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?

(A) *result = avg(n1, n2)*

This is not valid because it is missing one parameter.

(B) *avg(n1, n2, n3)*

Print is missing here.

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

Extra parameters is added in here.

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

This is the valid one.

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

Extra parameters are added here.

Part 2

Define a function:

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

```
def types(val):
    a = float(val)
    b = int(val)
    print('an float:',a)
    print('an integer:',b)
print(types(2))
```

```
an float: 2.0
an integer: 2
None
```

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

```
def types(val):
    a = float(val)
    b = int(val)
    print('an float:',a)
    print('an integer:',b)
print(types(2))

def Squared(val):
    return val*2
print('squared value is :',Squared(2))
```

```
squared value is : 4
macbookpro@Avishek-macbook-pro untitled fol
```

(C) *int_to_string( )* that takes an integer value and returns it as a string.

```
def int_to_string(num):
    return str(num)

print(int_to_string(2))
```

```
2
```

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

```
def hello_world():
    """hello_world"""
print(hello_world.__doc__)

def name():
    name = input("My name is Abhishek")
    return name
print(name())
```

```
hello_world
My name is Abhishek
```

(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.

```
def print_ast(n, symbol = '+'):
    print(symbol * n)

print_ast(4)
```

```
++++
macbookpro@Avishek-macbook-pro untitle
```

(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"

```
def either_side(n):
    less = "You typed ",n," the value less than ",n, " is ", n-1
    more = "You typed ",n," the value more than ",n, " is ", n+1
    return less,more


print(either_side(2))
```

```
(('You typed ', 2, ' the value less than ', 2, ' is ', 1), ('You typed ', 2, ' the value more than ', 2, ' is ', 3))
```

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.

```
def divide():
    num3 = int(input("Enter a number: "))
    num4 = int(input("Enter another number: "))
    answer = round(num3/num4,2)
    print(num3, "/", num4 ,"is", answer)

divide()
```

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

```python
def add(a,b):
    return a+b
def sub(a,b):
    return a-b
def multi(a,b):
    return a*b
def divide(a,b):
    return a/b
def truncated_division(a,b):
    return a // b
def modulus(a,b):
    return a % b
def exponent(a,b):
    return a ** b

n1 = int(input('enter 1st number:'))
n2 = int(input('enter 2nd number:'))

print(n1,'+',n2,'=',add(n1,n2))
print(n1,'-',n2,'=',sub(n1,n2))
print(n1,'*',n2,'=',multi(n1,n2))
print(n1,'/',n2,'=',divide(n1,n2))
print(n1,'//',n2,'=',truncated_division(n1,n2))
print(n1,'%',n2,'=',modulus(n1,n2))
print(n1,'**',n2,'=',exponent(n1,n2))
```

```
enter 1st number:6
enter 2nd number:5
6 + 5 = 11
6 - 5 = 1
6 * 5 = 30
6 / 5 = 1.2
6 // 5 = 1
6 % 5 = 1
6 ** 5 = 7776
macbookpro@Avishek-macbook-pro
```

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.

at and int.py > ...

```
    return a+b
def sub(a,b):
    """this sub a-b"""
    return a-b
def multi(a,b):
    """this multi a*b"""
    return a*b
def divide(a,b):
    """this divide a/b"""
    return a/b
def truncated_division(a,b):
    """this runcated_division a//b"""
    return a // b
def modulus(a,b):
    """this modulus a % b"""
    return a % b
def exponent(a,b):
    """this exponent a ** b"""
    return a ** b
def name():
    print(add.__doc__)
    print(sub.__doc__)
    print(multi.__doc__)
    print(divide.__doc__)
    print(truncated_division.__doc__)
    print(modulus.__doc__)
    print(exponent.__doc__)
name()
```

```
this add a+b
this sub a-b
this multi a*b
this divide a/b
this runcated_division a//b
this modulus a % b
this exponent a ** b
macbookpro@Avishek-macbook-pro
```

