CS101 Homework #4
Soccer player data processing

Read the homework description carefully and follow the instructions. Please be fully aware that this homework is an individual task; you can discuss the problem with your friends, but you must not implement your idea together. You will fail the entire course (that is, your CS101 grade will be an F) if you are found to be involved in any attempt of plagiarism.

Objectives

- Read and organize data from csv files
- Define a class and create class objects

Preliminaries

- In this homework, you will use a dataset that contains 18207 soccer players in 2019. You can find the detailed information of the dataset from the link below (if you're interested). We will provide a simplified and shuffled version of the dataset in this homework as a csv file.
  https://www.kaggle.com/karangadiya/fifa19
- The csv file of the soccer player dataset we are providing contains the following five columns.
  - Name
  - Age
  - Nationality
  - Overall (rating of the player)
  - Position (soccer players have their main position in the game)
- For your information, here are some examples of the data, extracted directly from the csv file.

```
(...)
M. Martínez,34,Argentina,68,CB
L. Vigen Christensen,23,Denmark,68,RCM
M. Ogasawara,39,Japan,65,CDM
T. Hamed,29,Egypt,67,LDM
D. Buballa,28,Germany,68,LB
C. Eriksen,26,Denmark,88,CAM
(...)
```

- You can use Python's csv module in this homework. Please refer to the link below to understand how to use the module.
  https://docs.python.org/3/library/csv.html
  We also provide some skeleton code as a starting point for using the csv module. Note that using the csv module is not mandatory, but you will find it easy to process the given csv file when using the module.

Problem 1 [5 points]

In this problem, you are asked to implement the function `is_nth_player_world_class(n)`.

Function `is_nth_player_world_class(n)`

- Parameter - n: int
  - An integer that specifies the row number in the csv file.
  - We assume the first row in the file has the row number of 0, not 1.
- Return: tuple(str, bool)
  - A tuple containing a string, which is the specified player's name, and a boolean, indicating whether the player is a world class player.
  - A world class player has an overall rating greater than or equal to 80.
- Exceptions
  - If n is not in a valid range of row numbers, return the string `'No such player'`.

Examples

- `is_nth_player_world_class(-1)` returns `'No such player'`
- `is_nth_player_world_class(0)` returns `('M. Coronel', False)`
- `is_nth_player_world_class(1750)` returns `('L. Messi', True)`

Problem 2.1 [10 points]

In this problem, you are asked to define `Country` and `Player` classes, and then implement the function `convert_csv_file_to_objects()`.

Class `Country`: This class should contain two class variables as follows.

- `country_name`: str
- `player_list`: list(`Player`)

Class `Player`: This class should contain five class variables as follows.

- `name`: str
- `age`: int
- `nationality`: str
- `overall`: int
- `position`: str

Function `convert_csv_file_to_objects()`

- No parameters
- Return: list(`Country`)
    - Read through the given csv dataset file and form a list of Country objects, where each country object is initialized with player objects that belong to that country.
    - Note that we are using 'Nationality' from the csv file into 'country_name' in Country class and 'nationality' in Player class.
    - The order of Country objects in the returning list must be sorted based on the csv file. In other words, the nationality that appears earlier in the csv file should be placed earlier in the list of Country objects.
    - The order of Player objects in player_list in each Country object must be sorted based on the csv file. In other words, the player that appears earlier in the csv file should be placed earlier in player_list for each Country object.

Examples

- `country_objects = convert_csv_file_to_objects()`
- `print(country_objects[0].country_name)`
    - `'Argentina'`
- `print(country_objects[1].country_name)`
    - `'Republic of Ireland'`
- `print(country_objects[0].player_list[50].name)`
    - `'F. Mussis'`
- `print(country_objects[1].player_list[30].name)`
    - `'D. Murphy'`

Problem 2.2 [5 points]

In this problem, you are asked to implement the function `get_country_object_by_country_name(objects, country_name)`.

