|  |  |
| --- | --- |
| Name: | Edward Eisenberger |
| ID#: | 1066164 |
| Assignment 1 | |
| Due Date: | Feb. 6, 2019 |
| Date of Submission: | Feb. 5, 2019 |

Contents

[Overview 3](#_Toc206624)

[Problem 1 3](#_Toc206625)

[Summary 3](#_Toc206626)

[Input 3](#_Toc206627)

[Output 3](#_Toc206628)

[Screenshot 4](#_Toc206629)

[Problem 2 4](#_Toc206630)

[Summary 4](#_Toc206631)

[Input 5](#_Toc206632)

[Output 5](#_Toc206633)

[Problem 3 6](#_Toc206634)

[Part a 6](#_Toc206635)

[Part b 7](#_Toc206636)

# Overview

Assignment 1 consists of three (3) problems outlined in the following sections. A single Visual Studio solution containing several Python modules was created to solve each problem. The main module, Assignment1.py, contains the main routine which calls

# Problem 1

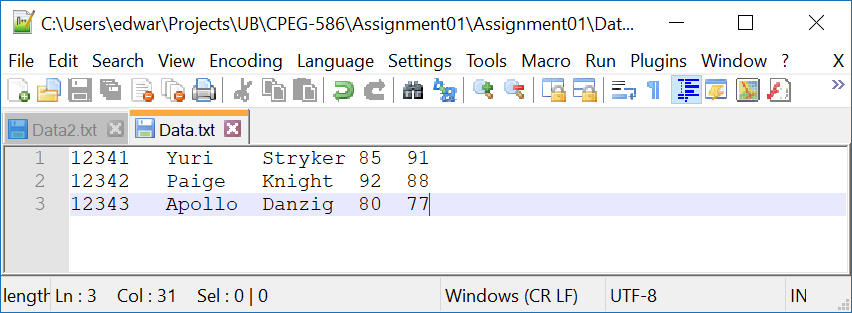
## Summary

Problem 1 provided introductory lessons to the Python programming language. A python module was created which:

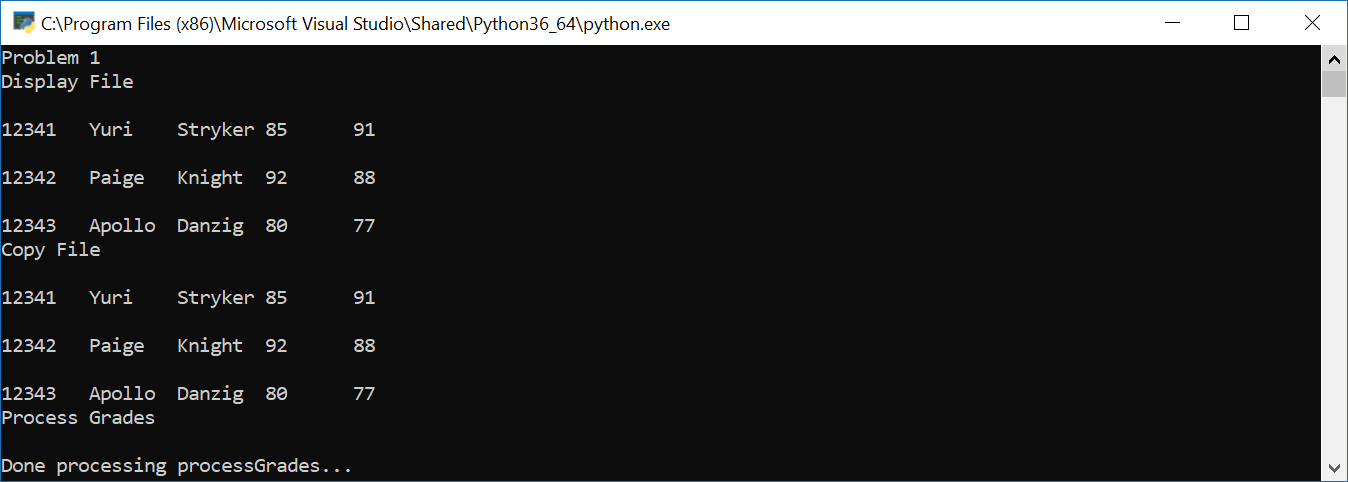
1. Reads students and their corresponding grades from a text file
2. Creates a new student and grades in the list
3. Writes the updated list to a file

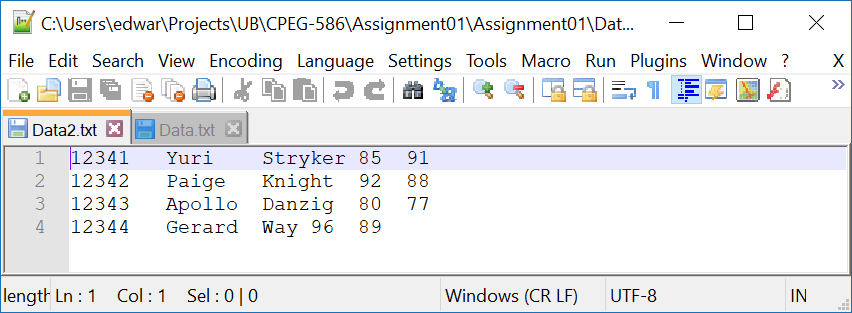
The file read into the module is shown in the Input section and the results of the module are shown in the Output section.

