

Assignment 1

This assignment has two components.

The first part are questions relating to totality.

Here I provide you with code, and you are asked to answer questions about it.

You may benefit from running the code that I give you with python, so you can test things yourself.

For the second part of this assignment, you are asked to write programs.

In many cases, we will have done these programs in class, or have written similar programs.

There is no problem with your code being identical to what was presented in class.

Your submission for this assignment should be a zipped folder with several files.

Submit your assignment on canvas. If you cannot do this, email me the assignment.

One file for the written problems, and one file for each programming problem.

This assignment is worth 100 points.

Part 1. 25 points

For each of the following functions, indicate if the function terminates for all integer values of x .

If it does not terminate on all integer values of x ,

give an example value of x for which it does not terminate.

```
def foo(x):  
    return x
```

```
def bar(x):  
    if x > 0:  
        return x  
    else:  
        return bar(x*-1)
```

```
def baz(x):
```

```
if x > 0:
    return baz(x-1) + 1
else:
    return 0
```

```
def quux(x):
    return quux(x-1)
```

```
def f(x):
    if x > 0:
        return f(x+1)
    else:
        return f(x-1)
```

Part 2. 75 points total

1. 20 points.

We have seen the factorial function a few times,
and implemented it both in a functional and imperative style.

Write a factorial function in python.

In other words, complete the following

```
def factorial(n):
    #your code here
```

2. 5 points.

Now that you have the factorial function, you might wish to test it.

Write a complete program that accepts an input number from the user (using the `input()` function),
computes that number factorial, and then outputs the result (using the `print()` function)

3. 20 points.

Write a function that prints the numbers 1 through n, inclusively.

That is, starting your code as follows:

```
def print_numbers(n):  
    #your code here
```

upon calling `print_numbers` with the argument 5, output should look like:

```
1  
2  
3  
4  
5
```

4. 5 points.

As with problem 2, write a complete program that accepts an input number from the user, and calls `print_numbers` with that number.

5. 20 points.

Write a function that accepts two arguments, and checks to see if the first is a multiple of the second.

```
def is_multiple(n,m):  
    #your code here
```

Thus `is_multiple(10,5)` should return `True`, while `is_multiple(14,3)` would return `False`.

6. 5 points.

write a complete program that accepts two input numbers from the user, and calls `is_multiple` with those numbers.