**Homework 3**

**Jamie Andrews**

**Problem 1:**

#--------------Homework 3 Question 1 by Jamie Andrews-----------

#1. Define the input tuple and call in 3 different arguments.

#2. Call in 5 different variables and assign it to the a.split(separator).

#3. Write and if/else statement to determine whether the fulltime is true or not.

#4. Call a variable called mytup[i] and assign to the 5 other variables.

#5. Create a for loop in which takes the tuples that are in range of the 5 variables.

#6. Set mytup equal to the tuple of mytup and return it.

#7. Create a def function that is supposed to take the input of the tuples entered into the program.

#8. Repeat steps 2-4, however the variable should be called 'raw' when assigning it to the other 5 variables.

#9. Set mylist to be assigned to the range of raw.

#10. Create a def function that reads the tuples into the program.

#11. Create a for loop in which handles the inputs into the program.

#12. Write a try/except statement in which the program will read the file and see if it equals

# to the tuple def function (the first one), if it does not then it will print 'adsf' and it will break.

#13. Outside of the 3 def functions, call file object and have it assigned to open a txt file.

#14. Assign type to the 5 variables.

#15. Assign separator to be a ','.

#16. Assign myfile to third def function and print myfile.

#1.

def input\_tuple(a, type, separator):

#2.

firstname, lastname, age, id, fulltime = a.split(separator)

#3.

if fulltime=='true' or fulltime=='1':

fulltime = '1'

else:

fulltime = '0'

#4.

mytup = [firstname, lastname, age, id, bool(eval(fulltime))]

#5.

for i in range(len(mytup)):

mytup[i] = type[i](mytup[i])

#6.

mytup = tuple(mytup)

return mytup

#7.

def input\_tuple\_lc(a, type, separator):

#8.

firstname, lastname, age, ID, fulltime = a.split(separator)

if fulltime=='true' or fulltime=='1':

fulltime = '1'

else:

fulltime = '0'

raw = [firstname, lastname, age, ID, bool(eval(fulltime))]

#9.

mylist = [type[i](raw[i]) for i in range(len(raw))]

return mylist

#10.

def read\_tuple(file\_obj,type, separator):

#11.

for line in file\_obj:

#12.

try:

myfile = input\_tuple(line,type,separator)

print(myfile)

except 'empty' in line:

print("adsf")

break

#13.

file\_obj = open('textfile.txt','r')

#14.

type = [str, str, float, int, bool]

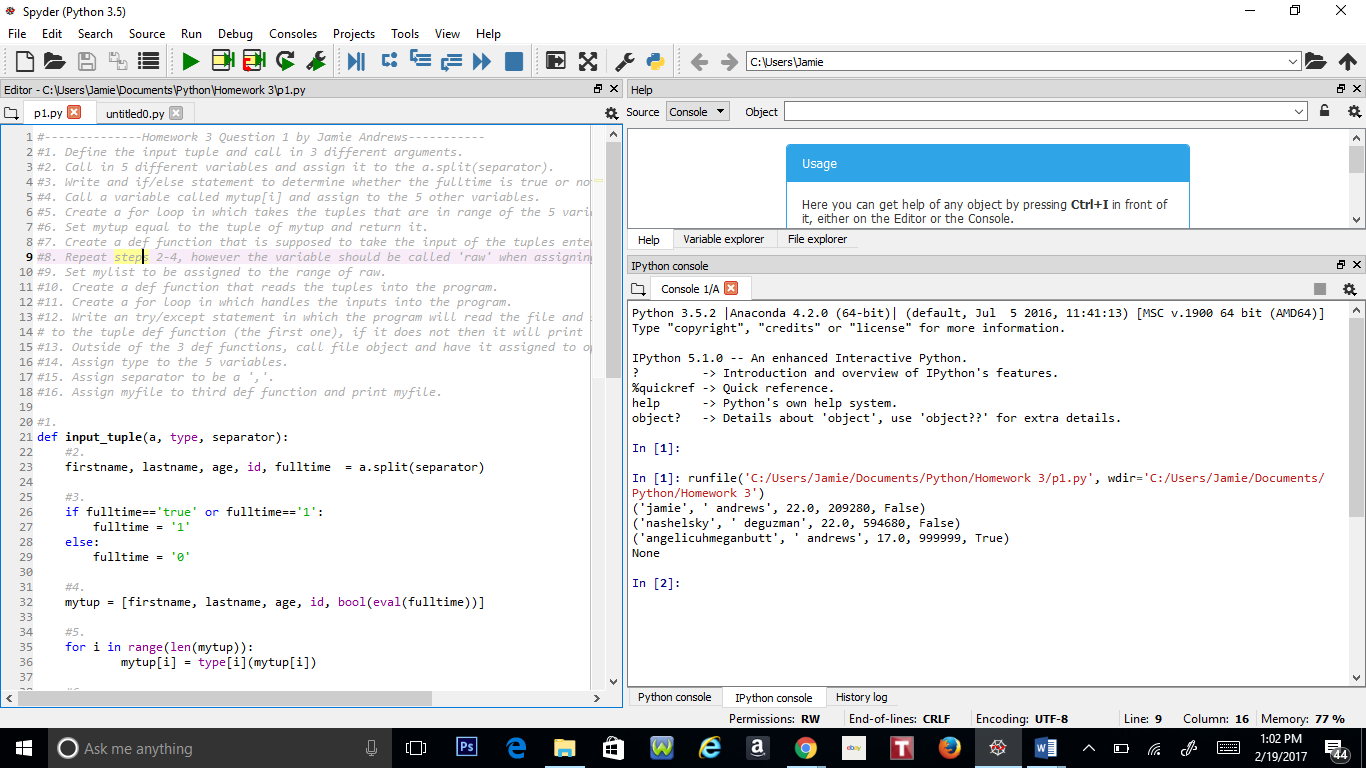
#15.

separator = ','

#16.

myfile = read\_tuple(file\_obj,type,separator)

print(myfile)



**Problem 2:**

#--------------Homework 3 Question 2 by Jamie Andrews----------

#1. Import the math library.

