**Part 1**

A python function “avg” returns the average of three values, as shown below:

*def avg (num1, num2, num3):*

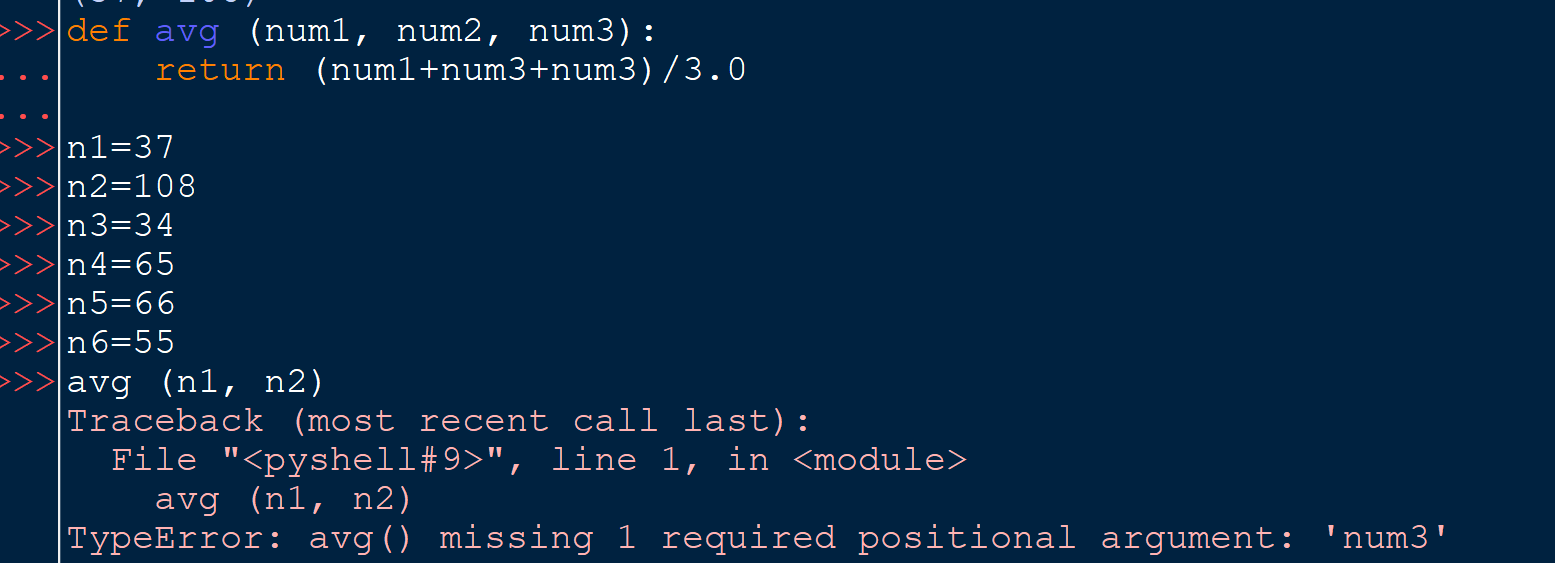
*return (num1 + num2 + num3) / 3.0*

