**ENGR 102 Sect 508 Lab 11b**

**100 points**

**Reading assignment:**

|  |  |
| --- | --- |
| **Lecture Slides** | **L11** |
| **zyBook chapter 11** | **Complete all participation and challenge activities** |

*Attention!!*

*Individual submission.*

*Submit* *your Py-files together with your word/pdf file with screenshots of your tests outputs. Include any derivations, comments and supplemental notes in your word/pdf files.*

*No pictures by the phone – it is impossible to read. You will be allowed to resubmit and reupload HW as many times as you want to within the due date/time, only last submission will be graded. No late submissions. For submission you may use this file as template: rename file including your name. Do not forget to put your name inside of this file as well. If it is a team work use Team Header, include the team number and all team members.*

**Activity #1: Programs to test writing functions – to be done individually.**

This activity is meant to give you more experience writing functions.

In all cases, the key idea is to write a single function. However, in all cases, **you should create a program that will essentially “test” the function**. For example, you might include several function calls and the results, or create a program where a user can enter values and see the results. You may feel free to make multiple functions within any one of these programs.

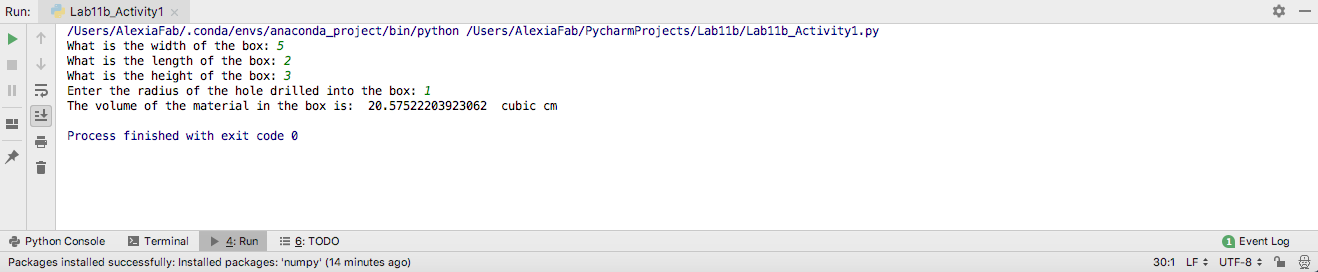
1. **[25 points]** Imagine that you have a block of material in which a hole has been drilled:

Write a function that will take in the dimensions of the box, length, width, and height, and the radius of the hole, and determine the volume of material remaining. Assume the hole has been drilled along the height direction. Note: first write the function assuming the hole has radius less than min(length/2, width/2) – you will still receive a majority of credit (more than 70%) for this result. For full credit, you will need to account for larger radii.

Code:

*# By submitting this assignment, I agree to the following:  
# “Aggies do not lie, cheat, or steal, or tolerate those who do”  
# “I have not given or received any unauthorized aid on this assignment”  
#  
# Name: ALEXIA PEREZ  
# Section: 508  
# Assignment: LAB 11B  
# Date: 09-11-2018***import** numpy  
**from** math **import** \*  
  
*# In this program we will calculate the volume of material inside a cube that has a hole drilled into it.  
  
# Inputs:*width = float(input(**'What is the width of the box: '**))  
length = float(input(**'What is the length of the box: '**))  
height = float(input(**'What is the height of the box: '**))  
r = float(input(**'Enter the radius of the hole drilled into the box: '**))  
  
*# Operations:***def** volume\_material(r):  
 volume\_box = width\*length\*height  
 volume\_hole = pi\*r\*\*2\*height  
 material = volume\_box-volume\_hole  
 **return** material  
  
  
*# Output (we assume the measurements are in cm):*print(**'The volume of the material in the box is: '**,volume\_material(r),**' cubic cm'**)

Output:

