Zelle 3e Chapter 11 Coding Assignment

General Instructions

My expectations for your work on coding assignment exercises will grow as we progress through the course. In addition to applying any new programming techniques that have been covered in the current chapter, I will be expecting you to follow all of the good programming practices that we have adopted in the preceding weeks. Here is a quick summary of good practices that we have covered so far:

- Include a single-line comment with name of program file.
- Include a single-line comment that describes the intent of the program.
- Place your highest-level code in a function named `main`.
- Include a final line of code in the program that executes the `main` function.
- Follow all PEP-8 Python coding style guidelines enforced by the PyCharm Editor. For example, place two blank lines between the code making up a function and the code surrounding that function.
- Output printed by the program (both prompts and results) should be polite and descriptive.
- Choose names for your variables that are properly descriptive.
- Choose names for your functions that are properly descriptive.
- Close all files before the conclusion of the program.
- Model your solution after the code that I demonstrate in the tutorial videos.
- Remember to test your program thoroughly before submitting your work.
- Your code must pass all relevant test cases. Make sure that it passes tests at the boundaries created by *if, else*, and *elif* conditions in your program (boundary value tests).

Exercise 1

Create a program named *country_population_lists.* This program reads a data file containing country names and a related population value for the country. Each line of the file contains the data for one country.

The program should create a list of tuples, each tuple containing a country name and a population value.

Having created the list of tuples, this list should be sorted into country name order and a list should be printed with the title COUNTRY NAME ORDER.

When the first list is complete, the list of tuples should be sorted into population order and a list should be printed with the title POPULATION ORDER.

When coding and testing this program, follow the approach that I take in the Part 1 tutorial video.

This program should process both test data files included in the starter files for the assignment and produce appropriate output. These files are named:

- empty_file.txt
- country_pop_data.txt

When this program is run, console sessions should look like this:

```
Please enter input file name: empty_file.txt

COUNTRY NAME ORDER

POPULATION ORDER


Please enter input file name: country_pop_data.txt

COUNTRY NAME ORDER
Bangladesh       165435000
Brazil           209772000
China           1394860000
India           1338820000
Indonesia        265015300
Japan            126440000
Nigeria          193392517
Pakistan         202477000
Russia           146877088
USA              328077000
```

POPULATION ORDER

Japan	126440000
Russia	146877088
Bangladesh	165435000
Nigeria	193392517
Pakistan	202477000
Brazil	209772000
Indonesia	265015300
USA	328077000
India	1338820000
China	1394860000

Exercise 2

Create a program named *distribute_race_awards_with_dictionary.* This program performs a simple lookup based upon integer input provided by the user at the console. This program is different from a similar program created in a previous chapter in that the lookup is performed using a Python dictionary.

When coding and testing this program, follow the approach that I take in the Part 2 tutorial video.

When this program is run, console sessions should look like this:

```
Please enter the place in which the participant finished (1, 2,
3, ...): 1
Participant place: 1
Participant award: Blue Ribbon


Please enter the place in which the participant finished (1, 2,
3, ...): 6
Participant place: 6
Participant award: Purple Ribbon


Please enter the place in which the participant finished (1, 2,
3, ...): 7
Participant place: 7
Participant award: Participant Ribbon


Please enter the place in which the participant finished (1, 2,
3, ...): 0
Participant place: 0
Participant award: Input Error - Place must be greater than 0
```

Exercise 3

Create a program named *analyze_slot_machine_tries_ignoring_duplicates*. This program should implement functionality that is the same as the Part 3 tutorial example, except for the following differences:

- This program should not include duplicate colors from a single try when accumulating counts for colors.

This program should process both test data files included in the starter files for the assignment and produce appropriate output. These files are named:

- empty_file.txt
- slot_values.txt

When this program is run, console sessions should look like this:

```
Please enter the name of the input file: empty_file.txt


Please enter the name of the input file: slot_values.txt
Blue         2990
Green        2983
Orange       3024
Purple       3015
Red          2959
Yellow       3011
```

Tools
Use PyCharm to create and test both python programs.

Submission Method
Follow the process that I demonstrated in the tutorial video on submitting your work. This involves:

- Locating the properly named directory associated with your project in the file system.
- Compressing that directory into a single .ZIP file using a utility program.
- Submitting the properly named zip file to the submission activity for this assignment.

File and Directory Naming
Please name your Python program files as instructed in each exercise. Please use the following naming scheme for naming your project:

YourLastName_YourFirstName_exercises_zelle_3e_chapter_11

When you have compressed your project directory into a .ZIP file, it should have the following name structure:

YourLastName_YourFirstName_exercises_zelle_3e_chapter_11.zip

Due By
Please submit this assignment by the date and time shown in the Weekly Schedule.