

CS 1301 Pair Homework 2 - Conversions

Due: Monday January 26th, before 11:55 PM

Out of 100 points

Files to submit: HW2.py

This is a Pair Programming assignment!

This is a pair programming problem! You are expected to work with the person you have been paired with in class, and you are both responsible for submitting the exact same code to T-Square. Your pair may collaborate with other students in this class. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. You should not exchange code or write code for other pairs. Collaboration at a reasonable level will not result in substantially similar code. For pair programming assignments, you and your partner should turn in identical assignments with both your names at the top.

NOTE: EACH partner is responsible for turning in their own assignment! If you fail to turn in your assignment, you will receive a zero, regardless of if your partner turned in their assignment or not.

For Help:

- TA Helpdesk - Schedule posted on class website.
- Email TA's or use T-Square Forums

Notes:

- Don't forget to include the required comments and collaboration statement (as outlined on the course syllabus).
- Do not wait until the last minute to do this assignment in case you run into problems.

Conversions / Calculations (100 points)

You will write a few python functions for practice with calculations. In your HW2.py file, include a comment at the top with your name, section, GTID/Email, and your collaboration statement. Here is an example of an appropriate collaboration statement (the # character in Python is for comments that are not part of the code):

```
#George P. Burdell 902112358
```

```
#gburdell3@gatech.edu
```

```
#My partner, Name, and I worked on this assignment alone, using only this semester's course materials.
```

```
***ALWAYS REMEMBER THE COLLABORATION STATEMENT***
```

Also include each of the following functions:

- touchdownsPrediction
- valentine
- northPole
- basketBallin
- cupHalfFull
- restroomTrips
- frenchToOrder
- travelCalculator

Function Name: **touchdownsPrediction**

Parameters:

numYards

Return Value:

numTouchdowns

Description:

In 2014, Justin Thomas rushed 1,086 yards and scored 8 touchdowns. With that knowledge, write a user-interactive function that will take in the number of yards you predict Justin Thomas to rush in the next football game and **return** the number of touchdowns you expect him to score, *rounded down to a whole number*. (Hint: you will need to find the average number of yards JT rushes per touchdown.)

Test Cases:

```
python>>> x = touchdownsPrediction(167)
```

```
Ok
```

```
python>>> print(x)
```

```
1
```

```
Ok
```

```
python>>> y = touchdownsPrediction(690)
```

```
Ok
```

```
python>>> print(y)
```

```
5
```

```
Ok
```

Function Name: **valentine**

Parameters:

nameOfCrush

Return Value:

None

Description:

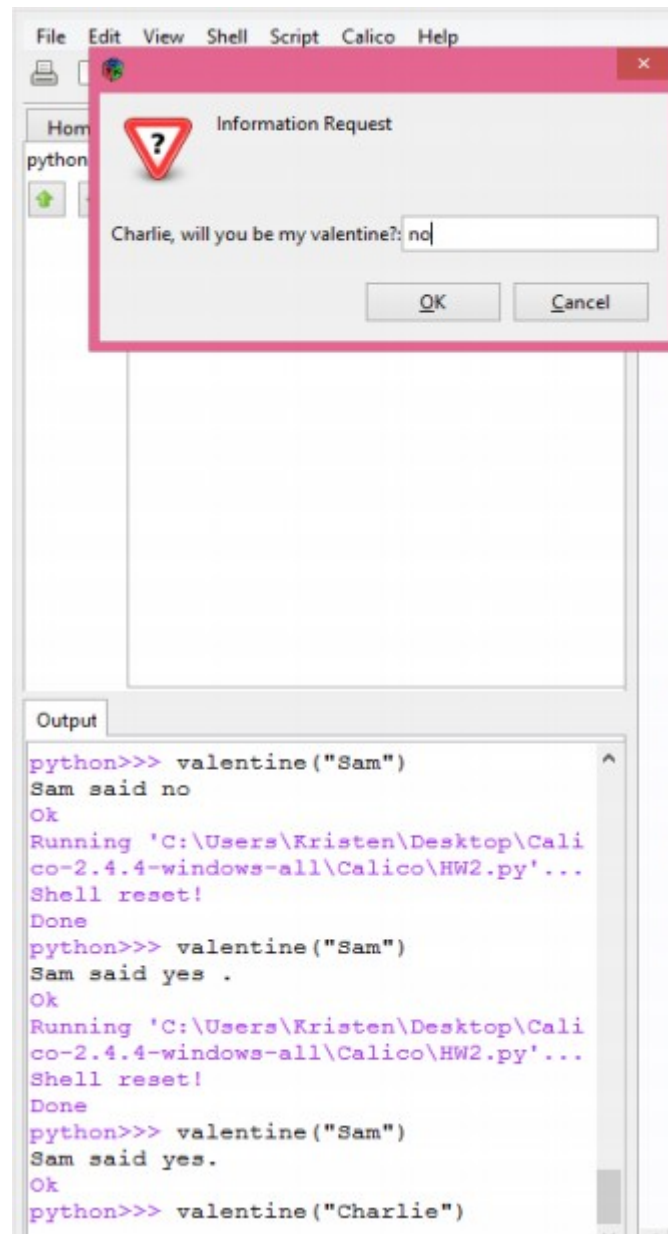
Write a user-interactive function that will take in the name of your crush as a parameter, ask your crush if he/she will be your valentine (via an input), and print his/her answer as printed below.