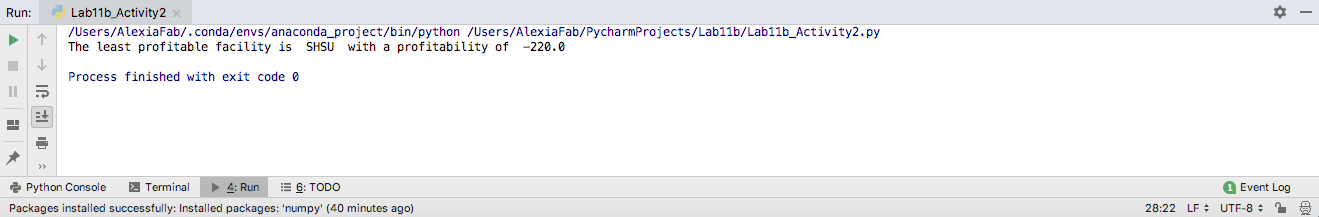


1. **[15 points]**Imagine that you have three parallel lists of the same length, one with the names of several production facilities, another with the annual cost to operate each of those facilities, and a third with the value of the products produced at each facility. Return the name and net profitability (profitability is the value of what’s produced minus the cost to operate) of the least profitable facility.

**Code:**

*# By submitting this assignment, I agree to the following:  
# “Aggies do not lie, cheat, or steal, or tolerate those who do”  
# “I have not given or received any unauthorized aid on this assignment”  
#  
# Name: ALEXIA PEREZ  
# Section: 508  
# Assignment: LAB 11B  
# Date: 09-11-2018***import** numpy  
**from** math **import** \*  
  
*# In this program we will calculate the profitability of a list of production facilities  
# and return the least productive facility's name and profitability:  
# (input isn't specified to be required in the doc file so we're just going to use pre-written lists)  
  
# Lists (I decided to name the production facilies as some of the biggest Texas Universities just because.)*facilities = [**'TAMU'**,**'UTA'**,**'SHSU'**,**'TXST'**,**'TTU'**]  
cost = [850,1020,420,200,580]  
value = [1020,850,200,420,580]  
  
*# Operations:  
  
# To calculate the profitability we need to calculate value-cost:***def** profitability(x):  
 **return** float(value[x])-float(cost[x])  
  
  
**for** i **in** range(len(facilities)-1):  
 **if** profitability(i)<profitability(i+1):  
 least\_profitable = facilities[i]  
 x = i  
  
print(**'The least profitable facility is '**,least\_profitable,**' with a profitability of '**,profitability(x))

**Output:**



1. **[15 points]**Write a function that takes as input a person’s name, city, state, zip code, and address, where the address is either one string (one line) or two strings (two lines), and prints the person’s information like a mailing label. Show that the routine works regardless of whether it is called with one address line or two address lines.

Code:

*# By submitting this assignment, I agree to the following:  
# “Aggies do not lie, cheat, or steal, or tolerate those who do”  
# “I have not given or received any unauthorized aid on this assignment”  
#  
# Name: ALEXIA PEREZ  
# Section: 508  
# Assignment: LAB 11B  
# Date: 09-11-2018***import** numpy **as** np  
**from** math **import** \*  
  
*# In this program we will print a person's information in mailing label format:  
  
# Input:*info = [input(**'Enter name: '**),input(**'Enter address: '**),  
 input(**'Enter address line 2: '**),  
 input(**'Enter city: '**),input(**'Enter State: '**)]  
  
*# Printing operations:***def** mailing\_label():  
 line = str(info[1]) + str(info[2])  
 info.pop(1)  
 info.pop(1)  
 info.insert(1, line)  
 **for** i **in** info:  
 print(i, sep=**' '**)  
 **return**mailing\_label()

Output:

