**CIS 122 Spring 2015**

**Project 3: Strings and Things**  
**Due: Monday, April 20, 2015 6p**

**Goals**

By the end of this assignment, you should have had
- more practice using the function design recipe
- more practice writing and calling Python functions
- started using Python strings, input/output, and for loops
- practice with the difference between Python print and return

**Grading Rubric**

This project will be worth 30 points (60 graded). Each function is worth 10 points, with 2 points for the function header, 3 points for a docstring demonstrating use of function design recipe (type contract, description, examples), 3 points for the body of the function including the return statement, and 2 points for correct output when the function is executed.

**Getting Started**

For this project you will be writing and editing Python functions.  
You should use ONLY Python code that has been covered in textbook readings and class lecture/lab to solve the given problems.

Remember to include the docstring in each function after the function header. The docstring should reflect the use of the function design recipe, i.e., the format of each function docstring should match the examples in the text, e.g., on p. 47, with (a) type contract, (b) descriptive comment, and (c) examples.

Examples (function design recipe step 1) are given to you as part of the problem specification for each problem in this problem set. Hints for some of the problems are given at the end of the problem set.

You will test your code using the example function calls: in the Python Shell, call the function using the example input, and compare the result to the example output. You do not need to save or submit the Shell session.

**Finishing & submitting your work**

When you have completed all of the problems, add comments to the editor window to make sure the functions for each problem are clear. Do a final Save command to save the code in the editor window as a file with the name cis122project3. (On Macs, IDLE will automatically add ".py" to the file name, so the complete name of your saved file will be cis122project3.py.)

To submit your project, login to Blackboard. From the menu on the left hand side of the screen, choose "Projects". In the projects folder, choose "Submit Strings and Things". In Section 2 of the page that is displayed, scroll down to "Attach file" and choose "Browse My Computer". Locate cis122project3.py (the file you just created), and double click on it. The file name will appear on the "Attached Files" list.

At the bottom of Section 2, you will see a "Comments" window. This is where you credit all of the sources of any help you may have received on this assignment, including your partner if you are working in a programming pair. This is also the place to include any feedback you may have about the assignment and/or any remaining questions you may have.

Scroll down to Section 3 and hit the "Submit" button. You may re-submit your project up until the project deadline. Only the final submission will be graded.
(0) (CS Circles) Part 1 There is a single road between the two cities. The road has three bridges with weight limits $a$, $b$, $c$, as shown in the picture below:

To drive along the route, your truck needs to drive first over the bridge with weight limit $a$, then the one with weight limit $b$, then the one with weight limit $c$. Your truck will crash if you overload any of the three weight limits. Write a function, `max_trans`, that returns the maximum weight that can be transported along this road. Your function should take the values of $a$, $b$, and $c$ as input. (That is, $a$, $b$, and $c$ will be function parameters.)

Test `max_trans1` as follows:

```python
>>> max_trans(1, 2, 3)
1
>>> max_trans(9, 6, 3)
3
>>> max_trans(0, 0, 0)
0
```

Part 2:
There is also a second route consisting of two bridges, the first with weight limit d, and the second with weight limit e, as illustrated:

Your truck can take either route. Using the function from part 1 as a start, write a function, max_trans2, that returns the maximum weight that can be transported between the two cities. Your function should take the values of a, b, c, d, and e as input.

Test max_trans2 as follows:

```python
>>> max_trans2(126, 238, 326, 413, 515)
413
>>> max_trans2(222, 110, 411, 54, 73)
110
>>> max_trans2(227, 337, 135, 56, 73)
135
```

(1) Better Tip Calculator

(Re-)write a function, tip_calc, that has one parameter, a restaurant bill total, and prints a tip amount that is 18% of the total bill, formatted in standard money format (e.g., $5.45). Function tip_calc should return the value None.

Test your function on the following examples:

```python
>>> tip_calc(28.60)
$5.15
>>> tip_calc(48)
$8.64
>>> tip_calc(10)
$1.8  # [see challenge problems]
```

(2) Fancy Name Display

Ever wanted to see your name in stars? For this part, you’re going to write a fancy name printing function, nice_name. The function should prompt the user for their name, and then display their name prominently. Function nice_name should return the value None. At a minimum, the function should display the name in uppercase letters surrounded by * characters. However, you are encouraged to make your display as fancy as you want. A sample name display function might produce the following:

```python
>>> nice_name()

What is your name? Hermione

***************
*   HERMIONE   *
***************
```
(3) **Monograms-R-Us**

The local bath and body store has decided to add an instant monogram feature (for a low, low price) for customers purchasing bath towels.

Write a function, monogram, that asks a user for three inputs: their first name, middle initial, and last name, and returns a string of three uppercase initials in "monogram order" (first, last, middle).

```python
>>> monogram()
Please enter your first name: University
Please enter your middle initial: O
Please enter your last name: Ducks
'UDO'
```

(4) **Counting characters** (GCM, p. 127, ch. 7 #12)

Write the body of the following function:

```python
def char_ct(s1, s2, ch):
    """ (str, str, str) -> int

    Precondition: len(ch) == 1

    Return the total number of times that ch occurs in s1 and s2.

    >>> char_ct('color', 'yellow', 'l')
    3
    >>> char_ct('red', 'blue', 'l')
    1
    >>> char_ct('green', 'purple', 'b')
    0

    ""
    pass  # your code replaces pass
    return #total_ct
```

**If you want to get started on next week's project: Sunshine on a Cloudy Day**

The diameter of the Sun is 1,392,000 km, while the equatorial diameter of the Earth is 12,756 km (a ratio of about 109:1). Using Turtle Graphics, write a function, `sun_and_earth`, to illustrate the relative size of the sun and the earth by drawing two circles.
(Challenge – Python format method)
Explore the Python string method format examples in the text (GCM p. 121-122) and then write a better_tip_calc function using format to improve the monetary display.

(Challenge - Oxford word challenge)
An anagram, says the Concise Oxford Dictionary, is 'a word or phrase formed by transposing the letters of another word or phrase'. Can you transpose these words or phrases to make new words or phrases?

a) Set up a variable to have the value of the original word.

b) Answer the riddle.

c) Use string operators and the original word to construct the answer string. (Extra challenge: use as few operations as possible.)

Example: Rearrange lemon to make something else to eat.

a) original = ‘lemon’

b) Answer: melon

c) print(original[2] + original[1] + original[0] + original[3:])

1. Rearrange the letters of the word eat to make something to drink.
2. Rearrange brush to make a kind of bush.
3. Rearrange rail to make a dishonest person.
4. Rearrange skate to make a poet.
5. Rearrange wand to make a time of day.
6. Rearrange Alice to make another woman's name.
7. Rearrange Dorothea to make another woman's name.
8. Rearrange supersonic to get part of an orchestra.
9. Rearrange hormone to get the name of a bird.
10. Rearrange spectrum to make some eatables.
11. Which food might you make out of stale lamb?
12. What do you have to be if you want to listen?

(Send me an email if you want me to give you the answers to the riddles.)

HINTS:
(1) Use round and a type coercion function.

(2) Don't forget that the "*" operator works for strings.

(3) Turtle graphics functions you will need: circle.
Turtle graphics functions you may want to explore: color, begin_fill, end_fill; also, possibly: hideturtle, penup, pendown, and setpos.