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]

```
def addCharacter():
    char = input("Enter a character: ")
    if char.isalpha(): #checks if it is a character from A_Z or not and returns true or false
        nextChar = ord(char) + 1  #ord gives ascci value and when we add one it gives another ascci value
        print("ord value of ",char, " is " ,nextChar) #prints the ascci value
        next_char = chr(nextChar)  #chr gives the character value
        print("next chars after",char," is", next_char)

    else:
        print() -> None acter from A-Z only")
addCharacter()
```

```
Enter a character: c
ord value of  c  is  100
next chars after c  is d
```

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

```
def add(a,b):
    """This function adds
    and returns a + b"""
    return a + b

def subtract(a,b):
    """This subtracts
    a -b"""
    return a - b

def multiply(a,b):
    """ This function Multiplies
    a* b"""
    return a * b

def divide(a,b):
    """ This function divides
    a/b"""
    return a / b

def truncated_division(a,b):
    """This function truncated division
    a//b"""
    return a // b

def modulus(a,b):
    """This function modulus
    a % b"""
    return a % b

def exponent(a,b):
    """This function exponent
    a**b"""
    return a ** b
```

```
while True:
        print("Select operation from 1-7.")
        print("1:add")
        print("2:subtract")
        print("3:multiply")
        print("4:Divide")
        print("5:truncated division")
        print("6:modulus")
        print("7:exponent")
        print("4:Divide")


        choice = input("Enter your choice (1/2/3/4/5/6): ")

        if choice in ('1', '2', '3', '4','5','6','7'):
            num1 = float(input("Enter the first number: "))
            num2 = float(input("Enter the second number: "))


        if choice == '1':
                print (num1 ,"+" ,num2 ,"=", add(num1,num2))

        elif choice == '2':
             print (num1 ,"-" ,num2 ,"=", subtract(num1,num2))

        elif choice == '3':
            print (num1, "*" ,num2 ,"=", multiply(num1,num2))

        elif choice == '4':
                print (num1 ,"/" ,num2 ,"=", divide(num1,num2))

        elif choice == '5':
                 print (num1 ,"//" ,num2 ,"=", truncated_division(num1,num2))

        elif choice == '6':
                print (num1 ,"%" ,num2 ,"=", modulus(num1,num2))

        elif choice == '7':
            print (num1 ,"**" ,num2 ,"=", exponent(num1,num2))
            break

        else:
                print("Wrong choice. Please chose from 1-7")
```

```
choice = input("Enter your choice (1/2/3/4/5/6): ")

if choice in ('1', '2', '3', '4','5','6','7'):
    num1 = float(input("Enter the first number: "))
    num2 = float(input("Enter the second number: "))


if choice == '1':
      print (num1 ,"+" ,num2 ,"=", add(num1,num2))

elif choice == '2':
    print (num1 ,"-" ,num2 ,"=", subtract(num1,num2))

elif choice == '3':
    print (num1, "*" ,num2 ,"=", multiply(num1,num2))

elif choice == '4':
      print (num1 ,"/" ,num2 ,"=", divide(num1,num2))

elif choice == '5':
       print (num1 ,"//" ,num2 ,"=", truncated_division(num1,num2))

elif choice == '6':
      print (num1 ,"%" ,num2 ,"=", modulus(num1,num2))

elif choice == '7':
    print (num1 ,"**" ,num2 ,"=", exponent(num1,num2))

else:
      print("Wrong choice. Please chose from 1-7")

ending = input("Do you wish to continue? yes or no")
if(ending == "yes"):
    continue
else:
    print("calculator batter dead")
    break;
```

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)

	01	02	03	04	05	06	07	08	09	10
01	01	02	03	04	05	06	07	08	09	10
02	02	04	06	08	10	12	14	16	18	20
03	03	06	09	12	15	18	21	24	27	30
04	04	08	12	16	20	24	28	32	36	40
05	05	10	15	20	25	30	35	40	45	50
06	06	12	18	24	30	36	42	48	54	60
07	07	14	21	28	35	42	49	56	63	70
08	08	16	24	32	40	48	56	64	72	80
09	09	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

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)

	01	02	03
01	01	01	01
02	02	04	08
03	03	09	28