Practical Questions

Question # 1

Determine the potential error(s) associated with each question (either logical or stylistic errors) and fix the problem(s) in the most efficient way possible.

a) X=1

b) my_var = “Hello’

c) 2 = ‘Brian Rocks’

d)
```
def Greeter():
print(‘Hello”)
```

e)
```
def mystery():
a = 2 b = 3  c = 4  d = 5
a = b + c   d = a + b
c = a + d   b = a + c
return (a, c, b, d)
```

f)
```
def greater_than(num1,num2,num3):
if(num1>num2):
    answer =  1
if(num2 < num1):
    answer =  2
if(num1 > num3):
    answer = 1
if(num3 > num1):
    answer =  3
if(num2 > num3):
    answer =  2
if(num3 > num2):
    answer =  3
return answer
```

g)
```
def greater_than2(num1,num2,num3):
    if(num1>num2):
        return 1
    elif(num2 < num1):
        return 2
    elif(num1 > num3):
        return 1
    elif(num3 > num1):
        return 3
    elif(num2 > num3):
        return 2
    elif(num3 > num2):
        return 3
    else:
        return 0
```

h)
```
def my_func(a,,b,c,d)
    return ((a)*4 + (d/b)*c – (a/d) * b + (a+d/c)*b)
```

Question # 2

Write the code for the function described for each questions. Remember to follow the design recipe as taught in lecture (you have to provide your own function description).

a) (str, float, str) -> bool
Create a function that takes in a student's gpa, name, and program code and returns whether a student should be accepted into a course. (You can use if statements)

b) Create a function that takes in a list of 5 names and returns whether the name "Brian" is one of them.

c) Create a function that takes in a temperature in Celsius and determines whether this is above the melting point of gold in Kelvin. (Melting point of gold is 1948^{o}F)

d) Create a function that takes in a string and returns the string repeated twice if the original string has 9 characters or fewer and returns the string with the cases swapped if the string is longer than 9 characters long.

e) Create a function that takes in a number and returns the string "Big" if it has more than 3 digits and returns the string "small" if it has less than 4 digits.

f) Create a function that takes in a string and an integer and returns the number of character corresponding to the number given, starting from the end of the string. The function should add underscores to the beginning of the string if the string's length is shorter than the number given.

g) **Challenge**

https://open.kattis.com/problems/measurement

h) **Challenge**

https://open.kattis.com/problems/billiard

Discussion Questions

Question # 3

When trying to differentiate on by more than one that one Boolean expression, you can either use nested if statements or use an if/elif/else statement. Is one better than the other? What's one case where nested if statements would be better? What's one case where an if/elif/else statement would be better?

Question # 4

As you have seen through if statements and the exam, Boolean expressions can be used for a wide range of tasks. Is there any other concepts that are taught in lecture that are related to Boolean expressions? Why is it that "a" == ("a" and "b") is False while "b" == ("a" and "b") is True? Why is it that "a" == ("a" or "b") is True while "b" == ("a" or "b") is false?

Question # 5, The "Logic Question" **(Challenge)**

Every Monday Brian takes a walk in the valley, by the Miller Lash House to the tennis courts; back up the Pan-Am path to his office. On Tuesdays he walk the exact same route in reverse. Brian always leaves his office at precisely 9:00am, and takes exactly 1 hour to walk the route, regardless of direction. The route is exactly 3km[1]. Last week, Brian noticed something interesting. One particular shadow made him think 'I saw that in the exact same spot yesterday', I must've been standing exactly here at exactly this time yesterday.

Brian likes to vary his speed, some days he sprints the first half and walks the 2nd half of his route, other days he walks consistently the same speed for the full hour, but each day the total journey takes exactly the same amount of time.

The thing Brian wants to know is: How often does this happen? How often will he wind up in a spot at some time on Tuesday such that he was in the exact same spot at the same time the day before? Let's assume that by 'same spot, same time', we mean within 1m and within 1 second.

[1] Hey… if I'm going to create a fictional universe where I'm in better shape, might as well create one where I'm more punctual as well ☺