## Input

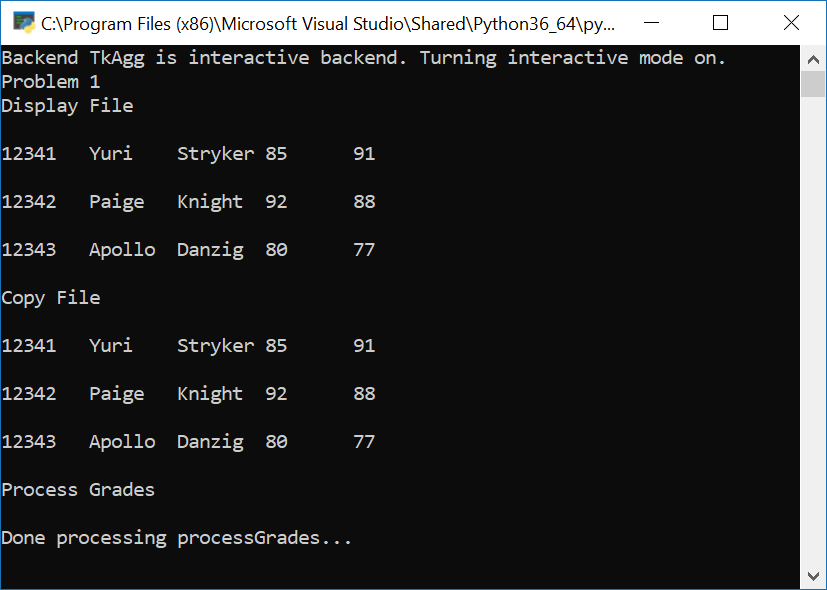


## Output





## Screenshot



# Problem 2

## Summary

Problem 2 required implementing a Python module which uses Linear Least Squares Optimization to estimate the model parameters of a line to fit the given data. The following Input section details the data used for the Linear Least Squares Optimization python module. The following Output displays the results.

## Input

The following input was hard coded into the Python Module.