Function `get_country_object_by_country_name(objects, country_name)`

- Parameter - objects: list(`Country`)
  - A list of Country objects created from problem 2.1
- Parameter - country_name: str
  - A string that specifies the name of the country.
- Return: `Country` object
  - Return a Country object that is specified by the given country_name.
- Exceptions
  - If there are no countries that match with the given country_name, return a string `'No such country'`.

Examples

- `country_objects = convert_csv_file_to_objects()  # Problem 2.1`
- ```
  south_korea = get_country_object_by_country_name(
                country_objects, 'Korea Republic')
  ```
- `print(south_korea.player_list[12].name)`
  - `'Lee Seung Woo'`
- `b612 = get_country_object_by_country_name(country_objects, 'b612')`
- `print(b612)`
  - `'No such country'`
  ```

Problem 3 [15 points]

In this problem, you are asked to define a member function of the `Country` class named `get_best_player_and_number_of_world_classes()`.

Function `get_best_player_and_number_of_world_classes()`

- Note that this function is a member function of the `Country` class, not a standalone function.
- No parameters
- Return: tuple(str, int)
  - A tuple containing a string, which is the name of the player who has the highest overall rating among world class players in the country, and an integer, which is the number of world class players in the country.
  - A world class player has an overall rating greater than or equal to 80.
  - If there's a tie on the highest overall rating, return the player's name that appears **the earliest** in the list (the lowest index in the list).
- Exceptions
  - If there are no world class players in the country, return string `'No world class'`.

Examples

```
- country_objects = convert_csv_file_to_objects()        # Problem 2.1
- south_korea = get_country_object_by_country_name(
                country_objects, 'Korea Republic')       # Problem 2.2
- print(south_korea.get_best_player_and_number_of_world_classes())
    - ('H. Son', 1)
- argentina = get_country_object_by_country_name(
                country_objects, 'Argentina')            # Problem 2.2
- print(argentina.get_best_player_and_number_of_world_classes())
    - ('L. Messi', 33)
- angola = get_country_object_by_country_name(
                country_objects, 'Angola')               # Problem 2.2
- print(angola.get_best_player_and_number_of_world_classes())
    - 'No world class'
```

Problem 4 [15 points]

In this problem, you are asked to define a member function of the `Country` class named `get_best_players_for_each_position()`.

Function `get_best_players_for_each_position()`

- Note that this function is a member function of the `Country` class, not a standalone function.
- No parameters
- Return: tuple(str, str, str, str)
  - A tuple containing four strings. Each string is the name of the player who has the highest overall rating within each position.
    - First string: Forward.
      This position includes LW, RW, LF, RF, LS, RS, CF, and ST.
    - Second string: Midfielder.
      This position includes LDM, CDM, RDM, LAM, CAM, RAM, LCM, CM, RCM, LM, and RM.
    - Third string: Defender.
      This position includes LCB, CB, RCB, LWB, RWB, LB, and RB.
    - Fourth string: Goalkeeper.
      This position is GK.
  - If there's a tie on the overall rating within the position, choose **the youngest** player. If the ages are also same, choose the player that appears **the earliest** in the list (the lowest index in the list)
  - If the player with the highest overall rating within the position has the overall rating that is less than 70, the player is disqualified.
  - If there are no players that satisfy the condition, return None for that position.

Examples

- `country_objects = convert_csv_file_to_objects()  # Problem 2.1`
- `south_korea = get_country_object_by_country_name(country_objects, 'Korea Republic')  # Problem 2.2`
- `print(south_korea.get_best_players_for_each_position())`
  - `('Suk Hyun Jun', 'H. Son', 'Kim Min Jae', 'Cho Hyun Woo')`
- `germany = get_country_object_by_country_name(country_objects, 'Germany')  # Problem 2.2`
- `print(germany.get_best_players_for_each_position())`
  - `('L. Sané', 'T. Kroos', 'M. Hummels', 'M. ter Stegen')`
- `iraq = get_country_object_by_country_name(country_objects, 'Iraq')  # Problem 2.2`
- `print(iraq.get_best_players_for_each_position())`
  - `(None, 'A. Adnan', None, None)`