Computer Project 05

This assignment is worth 40 points (4.0% of the course grade) and must be **completed no later than 11:59 PM on Monday, February 27, 2023.**
Note that there will be zero penalties for assignments handed in on the following two days: February 28 & March 1.
Note that you must click on the "Submit" button. Otherwise, it will not be considered as submitted.

Assignment Overview

In this assignment, you will practice with files and error-handling using exceptions. You are not allowed to use the split method. You are not allowed to use any advanced data structure (e.g., Lists, Dictionaries, Classes, etc.). Using any of these will result in a zero.

Assignment Background

Anime like One Piece, Dragon Ball, and Pokémon have been seen by million of viewers worldwide. Anime-Planet.com is a website which contains records of many anime titles and allows people to review and score them based on popularity, story, animation, etc.

This project will read files with these records and address the following questions: (1) What anime title(s) has the highest score? (2) What title(s) have the highest number of episodes? (3) What title(s) have the lowest scores?

The files used in this project were adapted from the following source data:
https://www.kaggle.com/datasets/vishalmane10/anime-dataset-2022

Assignment Specifications

1. You must implement the following functions:

   a) **`open_file() --→ fp`**
   This function repeatedly prompts the user for a filename until the file is opened successfully. An error message should be shown if the file cannot be opened. It returns a file pointer. Use try-except to determine if the file can be opened. Check the `strings.txt` file for the appropriate strings to use in the prompt and error message to match the tests. When opening the file, you need to use `encoding="utf-8".` For example,

```
fp = open("anime_tiny.txt", "r", encoding="utf-8")
```

   b) **`find_max(value, name, max_value, max_name) -→ float,str:`**
   This function returns the value and the formatted name whose value is greater than `max_value`. If value is less than `max_value` return `max_value` and `max_name`. If `value` is higher than `max_value` return `value` and formatted `name.` Otherwise, concatenate name into `max_name` and return `max_value` and `max_name`. This function is a single comparison between the parameters.

   Use the following formatting to format the returned strings for names:
   `"\n\t{}"` (when returning a new max name) and `"{}\n\t{}"` (when concatenating the `name` after `max_name`)

      i. **Input:**
         1. value (float)

2. **name (str)**
3. **max_value (float)**
4. **max_name (str)**

ii. **Output:**

1. **Maximum value (float)**
2. **Maximum name (str)**

c) **`find_min(value, name, min_value, min_name) -→ float,str:`**

This function returns the value and name whose value is less than `min_value.` If value is greater than `min_value` return `min_value` and `min_name`. If `value` is lower than `min_value` return `value` and formatted `name.` Otherwise, concatenate `name` into `min_name` and return `min_value` and `min_name`. This function is a single comparison between the parameters.
Use the following formatting to format the returned strings for names:
`"\n\t{}"` (when returning one name) and `"{}\n\t{}"` (when concatenating the `name` after `min_name`)

i. **Input:**

1. **value (float)**
2. **name (str)**
3. **min_value (float)**
4. **min_name (str)**

ii. **Output:**

1. **min_value (float)**
2. **min_name (str)**

d) **`read_file(data_fp) → float,str,float,str,float,str,float`**

This function reads a file with the anime title data. While reading the file, this function finds the highest scoring title, the highest episode count title, and the lowest scoring title. Also computes the average score for all titles in the file. For this project, we are interested in the following columns:

```
title = line[0:100]
score = line[100:105]
episodes = line[105:110]
```

When computing the average, round the value to two decimal places. Only consider the anime title's score and episodes that are not "N\A" for the max, min, and average computations. For example, if One Piece has a score of 8.66 and episode count is "N/A" (because is still airing), then compute the max score, min score, and average score but do not compute the max episode count evaluation. If the count of scores is 0 (e.g., all scores are N/A), return `0.0`

Important: You are required to use string slicing as shown above. Not doing so can affect reading the data from the file.

i. **Input:**

1. **`fp (File pointer object)`**

ii. **Output (return values):**

1. **`max_score (float)`**
2. **`max_score_name (str)`**
3. **`max_episodes (float)`**
4. **`max_episode_name (str)`**
5. **`min_score (float)`**
6. **`min_score_name (str)`**

