PROJECT: Data Processing. DESCRIPTION, INCLUDING SUBMISSION INSTRUCTIONS

READ ALL THIS DOCUMENT

This is an up to 3 people team work (you can also work in a team of 2 people or individually if you prefer so). While you may discuss generalities with colleagues from other teams about this exercise, you cannot develop the same code, nor share code among teams, nor obtain code from other sources.

Being a team exercise it places a big responsibility on each individual. You want to respect and be honest with your partners and with yourself: DO YOUR SHARE, and BE KNOWLEADGEABLE OF THE WHOLE ASSIGNMENT. Working on this assignment is also a way for you to prepare for the final exam.

Deadline: Sunday December 3, 12:00 NOON!!!!!!!! . No extensions are possible. A solution will be posted after the deadline allowing time for reviewing and preparing for the exam

A. PROBLEM SOLVING

1. FIRST: Read the Problem Solving Suggestions document. IT IS BRIEF AND HIGHLY RECOMMENDED.

2. **UNDERSTAND WHAT YOU ARE ASKED TO DO** AND WHAT YOU ARE ASKED TO SUBMIT.

3. START WORKING ON THIS AS SOON AS POSSIBLE

B. GENERAL PROGRAM DESCRIPTION

You are asked to implement a Python program which calculates the points that students obtain in a multiple choice questions test, based on data with their detailed responses, the answer key, and points associated to each question.
The user will have a choice to run the program to process all the students or only selected students, one student at a time, showing the totals for each student after processing each student.
The program not only calculates the points that each student gets, but it also provides several statistics. More details are provided below and check the sample runs.

The input to the program are text files in an explicit format as detailed below. Concrete data files to help you test your program are provided; you may create different data files to best test your program as long as you follow the prescribed formatting. The program should also produce a data file as output information, with a specific format, as described below.

C. DEVELOPING YOUR PROGRAM IN STAGES

You are recommended to develop this program gradually and testing it often. It is recommended that you advance in stages as follows:
Stage I: process all the students data, only do the most basic stats, do a basic user interaction but do not yet validate the user's input.
Stage II. Incorporate validations of the user input
Stage III. Provide the option of processing specific selected students
Stage IV. Provide additional statistics, incorporate graphics.

D. FILES DESCRIPTIONS

1) Input to the program, ANSWER KEY AND POINTS **file: IN_key+pts.txt**

This file will include only two lines.

- **first line**: a string with the answer key, not including any space.
- 1 stands for A, 2 for B, … 5 stands for E
- **second line**: points associated to each question, each separated by at least one space
- There will be as many responses to questions (q) as points, ordered by question number

Example (q =10 in this example):
1134215132
1 1.5 1 1 1 1 1 1.5 2 1.5

Viewed as a string this example would be "1134215132\n1 1.5 1 1 1 1 1 1.5 2 1.5\n"

2) Input to the program, STUDENTS ANSWERS **file: IN_data_studs .txt**

- This file includes the answers from a number n of students
- There is one line per student.
- Each line contains the student's name, then at least one space, and then (q) student's answers as a string of q character digits 1,2,3,4,5, ordered by question number
- Assume that the students' names do not include spaces (e.g the name is one word only)
- Assume that there are no spaces in the answers (i.e. everyone responded all the questions)

Example (n=3 students, q=10 answers each in this example)
name 4151121513
otherName 3113111552
anotherName 3113121532

3) Output from the program. POINTS AND PERCENTAGE PER STUDENT: **OUT_results.csv**

The output file will be one line per student, each line including the name, the total points and the percentage of those points with respect to the maximum points, separated by commas (and each line ending in "\n")

Example (continuing the previous example, the maximum is 12.5 points)
name,1.5,12
otherName,4,32
anotherName,3.5,28

As a string this would be: "name,1.5,12\notherName,4,32\nanotherName ,3.5,2\n"

E. HOW DATA IS READ FROM AND WRITTEN TO THE FILES

Python code (a function named *read_string_list_from_file(…)*) will be provided to read data from a text file, which reads a text file assuming that each line is separated from the next with a return "\n". You should incorporate this function verbatim in your code. When called, the function will return a list of strings, where each string contains the data associated on one student, in the same order that student data is placed in the data file.

Example (following the previous example), when calling the function, and providing the file name as argument, the function provided will return:

["name 4151121513", "otherName 3113111552", "anotherName 3113121532"]

The function *write_result_to_file(…)* is also provided. Check the assumptions.

