Summer 2016 Bootcamp Problem Set 02

Function Created: print_multiplication_table

- Your code should print the multiplication table right-justified as shown below

```
   1     2     3     4     5     6     7     8     9    10    11    12
   2     4     6     8    10    12    14    16    18    20    22    24
   3     6     9    12    15    18    21    24    27    30    33    36
   4     8    12    16    20    24    28    32    36    40    44    48
   5    10    15    20    25    30    35    40    45    50    55    60
   6    12    18    24    30    36    42    48    54    60    66    72
   7    14    21    28    35    42    49    56    63    70    77    84
   8    16    24    32    40    48    56    64    72    80    88    96
   9    18    27    36    45    54    63    72    81    90    99   108
  10    20    30    40    50    60    70    80    90   100   110   120
  11    22    33    44    55    66    77    88    99   110   121   132
  12    24    36    48    60    72    84    96   108   120   132   144
```

Function Created: pyramid

- Prompt the user for a positive integer x and verify
- Print a pyramid that is x lines tall

```
      *
    *   *
  *   *   *
*   *   *   *
```

Create a pyramid of * characters. The leftmost star at the base of the pyramid should touch the left margin of the runtime window. Shown is an example when 4 is entered.

Function Created: factorial

Factorials are very simple things. They're just products, indicated by an exclamation mark. For instance, "four factorial" is written as "4!" and means $1 \times 2 \times 3 \times 4 \;=\; 24$. In general, n! means the product of all the whole numbers from 1 to n; that is, $n! \;=\; 1 \times 2 \times 3 \times \ldots \times n$.

- Prompt the user for a positive integer x and verify
- Calculate and print out $x!$

Summer 2016 Bootcamp Problem Set 02

Function Created: fibonacci

The Fibonacci Series is the series of numbers: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34 and so on. The next number is found by adding up the two numbers before it. For example, the 3 is found by adding the two numbers before it (1+2).

- Prompt the user for a positive integer x and verify
- Write the first x numbers in the fibonacci series

Function Created: tip_calculator (bill_amt)

- You should code a function that accepts a bill_amt float variable and prints out various suggested tip amounts as shown below.
- Make sure to round to two decimal places (one option is to use the **round** function)
- Try to get it to display with two decimal places (one option is to use a string format like **%.2f**. Try Google or stackoverflow to find how to do this.

```
Please enter your total bill: 7
Your bill amount is $7.00
A 10 percent tip: $0.70 totalling $0.70
A 15 percent tip: $1.05 totalling $1.05
A 20 percent tip: $1.40 totalling $1.40
An excellent tip: $7.00 totalling $14.00
```

- Your function signature should look like this since you are accepting a parameter:
  `def tip_calculator(bill_amt):`
- When you are invoking the tip_calculator function in your code, there are a couple of ways you could do that while passing a float for the bill_amt. Either of the following would work. One is more explicit and takes three lines. The other just uses one line.

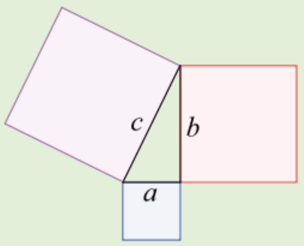
```
# EITHER INVOKE THIS WAY
bill = raw_input("\n\nPlease enter your total bill: ")
bill_as_float = float(bill)
tip_calculator(bill_as_float)

# OR THIS WAY
tip_calculator(float(raw_input("\n\nPlease enter your total bill: ")))
```

Summer 2016 Bootcamp Problem Set 02

Function Created: is_pythagorean(a, b, c)

