1. Write a function named *right\_justify* that takes a string named *s* as a parameter and prints the string with enough leading spaces so that the last letter of the string is in column 70 of the display:  
   >>> right\_justify('monty')

monty

Hint: Use string concatenation and repetition. Also, Python provides a built-in function called *len* that returns the length of a string, so the value of len('monty') is 5. This function does not need to return a value.

**def right\_justify(s):**

**sLen = len(s)**

**len(‘monty’) \* 14**

**spaceRequired = len(‘monty’) - sLen**

**spaces = ' ' \* spaceRequired**

**return spaces + s**

1. Write the function volOfCylinder which takes two arguments r and h and returns the volume of a cylinder with base radius r and height h. Include help comments. Use the formula V = pi \* r\*\*2 \* h

**def volOfCylinder (r, h):**

**# for volume of a cylinder**

**volume = pi \* r \*\* 2 \* h**

**return result**

1. If you are given three sticks, you may or may not be able to arrange them in a triangle. For example, if one of the sticks is 12 inches long and the other two are one inch long, you will not be able to get the short sticks to meet in the middle. For any three lengths, there is a simple test to see if it is possible to form a triangle:

If any of the three lengths is greater than the sum of the other two, then you cannot form a triangle. Otherwise, you can. (If the sum of two lengths equals the third, they form what is called a “degenerate” triangle.)

Write a function named *is\_triangle* that takes three integers as arguments, and that prints either “Yes” or “No”, depending on whether you can or cannot form a triangle from sticks with the given lengths.

**def is\_triangle (a, b, c)**

**if is\_triangle (a + b < c ) or (a + c < b ) or (b + c < a):**

**print ("Yes, it is a triangle")**

**else :**

**print ("No, it is not a triangle")**

1. Write the function *getGrade* which takes one argument x and returns (not prints) a letter grade. Grading scheme. A: x >= 88, B: 76 <= x < 88, C: 64 <= x < 76, F: x < 64.

**def getGrade (A, B, C, F)**

**if grade >= 88:**

**return (A)**

**elif grade 76 <= x < 88:**

**return (B)**

**elif grade 64 <= x < 76:**

**return (C)**

**else:**

**return (F)**