

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 an 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]
```

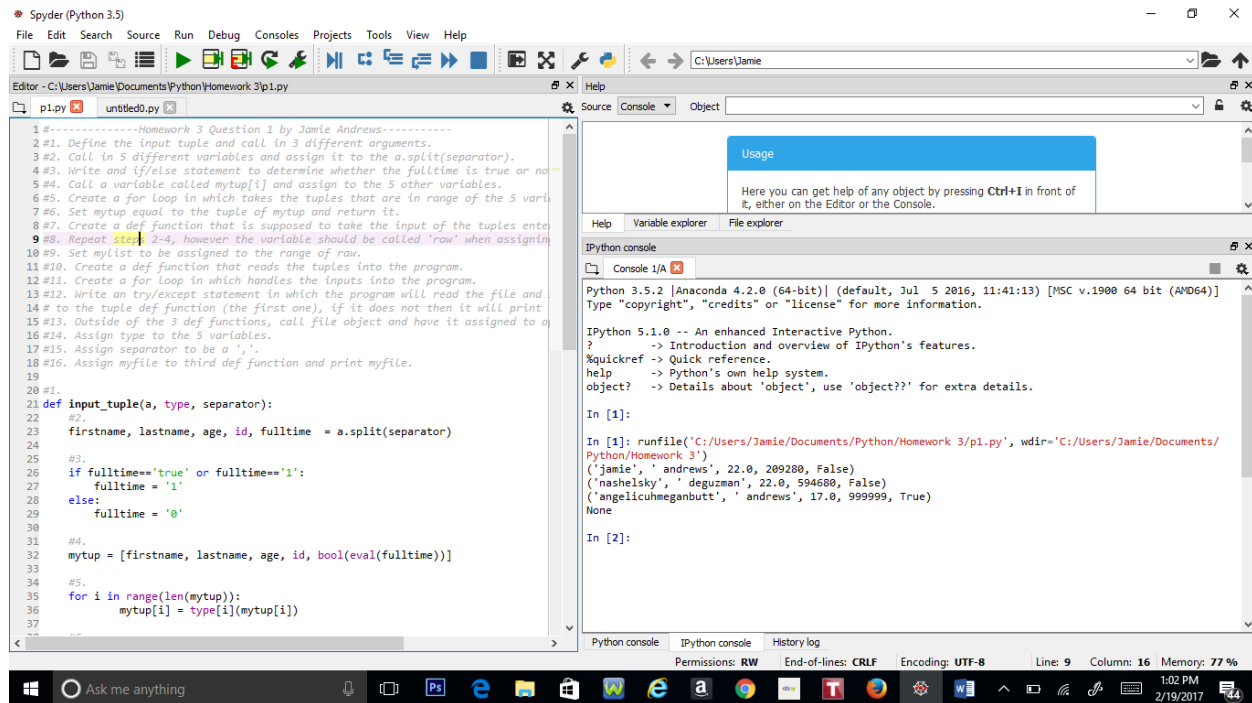
#15.

```
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 supposed 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]
```

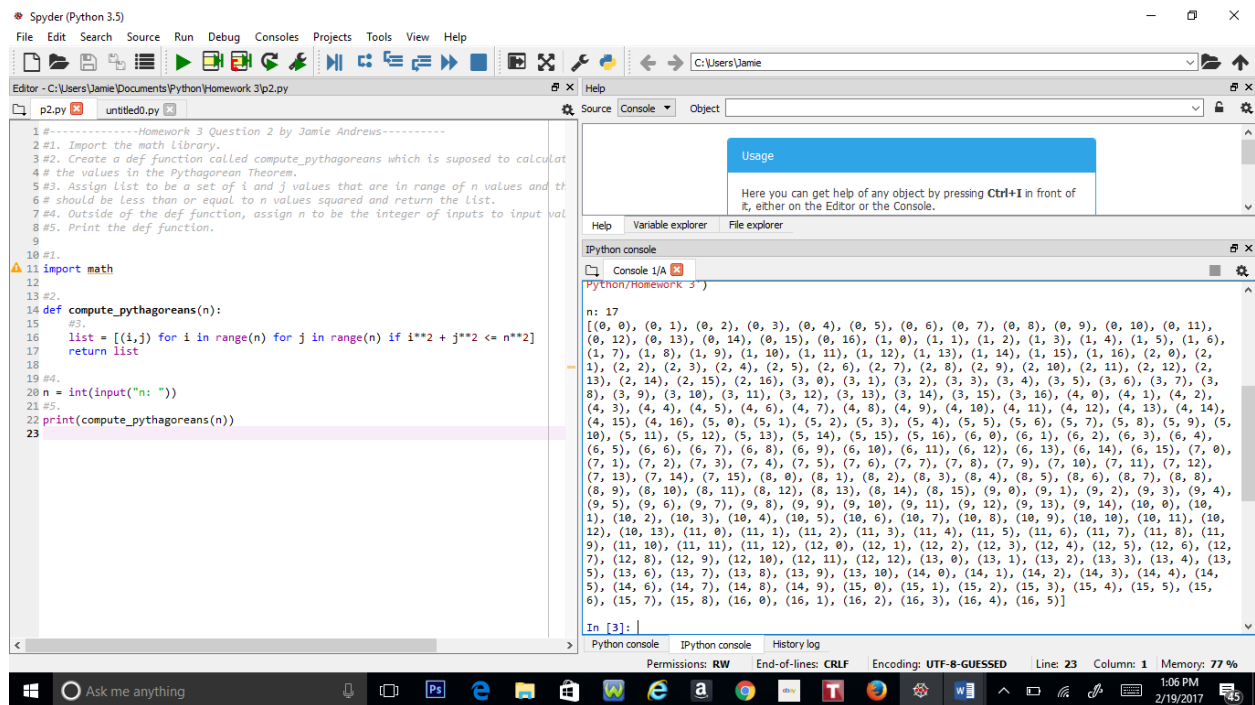
```
return list
```

#4.

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

#5.

