Exercise for Python Tutorials

Unit 1

Practice working with the Python interpreter in both the interactive mode and the scripting mode.

I(a). Draw a grid (using User Defined Functions):

```
+ - - - - + - - - - +
|         |         |
|         |         |
|         |         |
+ - - - - + - - - - +
```

(b). Develop a program which uses this grid as basic block to build a larger grid as shown below:

```
+ - - - - + - - - - ++ - - - - + - - - - +
|         |         ||         |         |
|         |         ||         |         |
|         |         ||         |         |
+ - - - - + - - - - ++ - - - - + - - - - +
+ - - - - + - - - - ++ - - - - + - - - - +
|         |         ||         |         |
|         |         ||         |         |
|         |         ||         |         |
+ - - - - + - - - - ++ - - - - + - - - - +
```

II(a). Write a function named is_triangle that takes 3 integers as arguments and prints either 'YES' or 'NO', depending on whether you can or cannot form a triangle from the 3 integers.

(b). Write a function that prompts the user to input the length of the 3 sides, converts them into integers. [Hint: any one length cannot be greater than sum of the other two]

III. Write a Python program that reads an number from the user and prints the square of the number if and only if: the number is a positive odd integer.

IV. The built-in function 'eval' takes a string and evaluates it using the Python interpreter. Write a function called 'eval_loop()' that iteratively prompts the user, takes the resulting input and evaluates it using 'eval' and prints the result.

V. Write a function test_sqroot that prints a table like this:

```
num    math.sqrt(num)        Newton's method
1.0    1.0                   1.0
2.0    1.414...              1.414...
3.0    ...          ...
```

The first column is the number, second column is the square root of number using math.sqrt() and the third column is the square root of number estimated from Newton's method.

VI. The Ackermann function, A(m,n) is defined as:

```
n+1                    if m=0
A(m,n) =  A(m-1,1)     if m >0 and n =0
A(m-1, A(m,n-1))       if m > 0 and n >0
```

Write a function named ack() that evaluates Ackermann's function

VII. A number, a, is a power of b if it is divisible by b and a/b is a power of b. Write a function called is_power() that takes parameters a and b and returns True if a is a power of b.
[Note: you will have to think about the base case.]

VIII. Find the factorial of a positive integer using recursion

IX. Find the sum of Natural numbers using recursion

X. Find the gcd of 2 numbers using recursion

XI. Find the tribonacci series (0, 1, 1, 2, 4, 7, 13, 24, 44, ...)

(a) Upto N terms

(b) Last term <= N