LAB4 Quiz
Assoc. Prof. Dr. Hacer YALIM KELEŞ

Date: 20/11/2020

Write a Python program which takes a sequence of inputs that contain lists of grades of some students (each student info is presented in one line) and some query items. Input data will contain the total number of students, i.e. M_STUDENTS, total lab count, i.e. LAB_COUNT, and homework count, i.e. HW_COUNT, that are provided at the beginning, and then followed by records of each student in the order of ***student id, name, surname, lab grades, homework grades, midterm grade, and final grade.*** After the student records, a set of query items, which contains a sequence of student ids, will be provided. Write a program that calculates the cumulative averages of all the grades of the students in the query items. The output format is specified below. Please also look at the sample I/O files.

Specification details:

- You can assume that there will be at least one student and one query string. Each query string will include a student id.

- Each student record will be provided in different lines.

- The number of lab/homework grades that are provided for each student can be different; some students may miss some of their assignments. Hence, the number of grades can change in the records of different students. However, each record item, such as lab, hw, etc., is separated from the others using a pair of angle brackets, i.e. $<>$. Midterm and final will be held once. If the student has not taken the exam (midterm / final / lab / homework), no grade will be provided for the relevant part between left and right angle brackets.

- If the query student id is not found in the student records, do not print anything for that query. If none of the query items are found in the records, print **None**.

- Average grades will be calculated as shown below.

$$\text{AVERAGE GRADE} = \frac{\text{LAB AVG}}{10} + \frac{\text{HW AVG}}{10} + \frac{\text{MIDTERM}}{10} + \text{FINAL} * \frac{8}{10}$$

In the equation above, LAB_AVG represents the averages of the labs, HW_AVG represents the averages of the homeworks. Note that, while computing these averages, LAB_COUNT and HW_COUNT parameters should be used, respectively.

- Print all the averages as floating point numbers using 2 digit precision after the decimal point.
  Hint: You can use format method as shown below.

  print("{:.2f}".format(average_grade))

I/O format:
input:
<M_STUDENTS >[Space]<LAB_COUNT>[Space]<HW_COUNT>[Newline]
<STUDENT_ID_1>[Space]<NAME>[Space]<SURNAME>[Space]<LEFT_BRACKET>[Space]<LAB_GRADES
>[Space]<RIGHT_BRACKET>[Space]<LEFT_BRACKET>[Space]<HW_GRADES>[Space]<RIGHT_BRACKET
>[Space]<LEFT_BRACKET>[Space]<MIDTERM>[Space]<RIGHT_BRACKET>[Space]<LEFT_BRACKET>[Spa
ce]<FINAL>[Space]<RIGHT_BRACKET>[Newline]
…
<STUDENT_ID_M>[Space]<NAME>[Space]<SURNAME>[Space]<LEFT_BRACKET>[Space]<LAB_GRADES
>[Space]<RIGHT_BRACKET>[Space]<LEFT_BRACKET>[Space]<HW_GRADES>[Space]<RIGHT_BRACKET
>[Space]<LEFT_BRACKET>[Space]<MIDTERM>[Space]<RIGHT_BRACKET>[Space]<LEFT_BRACKET>[Spa
ce]<FINAL>[Space]<RIGHT_BRACKET>[Newline]
<QUERY1>[Space] <QUERY2>[Space]…. <QUERYN>

output:
<STUDENT_NAME_1>[Space] <STUDENT_SURNAME_1>[Space]<AVERAGE_GRADE>[Newline]
…
<STUDENT_NAME_N>[Space]<STUDENT_SURNAME_N>[Space]<AVERAGE_GRADE>[Newline]

if no students are found in searched queries: None

Testing: You are provided with some sample I/O files. Assume that input.txt stands for a sample input and output.txt stands for the corresponding output file and your source code is named as Lab4.py; you can test your program from the command line using the following commands.

>>> python Lab4.py < input.txt > my_output.txt

This command redirects the inputs in the input.txt file as if they are provided from the command line to your python code. The outputs, which you generate using the print() function in your source codes, are redirected to my_output.txt file. You can then check if my_output.txt file is exactly the same with the provided output.txt file, using diff command from the command prompt:

>>> diff output.txt my_output.txt

Submission:

1. Name your Python source file as <student_id>.py; replace <student_id> using your student id number.

2. Upload your python file using the interface provided in e-kampüs course page.