IT1007 Introduction to Programming with Python and C
Lab Exercise 01

Submission instructions:

1. There are two parts in this lab. For Part A, you have to submit within the same day of your lab session. For your Part B, you will submit it before the deadline announced. There are two separate skeleton files for each part of the lab. Please submit them separately. **Please do NOT merge the two parts into one single file**.
2. Please do **NOT** change the names of your files.
3. Submit your solution to the **CORRECT** group folder in the IVLE Workbin.
4. You have to name your **functions** exactly (case sensitive) as the questions stated.

Failure to follow each of the instruction will result in 10% deduction of your marks.

Part A (Deadline: Same day of your Tlab)

Question 1

a. Write a function, **`sigma`**, that takes in an integer n, and returns the sum of the first n positive integers.

Sample output:

```
>>> sigma(1)
1
>>> sigma(2)
3
>>> sigma(5)
15
>>> sigma(100)
5050
```

b. Write another function, **`sigma_squared`**, that takes in an integer, n, and returns the sum of the first n-squared positive integers. Example: Suppose n is 4.

$$1^2 + 2^2 + 3^2 + 4^2 = 1 + 4 + 9 + 16 = 30$$

Sample output:

```
>>> sigma_squared(1)
1
>>> sigma_squared(2)
5
>>> sigma_squared(5)
55
>>> sigma_squared(100)
338350
```

Question 2

a. Write 2 functions, **`is_even`** and **`is_odd`**, which takes in an integer, n, and returns `True` or `False` accordingly. You must implement the functions using if-else.

Sample output:

```
>>> is_even(0)
True
>>> is_even(1)
False
>>> is_even(4)
True
>>> is_odd(0)
False
```

```
>>> is_odd(1)
True
>>> is_odd(4)
False
>>> is_odd(5)
True
>>> is_odd(-1)
True
```

b. Same question as 2a, except that you must implement the functions in one line. Please name your functions **`is_even2`** and **`is_odd2`** respectively. You cannot call another **`is_odd`** or **`is_even`** functions

c. Same question as 2a, except that you only need to write 1 function, **`is_odd3`**, and you are only allowed to implement it by calling `is_even`.

Part B (Deadline: Announced in IVLE)

Question 3

A number n is *prime* if it is divisible by itself and 1 only. Write a function, **`is_prime`**, which takes in an integer n > 1, and returns `True` if the number is a prime number and `False` otherwise.

Sample output:

```
>>> is_prime(2)
True
>>> is_prime(4)
False
>>> is_prime(7)
True
>>> is_prime(331)
True
```