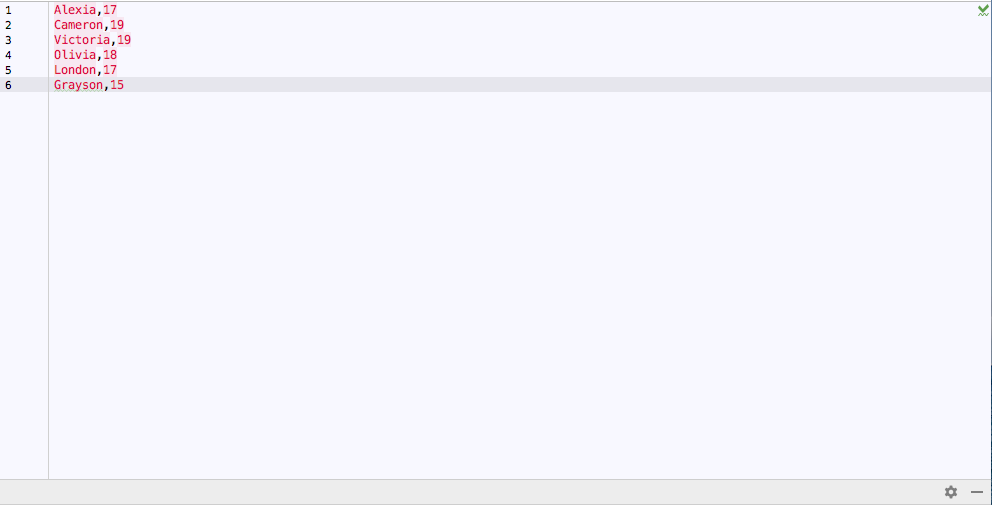


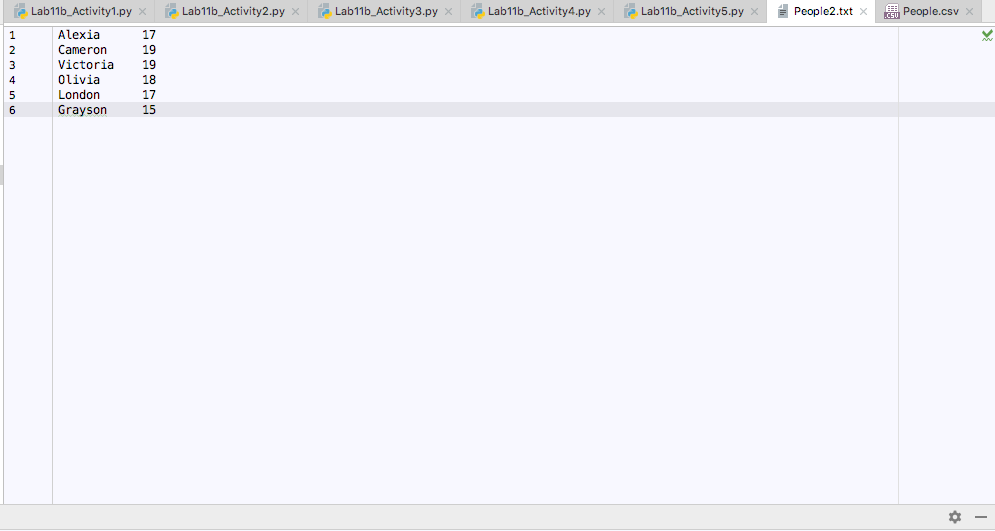
1. **[15 points]**Write a function that takes the name of a file with a .csv extension (a comma-separated value file), and writes a new file that is equivalent (same name and same data) but with a .tsv extension (a tab-separated value file: like a CSV but with tabs instead of commas separating the elements of the file). Note: the character used to represent a tab is ‘\t’.

Code:

*# By submitting this assignment, I agree to the following:  
# “Aggies do not lie, cheat, or steal, or tolerate those who do”  
# “I have not given or received any unauthorized aid on this assignment”  
#  
# Name: ALEXIA PEREZ  
# Section: 508  
# Assignment: LAB 11B  
# Date: 09-11-2018***import** numpy **as** np  
**from** math **import \*  
import** re  
  
**def fconv**()**:  
 with** open(**"People.csv"**, **'r'**) **as** myfile**:  
 with** open(**"People2.txt"**, **'w'**) **as** file**:  
 for** line **in** myfile**:** fileContent **=** re.sub(**","**, **"**\t**"**, line)  
 file.write(fileContent)  
  
fconv()

Output:



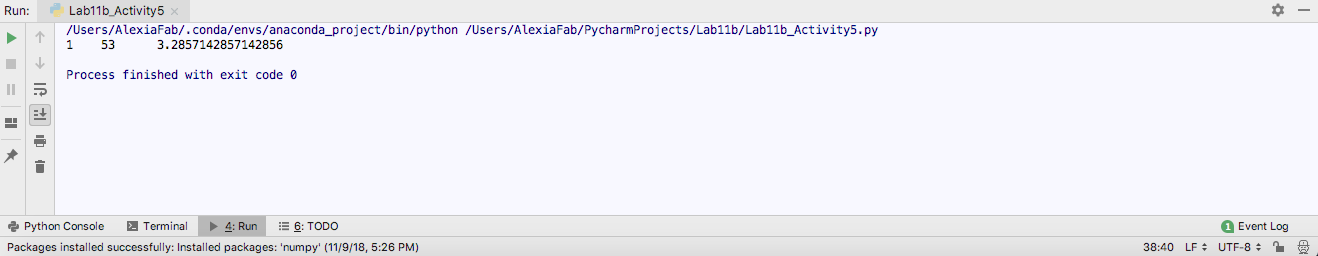


1. **[15 points]**Write a single function that takes in a list and returns the minimum, mean, and maximum value from the list.