Test Cases:

```
python>>> valentine("Charlie")
```

Charlie said no.

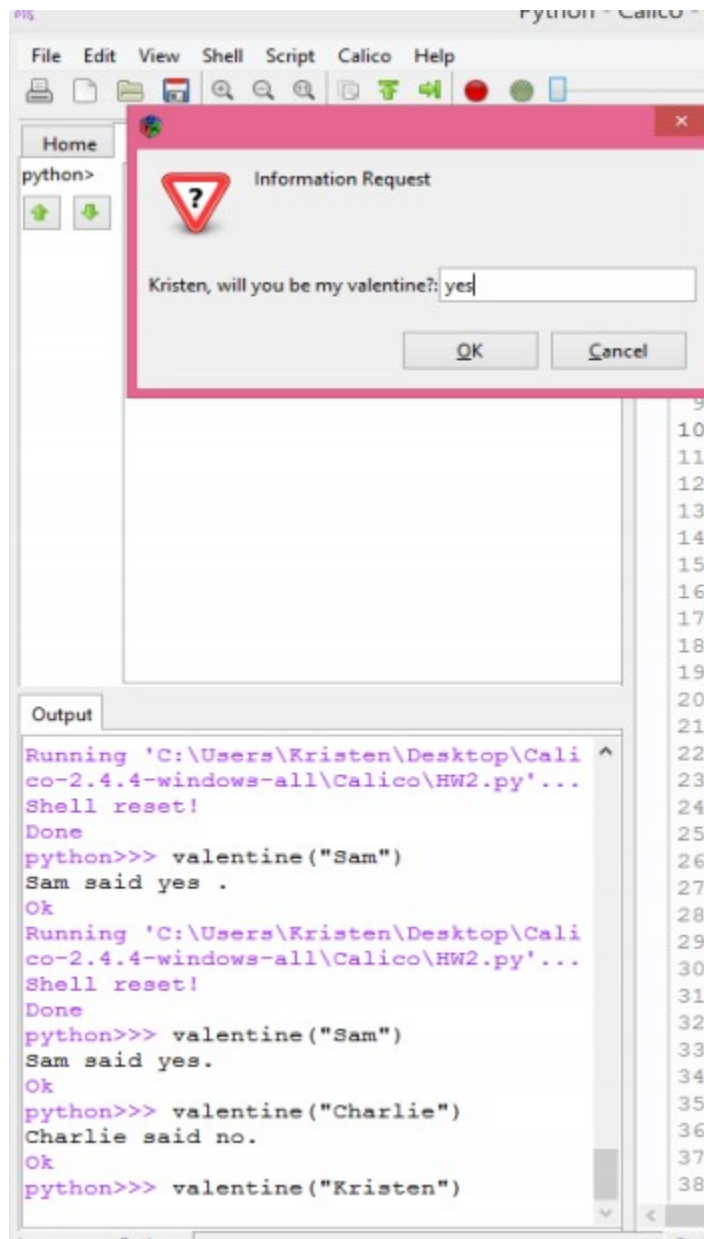
Ok



`python>>> valentine("Kristen")`

Kristen said yes.

Ok



Function Name: **northPole**

Parameters:

numChildren

Return Value:

None

Description:

Santa is looking at hiring elves for this upcoming Christmas season. He needs you to write a user-interactive function to calculate the number of elves Santa needs, depending on the number of children he will visit. Each child will get 3 presents, and elves can only make 60 presents per Christmas season. (Hint: round the number of elves up to the nearest whole number using `math.ceil()` - you will have to import `math`.)

