1. **(25 Points)** Write a complete, working Python 3 program called euidA.py (where “euid” is your EUID) that does the following:

* Write your EUID in comments at the top of the file.
* Prompt for and read in a single word string to a variable called word.
* Create a set called my\_set from word.
* Since a set omits duplicates, compare the lengths of my\_set and word. If the lengths are the same, print out a message that word contains no duplicate letters. Otherwise, print out a message that word contains duplicate letters.
* The lowercase letters a through z can be represented by their equivalent ASCII values 97 through 122, respectively. Generate a random number between 97 and 122, inclusively, and assign the result to the variable num.
* Convert num to its one-character string equivalent (i.e., the lowercase letter) and assign the result to the variable called mystery.
* If mystery is in my\_set, remove mystery from my\_set. Otherwise, if mystery is not in my\_set, add mystery to my\_set.
* Create a new set called new\_set using the literal values for the vowels of the alphabet (i.e., 'a', 'e', 'i', 'o', and 'u')
* Find and then print the set difference from my\_set and new\_set.
* You may assume the user enters all elements using the appropriate data type. Due to time constraints, no further comments are required.

Here is a sample output to help you write the code. The items in bold are entered by the user.

$ **python3 mat0299A.py**

Enter a single word: **surprise**

surprise contains duplicate letters

{'s', 'd', 'r', 'p'}

$ **python3 mat0299A.py**

Enter a single word: **value**

value contains no duplicate letters

{'k', 'v', 'l'}

1. **(25 Points)** Write a complete, working Python 3 program called euidB.py (where “euid” is your EUID) that does the following:

* Write your EUID in comments at the top of the file.
* Create a stock prices dictionary called stock\_prices that have the following names of stocks and current prices:
  + ABC 5.17
  + XYZ 2.38
  + PQR 7.84

where the names of the stock are strings and the current prices are floating-point numbers.

* Prompt for and read in the name of a stock to delete from stock\_prices.
* If the name of the stock is found in the stock\_prices dictionary, delete that stock from the dictionary.
* Prompt for and read in the name of a stock to update its price.
* If the name of the stock is found in the stock\_prices dictionary:
  + Retrieve the current price of the stock from the stock\_prices dictionary and assign the result to the variable called current\_price.
  + Compute the loss in value of the stock to be the square root of the current\_price and assign the result to the variable called loss.
  + Update the price of the stock in the stock\_prices dictionary to be the difference of current\_price and loss.
* Otherwise, if the name of the stock was not found in the stock\_prices dictionary:
  + Prompt for and read in the price of the new stock as a floating-point number.
  + Add the new stock and price to the stock\_prices dictionary.
* Print the stock\_prices dictionary (both keys and values). Do not worry about formatting the price to two decimal places.
* You may assume the user enters all elements using the appropriate data type. Due to time constraints, no further comments are required.

Here is a sample output to help you write the code. The items in bold are entered by the user.

$ **python3 mat0299B.py**

Enter name of stock to delete: **XYZ**

Enter name of stock to update price: **ABC**

{'ABC': 2.8962365998195856, 'PQR': 7.84}

$ **python3 mat0299B.py**

Enter name of stock to delete: **GHI**

Enter name of stock to update price: **PQR**

{'ABC': 5.17, 'XYZ': 2.38, 'PQR': 5.04}