Math 308: Bridge to Advanced Math
Programming Assignment 2

Instructions.

- Use IDLE to create a text file entitled `pa2.py` containing the functions described in the problems listed below. It is okay to include other functions to help you do the problems.
- After you create each function you should test that it works correctly.
- Upload this file to blackboard. To do this, log in to blackboard. Click on "Programming Assignments" in the left panel, then click "Programming Assignment 2". Click on the button "Browse My Computer" and select your `pa2.py` file. Then click the "Submit" button at the bottom of the page.
- You do not need to record your name in the `pa2.py` file. Blackboard knows your name and relays that information to me.

Grading. You work will be graded based on the following criteria:

- If the file generates an error when loaded into Python, your grade will be a zero. (You should test your file!)
- I will call the functions listed in the problems with the *exact* names listed below with test cases. Your program should return a correct answer. You will get full credit for that case if it returns a correct answer, and zero credit if it returns a wrong answer, returns no answer, or generates an error.
- The correct response must be returned from each function. (Printing it out will not count. Use the `return` statement!)

Problems. Inside the file `pa2.py` place functions as described below which solve each of the following problems. Only include these three functions, and be sure to title them as described in the problems. (Improperly titled functions will not be called properly.) To write these functions, it should be sufficient to understand the documents on Logic and on Tuples and Sets listed on the course programming page.

1. Write a function named `implies(P,Q)` that takes as input two boolean values P and Q and returns the truth value of the statement "P implies Q." (You can test your function using the `print_truth_table` function given on the Logic page.)
2. Write a function named `iff(P,Q)` that takes as input two boolean values P and Q and returns the truth value of the statement "P iff Q."
3. Say that a *2-input truth function* is a function which takes as input two boolean values P and Q and returns a boolean value.(Examples include xor above, as well as implies and iff from the problems above.)

   Write a function named `logically_equivalent(tf1, tf2)` that takes as input two 2-input truth functions `tf1` and `tf2` and returns the boolean value of the statement "The two truth functions are logically equivalent." Your function should return `True` if and only if the two functions `tf1` and `tf2` return the same output whenever they are passed the same input values.

Examples of successful input/output:

```
>>> def or_not(P,Q):
    return (not P) or (not Q)

>>> def not_and(P,Q):
    return not (P and Q)

>>> logically_equivalent(or_not, not_and)
True
>>> def or_function(P,Q):
    return P or Q

>>> logically_equivalent(or_function, not_and)
False
```

4. Write a function `planar_distance(p,q)` which takes as input two points in the plane and returns the distance between them. Here a *point in the plane* should be interpreted as a 2-tuple whose entries are both real numbers.

   Examples of successful input/output:

```
>>> print(planar_distance( (0,0), (1,1) ))
1.4142135623730951
>>> print(planar_distance( (1,1), (4,5) ))
5.0
```

5. Suppose A and B are two sets of numbers. Their *sumset* is the set

$$\{a+b \ : \ a \in A \text{ and } b \in B\}.$$

   Write a function called `sumset(A,B)` which takes as input two sets of numbers A and B and returns their sumset.

   Examples of successful input/output:

```
>>> sumset({1,2,3},{1,2,3})
{2, 3, 4, 5, 6}
>>> sumset({-1,1},{3,10})
{9, 2, 11, 4}
```