Welcome to the 2017 Spring semester of CS 1371! Before beginning your first homework, you should take a look at the **CS1371_HomeworkGuide.pdf** and **testingYourCode.pdf** files included in the .zip file for this homework. These documents detail everything you need to know about completing this and future homeworks. The document about testing your code also includes information about testing functions with output types that have not been taught yet. You can ignore this part of the document for now, but will probably want to save it as a reference for later in the semester.

If you have not yet downloaded MATLAB you can click here to download it from GT OIT. Once you have MATLAB installed, and you have read and understood the documents above, you can start on this homework!

Also note, because you only have 5 days to complete this homework it is a little shorter than usual. For the rest of the semester, you will have a full week to complete the homeworks.

Happy coding,
~Homework team

Function Name: `f`

Inputs:

1. *(double)* An x value
2. *(double)* A y value
3. *(double)* A k value

Outputs:

1. *(double)* The resulting value of the function

Function Description:

This problem is designed to illustrate the fact that MATLAB functions are similar to math functions that you are familiar with. Write a function in MATLAB that will evaluate the function $f(x, y, k)$.

$$f(x,y,k) = \left\lfloor rem\left(\left\lfloor \frac{y+k}{17} \right\rfloor 2^{-17*x-rem((y+k),17)}, 2\right)\right\rfloor$$

There are built-in MATLAB functions for floor, remainder and exponentiation (`floor()`, `rem()`, and `.^`, respectively). Feel free to use the `help` function in the Command Window to look up these or any other built-in functions you may be confused about.

If this functions seems needlessly complex, just wait. It has a very interesting property that will be explored in a later homework :)

Function Name: `cartDist`

Inputs:

1. *(double)* The first point's x-coordinate (x_1)
2. *(double)* The first point's y-coordinate (y_1)
3. *(double)* The second point's x-coordinate (x_2)
4. *(double)* The second point's y-coordinate (y_2)

Outputs:

1. *(double)* The distance between the two points

Function Description:

This function will take in two points defined in 2 dimensional space and calculate the distance between them. The points will be represented using Cartesian coordinates in the form [x_1, y_1] and [x_2, y_2]. In case you are rusty on your geometry, the distance between two points can be calculated using the following formula:

$$distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Round the distance to the nearest hundredth.

Function Name: `freefall`

Inputs:

1. *(double)* An amount of time in seconds

Outputs:

1. *(double)* The final position of the falling object in meters
2. *(double)* The final velocity of the object in meters/second

Function Description:

You've gotten tired of solving the same kinematics problems in physics, so you decide to automate the process using MATLAB. This function will determine the position and velocity of an object free-falling from rest. You can assume a constant acceleration of 9.807 m/s and no air resistance. The following are the formulas for position and velocity.

$$p_f = \frac{a*t^2}{2}$$

$$v_f = a * t$$

where p_f is the final position, v_f is the final velocity, a is the acceleration of the object, and t is the elapsed time.

Round your final answer to the thousandths place.

Function Name: `candy`

Inputs:

1. *(double)* Number of pieces of candy in a bag
2. *(double)* Number of kids

Outputs:

1. *(double)* Pieces of candy per kid
2. *(double)* Pieces of candy wasted

Function Description:

You are at a birthday party and buy a bag of candy to hand out to each of the kids who attend. But in order to be fair, every kid has to get the same number of pieces, and any pieces left over in the bag are considered to be wasted.

This function will take in the number of pieces of candy in a given bag and determine how many pieces of candy each kid gets, and how many pieces of candy are wasted. For example, if the size of the bag was 50 pieces, and there were 4 kids at the party, each kid would get 12 pieces of candy and 2 pieces of candy would be wasted. So the first output would be 12 and the second would be 2.

Notes:

- You may find the `floor()` and/or `mod()` functions useful.