Code:

*# By submitting this assignment, I agree to the following:  
# “Aggies do not lie, cheat, or steal, or tolerate those who do”  
# “I have not given or received any unauthorized aid on this assignment”  
#  
# Name: ALEXIA PEREZ  
# Section: 508  
# Assignment: LAB 11B  
# Date: 09-11-2018***import** numpy **as** np  
**from** math **import \***numbers **=** [1,12,2,53,23,6,17]  
  
  
**def min\_function**(*numbers*)**:** min\_value **= None  
 for** value **in** *numbers***:  
 if not** min\_value**:** min\_value **=** value  
 **elif** value **<** min\_value**:** min\_value **=** value  
 **return** min\_value  
  
  
**def max\_function**(*numbers*)**:** max\_value **= None  
 for** value **in** *numbers***:  
 if not** max\_value**:** max\_value **=** value  
 **elif** value **>** max\_value**:** max\_value **=** value  
 **return** max\_value  
  
  
**def mean\_function** (*numbers*)**:  
 for** i **in** range (len(*numbers*)**-**1)**:** total **=** *numbers*[i]**+***numbers*[i**+**1]  
 mean **=** total **/** len(*numbers*)  
 **return** mean  
  
  
print(min\_function(numbers),**'**\t**'**,max\_function(numbers),**'**\t**'**,mean\_function(numbers))

Output:

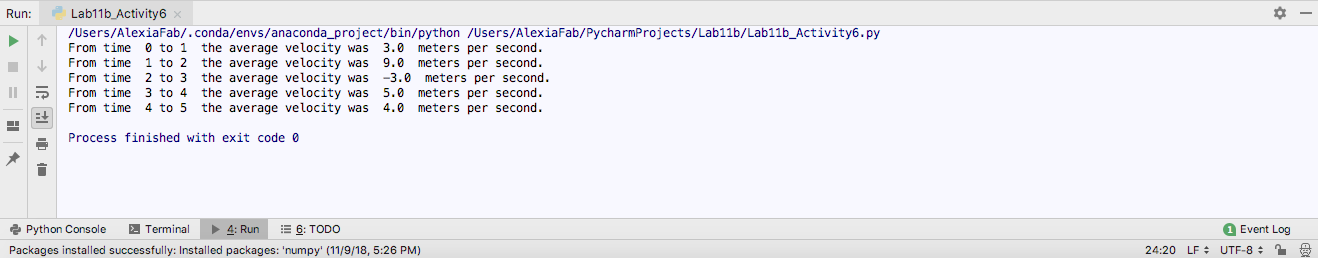


1. **[15 points]**Write a function that takes in two parallel lists: a list of times (in increasing order), and a list of distance traveled by that point in time. The function should return a new list giving the average velocity between consecutive time measurements. The new list should have length one less than the original lists.

Code:

*# By submitting this assignment, I agree to the following:  
# “Aggies do not lie, cheat, or steal, or tolerate those who do”  
# “I have not given or received any unauthorized aid on this assignment”  
#  
# Name: ALEXIA PEREZ  
# Section: 508  
# Assignment: LAB 11B  
# Date: 09-11-2018***import** numpy **as** np  
**from** math **import \***times **=** [0,1,2,3,4,5]  
distance **=** [0,3,12,9,14,18]  
  
  
**def average\_velocity** ()**:  
 for** i **in** range (len(distance)**-**1)**:** time\_interval **=** str(times[i]) **+ ' to ' +** str(times[i **+** 1])  
 avg\_velocity **=** (int(distance[i**+**1])**-**int(distance[i]))**/**(int(times[i**+**1])**-**int(times[i]))  
 print(**'From time '**,time\_interval,**' the average velocity was '**,avg\_velocity,**' meters per second.'**)  
  
  
average\_velocity()

Output:

**