CS 1301

Pair Homework 2 – Conversions/Complex Functions

Due: Friday, Feb. 4th, before 6 PM

Out of 100 points

Files to submit: 1. hw2.py

You will be writing several functions, but they will all be saved in one file: hw2.py. Please save all of the functions in this one file or you will lose points. **This is a pair programming assignment!** You are expected to work with the person you have been pared 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.

For Help:

- TA Helpdesk – Schedule posted on class website.
- Email TAs

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.
- If you find a significant error in the homework assignment, please let a TA know immediately.

Part 1 – Conversions (50 pts)

Function name: mass (15 pts)

Write a function called **mass()** which takes in a numerical value in pounds and coverts that value into weys. For reference, there are 256 pounds in a wey (yes, that's a real unit!). Please **print** the value. Save this function in your **hw2.py** file as well. Note that this function should NOT return a value.

Sample Output:

```
>>> mass(500)
1.953125
>>> mass(90.0)
0.3515625
```

Function name: volume (15 pts)

In this function, you will need to convert volumes. There are 0.264 US gallons in one liter.
Write a function called **volume()** that takes in a numerical value in liters and converts it to US gallons.

Since we want to know what we're dealing with, **print** the value out with a nice little sentence explaining the conversion that just happened. Please print the values to 2 decimal places. See the sample output below. Save this function in your **hw2.py** file as well. Note that this function should not return a value.

Sample Output:

```
>>> volume(5)
There are 1.32 US gallons in 5.00 liters
>>> volume(7)
There are 1.85 US gallons in 7.00 liters
```

Function name: bears (15 pts)

Just like the previous two problems, it's conversion time. One male polar bear is equal to 1151.430565248 cans of soda. Seriously. Write a function called **bears()** that takes in a number of polar bears and **prints** out the resultant conversion into cans of soda. This time, we want you to specify how many decimal places to round to as an **integer parameter**. See the sample output for more clarification. Save this function in your **hw2.py** file as well.

Sample Output:

```
>>>bears(2, 4)
There are 2302.8611 cans of soda in 2.0000 male polar bears
>>>bears(6.5, 3)
There are 7484.299 cans of soda in 6.500 male polar bears
>>>bears(10, 1)
There are 11514.3 cans of soda in 10.0 male polar bears
```

Source for the conversion factor: http://www.weirdconverter.com/

You should totally go there and play around with conversions that don't make any sense. Impress your professors! Entertain your friends! Dazzle your family!

Part 2 – Complex Functions

Now you will write a few python functions that return data (instead of printing it!). Also include each of the following functions in your hw2.py file.

Function Name: calcEnergy (25 points)

Parameters:

1. time – a number representing how many hours you will use your A/C unit
2. kwh – an integer representing how many kilowatts of electricity your A/C unit will consume per hour
3. ppkw – a floating point number representing price of one kilowatt of electricity
4. discount – a boolean, either True or False, that indicates if you have a discount of 20% on the total price of the electricity that you use.

Return Value:

A floating point number representing the total cost of energy consumed.

Sample Output:

```
>>>calcEnergy(100, 5, 1.2, True)
480
>>>a = calcEnergy(35, 3, 2.65, False)
>>>a
278.25
```

Description:
Write a function calcEnergy that will calculate and return the amount of money you will spend on your A/C unit given the number of hours you use it, the number of kilowatts of electricy your A/C unit will consumer per hour, the price of one kilowatt of electricity, and whether or not you have a 20% discount on your bill.

Function Name: makeSandwiches (25 points)

Parameters:

1. bread - an integer representing the number of slices of bread
2. peanutButter - an integer representing the number of spoons of peanut butter
3. jelly - an integer representing the number of spoons of strawberry jelly

Return Value:

An integer representing the amount of sandwiches you can make.

Test Cases:

```
>>>makeSandwiches(6, 3, 3)
2
>>>a = makeSandwiches(1, 2, 2)
>>>a
0
>>>makeSandwiches(11, 20, 20)
5
```

Description:
Write a function that returns the number of sandwiches you can make based on the number of slices of bread, spoons of peanut butter, and spoons of strawberry jelly. Assume that it takes the following to make 1 sandwich:

- 2 slices of bread
- 1.5 spoons of peanut butter
- 0.5 spoons of strawberry jelly

Note that the number of sandwiches must be a whole number; you cannot have 1 1/2 of a sandwich. For example, if you have 12 slices of bread, 9 spoons of peanut butter, but only 2 spoons of jelly, you can only make 4 sandwiches (despite the fact that you have enough bread and peanut butter to make 6 sandwiches). Likewise, if you have 3 slices of bread, 100 spoons of peanut butter, and 300 spoons of strawberry jelly, you can still only make 1 sandwich.

Hint: Python has a built-in function called min that takes in a comma separate list of numbers and returns the minimum value. E.g. min(5, 3, 7) returns the number 3. This may be useful in coming up with your solution.

Grading Rubric

Part 1 – Conversions – 50 points

- mass – 15 points
  - Function named correctly (mass) – 5 points
  - Performs correct conversion – 5 points
  - Prints the value – 5 points
- volume – 15 points
  - Function named correctly (volume) – 5 points
  - Performs correct conversion – 5 points
  - Print statement containing values to 2 decimal points – 5 points
- bears – 20 points
  - Function named correctly (bears) – 5 points
  - Performs correct conversion – 5 points
  - Print statement containing values to specified decimal points – 10 points

Part 2 - Complex Functions - 50

- calcEnergy – 25 points
  - Function named correctly (calcEnergy) – 5 points
  - Performs correct calculation – 10 points (Without discount)
  - Takes discount into account – 5 point
  - **Returns** the value – 5 points
- makeSandwiches – 25 points
  - Function named correctly (makeSandwiches) – 5 points
  - Performs correct calculation – 10 points
  - Does not return a floating value – 5 points
  - **Returns** the value – 5 points