Test Cases:

```
python>>> northPole(1234567)
```

Santa needs 61729 elves to make 3703701 presents.

Ok

```
python>>> northPole(9876)
```

Santa needs 494 elves to make 29628 presents.

Ok

Function Name: **basketBallin**

Parameters:

pointsScoredAgainst, moneyNeeded

Return Value:

None

Description:

In order to raise some money to buy a few things on your wish list, you are placing a bet on a basketball game this Friday. There is a \$10 entry fee, and the rules of the bet are that you receive \$5 per marginal point (i.e. the number of points your team wins by). Assuming that your team wins and you know the average score of the opponent's team, write a user-interactive function to calculate how many points your basketball team will have to score in order to raise enough money to buy something on your wish list.

Your parameters will include the number of points the opponent scores and the amount of money you intend to raise.

You will need to calculate how many marginal points are needed to raise the money you want (including getting your \$10 entry fee back!), and then print how many points your team will have to score total.

Test Cases:

```
python>>> basketBallin(45, 250)
```

Your team will need to score 97 points.

Ok

```
python>>> basketBallin(45, 10)
```

Your team will need to score 49 points.

Ok

Function Name: **cupHalfFull**

Parameters:

radius, height, percentage

Return Value:

numOunces

Description:

You are a very curious person, so you want to know exactly how many ounces of liquid are in your drink at all times. You will calculate this by knowing the dimensions of your cup in inches and guesstimating the percentage of drink you have left. Write a user-interactive function to calculate the volume that your cup can hold, taking into account the percentage of drink in the cup, to estimate the number of ounces of liquid you have left. **Return** the number of ounces.

Volume of a cylinder, inches³ (your cup): $\pi * radius^2 * height$

Note: Your percentage should be a number between 0 and 100.

Conversion: 1 inch³ = 0.554113 ounces

Test Cases:

```
python>>> a = cupHalfFull(2.5, 6, 80)
```

Ok

```
python>>> print(a)
```

```
52.22391990175803
```

Ok

```
python>>> b = cupHalfFull(1.5, 8, 35.6)
```

Ok

```
python>>> print(b)
```

```
11.155029291015516
```

Ok

Function Name: **restroomTrips**

Parameters:

None

Return Value:

None

Description:

You are road-tripping across the country this Spring Break with a group of your friends, and you are planning your route, including stops for restroom trips and meals. You have this one friend who LOVES Coca-Cola and always has a can of soda with him. You know he will probably drink a couple of cans of soda on the trip, so you want to calculate how many times you will have to stop for him to use the restroom. You assume that he will have to use the restroom for every 16 ounces of soda he consumes. To figure this out, write a user-interactive function to calculate the number of times he will need to stop to use the restroom. Do this by asking your friend (the user) how many cans of soda he plans on drinking, then calculate the number of ounces that he will consume total by **calling the cupHalfFull function**, and then calculate the number of times you will have to stop according to your assumption (1 restroom trip per 16 ounces). **Print** your result.

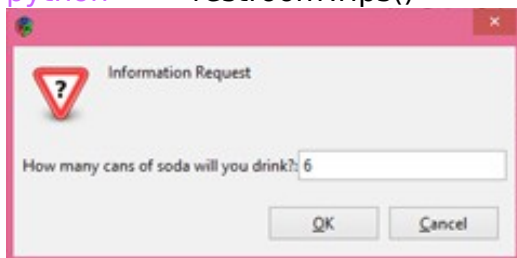
Note: a can of soda has a radius of 1.2 inches and a height of 4.8 inches

Hint 1: How *full* will a drink be if he hasn't consumed any yet? (not a trick question)

Hint 2: You will want to round the number of restroom trips up to the next whole number, using the `math.ceil()` function (so import `math`)

Test Cases:

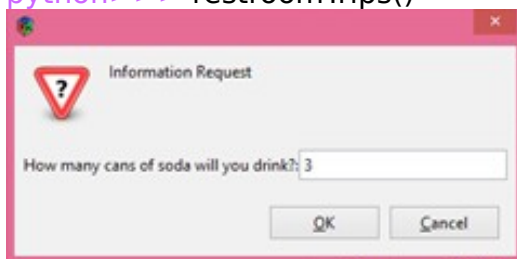
```
python>>> restroomTrips()
```



You will need to make 5 restroom trips.

Ok

```
python>>> restroomTrips()
```



You will need to make 3 restroom trips.