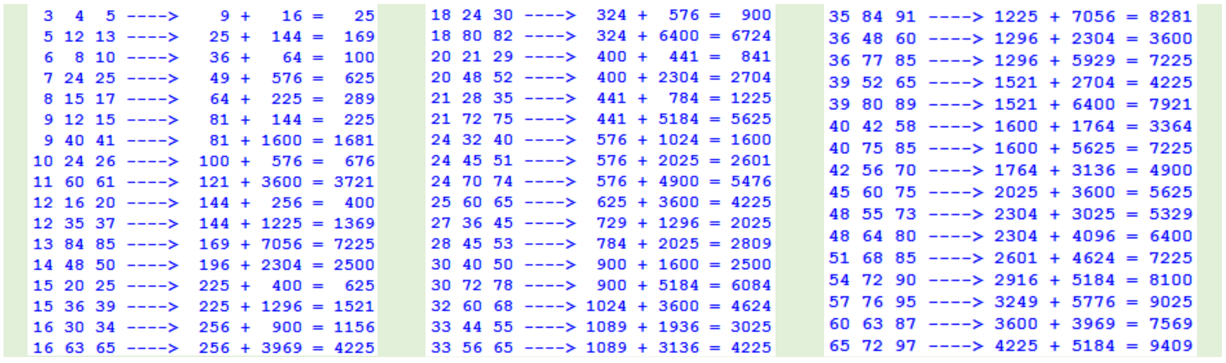
- Write a function that accepts three integers (a, b, c) and prints "IS PYTHAGOREAN" or "IS NOT PYTHAGOREAN" based on the integers satisfying the Pythagorean equation. $a^2 + b^2 = c^2$

Wikipedia states "In mathematics, the **Pythagorean theorem**, also known as **Pythagoras' theorem**, is a fundamental relation in Euclidean geometry among the three sides of a right triangle. It states that the square of the hypotenuse (the side opposite the right angle) is equal to the sum of the squares of the other two sides. The theorem can be written as an equation relating the lengths of the sides a, b and c, often called the "Pythagorean equation": $a^2 + b^2 = c^2$



Function Created: print_pythagoreans_under_100

- Your code should print all Pythagorean triples under 100.
- **Note**: If we identify 3, 4, 5 as a Pythagorean triple, we don't also identify 4, 3, 5 as these are the same.
- **Note**: Your answer will just go down the page vertically. The screenshot below has been cut and pasted repeatedly to reduce the space taken for display.



```
 3  4  5 ---->    9 +   16 =   25    18 24 30 ---->  324 +  576 =  900    35 84 91 ----> 1225 + 7056 = 8281
 5 12 13 ---->   25 +  144 =  169    18 80 82 ---->  324 + 6400 = 6724    36 48 60 ----> 1296 + 2304 = 3600
 6  8 10 ---->   36 +   64 =  100    20 21 29 ---->  400 +  441 =  841    36 77 85 ----> 1296 + 5929 = 7225
 7 24 25 ---->   49 +  576 =  625    20 48 52 ---->  400 + 2304 = 2704    39 52 65 ----> 1521 + 2704 = 4225
 8 15 17 ---->   64 +  225 =  289    21 28 35 ---->  441 +  784 = 1225    39 80 89 ----> 1521 + 6400 = 7921
 9 12 15 ---->   81 +  144 =  225    21 72 75 ---->  441 + 5184 = 5625    40 42 58 ----> 1600 + 1764 = 3364
 9 40 41 ---->   81 + 1600 = 1681    24 32 40 ---->  576 + 1024 = 1600    40 75 85 ----> 1600 + 5625 = 7225
10 24 26 ---->  100 +  576 =  676    24 45 51 ---->  576 + 2025 = 2601    42 56 70 ----> 1764 + 3136 = 4900
11 60 61 ---->  121 + 3600 = 3721    24 70 74 ---->  576 + 4900 = 5476    45 60 75 ----> 2025 + 3600 = 5625
12 16 20 ---->  144 +  256 =  400    25 60 65 ---->  625 + 3600 = 4225    48 55 73 ----> 2304 + 3025 = 5329
12 35 37 ---->  144 + 1225 = 1369    27 36 45 ---->  729 + 1296 = 2025    48 64 80 ----> 2304 + 4096 = 6400
13 84 85 ---->  169 + 7056 = 7225    28 45 53 ---->  784 + 2025 = 2809    51 68 85 ----> 2601 + 4624 = 7225
14 48 50 ---->  196 + 2304 = 2500    30 40 50 ---->  900 + 1600 = 2500    54 72 90 ----> 2916 + 5184 = 8100
15 20 25 ---->  225 +  400 =  625    30 72 78 ---->  900 + 5184 = 6084    57 76 95 ----> 3249 + 5776 = 9025
15 36 39 ---->  225 + 1296 = 1521    32 60 68 ----> 1024 + 3600 = 4624    60 63 87 ----> 3600 + 3969 = 7569
16 30 34 ---->  256 +  900 = 1156    33 44 55 ----> 1089 + 1936 = 3025    65 72 97 ----> 4225 + 5184 = 9409
16 63 65 ---->  256 + 3969 = 4225    33 56 65 ----> 1089 + 3136 = 4225
```

