Information and Communication Technologies — S6 — Work n°11

Handling data — Part 3

In last work, we have discovered the ".csv" format. Data encoded in this fashion can be seen as an array of arrays (or a 2d array, a matrix, or simply put, data that can be viewed as a tabular in a spreadsheet). Usually, each line in the file represents an entity (for example, a city), and each entity has different characteristics (for example: postal code, name…): one per column (two consecutive column are separated by commas (",") or, for French, by semi-colons (";")).

First, we recall that we work with a file containing GPS coordinates of French cities. The file is in French, but it should not be a problem. You can see the beginning of the file on Figure 1, and you can download it at http://www.barsamian.am/2023-2024/S6ICTA/TP10_Cities.csv.

```
Code_commune_INSEE;Nom_commune;Code_postal;Libelle_acheminement;Ligne_5;coordonnees_gps
80355;FRESNEVILLE;80140;FRESNEVILLE;;49.9469630616, 1.753960976
80365;FRICAMPS;80290;FRICAMPS;;49.7720118421, 1.95186211928
80368;FRIVILLE ESCARBOTIN;80130;FRIVILLE ESCARBOTIN;;50.0912781795, 1.52364516053
80379;GLISY;80440;GLISY;;49.8341850031, 2.39954269272
80387;GRATTEPANCHE;80680;GRATTEPANCHE;;49.8142899245, 2.29952854065
```

Figure 1: File containing GPS coordinates of French cities.

The names of cities in this file are without accents, without special characters (no hypen, no apostrophe…), and use capital letters only. A python code to manipulate this file, as it was needed in the last work, is given in Listing 1, and can be downloaded from:
http://www.barsamian.am/2023-2024/S6ICTA/TP10_Cities_bis.py.

```python
city = input("What city are your searching for? ")

# "iso-8859-1" is a common encoding. On Linux, the standard encoding is
# "utf-8" and on Windows you can also encounter "cp1252".
f = open("TP10_Cities.csv", "r", encoding="iso-8859-1")
# strip() removes blank characters at the beginning and the end of the string,
# here in particular the end of line characters left by readline()
for line in f:
    values = line.strip().split(";")
    name = values[1]
    if (city.upper() == name):
        print("The GPS coordinates of " + name + " are: " + values[5])
f.close()
```

Listing 1: Skeleton code to handle our cities file.

As last time, there is a first problem: different French cities have the same name! For example "Annay" is the name of two different cities.

1. If the name, alone, is not enough to uniquely distinguish French cities, what do you suggest to uniquely distinguish them?

To help the users of your program, we will add information about the "département" where the city lies. Each French département has an identifier (identifiers that are recalled in another file). The file is in French, but it should not be a problem. You can see the beginning of the file on Figure 2, and you can download it at http://www.barsamian.am/2023-2024/S6ICTA/TP11_Departements.csv.

2. Write a function dpt_id_from_zipcode that takes as input a string which contains the zipcode of a French city ("Code postal", 3rd column) and outputs the id of the département where the city lies. In this question, we will assume that the id of the département is contained in the two first digits of the zipcode. For instance:

    - the call dpt_id_from_zipcode("62880") should return "62"

3. Modify the program to add the id of the département of the city, when you print it. For instance if you search for "Annay", the program should print:

   ```
   The GPS coordinates of ANNAY (58) are: 47.5311404118, 2.91887895811
   The GPS coordinates of ANNAY (62) are: 50.470091244, 2.86997706713
   ```

```
code_departement,nom_departement,code_region,nom_region
01,Ain,84,Auvergne-Rhône-Alpes
02,Aisne,32,Hauts-de-France
03,Allier,84,Auvergne-Rhône-Alpes
04,Alpes-de-Haute-Provence,93,Provence-Alpes-Côte d'Azur
05,Hautes-Alpes,93,Provence-Alpes-Côte d'Azur
```

Figure 2: File containing identifiers of French départements.

4. Write a function `dpt_name_from_id` that takes as input a string which contains the id of a French département, and outputs the name of this département. This function should:

   - open `TP11_Departements.csv`
   - read it line by line
   - when it finds the line with the correct id, return the name
   - at the end of the file reading (this only happens if we don't find the given id in the file), return an empty string

   For instance:

   - the call `dpt_name_from_id("62")` should return `"Pas-de-Calais"`

5. Modify the program to also add the name of the département of the city, when you print it. For instance if you search for "Annay", the program should print:

   ```
   The GPS coordinates of ANNAY (58, Nièvre) are: 47.5311404118, 2.91887895811
   The GPS coordinates of ANNAY (62, Pas-de-Calais) are: 50.470091244, 2.86997706713
   ```

6. Search for the city "Justiniac". What happens?

7. The zipcode of Justiniac (département 9, Ariège) is 9700. What is the result of the call `dpt_id_from_zipcode("9700")`? What should be the result? Modify the function `dpt_id_from_zipcode` to take zipcodes of 4 digits into consideration. This function should always return a string with two characters ; in the case of Justiniac, it should return `"09"`.

8. Search for the city "Cayenne". What happens?

9. The zipcode of Cayenne (département 973, Guyane) is 97300. What is the result of the call `dpt_id_from_zipcode("97300")`? What should be the result? Modify the function `dpt_id_from_zipcode` so that when the zipcode is greater than 96000, it returns the 3 first characters instead of the 2 first characters. In the case of Cayenne, it should return `"973"`.

10. List all the cities in the cities file that have the same name of another French city.

11. BONUS: Given the name of a French "région", list all the départements that are part of this région.

12. EXTRA BONUS: List all the French cities that are south of Valence.