*n1 = 37, n2 = 108, n3 = 67*

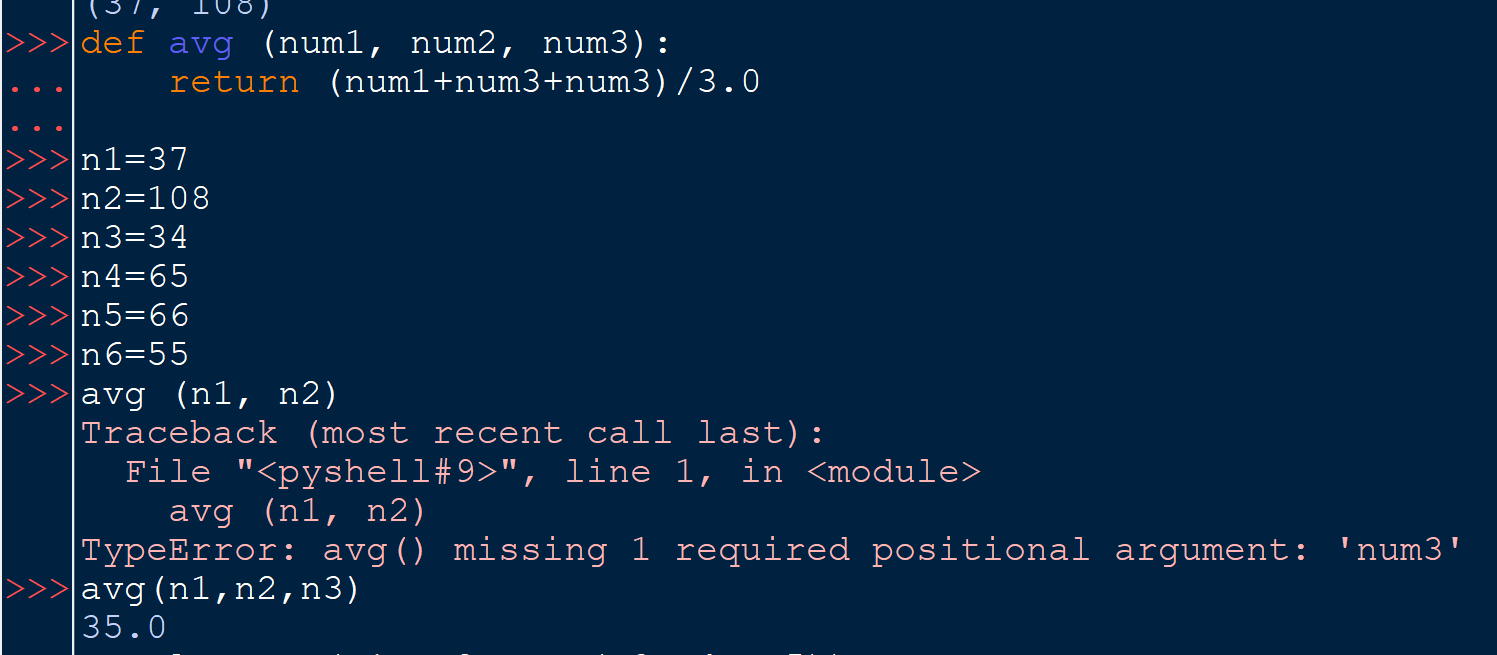
Define the function and variable declarations given above in IDLE shell and execute the following expressions. Which of the statements are valid?