```
print(compute_pythagoreans(n))
```



```
1 #-----Homework 3 Question 2 by Jamie Andrews-----
2 #1. Import the math library.
3 #2. Create a def function called compute_pythagoreans which is supposed to calculate
4 # the values in the Pythagorean Theorem.
5 #3. Assign list to be a set of i and j values that are in range of n values and that
6 # should be less than or equal to n values squared and return the list.
7 #4. Outside of the def function, assign n to be the integer of inputs to input value.
8 #5. Print the def function.
9
10 #1.
11 import math
12
13 #2.
14 def compute_pythagoreans(n):
15     #3.
16     list = [(i,j) for i in range(n) for j in range(n) if i**2 + j**2 <= n**2]
17     return list
18
19 #4.
20 n = int(input("n: "))
21 #5.
22 print(compute_pythagoreans(n))
23
```

Console 1/A

```
Python/Homework 3
n: 17
[(0, 0), (0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (0, 7), (0, 8), (0, 9), (0, 10), (0, 11),
(0, 12), (0, 13), (0, 14), (0, 15), (0, 16), (1, 0), (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6),
(1, 7), (1, 8), (1, 9), (1, 10), (1, 11), (1, 12), (1, 13), (1, 14), (1, 15), (1, 16), (2, 0), (2,
1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (2, 8), (2, 9), (2, 10), (2, 11), (2, 12), (2,
13), (2, 14), (2, 15), (2, 16), (3, 0), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6), (3, 7), (3,
8), (3, 9), (3, 10), (3, 11), (3, 12), (3, 13), (3, 14), (3, 15), (3, 16), (4, 0), (4, 1), (4, 2),
(4, 3), (4, 4), (4, 5), (4, 6), (4, 7), (4, 8), (4, 9), (4, 10), (4, 11), (4, 12), (4, 13), (4, 14),
(4, 15), (4, 16), (5, 0), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5), (5, 6), (5, 7), (5, 8), (5, 9), (5,
10), (5, 11), (5, 12), (5, 13), (5, 14), (5, 15), (5, 16), (6, 0), (6, 1), (6, 2), (6, 3), (6, 4),
(6, 5), (6, 6), (6, 7), (6, 8), (6, 9), (6, 10), (6, 11), (6, 12), (6, 13), (6, 14), (6, 15), (7, 0),
(7, 1), (7, 2), (7, 3), (7, 4), (7, 5), (7, 6), (7, 7), (7, 8), (7, 9), (7, 10), (7, 11), (7, 12),
(7, 13), (7, 14), (7, 15), (8, 0), (8, 1), (8, 2), (8, 3), (8, 4), (8, 5), (8, 6), (8, 7), (8, 8),
(8, 9), (8, 10), (8, 11), (8, 12), (8, 13), (8, 14), (8, 15), (9, 0), (9, 1), (9, 2), (9, 3), (9, 4),
(9, 5), (9, 6), (9, 7), (9, 8), (9, 9), (9, 10), (9, 11), (9, 12), (9, 13), (9, 14), (10, 0), (10,
1), (10, 2), (10, 3), (10, 4), (10, 5), (10, 6), (10, 7), (10, 8), (10, 9), (10, 10), (10, 11), (10,
12), (10, 13), (11, 0), (11, 1), (11, 2), (11, 3), (11, 4), (11, 5), (11, 6), (11, 7), (11, 8), (11,
9), (11, 10), (11, 11), (11, 12), (12, 0), (12, 1), (12, 2), (12, 3), (12, 4), (12, 5), (12, 6), (12,
7), (12, 8), (12, 9), (12, 10), (12, 11), (12, 12), (13, 0), (13, 1), (13, 2), (13, 3), (13, 4), (13,
5), (13, 6), (13, 7), (13, 8), (13, 9), (13, 10), (14, 0), (14, 1), (14, 2), (14, 3), (14, 4), (14,
5), (14, 6), (14, 7), (14, 8), (14, 9), (15, 0), (15, 1), (15, 2), (15, 3), (15, 4), (15, 5), (15,
6), (15, 7), (15, 8), (16, 0), (16, 1), (16, 2), (16, 3), (16, 4), (16, 5)]
```

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)
```