7. **`avg_score (float)`**

e) **`search_anime (data_fp, search_str) -→ int, str`**

This function reads through a file and returns the title and release season if it contains the search string passed as parameter. Concatenate all the formatted strings of titles and release season, if there are multiple titles that includes the `search_str.` For this project, we are interested in the following columns:

```
title = line[0:100]
release_season = line[110:122]
```

Each entry for title and release season should be formatted using the following string formatting:
`"\n\t{}{}"` (first string is the title in a field width of 100 and then release season as a string in a field width of 12)

i. **Input:**
   1. **data_fp (file pointer object)**
   2. **search_str (str)**

ii. **Output:**
   1. **count (int): number of titles with the search string**
   2. **out_str (str): Title and release season output string**

f) **main():**

This function is the starting point of the program. The program prompts the user for an option until the user decides to stop the program:

i. **Option 1:**
   1. Prompt the user for a file to open.
   2. Call the `read_file` function and receive the `max, min`, and `average` values.
   3. Display the highest score value and all anime title(s) with the same score.
   4. Display the highest number of episodes and all anime title(s) with the same number of episodes.
   5. Display the lowest score value and all anime title(s) with the same score. Check the `strings.txt` file for the appropriate strings to use to match the tests.
   6. **Close the file.**

ii. **Option 2:**
   1. Prompt the user for a file to open.
   2. Prompt the user for a search string to look for in each title within the file.
   3. Call the `search_anime` function and receive the number of titles found with the title and release season of all elements found.
   4. Display the results. Check the `strings.txt` file for the appropriate strings to use to match the tests.
   5. **Close the file.**

iii. **Option 3:** Stop the program.

If the user does not enter any of these options, the program needs to display an error message and prompt again until a valid option is selected. Check the `strings.txt` file for the appropriate strings to use to match the tests.

Assignment Notes and Hints

1. The coding standard for CSE 231 is posted on the course website:
   http://www.cse.msu.edu/~cse231/General/coding.standard.html

   Items 1-9 of the Coding Standard will be enforced for this project.

2. The program will produce reasonable and readable output, with appropriate labels for all values displayed.

3. Be sure to prompt the user for the inputs in the correct order. And, your program cannot prompt the user for any supplementary inputs.

4. We provide a *proj05.py* program for you to start with.

5. You are not allowed to use advanced data structures such as list, dictionaries, classes, etc. However, you are allowed to read ahead and use try-except.

6. If you “hard code” answers, you will receive a grade of zero for the whole project. An example of hard coding is to simply print the correct highest scores of a file rather than having the program find it in the file and then printing it.

Suggested Procedures

- Solve the problem using pencil and paper first. You cannot write a program until you have figured out how to solve the problem. This first step can be done collaboratively with another student. However, once the discussion turns to Python specifics and the subsequent writing of Python statements, you must work on your own.

- Cycle through the following steps to incrementally develop your program:
  - Edit your program to add new capabilities.
  - Run the program on Spyder and fix any errors.
  - Use the Coding Rooms system to submit the current version of your program.

- Be sure to use the Coding Rooms system to submit the final version of your program.

- Be sure to log out when you leave the room, if you’re working in a public lab.

The last version of your solution is the program which will be graded by your TA. You should use the Coding Rooms system to back up your partial solutions, especially if you are working close to the project deadline. That is the easiest way to ensure that you won’t lose significant portions of your work if your machine fails or there are other last-minute problems.

Assignment Deliverable

The deliverable for this assignment is the following file:

proj05.py – the source code for your Python program

Be sure to use the specified file name and to submit it for grading via the Coding Rooms system before the project deadline.

Grading Rubrics

```
Computer Project #05                                  Scoring Summary

General Requirements:
  (4 pts) Coding Standard 1-9 (descriptive comments, function headers,
etc...)

Implementation:
  (3 pts) open_file (No CodingRooms Test)
     -1 point Did not use try/except
     -1 point Did not use while loop

  (6 pts) read_file function test

  (5 pts) search_anime function test

  (3 pts) find_max function test

  (3 pts) find_min function test

  (4 pts) Pass Test1

  (4 pts) Pass Test2

  (4 pts) Pass Test3

  (4 pts) Pass Test4

Note: hard coding an answer earns zero points for the whole project
-10 points for not using main() in a meaningful way.
```