**Note down the response to each. Do they differ from what you would expect?**

1. *result = avg(n1, n2)*



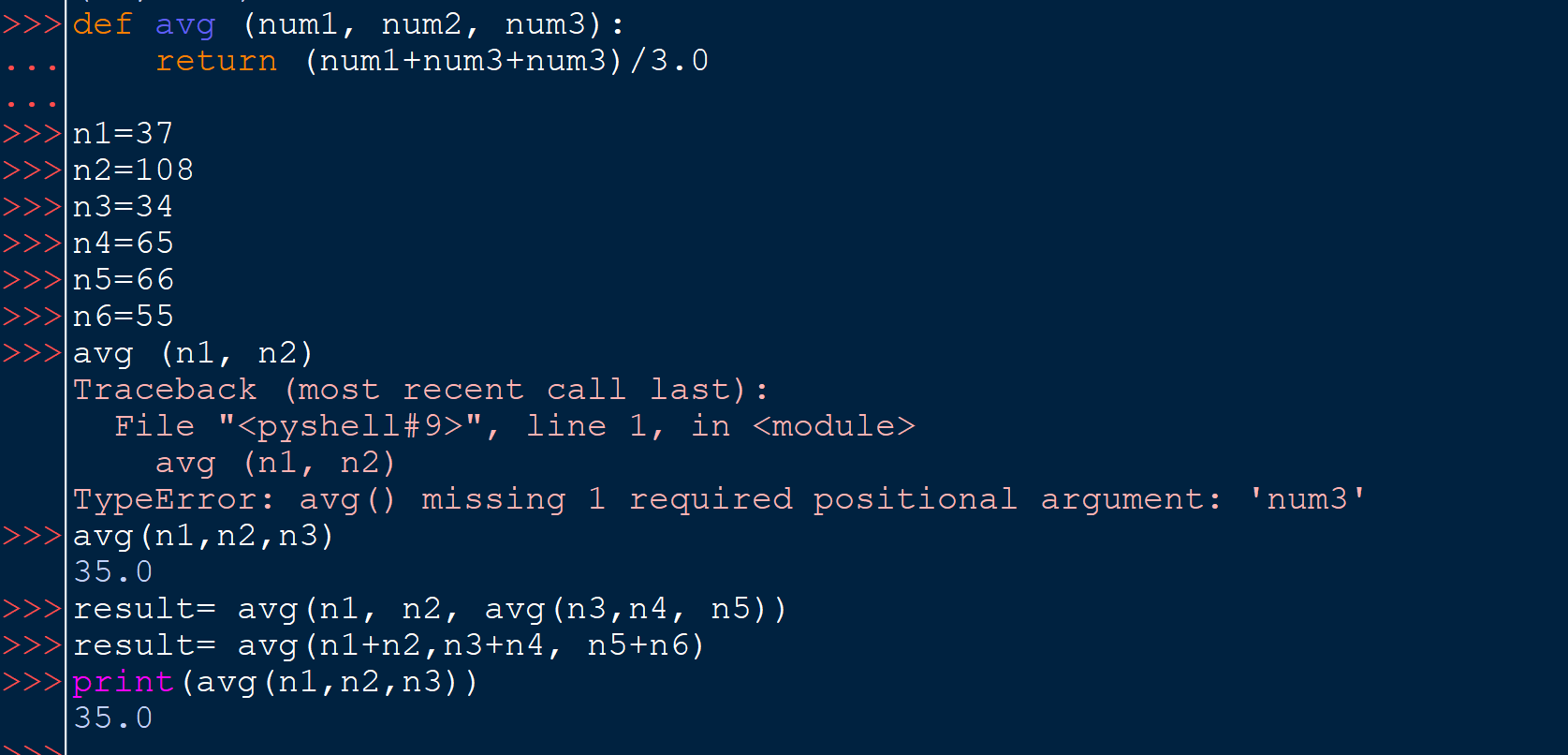
1. *avg(n1, n2, n3)*

**

1. *result = avg(n1 + n2, n3 + n4, n5 + n6)*

(D) *print(avg(n1, n2, n3))*

(E) *result = avg(n1, n2, avg(n3, n4, n5))*



**Part 2**

Define a function:

(A) *types( )* that prints a given value both as a float and an integer

(B) *squared( )* that take an integer and returns the value squared.

(C) *int\_to\_string( )* that takes an integer value and returns it as a string.

(D) *hello\_world( )* that takes a parameter name and displays the following output to the console: "Hello World, my name is name".

(E) *print\_ast( )* that takes an integer value *n* and a string value symbol, with a default value of "\*". This character should be printed *n* times to the console.

(F) *improved\_average( )* that takes five integer parameters. It should return the mode, median and mean values of the numbers passed to the function.

(G) *either\_side( )* which when passed an integer value also prints the values which are one less and one more than that value e.g.

*"You typed 4, one less than 4 is 3, one more than 4 is 5"*

**Part 3**

1. Create a function that prompts the user for two integer values and displays the results of the first number divided by the second to two decimal places.

2. Create a Python program called calculator with functions to perform the following arithmetic calculations, each should take two decimal parameters and return the result of the arithmetic calculation in question.

