Important

1. Due Date: **Tuesday, January 21st, 11:59 pm.**
2. This homework will be graded out of 100 points.
3. This is an individual assignment. Collaboration is encouraged, but **your submission must be uniquely yours.**
4. For Help:
   - TA Helpdesk (Schedule posted on class website)
   - Email TA's or use Piazza Forums Notes
   - How to Think Like a Computer Scientists
   - [http://openbookproject.net/thinkcs/python/english3e/]
   - CS 1301 Python Debugging Guide
   - [http://www.cc.gatech.edu/classes/AY2016/cs1301_spring/CS-1301-Debugging-Guide/index.html]

6. Comment out or delete all your function calls. Only import statements, global variables, and comments are okay to be outside the scope of a function.
7. **Read the entire document before starting this assignment.**

Introduction

The goal of this homework is to learn the difference between print and return as well as give you practice with conditionals. The homework will consist of 5 functions for you to implement. You have been given HW02.py to fill out with instructions in the docstrings. However, below you will find more detailed information to complete your assignment. Read it thoroughly before you begin. You have until **Tuesday, January 21st, 11:59 pm** to complete this assignment.

Print vs Return

Two concepts that may be difficult for beginner programmers to differentiate between are the print function and the return statement. While it may appear that they do the same thing, it is important to note that they are quite different. The print function as its name states, is a function just like round or any other predefined and user defined functions. The purpose of the print function is to display information to the user. You cannot save what you print. The return statement, on the other hand, is part of a function definition. All functions have a return value, whether you explicitly write a return statement or not; functions that do not explicitly have a return statement always return the value None. The return statement is useful because it allows you to give a value to a function, thus allowing you to either save it for later, use it in a more complex expression, or print it for human consumption. Important note to remember: print is for the user whereas return is for the program.

Juliette (julietteshelton@gatech.edu) & Brae (bdavies34@gatech.edu)

CS1301 - HOMEWORK 02: CONDITIONALS

Example:
Let's say we have the following two functions below in a file called file.py:

```
def printFunc():
    print(2)

def returnFunc():
    return 2
```

This is what would happen if we ran the file, and typed the following into the python shell.

```
>>> a = printFunc()
2
>>> print(a)
None
```

```
# notice that although the number 2 is printed to the screen the
# variable we assigned the function call to, a, has the value None
# (NoneType)
```

```
>>> b = returnFunc()
>>> print(b)
2
# when we call returnFunc() and assign it to a variable, nothing is
# printed to the screen because there are no print statements inside
# the function
# however the variable, b, now holds the value 2 (int)
```

PART 1: FUNCTIONS

Function name: `co2emissions()`
Parameters: number of trolleys (int), number of buses (int), number of cars (int)
Returns: string
Description: You are trying to get to the fair and you want to determine the best route to take! You will be taking many different modes of transportation, and your goal is to determine which combination of these will produce the least amount of greenhouse gas emissions! You are traveling exactly one mile, and you know that trolleys release an average of 233 grams of CO_2 per passenger mile, transit buses release an average of 299 grams, and cars (with 1 person) release an average of 371 grams. You may have to take multiple trolleys, busses, or cars (assume 1 person in the car) to get there. Determine the total amount of CO2 emissions that your total trip will release. If the total

Juliette (julietteshelton@gatech.edu) & Brae (bdavies34@gatech.edu)

amount is less than or equal to 250 return "Great", if it is between 250 and 1000 return "Not Good", if it is greater than or equal to 1000 return "Bad!!".

Test Cases:
```
>>> co2emissions(1,0,0)
Great

>>> co2emissions(0,0,3)
Bad!!
```

Function name: `supportRally()`
Parameters: percent of decided students who support your idea (float), percent of total students who are undecided (float), desired_level(str)
Returns: string or float
Description:

Georgia Tech is looking for a way to make its campus more sustainable. You've come up with the idea to introduce reusable containers at West Village. Some students support your idea while others are undecided. You will hold a rally to gain the support of undecided students to reach an ideal support level. Write a function that takes in the percentage of decided students that support your idea (float), the percentage of students that are undecided (float), and the desired support percentage you want to reach (str). All of the undecided students will go to the rally. Assume the support levels are the following:

'Outstanding support' >= 90.0
'Significant support' >= 80.0
'Moderate support' >= 70.0

Return the float value of the minimum percentage of students at the rally you need to convince in order to achieve your desired support level rounded to 2 decimal places. If achieving the desired level of support requires a percent surpassing 100.0, return 'Not possible'.

