Problem Wk.2.1.1: Fun with Functions

Part 1: Function Types

For each of the following functions, specify the type of the output. If it can be either an int or a float, use `num`, which isn't a real Python type, but which we'll use to indicate that either basic numeric type is legal.
In fact, in Python, booleans `True` and `False` can be operated on as if they were the integers 1 and 0; but it is ugly and confusing to take advantage of this fact, and we will resolutely pretend that it isn't true.

1.
```
def a(x):
    return x + 1
```
Result:
noneType
num
int
float
boolean

2.
```
def b(x):
    return x + 1.0
```
Result: noneType

3.
```
def c(x, y):
    return x + y
```
(Assume that `x` and `y` are ints or floats.) Result: noneType

4.
```
def d(x,y):
    return x > y
```
Result: noneType

5.
```
def e(x, y, z):
    return x >= y and x <= z
```
Result: noneType

6.
```
def f(x, y):
    x + y - 2
```
Result: noneType

Part 2: Transcript

Below is a transcript of a session with the Python shell. Assume the functions from part 1 have been defined. Provide the type and value of the expressions being evaluated. If evaluating an expression would cause an error, select noneType and write error in the box. If the value of an expression is a function, select function as the type and write function in the box.

1. a(6)

This problem contains one or more multiple-choice questions. When a given set of choices is used for the first time, all choices are displayed. If there are several related questions, you can assume that the same choices are available for all questions.

noneType
num
int
float
boolean
function

2. a(-5.3)
   noneType
3. a(a(a(6)))
   noneType
4. c(a(1),b(1))
   noneType
5. d(10, 11.1)
   noneType
6. e(a(3),b(4),c(3, 4))
   noneType
7. e
   noneType

6.01SC Introduction to Electrical Engineering and Computer Science

Spring 2011