Programming and Data Structures with Python 2026

Assignment 3

26 Oct 2026, due 2 Nov 2026

Hogwarts grades

The academic office at the Hogwarts School of Witchcraft and Wizardry has compiled data about students' grades. The data is provided as text from standard input in three parts: information about courses, information about students and information about grades. Each part has a specific line format, described below..

1. Information about courses

   Line format: `Course Code~Course Name~Semester~Year~Instructor`

2. Information about students

   Line format: `Roll Number~Full Name`

3. Information about grades

   Line format: `Course Code~Semester~Year~Roll Number~Grade`

The possible grades are A, AB, B, BC, C, CD, D with corresponding grade points 10, 9, 8, 7, 6, 5 and 4. The grade point average of a student is the sum of his/her grade points divided by the number of courses. For instance, if a student has taken two courses with grades A and C, the grade point average is $(10 + 7) \div 2 = 8.50$. If a student has not completed any courses, the grade point average is defined to be 0.

You may assume that the data is internally consistent. For every grade, there is a corresponding course code and roll number in the input data.

Each section of the input starts with a line containing a single keyword. The first section begins with a line containing `Courses`. The second section begins with a line containing `Students`. The third section begins with a line containing `Grades`. The end of the input is marked by a line containing `EndOfInput`.

Write a Python program to read the data as described above and print out a line listing the grade point average for each student in the following format:

`Roll Number~Full Name~Grade Point Average`

Your output should be sorted by Roll Number. The grade point average should be rounded off to 2 digits after the decimal point. Use the built-in function `round()`.

Here is a sample input and its corresponding output.

Sample Input

```
Courses
TRAN~Transfiguration~1~2011-2012~Minerva McGonagall
CHAR~Charms~1~2011-2012~Filius Flitwick
Students
SLY2301~Hannah Abbott
SLY2302~Euan Abercrombie
SLY2303~Stewart Ackerley
SLY2304~Bertram Aubrey
SLY2305~Avery
SLY2306~Malcolm Baddock
SLY2307~Marcus Belby
SLY2308~Katie Bell
SLY2309~Sirius Orion Black
Grades
TRAN~1~2011-2012~SLY2301~AB
```

```
TRAN~1~2011-2012~SLY2302~B
TRAN~1~2011-2012~SLY2303~B
TRAN~1~2011-2012~SLY2305~A
TRAN~1~2011-2012~SLY2306~BC
TRAN~1~2011-2012~SLY2308~A
TRAN~1~2011-2012~SLY2309~AB
CHAR~1~2011-2012~SLY2301~A
CHAR~1~2011-2012~SLY2302~BC
CHAR~1~2011-2012~SLY2303~B
CHAR~1~2011-2012~SLY2305~BC
CHAR~1~2011-2012~SLY2306~C
CHAR~1~2011-2012~SLY2307~B
CHAR~1~2011-2012~SLY2308~AB
EndOfInput
```

Sample Output

```
SLY2301~Hannah Abbott~9.5
SLY2302~Euan Abercrombie~7.5
SLY2303~Stewart Ackerley~8.0
SLY2304~Bertram Aubrey~0
SLY2305~Avery~8.5
SLY2306~Malcolm Baddock~6.5
SLY2307~Marcus Belby~8.0
SLY2308~Katie Bell~9.5
SLY2309~Sirius Orion Black~9.0
```

Instructions

- You should assume that the input is provided through the keyboard, so you can read the input line by line using `input()`. Similarly, you should generate your output using `print()`. You do not need to read or write files.
- Submit your solution through Moodle as a standalone Python program (i.e. a `.py` file).
- Add documentation in the form of comments to explain at a high level what your code is doing.
- A zip file is enclosed with some sample inputs and outputs to test your code. You can provide the input from a text file to your program using *input redirection*. In a terminal, type

  `python mysolution.py < inputfile.txt`

  to feed the contents of `inputfile.txt` to `mysolution.py`.
