

Problem I.10

Pythagorean Triples

Due Date: 3/1/2019

Folder: IntroToProgramming

File Name: I10_PythagTriple_Name.py

Points: 5 points

Learning Objectives

- Programming Skills
 - ★ Double Loops
 - ★ 2D Numpy array
- Finding Pythagorean triples

Problem Background

A Pythagorean triple is a set of natural numbers $a < b < c$ for which,

$$a^2 + b^2 = c^2.$$

For example, $3^2 + 4^2 = 9 + 16 = 25 = 5^2$, so $(3, 4, 5)$ is a Pythagorean triple. Obviously, there are an infinite number of **real** numbers that satisfy this relation, but Pythagorean triples are interested in the **natural** numbers that satisfy it.

Also for this project, you will be working with nested loops. These are loops that exist within a loop. The syntax for this will look something like:

```
for ii in range(N):
    for jj in range(ii,N):
        code...
```

The `ii` loop is called the *outer* loop, and the `jj` loop is called the *inner* loop. Hence, for each time the code executes one iteration of the `ii` loop, it will execute **all** iterations of the `jj` loop. Note that the inner loop starts at `ii`. This is not at all required, but is an example of what can be done if the inner loop wants the start of the iteration to depend of which iteration of the outer loop we are on. Of course, we could also have the end of the inner loop iteration depend on the outer loop iteration with code like, `for jj in range(0,ii):`.

Finally, you will be using 2-dimensional numpy arrays in this code. In order to create a 2D numpy array with specific elements, use code like:

```
x = np.array([[1,2,3],[4,5,6],[7,8,9]])
```

where the input is a list of lists. This will create a numpy array that essentially stores the matrix,

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}.$$

To create a 2D numpy array consisting of all zeros, use code like:

```
y = np.zeros( [num_rows, num_cols] )
```

It is also possible to append a row or column to a 2D array, but before we discuss that, we need to talk about how to create an empty array. For example, say you want to create an empty array that will eventually have 3 columns and some undetermined number of rows, depending on how much appending you will do. You can create this array using

```
x = np.zeros([0,3])
```

This array is empty because it has 0 rows. But now you can append a row with 3 columns. To do this, you use the following code:

```
x = np.vstack( [ x,np.array([-1,4,2]) ] )
```

Note this is similar to the `np.append` function for 1D arrays. It will return a new array with the row appended. If you run the above code twice, your 2D array will look like the matrix:

$$\begin{bmatrix} -1 & 4 & 2 \\ -1 & 4 & 2 \end{bmatrix}$$

And you can continue doing this to add more rows to your array.

This whole process can be done with columns as well. To add columns to a 2D array, use the `np.hstack` function, which works in exactly the same way, but adds columns to the end of your array.

One thing to be careful about when adding rows or columns, the row or column you add must be of the same dimension as the array you are appending to. For instance, if you add a row with 4 elements in it to a 2D array with 3 columns, Python will give you an error.

Finally, since we are concerned only with natural numbers that satisfy the Pythagorean triple identity, we need a way to check if a variable is a whole number. To do this, you can use the `is.integer()` function. This returns true if the variable is an integer, and false otherwise.

Programming Reminders

- Check if a variable is an integer: `c.is_integer() == True`
- Loops syntax: `for ii in range(N)`
- Create 2D numpy array of zeros: `my_array = np.zeros((num_rows, num_cols))`
- Append a row to a 2D numpy array: `x = np.vstack([x, new numpy array])`

Program Criteria

Write a program that does the following:

- There is exactly one Pythagorean triple for which $a + b + c = 1026$. Write code to find this Pythagorean triple (a, b, c) . Recall, the Pythagorean triples should satisfy the ordering $a < b < c$ and all are natural numbers.
- Print out this Pythagorean triple, with appropriate descriptive text.
- In addition, store all Pythagorean triples that you happen find along the way to finding this particular triple. Store them in a $N \times 4$ numpy array, where N is the number of triples you find. In the first three columns of each row, store each of the triple values a, b , and c respectively. In the fourth column of each row, store the sum $a + b + c$.

Note: Since each person might go about finding the correct triple differently, the triples that are stored might be different for each person.

Deliverables

Place the following in a folder named `IntroToProgramming` in your repository:

- A Python file `I10_PythagTriple_Name.py` that satisfies the program criteria.