#2. Create a def function called compute\_pythagoreans which is suposed to calculate

# the values in the Pythagorean Theorem.

#3. Assign list to be a set of i and j values that are in range of n values and the i and j values squared

# should be less than or equal to n values squared and return the list.

#4. Outside of the def function, assign n to be the integer of inputs to input values into the program.

#5. Print the def function.

#1.

import math

#2.

def compute\_pythagoreans(n):

#3.

list = [(i,j) for i in range(n) for j in range(n) if i\*\*2 + j\*\*2 <= n\*\*2]

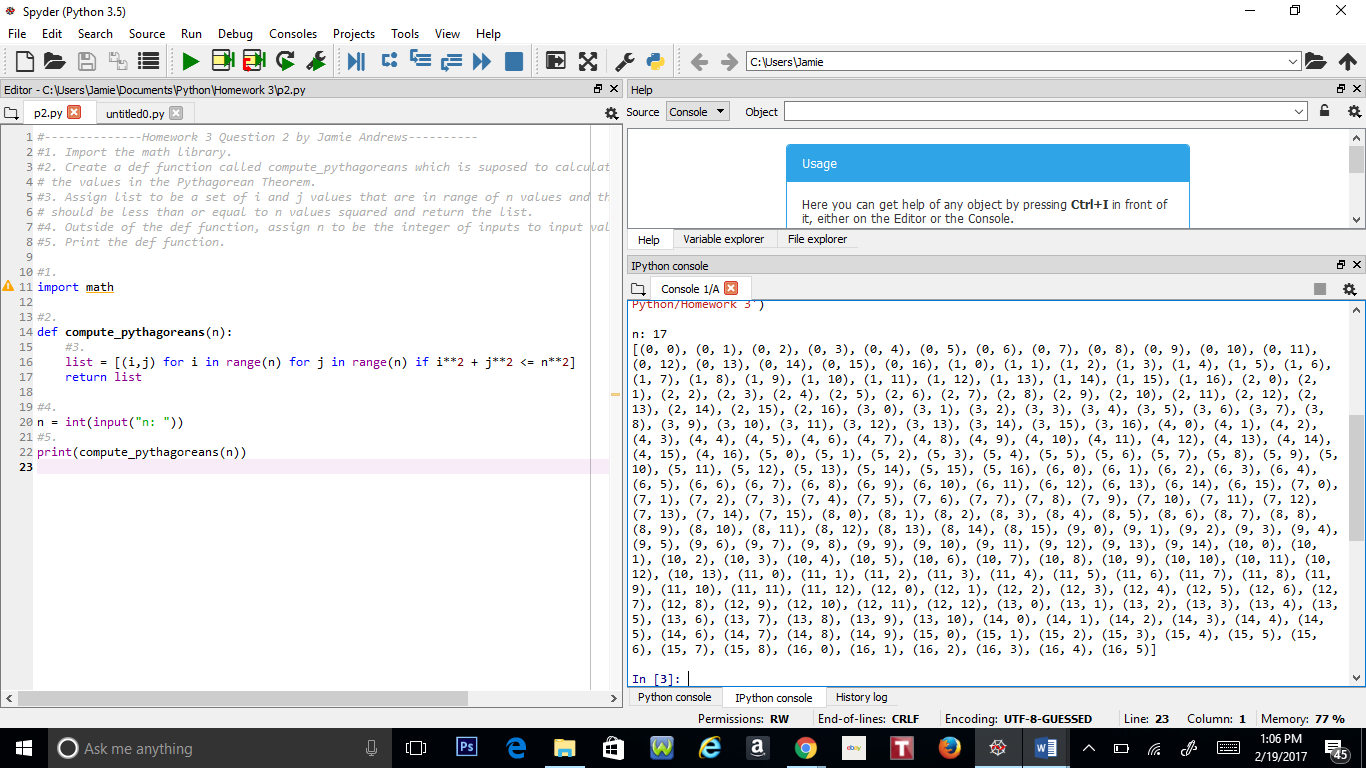
return list

#4.

n = int(input("n: "))

#5.

print(compute\_pythagoreans(n))



**Problem 3:**

#----------Homework 3 Question 3 by Jamie Andrews ------------

#1. Create def function that collects the data from the string

#2. Write a for loop that splits the string.

#3. Create another for loop that prints the string in range of the def functions length.

#4. Declare a variable that opens the csv file in the program.

#5. Assign argument string\_pos\_1st to array [0,2,3,4] so that it only prints out the 1, 3, 4, and 5 set of strings

#6. Assign sep to ",".

#7. Assign mydata to the def function.

#1.

def get\_csv\_data(f,string\_pos\_1st,sep):

#2.

for line in f:

myline = [line.split(sep) for i, line in enumerate(f)]

#3.

for i in range(len(string\_pos\_1st)):

print(myline[string\_pos\_1st[i]])

#4.

f = open("lb-james.csv", "r")

#5.

string\_pos\_1st = [0,2,3,4]

#6.

sep= ","

#7.

mydata = get\_csv\_data(f,string\_pos\_1st,sep)