|  |  |
| --- | --- |
| X | Y |
| 1 | 3.2 |
| 2 | 6.4 |
| 3 | 10.5 |
| 4 | 17.7 |
| 5 | 28.1 |
| 6 | 38.5 |

## Output

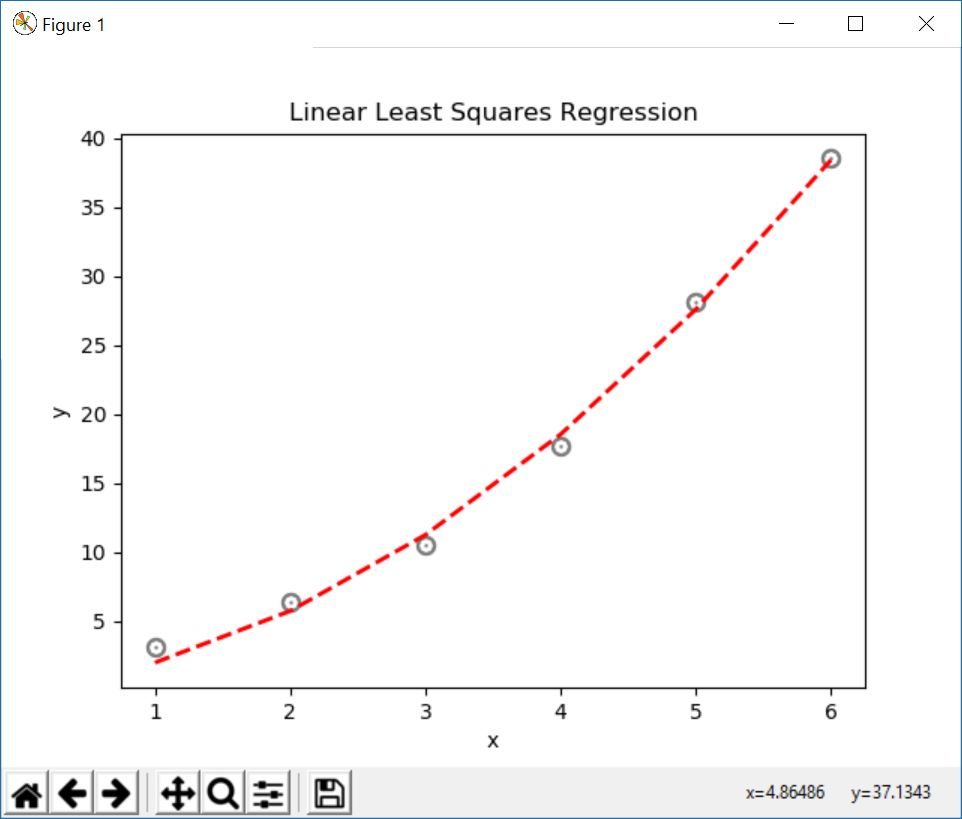


Figure - Linear Least Squares Regression Curve

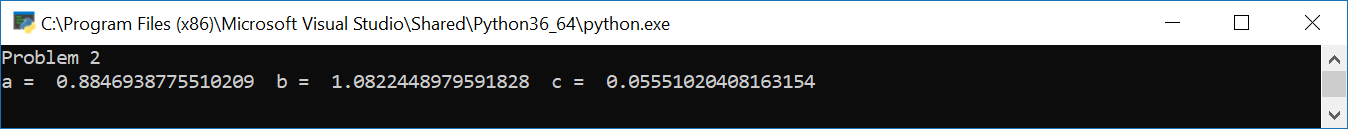


Figure - Linear Least Squares Regression Coefficients

# Problem 3

## Part a

For the single neuron network described in lecture 1, show derivations for computing gradients w1, w2 and b. Assume the network will be classifying using the decision boundary given by:  
 ­

## Part b

A python class, Neuron2.py, was implemented to represent a single neuron model which takes two (2) inputs. The gradient equations developed in Part a are used to train the Neuron using the following data for 100000 Epochs.

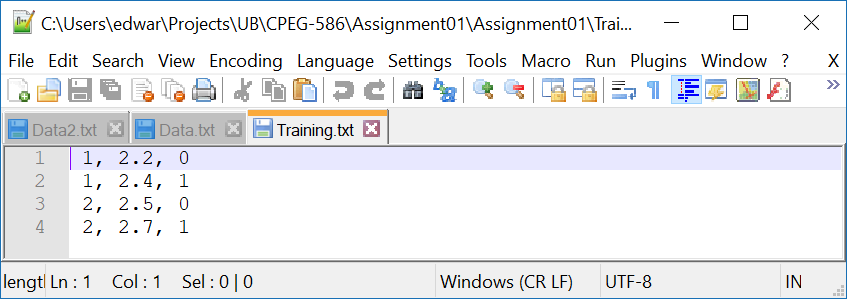


Figure - Training Data

The Neuron is then tested with the following test data:

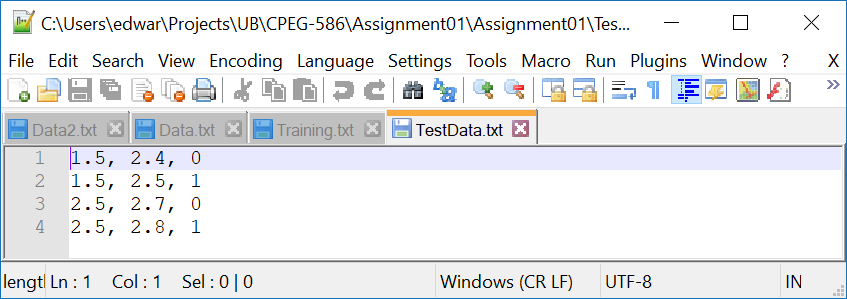


Figure - Test Data

The results of the training and test data are:

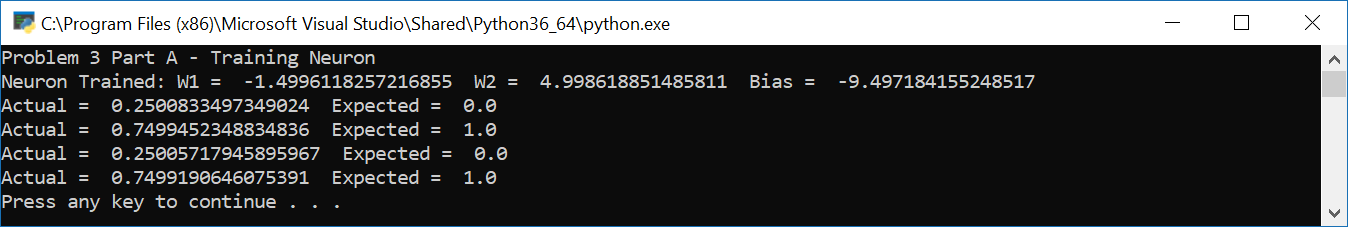


Figure - Neuron Results