Computer Project #5

Assignment Overview

This assignment focuses on the design, implementation and testing of Python programs to process data files, as described below.

It is worth 45 points (4.5% of course grade) and must be completed no later than 11:59 PM on Monday, October 19.

Assignment Deliverables

The deliverables for this assignment are the following files:

> `proj05a.py` – your source code program for Part A
> `proj05b.py` – your source code program for Part B

Be sure to use the specified file names and to submit them for grading via the **handin system** before the project deadline.

Assignment Specifications

1. The World Health Organization (WHO) compiles data about immunization levels around the world. The file named "`polio.txt`" contains data about the level of polio vaccinations in various countries over time.

Each line of the file contains the following fields, where there is exactly one space between fields:

> Income Level (6 characters)
> Region (4 characters)
> Country (55 characters)
> Year (4 characters)
> Percent Vaccinated (3 characters)

The "Income Level" field identifies the category assigned to that country by the World Bank:

> `WB_LI`   low income
> `WB_LMI`   lower middle income
> `WB_UMI`   upper middle income
> `WB_HI`   high income

The "Region" field identifies the region assigned to that country by WHO:

AFR	Africa
AMR	Americas
EMR	Eastern Mediterranean
EUR	Europe
SEAR	South-East Asia
WPR	Western Pacific

The "Country" field contains the name of the country.

The "Year" field contains the year for which the data was compiled.

The "Percent Vaccinated" field contains an integer number representing the percentage of children in that country who have received polio vaccine by the age of one.

2. You will develop the two Python programs described below. Each program will be in a separate source code file (the names are given under "Assignment Deliverables").

3. Your programs may not use lists, tuples, sets or dictionaries.

4. Your programs may use any built-in or standard library functions.

Part A

1. The program in "proj05a.py" will copy selected lines from "polio.txt" into a file selected by the user.

a) The program will always read from "polio.txt" (it will not prompt the user for the name of the input file). If it is unable to open that file, the program will halt.

b) The program will prompt the user for the name of the output file. If that file does not exist, the program will create it and continue. If that file does exist, the program will discard the current contents of the file and continue.

c) The program will prompt the user to enter a year, and will copy all lines of "polio.txt" selected by the user's response. A line is selected if the user's response matches the Year field or any of its prefixes. All lines are selected if the user's response is any of the values in the set {"", "all", "ALL"}. Note that "" is the empty string.

For example, a line whose Year field contains "1987" would be selected by any of the following user responses: {"1", "19", "198", "1987", "", "all", "ALL"}.

2. The output file created by the program will have the same format as the input file (same field widths and spacing). Note that when the user selects all lines, the output file will be identical to the input file.

3. The program will display appropriate messages to inform the user about any unusual circumstances.

Part B

1. The program in "`proj05b.py`" will display one summary report to the user.

a) The program will prompt the user to enter the name of the input file. If it is unable to open that file, the program will prompt the user again until the user enters a valid file name.

b) The program will prompt the user to enter a year, and will then prompt the user to enter an income level. The income level must be one of the characters in the set {1, 2, 3, 4}, where 1 corresponds to "low income", 2 corresponds to "lower middle income", 3 corresponds to "upper middle income" and 4 corresponds to "high income".

c) The program will identify all records (lines) in the input file which match the user's criteria for year and income level, and the program will display a report with the following information:

   The count of records in the input file which match the user's criteria
   The average percentage for those records (displayed with one fractional digit)
   The country with the lowest percentage for those records
   The country with the highest percentage for those records

If more than one country has the lowest or highest percentage, the report will display the information for the last of those countries found by your program.

For the last two items (lowest percentage and highest percentage), the program will display all information about the country: the name of the country, the World Bank income level (not the code), the WHO region (not the code), the year, and the percentage vaccinated.

2. The program will display appropriate messages to inform the user about any unusual circumstances.

3. The program will contain the following functions (you may develop additional functions):

```
open_file() → file object
process_file( file object ) → None
main() → None
```

The notation above gives the name of each function, the number and type of its argument(s), and the type of its return value.

a) The function names will be spelled exactly as shown (for example, `open_file`).

b) Function `open_file` has no parameters. It returns a file object after prompting the user to enter the name of the input file (see above).

c) Function `process_file` has one parameter (a file object). It performs the processing to read the input file and display the report.

d) Function `main` has no parameters. It invokes the other two functions and then closes the file object.

Assignment Notes

1. The data in "`polio.txt`" is from the World Health Organization's website and is used with permission:

   http://apps.who.int/gho/data/node.main.A824?lang=en

2. Documentation about the built-in function `open` is on the Python website:

   http://docs.python.org/3/library/functions.html#open

In particular, the information about modes "r" and "w" may be useful.

3. A simple approach to determining the maximal value in a set of values is described below.

   Initialize the current maximal value to something smaller than all valid values
   For each value in the set:
       If the value is larger than the current maximal value, save it as the current maximal value

A similar procedure can be used to determine the minimal value in a set of values.

Suggested Procedure

- *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 may 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.

- Create a small data file by taking representative lines from the instructor-supplied data file, then use that small data file for the initial testing of your programs. That approach will allow you to more easily determine if your logic is correct before using the large instructor-supplied data file.

- Complete the program for Part A before working on the program for Part B.

- Develop a simple version of the program for Part A, then run the program and track down any errors.

- Use the **handin system** to turn in the first version of your solution.

- Cycle through the steps to incrementally develop your program:

  - Edit your library program to add new capabilities.
  - Run the program and fix any errors.
  - Use the **handin system** to submit the current version of your solution.

- Use the **handin system** to submit the final version of your solution.

- Once Part A is complete, work on Part B in the same way: incrementally develop your program and use the **handin system** to submit your work

- You would be wise to back up your files on your H: drive, also.

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