Test 1:

```
Anime-Planet.com Records
Anime data gathered in 2022

Options
     1) Get max/min stats
     2) Search for an anime
     3) Stop the program!
     Enter option: 3

Thank you using this program!
```

Test 2:

```
Anime-Planet.com Records
Anime data gathered in 2022

Options
      1) Get max/min stats
      2) Search for an anime
      3) Stop the program!
      Enter option: 1

Enter filename: anime_all.txt


Anime with the highest score of 9.14:

      Kaguya-sama: Love is War - Ultra Romantic


Anime with the highest episode count of 3,057:

      3000 Whys Of Blue Cat


Anime with the lowest score of 2.25:

      Nami


Average score for anime in file is 6.72

Options
      1) Get max/min stats
      2) Search for an anime
      3) Stop the program!
      Enter option: 1

Enter filename: anime_action.txt


Anime with the highest score of 9.13:

      Fullmetal Alchemist: Brotherhood


Anime with the highest episode count of 500:

      Naruto Shippuden


Anime with the lowest score of 2.91:

      Ex-Arm
```

```
Average score for anime in file is 6.91

Options
        1) Get max/min stats
        2) Search for an anime
        3) Stop the program!
        Enter option: 3

Thank you using this program!
```

Test 3:

```
Anime-Planet.com Records
Anime data gathered in 2022

Options
        1) Get max/min stats
        2) Search for an anime
        3) Stop the program!
        Enter option: 2

Enter filename: anime_small.txt

Enter anime name: Dragon Ball

There are 11 anime titles with 'Dragon Ball'

        Dragon Ball: Curse of the Blood Rubies                                                              fall 1986
        Dragon Ball                                                                                         winter 1986
        Dragon Ball: Mystical Adventure                                                                     summer 1988
        Dragon Ball Z: Lord Slug                                                                            winter 1991
        Dragon Ball Z: The Return of Cooler                                                                 winter 1992
        Dragon Ball Z: Broly - Second Coming                                                                winter 1994
        Dragon Ball Z: Fusion Reborn                                                                        winter 1995
        Dragon Ball Z Kai                                                                                   spring 2009
        Dragon Ball Z: Battle of Gods                                                                       winter 2013
        Dragon Ball Z: Resurrection 'F'                                                                     spring 2015
        Dragon Ball Super: Super Hero                                                                       spring 2022

Options
        1) Get max/min stats
        2) Search for an anime
        3) Stop the program!
        Enter option: 2

Enter filename: anime_fantasy.txt

Enter anime name: Saint Seiya

There are 4 anime titles with 'Saint Seiya'

        Saint Seiya: Knights of the Zodiac                                                                  fall 1986
        Saint Seiya: Legend of Crimson Youth                                                                summer 1988
        Saint Seiya Omega                                                                                   spring 2012
        Saint Seiya: Legend of Sanctuary                                                                    spring 2014

Options
        1) Get max/min stats
        2) Search for an anime
        3) Stop the program!
        Enter option: 3

Thank you using this program!
```

Test 4:

```
Anime-Planet.com Records
Anime data gathered in 2022

Options
      1) Get max/min stats
      2) Search for an anime
      3) Stop the program!
      Enter option: a

Invalid menu option!!! Please try again!

Options
      1) Get max/min stats
      2) Search for an anime
      3) Stop the program!
      Enter option: A

Invalid menu option!!! Please try again!

Options
      1) Get max/min stats
      2) Search for an anime
      3) Stop the program!
      Enter option: 1

Enter filename: xxx

File not found!

Enter filename: anime_small.txt


Anime with the highest score of 9.13:

      Fullmetal Alchemist: Brotherhood


Anime with the highest episode count of 3,057:

      3000 Whys Of Blue Cat


Anime with the lowest score of 2.40:

      Puzzle of Autumn


Average score for anime in file is 6.72

Options
      1) Get max/min stats
      2) Search for an anime
      3) Stop the program!
      Enter option: 2

Enter filename: anime_small.txt

Enter anime name: yyy

No anime with 'yyy' was found!

Options
      1) Get max/min stats
      2) Search for an anime
      3) Stop the program!
      Enter option: 3

Thank you using this program!
```