Problem 1 (35 minutes)

Write a python script that parses a CSV file. A CSV file is formed out of lines where each word (column) is separated by a comma (“,”) character. Example:

```
Name,Age,Weight
John,30,94
Maria,28,60
Carl,29,88
```

This will translate into a table that looks like this: →

Name	Age	Weight
John	30	94
Maria	28	60
Carl	29	88

The first line is always the header. The “,” (comma) character is used just to separate words (it can not be part of a word).
The script should receive two parameters:

csvparser.py <path_to_csv_file> <column_name>

And should output one of the following:

- “[ERROR] - unknown column name” → if the column name is not found in the CSV file (the test will be done with case insensitive → e.g. “AGE” = “age” = “Age” …
- “[ERROR] - invalid format - different words on line <number>” → if a line from the CSV file (that is not empty) has fewer or more words than what the header has. The reply should also contain the number of the line where the problem is.
- “[ERROR] - invalid format - duplicate columns” -> if there are multiple columns with the same name (case insensitive)
- “[ERROR] - file not found” → if the file sent as a parameter to the script does not exist.
- “[ERROR] - IO error” → if for some reason, reading that file is not possible !
- “[OK]”, followed by a list of all **UNIQUE** (***lowered cased***) words from the selected column sorted by their count (from the highest to the lowest). If two words have the same count, then they will be counted alphabetically (from A to Z, with ignore case).

Example:

```
Name,Age,Weight
John,30,94
Maria,28,60
CaRl,29,88
Marta,25,80
MARIA,29,65
Carl,35,90
Maria,30,87
maRta,37,80
```

Running
csvparser.py <file> Name
Should output →

```
[OK]
maria
carl
marta
john
```

In this example, Maria is found 3 times, Marta and Carl two times and John one time. Since Marta and Carl have the same number of occurrences, they will be ordered alphabetically (so Carl comes first as it starts with letter “C”).

OBS: *you are not allowed to use any external parsing library for CSV format (only Python string functions and RE library). If this happens your score will be null.*
OBS2: *Errors must be intercepted and dealt with accordingly (allowing an error to be catched by the python framework will not be considered valid).*

Grading:

5p	**Program checks for a valid number of command line parameters and prints an error otherwise**
8p	**Program opens the requested key-value file and prints an error in case of I/O issues.** *[4p for opening and reading from file, 4p interception of possible I/O errors]*
5p	**Program checks to see that there are no duplicate column names**
12p	**Program checks to see if the file format is OK** *4p - reads the header and make the column list* *8p - checks that each line has the same number of values as the ones indicated by the header (OBS: empty lines should be omitted)*
5p	**Program checks to see if the second command line parameter is a valid column name**
9p	**Program prints the result in the expected order (this also includes sorting the objects)** *[5p all items are unique and sorted base on their number of occurrences, 3p items with the same number of occurrences are sorted alphabetically, 1p → output to the screen]*
8p	**Program compiles and runs as expected.**

Scrieti un script in Python care sa parseze un fisier CSV (fisier format din linii cu cuvinte separate prin virgula). Exemplu:

```
Name,Age,Weight
John,30,94
Maria,28,60
Carl,29,88
```

Acest fisier ar putea fi vizualizat ca o tabela in felul urmator →

Name	Age	Weight
John	30	94
Maria	28	60
Carl	29	88

Prima linie din fișier este mereu hederul. Virgula nu poate face parte dintr-un cuvânt.
Scriptul primeste 2 parametri, astfel:

csvparser.py <path_to_csv_file> <column_name>

Scriptul va afisa pe ecran:

- "[ERROR] - unknown column name" → daca numele coloanei nu este gasit in hederul fisierului CSV (testarea se va face ignorand casing-ul) → e.g. "AGE" = "age" = "Age" …
- "[ERROR] - invalid format - different words on line <number>" → daca o linie din fisieul CSV (care nu este goala) are un numar diferit cuvinte fata de cele din header. Cand este afisata aceasta eroare, trebuie scris si numarul liniei unde a aparut eroarea
- "[ERROR] - invalid format - duplicate columns" -> daca sunt mai multe coloane cu acelasi nume (verificarea se face ignorand casing-ul).
- "[ERROR] - file not found" → daca fisierul CSV nu exista.
- "[ERROR] - IO error" → daca nu s-a putut citi din fisierul CSV
- "[OK]", urmata de o lista de cuvinte unice (scrise cu litera mica) selectate din coloana aleasa, si sortate dupa numarul lor de aparitii (de la cel mai mare numar la cel mai mic). Daca dupa cuvinte apar de acelasi numar de ori, atunci vor fi aranjate alfabetic (comparatia se va face ignorand casing-ul).

  Examplu:

```
Name,Age,Weight
John,30,94
Maria,28,60
CaRl,29,88
Marta,25,80
MARIA,29,65
Carl,35,90
Maria,30,87
maRta,37,80
```

Executia
csvparser.py <file> Name
Afiseaza pe ecran →

```
[OK]
maria
carl
marta
john
```

In acest exemplu, Maria apare de 3 ori, Marta si Carl de 2 ori si John o singura data. Cum Marta si Carl au același numar de aparii, ei vor fi sortati alfabetic (mai intai Carl, apoi Marta).

OBS: *nu aveți voie cu librării externe pentru parsarea formatului CSV. Aveți voie doar cu libraria*

RE sau utilizarea functiilor standard pentru stringuri. Utilizarea unor librării externe duce la anularea punctajului la aceasta probă (punctaj = 0)
OBS2: *Erorile trebuie interceptate si tratate (daca o eroare nu e interceptata si tratata nu vom puncta anumite puncte din barem).*

Barem:

5p	**Se verifica daca numarul de parametri e corect, iar daca nu se afiseaza o eroare**
8p	**Programul citeste fisierul sursa si afiseaza un mesaj de eroare in caz de erori legate de accesul la fisier** *[4p deschiderea fisierului, 4p interceptie I/O errors]*
5p	**Se verifica sa nu fie coloane cu acelasi nume**
12p	**Verificare daca formatul e corect** *4p - citire header si crearea liste de coloane* *8p - verificare daca fiecare linie are cate cuvinte sunt in header. Liniile goale se ignora.*
5p	**Verificare daca al doilea parametru e un nume de coloana valid**
9p	**Afisare rezultat conform specificatiilor** *[5p item-urile unice sunt sortate dupa numarul de apartii descrescator, 3p itemurile cu acelasi numar de aparitii sunt sortate alfabetic, 1p → afisare pe ecran]*
8p	**Programul compileaza corect si ruleaza conform specificatiilor**