F. OUTPUT TO BE SHOWN TO THE USER:

The information that your program shows to the user should be analogous to what is shown in the sample runs.

Notice how there is general information shown to the user at the beginning of the processing, during the processing, and after the processing.

STATISTICS SHOWN TO THE USER

The statistics are shown at the end. Some of the statistical results are calculated as the processing takes place. **You are highly recommended to plan which variables and structures (consider lists) you need to accomplish these calculations**. Some statistics are calculated after the processing takes place.

The "**distance between two questions**" is the sum of the distances between the two questions considering all students in the whole data file .
The distance between two questions for one student is 0 if the student responded the two questions both correctly or both incorrectly. The distance for one student is 1 if the student responded one of the two questions correctly and the other question incorrectly.
This distance is calculated considering all students' data in both cases: when processing all the students or when processing only selected students.

Other statistics should be self-explanatory from the sample runs. Ask if in doubt.

G. REQUIREMENTS IN DETAIL (anything required gets points)

Requirements – general and validation

a) The program should have a dialog and options analogous to the one presented in the sample runs
b) The results obtained by your program with the data files provided should be the same as the results shown in the sample runs. The turtle graphic component is optional, for bonus points.
c) Your program should work well with other data files as long as the formatting conventions in the data files are respected
d) The program should validate that the user types numbers when asked to do so. (e.g. questions numbers), questions numbers are valid, options ALL or SEL are typed correctly
e) The program should validate that when the user asks to calculate selected students the student name is indeed present in the student data

Requirements – coding details and style

f) Your program should have at least 5 "fruitful" or "productive" functions
g) Your program should have at least 3 "void functions"
h) Your program should have at least 5 functions receiving parameters (and so that the parameters are correctly used inside the function) (these functions may be productive or void)
i) Your program should have a reasonable main level which shows the general structure of the program. The main level can be the program top level or it can be inside a "main" function.
j) You may use some variables as global (i.e. defined at the top level and not passed as parameters). The reason to have these variables as global would be that they are frequently used by many functions. Yet, given the requirements above you will have to use variables in functions that are not global.
k) All the global variables used in the program should be initialized at the top of the code (even before the functions are defined) including a brief comment of what each variable role is. No comments are needed if the names of the variables are self-explanatory.
l) Name your variables and functions appropriately
m) At the top of the program file include as comment the authors names and dates of the versions

n) Include comments, with general descriptions of functions, special situations being true at a certain place in the program, etc. On the other hand, do not include redundant comments. For example the statement "i = i + 1" does not need the comment "i is increased by 1". Keep in mind that good naming of variables and functions reduce the need of comments.

o) Include "Trace printing" as you debug your code. When you submit your solution you may comment out some tracing prints. **However, you need to leave in your program tracing print analogous to the sample runs.**

p) Bonus points will be given if you do not break loops (with break or return statements), and rather use while statements with one or more coditions.

Requirement –clarification file (txt)

q) You need to submit an admin file (as in previous assignments)

r) Name your file **"group.txt"** when you are a group of 2 or 3 members. In this case you need to include the group members names and clarify how you distributed tasks among the team members. If you are working individually, name the file **"individual.txt"**. In this latter case the file may be empty or it may include any comments about how you worked with this exercise.

q) It would be useful for you if you keep track of the time you spend on this exercise. You are asked but not required to share this information, including the total hours dedicated, considering all the team members. If you submit this information, include it in the admin file.

Requirement – Flowchart for top level

r) Submit a flowchart describing only the main/top level possibly referring to some of the global variables. You do NOT need to do a flowchart for all the details!! You may use flowgorithm or draw the flowchart by hand and take a picture/capture the screen and submit a jpg or png file.

H. WHAT YOU ARE PROVIDED

a. This description
b. Code with two functions, *read_string_list_from_file(…)* and *write_result_to_file(…)*
c. Sample runs
d. Input data files for you to test
e. Make sure that you check email and Canvas announcements in case that additional clarifications are provided.

I . WHAT YOU NEED TO SUBMIT

a. The code (Python file) of your final submission (allowing to see the Trace printing of the data files)
b. Flowchart of your main level program
c. A sample run processing all students, a sample run processing selected students.
d. Captured screens of the associated turtle graphics (if you implemented such)
e. The admin text file (group.txt or individual.text)

If you have any questions consult with the Teaching Team.
End of description of the data processing project.