Function Created: triangle_classifier

- **Note: Please ignore the absolute rules of geometry and reality and just go by the rules we have here for triangle classification. This is not about geometry. It is just to help you practice coding a set of rules in Python.**
- Your code should prompt the user to input three side lengths as integers: a, b, and, c
- If all three integers are 0, then print "STOPPING" and stop
- If any of the integers are negative, then print "Please enter positive values." and keep prompting the user for new side lengths
- If any of the three lengths is greater than the sum of the other two, then you cannot form a triangle. Otherwise, you can. Your code should print the following, as applicable, and then keep prompting the user for new side lengths:
  - NO – if the three integers cannot form a triangle using the rule above
  - YES – if the three integers can form a triangle using the rule above
  - EQUILATERAL – if the sides are equal
  - ISOSCELES – if two sides are equal
  - DEGENERATE - If the sum of two lengths equals the third.
  - PYTHAGOREAN – if this equation is satisfied $a^2 + b^2 = c^2$
- **Note**: A triangle can meet the requirements of more than one classification:
  - We will consider all equilateral triangles to also be isosceles triangles, but you should also separately identify isosceles triangles that are not equilateral triangles
  - It may also be possible sometimes that degenerate triangles are also isosceles, but this is not always the case
  - See some of the test cases below for clarification

```
Please enter side a: 3
Please enter side b: 3
Please enter side c: 9
  3   3   9 NO
```

```
Please enter side a: 22
Please enter side b: 22
Please enter side c: 12
 22  22  12 YES ISOSCELES
```

```
Please enter side a: 3
Please enter side b: 4
Please enter side c: 5
  3   4   5 YES PYTHAGOREAN
```

```
Please enter side a: 3
Please enter side b: 4
Please enter side c: 6
  3   4   6 YES
```

```
Please enter side a: 3
Please enter side b: 4
Please enter side c: 7
  3   4   7 YES DEGENERATE
```

```
Please enter side a: 0
Please enter side b: -1
Please enter side c: 0
Please enter positive values.
```

```
Please enter side a: 9
Please enter side b: 9
Please enter side c: 9
  9   9   9 YES EQUILATERAL
  9   9   9 YES ISOSCELES
```

```
Please enter side a: 3
Please enter side b: 3
Please enter side c: 6
  3   3   6 YES ISOSCELES
  3   3   6 YES DEGENERATE
```

```
Please enter side a: 0
Please enter side b: 0
Please enter side c: 0
STOPPING
```

Summer 2016 Bootcamp Problem Set 02

More Practice

- Remember to use Python Coding Bat or Learning Python the Hard Way if you would like more practice with coding.
- I will post the solutions and another problem set in a week after everyone has had a chance to attempt these problems.
- If you have other recommendations for coding practice resources, feel free to post to our Piazza site.
- Have fun! ☺