting at 0.

Example: `print find_errors(format_file('inifileB.ini'))` would print `[5,7,8,9]`

4 Generate an error-free file (3p)

Write a function `error_free(F)` that receives a filename and generates a new file, with the original name plus `ErrorFree` and the same extension. This new file is a **formatted**, error-free copy of `F`.

Example: `errors_free('inifileB.ini')` will generate a file named `inifileBErrorFree.ini` with the following content:

```
[alpha]
base  = moon
ship= alpha 3
[earth]
base=London
[beta]
```

5 statistics (3p)

Write a function `statistics(F)` that receives an error-free file filename `F` and prints out some lines with key-value pairs separated by `=`. The pairs are:

- `total sections` plus the number of sections defined
- `section name` plus the number of variables defined in this section

Example: `statistics('inifileBErrorFree.ini')` will print:

```
alfa=2
earth=1
beta=0
total sections=3
```

You may find helpful to implement `getSectionNames` that returns a list containing section names and `getSectionLines` that returns a list containing the number of the lines where the section is written.

6 When finish...

Write docstrings in all the functions.
Identify the author in the first lines.
Save your python file named **test3B.py**.
Upload this only file to the corresponding task in Campus Virtual.