Test Cases:

With 93% percent of the **decided** student body already supporting you, you need to convince 84.89% of the 37% of students that are undecided in order to reach the 90.0% total support level.

```
>>> supportRally(0.93, 0.37, 'Outstanding support')
84.89
```

With 70.2% percent of the **decided** student body already supporting you, it is not possible to reach the 80.0% total support level with only 15.4% of students being undecided.

```
>>> supportRally(0.702, 0.154, 'Significant support')
'Not Possible'
```

Function name: `isRallySuccessful()`
Parameters: level of evidence (int), percent of decided students who support your idea (float), percent of total students who are undecided (float), desired_level(str)
Returns: boolean
Description: Since you have now determined the percentage of students you must convince at the rally, we need to make sure you can actually convince these students to support the movement. Georgia Tech students LOVE their evidence! Write a function that takes in the level of evidence in support of your idea, the percent of decided students supporting the reusable containers before the rally, the percent of students who are undecided, and the desired support level. If the level of evidence is greater than 6 and it's possible to convince enough students to reach the desired level, return True. If the evidence is below or equal to 6 or it is not possible to convince enough students, return False. You must call supportRally() in this function!

Test Cases:
```
>>> isRallySuccessful(5, 0.88, 0.2, 'Significant support')
False
```

```
>>> isRallySuccessful(7, 0.91, 0.25, 'Outstanding support')
True
```

Function name: `spreadingFire()`
Parameters: square miles on fire (int), endangerment limit (int)
Returns: string
Description: More fires are starting to pop up in the forests of California. Given the square miles of fire and the endangerment limit (the number of square miles that are on fire that poses risk), write a function to determine which kind of extinguishing method should be employed. For every 100 square miles over the endangerment limit, the danger level increases by 1. For each case, return the danger level and extinguishing method. If the danger level is between 1 and 4, the extinguishing method is "Firebreak and Airdrop." If the danger level surpasses 4, return "Evacuate and Full Protocol". If the square miles of fire is less than 100 miles over the endangerment level, return "Proceed with caution". **Note:** Danger Level does not go below 0.

For example, if a fire had burned 250 square miles in a forest with a 50 square mile endangerment level, the danger level would be 2, and the function would return "Danger Level: 2; Firebreak and Airdrop".

Test Cases:
```
>>> spreadingFire(250, 50)
'Danger Level: 2; Firebreak and Airdrop'

>>> spreadingFire(775, 35)
'Danger Level: 7; Evacuate and Full Protocol'
```

Function name: `recycle()`
Parameters: number of soda bottles (int), number of grocery bags (int), number of plastic forks (int), number of bottle caps (int)
Returns: string
Description: You and your friends are walking on the beach, and sadly, you see plastic all over the shore. You decide to pick up all of the plastic so you can recycle it! However, before you can recycle it, it needs to be sorted.

You will group all of the items you find by their Type of Plastic. Then you will need to determine the number of items and total weight for each Type of Plastic category. (See Rules for Sorting) After you have finished sorting, the recycling center asks that you bag all of the items. The recycling center provides you with three different bags, each determined by the type of plastic, number of items for each type, and total weight. (See Rules for Bagging) You want to count the final number of items in each bag and print them out in the following format: "Green: (num of items), Yellow: (num of items), Red: (num of items)"

Rules for Sorting:

Item	Type of Plastic	Item Weight
An Empty Soda Bottle	PETE (1)	37.3 grams
Grocery Bag	HDPE (2)	5 grams
Plastic Fork	PP (5)	2.5 grams
Bottle Caps	PP (5)	2.4 grams

Rules for Bagging:
Green Bag: Type: PETE, Number of Items: < 5, Total Weight: > 60 grams
           Type: PP, Number of Items: > 10, Total Weight: > 60 grams

Yellow Bag: Type: PETE, Number of Items: > 5, Total Weight: > 60 grams
           Type: HDPE (All)
           Type: PP, Total Weight: < 60 grams

Juliette (julietteshelton@gatech.edu) & Brae (bdavies34@gatech.edu)

CS1301 - HOMEWORK 02: CONDITIONALS

Red Bag: Type: PETE , Total Weight: < 60 grams
Type: PP, Number of Items: < 10, Total Weight: > 60 grams

<u>Test Cases:</u>
```
>>> recycle(2,1,0,1)
Green:2, Yellow:2, Red:0

>>> recycle(1,0,3,3)
Green:0, Yellow:6, Red:1
```

Grading Rubric

```
co2emissions:                          15 pts
supportRally:                          20 pts
isRallySuccessful:                     15 pts
spreadingFire:                         25 pts
recycle:                               25 pts
_____
Total                                  100 pts
```

Provided

The following file(s) have been provided to you.

1. HW02.py
   This is the file you will edit and submit to GradeScope. All instructions for what the functions should do are in the docstrings.

Deliverables

For the homework, we will be using GradeScope for submissions and automatic grading. When you submit your HW02.py file to the appropriate assignment **on GradeScope**, the autograder will run automatically. The grade you see on GradeScope will be the grade you get on the assignment, unless it is late or your grading TA sees signs of you trying to defeat the system in your code. You may re-submit this assignment unlimited times until the deadline, but spamming GradeScope with a lot of submissions in a short amount of time should be avoided; you should resubmit only after substantial changes to your code have been made since your last submission.

1. HW02.py

Juliette (julietteshelton@gatech.edu) & Brae (bdavies34@gatech.edu)

Juliette (julietteshelton@gatech.edu) & Brae (bdavies34@gatech.edu)