APS106 – Lab #3

Preamble

This week you will practice manipulating strings by writing four separate functions to extract experimental information from strings.

By completing this lab, you will develop, utilize, and practice the following skills:

- *setup, debug, and run a program*
- *interpret real world parameters as variables*
- *utilize string methods*
- *convert between variable types*
- *writing and calling functions*
- *develop effective test cases*

Use appropriate variable names and place comments throughout your program.

The name of the source file must be "lab3.py".

Deliverables

For this lab, you must submit four functions within a single file named 'lab3.py' to MarkUs by the posted deadline. The functions to be included are:

1. `email_to_name`
2. `count_measurements`
3. `calc_control_average`
4. `generate_summary`

Details and requirements of each of these functions are provided within the document.

Five test cases are provided on MarkUs to help you prepare your solution. **Passing all these test cases does not guarantee your code is correct.** You will need to develop your own test cases to verify your solution works correctly. Your programs will be graded using ten secret test cases. These test cases will be released after the assignment deadline.

IMPORTANT:

- Do not change the file name or function names
- Do not use input() inside your program

Problem

For this week, we will imagine you are an engineer working on a treatment process for wastewater from a mine. To evaluate whether your design is effectively desalinating the wastewater, you hire several technicians to measure the pH of treated water at multiple locations around the mine every day.

Everyday, the measurements from each technician are recorded and uploaded to a database as a **string** with the following format:

Technician email, date, site name, pH measurement, site name, pH measurement, site name, pH measurement, ...

Where:

- technician email is the email of the technician
- date is the day the measurements were taken
- site name is the name of the site where a measurement was taken
- pH measurement is the measured pH at that site. Note that there are multiple site-measurement pairs. The number of measurements taken by technicians can vary each day.

Example strings are given below:

```
dina.dominguez@company.com, 01/11/20, A, 4.2, B, 6.7, Control,
7.1, B, 6.5, Control, 7.8, Control, 6.8, A, 3.9,
```

```
jamie.riggs@company.com, 14/12/20, B, 5.6, Control, 5.5, Db,
3.2, Control, 6.1, Control, 5.9,
```

The first string contains measurements by Dina Dominguez from Nov. 1, 2020. There are a total of seven measurements from three sites (A, B, and Control). The second string contains measurements by Jamie Riggs from Dec. 14, 2020. There are a total of five measurements from three sites (B, Db, and Control).

You are interested in extracting the following information from each string:

1. The name of the technician
2. The total number of measurements contained within the string
3. The average pH measurement at the Control site

After a few weeks, there are hundreds of measurements being added to the database every day and you are finding that manually analyzing and extracting information is very inefficient. Luckily, you have programming skills from APS106 that will help you efficiently extract the information you need. So, you decide to write a program to extract the information you need from each string and generate new stings that can be analyzed quickly.

For this week's lab, you will write a program with four functions: email_to_name, count_measurements, calc_control_average, and generate_summary. Instructions for each of **these** functions will be given in the subsequent sections.

Part 1 – Convert Email to Full Name

For the first part of this lab, you will complete the function `email_to_name.` This function accepts a string input parameter, `email,` and returns a string containing the last and first name in all uppercase letters and separated by a comma. You may assume that all input strings will be formatted as 'firstname.lastname@domain.com'.

Sample Test Cases

Input	Output
`ada.lovelace@emaildomain.com`	`LOVELACE,ADA`
`guido.vanrossum@companyX.com`	`VANROSSUM,GUIDO`

Part 2 – Count the Total Number of Measurements

For the second part of this lab, complete the function `count_measurements` which accepts a string input parameter containing comma separated site-measurement pairs and returns the total number of measurements contained within the string. You may assume the string contains at least one measurement.

Hint: There is a mathematical relationship between the number of commas in the string and the number of measurements.

Sample Test Cases

Input	Output	Description
`B, 5.6, Control, 5.5, Db, 3.2,`	`3`	Multiple measurements
`Control, 7.5,`	`1`	Single measurement

Part 3 – Calculate the Average pH at the Control Site

For the third part of this lab, complete the function `calc_control_average` which accepts a string input parameter containing comma separated site-measurement pairs and returns the average pH measurement at the control site. **Each input string will contain exactly three measurements from the control site** (i.e. the site named `Control`**).** You may further assume each measurement will contain exactly one decimal point of precision. You may also assume that the last measurement in the string will be followed by a comma. The average should be returned as a float **rounded to one decimal place**.

Hint: You will need to slice the input string to extract the information you need. You may find the `find` method useful for determining the slice indices.

Sample Test Cases

Input	Output	Description
`Control, 7.4, Control, 7.2, Control, 7.6,`	`7.4`	All three measurements are from control site, all pH measurements < 10
`Control,  10.2, Control, 11.2, Control,9.6,`	`10.3`	All three measurements are from control site, some pH measurements > 10
`A, 4.2, B, 6.7, Control, 7.1, B, 6.5, Control, 7.8, Control, 6.8, A, 3.9,`	`7.2`	Not all measurements from control site

Note: There is no guarantee regarding the number of spaces following each comma. There may be zero, one, or multiple spaces following any comma in the string.

Part 4 – Put Everything Together

For the final part of this lab, complete the function `generate_summary.` This function accepts a string input with the following format:

Technician email, date, site name, pH measurement, site name, pH measurement, site name, pH measurement, …

and **returns a string** containing the technician name, number of measurements, and average of the control site pH measurements:

Last name (uppercase), first name (uppercase), total number of measurements, average of control site pH measurements

There should be no spaces before or after the commas. **This function should call the other functions you created in parts 1-3**. As with part 3, each string will contain exactly three measurements from the control site and the average pH at the control site should be rounded to one decimal place.

Sample Test Cases

Inputs	Output
`dina.dominguez@company.com, 01/11/20, A, 4.2, B, 6.7, Control, 7.1, B, 6.5, Control, 7.8, Control, 6.8, A, 3.9,`	`DOMINGUEZ,DINA,7,7.2`
`jamie.riggs@company.com, 14/12/20, B, 5.6, Control, 5.5, Db, 3.2, Control, 6.1, Control, 5.9,`	`RIGGS,JAMIE,5,5.8`

Note: No test cases for this function will be provided on MarkUS until after grading is complete. You are responsible for testing this function.