Ok

Function Name: **frenchToOrder**

Parameters:

numSlicesRoastBeef, numFrenchRolls

Return Value:

numDips

Description:

You are the proud owner of a restaurant called French to Order. You specialize in one sandwich, the French Dip. For each sandwich, you use 5 slices of roast beef, 2 French rolls, and one cup of au jus dip. You make the au jus dip fresh every morning, and for cost-purposes, only want to make enough au jus dip to serve one cup with each sandwich. Although you have a very specific roast beef to French roll ratio, your supplier usually brings you random amounts of each, so you need to figure out how many sandwiches you can possibly make with the ingredients he brings you. Then you will know how many cups of au jus dip to make. Write a user-interactive function to calculate how many cups of au jus dip you will need, based on the number of slices of roast beef and French rolls you have.

Hint 1: you can only make as many sandwiches as your least amount of ingredients (according to the recipe) will make... i.e. if you have 5 slices of roast beef and 4 French Rolls, you will only be able to make one sandwich because of the roast beef constraint.

Hint 2: you will need to use a simple if / else statement to figure out which of your ingredients is your constraint

Test Cases:

```
python>>> s = frenchToOrder(18, 9)
```

Ok

```
python>>> print(s)
```

3

Ok

```
python>>> t = frenchToOrder(2, 6)
```

Ok

```
python>>> print(t)
```

0

Ok

Function Name: **travelCalculator**

Parameters:

None

Return Value:

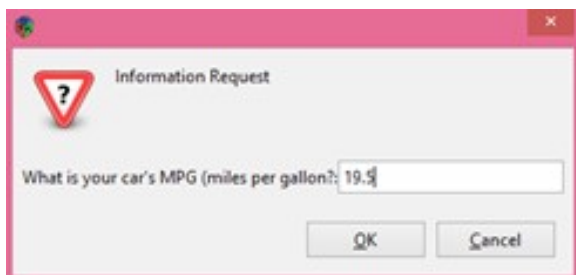
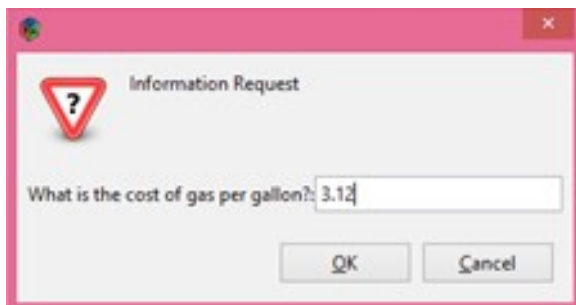
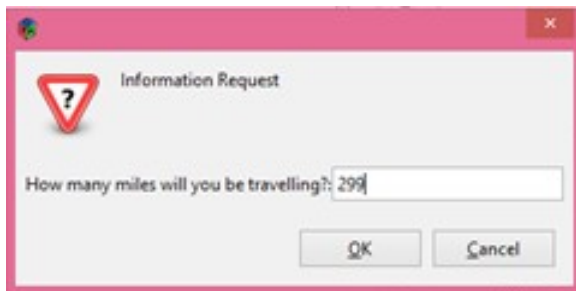
None

Description:

Write a user-interactive function to calculate how much a car-trip will cost you in gas. You will get the distance in miles, cost of gas per gallon, and MPG (miles per gallon) from the user. Then calculate how much that trip will cost the user and print the amount, rounded to two decimals.

Test Case:

```
python>>> travelCalculator()
```



This trip will cost you \$47.84.

Ok

Rubric

100 points

- **touchdownsPrediction** - 5 points
 - Function named correctly 2.5
 - Returns the correct result 2.5
- **valentine** - 5 points
 - Function named correctly 2.5
 - Prints the answer from the input correctly 2.5
- **northPole** - 15 points
 - Function named correctly 5
 - Calculates correct result 5
 - Prints properly 5
- **basketBallin** - 15 points
 - Function named correctly 5
 - Calculates correct result 5
 - Returns result properly 5
- **cupHalfFull** - 15 points
 - Function named correctly 5
 - Calculates correct result 5
 - Returns result properly 5
- **restroomTrips** - 15 points
 - Function named correctly 5
 - Calls cupHalfFull() with right parameters 5
 - Prints the correct result 5
- **frenchToOrder** - 15 points
 - Function named correctly 5
 - If / else statements are correct 5
 - Returns the correct result 5
- **travelCalculator** - 15 points
 - Function named correctly 5
 - Gets input from the user properly 5
 - Prints the correct result 5