A. Addition

B. Subtraction

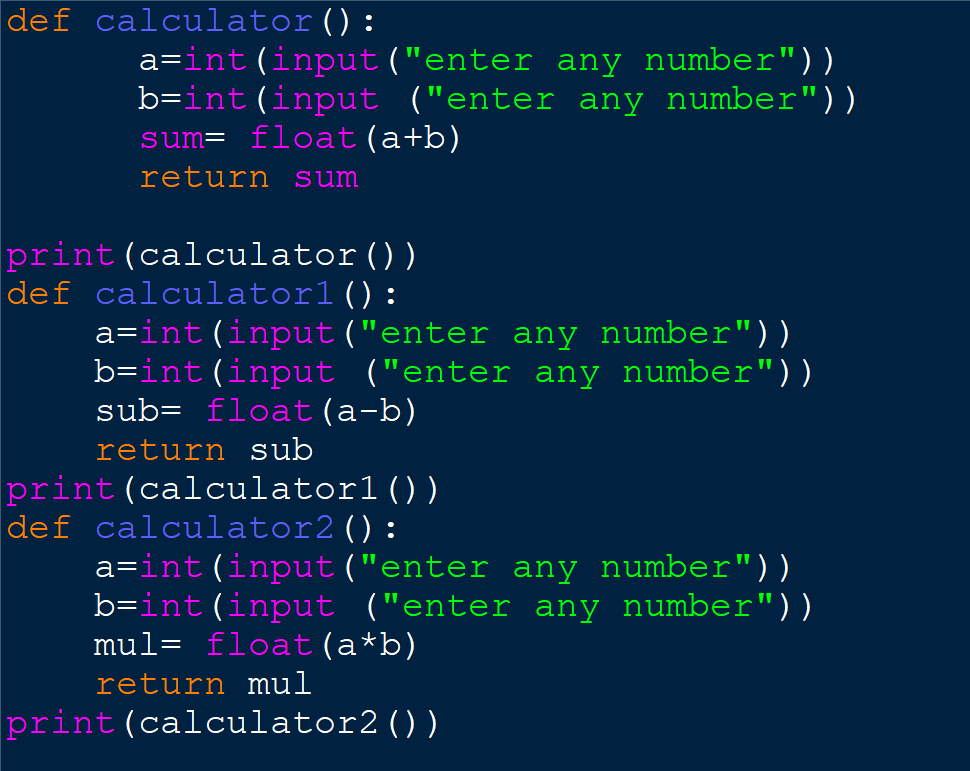
C. Multiplication

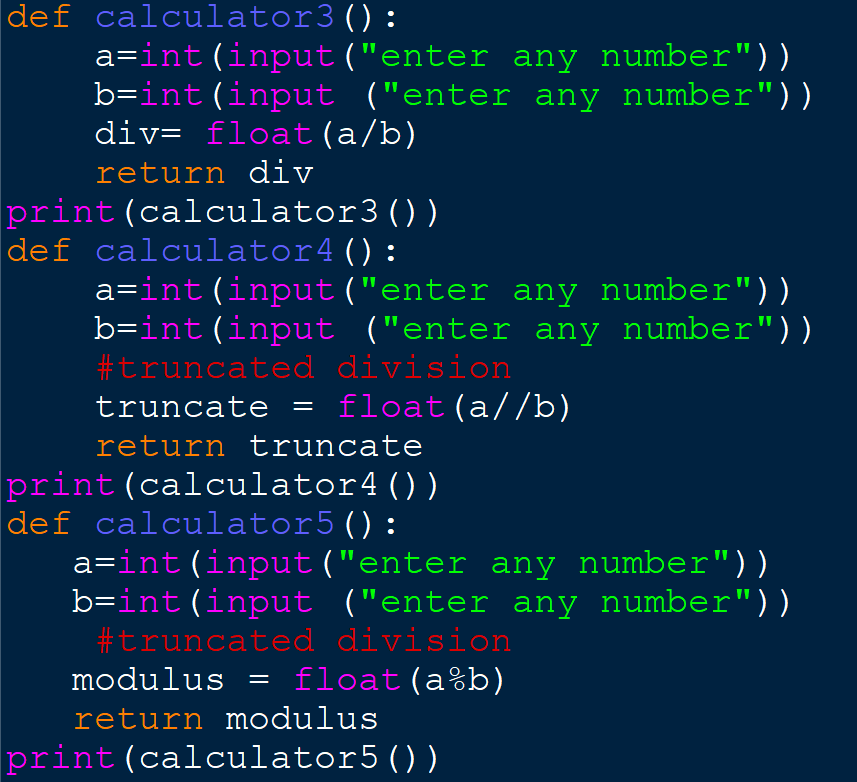
D. Division

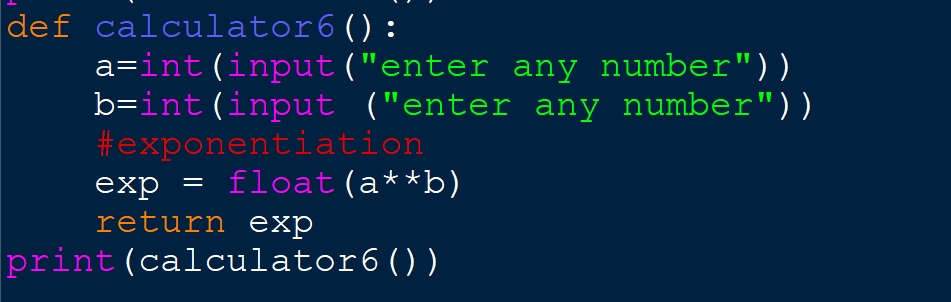
E. Truncated division

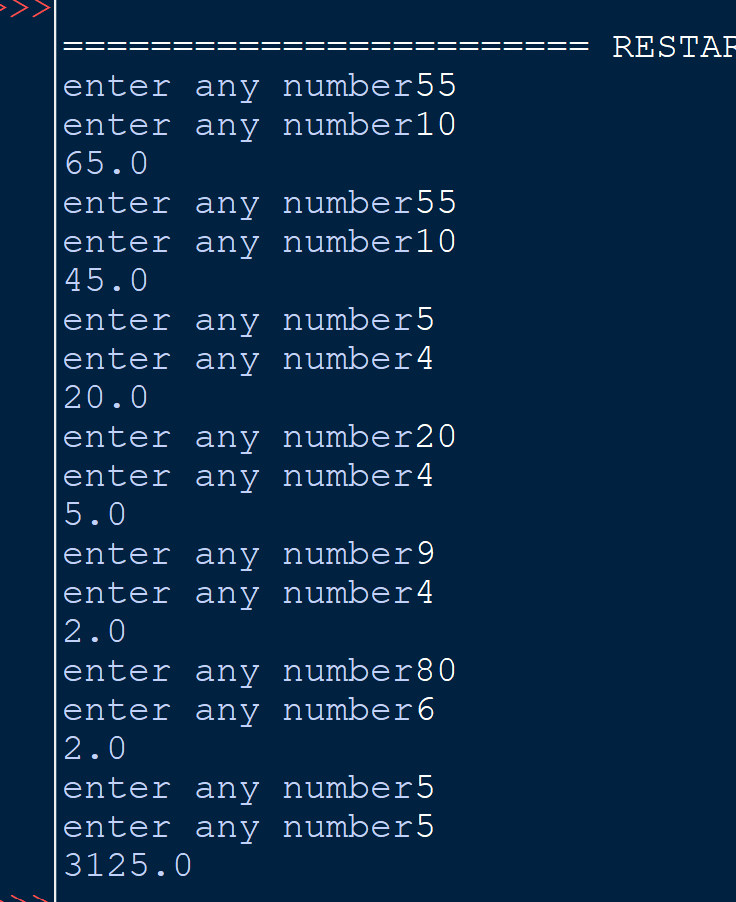
F. Modulus

G. Exponentiation









3. Go back and add multi-line Docstrings to each of the functions you defined in the previous question. Use the help function to check them afterwards.

4. Take a character input from the user and convert the character into next character in the alphabetical order. Use *ord( )* and *chr( )* ASCII functions.

[Hint: for input of ‘a’, print ‘b’ and so on]

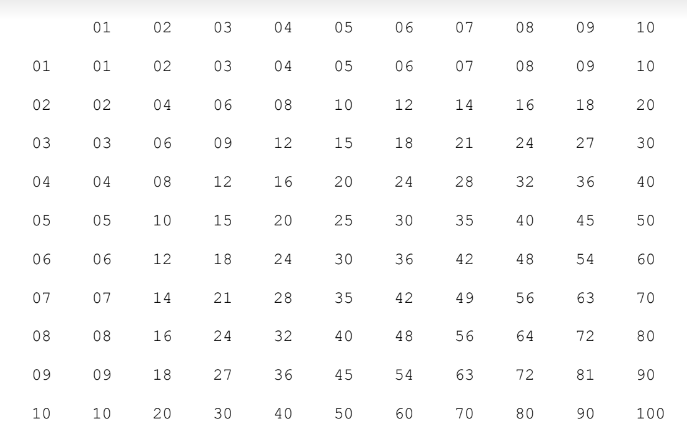
5. Use a looping statement to take user’s choice to continue for the above program.

**Part 4 (Optional)**

You will need to understand control structures to complete the following questions. Therefore, you should carry out some independent research before attempting this. However, it will also be covered next week in class.

1. Create a function *multiplication\_table( )*. It should take a single parameter *n*, which determines the size of the grid to be output e.g.

*multiplication\_table(10)*



2. Modify your existing function to take an additional parameter: *power*, with a default value of *False*. If a value of *True* is provided, your multiplication table should instead apply the top row as *powers* instead of multiplying the numbers.

multiplication\_table(3, True)