Spyder (Python 3.5)

File Edit Search Source Run Debug Consoles Projects Tools View Help

Editor - C:\Users\Jamie\Documents\Python\Homework 3\p3.py

```
1 #-----Homework 3 Question 3 by Jamie Andrews -----
2 #1. Create def function that collects the data from the string
3 #2. Write a for loop that splits the string.
4 #3. Create another for loop that prints the string in range of the def functions
5 #4. Declare a variable that opens the csv file in the program.
6 #5. Assign argument string_pos_1st to array [0,2,3,4] so that it only prints out
7 #6. Assign sep to ",".
8 #7. Assign mydata to the def function.
9
10 #1.
11 def get_csv_data(f,string_pos_1st,sep):
12     #2.
13     for line in f:
14         myline = [line.split(sep) for i, line in enumerate(f)]
15     #3.
16     for i in range(len(string_pos_1st)):
17         print(myline[string_pos_1st[i]])
18
19 #4.
20 f = open("lb-james.csv", "r")
21 #5.
22 string_pos_1st = [0,2,3,4]
23 #6.
24 sep= ","
25 #7.
26 mydata = get_csv_data(f,string_pos_1st,sep)
```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help Variable explorer File explorer

Python console

Console 1/A

```
[ '2005-06', '21', 'CLE', 'NBA', 'SF', '79', '42.5', '11.1', '23.1', '0.48', '1.6', '4.8',
'0.335', '9.5', '18.3', '0.518', '0.515', '7.6', '10.3', '0.738', '0.9', '6.1', '7', '6.6', '1.6',
'0.8', '3.3', '2.3', '31.4\n']
[ '2006-07', '22', 'CLE', 'NBA', 'SF', '78', '78', '40.9', '9.9', '20.8', '0.476', '1.3', '4',
'0.319', '8.6', '16.8', '0.513', '0.507', '6.3', '9', '0.698', '1.1', '5.7', '6.7', '6', '1.6',
'0.7', '3.2', '2.2', '27.3\n']
[ '2007-08', '23', 'CLE', 'NBA', 'SF', '75', '74', '40.4', '10.6', '21.9', '0.484', '1.5', '4.8',
'0.315', '9.1', '17.1', '0.531', '0.518', '7.3', '10.3', '0.712', '1.8', '6.1', '7.9', '7.2', '1.8',
'1.1', '3.4', '2.2', '30\n']

In [5]: runfile('C:/Users/Jamie/Documents/Python/Homework 3/p3.py', wdir='C:/Users/Jamie/Documents/Python/Homework 3')
[ '2003-04', '19', 'CLE', 'NBA', 'SF', '79', '79', '39.5', '7.9', '18.9', '0.417', '0.8', '2.7',
'0.29', '7.1', '16.1', '0.438', '0.438', '4.4', '5.8', '0.754', '1.3', '4.2', '5.5', '5.9', '1.6',
'0.7', '3.5', '1.9', '20.9\n']
[ '2005-06', '21', 'CLE', 'NBA', 'SF', '79', '79', '42.5', '11.1', '23.1', '0.48', '1.6', '4.8',
'0.335', '9.5', '18.3', '0.518', '0.515', '7.6', '10.3', '0.738', '0.9', '6.1', '7', '6.6', '1.6',
'0.8', '3.3', '2.3', '31.4\n']
[ '2006-07', '22', 'CLE', 'NBA', 'SF', '78', '78', '40.9', '9.9', '20.8', '0.476', '1.3', '4',
'0.319', '8.6', '16.8', '0.513', '0.507', '6.3', '9', '0.698', '1.1', '5.7', '6.7', '6', '1.6',
'0.7', '3.2', '2.2', '27.3\n']
[ '2007-08', '23', 'CLE', 'NBA', 'SF', '75', '74', '40.4', '10.6', '21.9', '0.484', '1.5', '4.8',
'0.315', '9.1', '17.1', '0.531', '0.518', '7.3', '10.3', '0.712', '1.8', '6.1', '7.9', '7.2', '1.8',
'1.1', '3.4', '2.2', '30\n']

In [6]:
```

Python console IPython console History log

Permissions: RW End-of-lines: CRLF Encoding: ASCII Line: 25 Column: 4 Memory: 81 %

3:29 PM 2/19/2017