Recursion:

What is recursion?

Recursion is the process of solving a problem by breaking it up into smaller sub-problems, and solving these. This may sound confusing at first, but is an important technique, and there are some problems (listed later as additional exercises), which can only be solved by thinking recursively! Often, when you are faced with a large task, you accomplish it by breaking the task into smaller, more manageable components and then completing these in order to complete the overall task. Recursion applies this idea to programming.

Excercise:

Read **example.py.** Open it using Notepad++ (Right click the file and select Edit with Notepad++) If you do not have Notepad++ installed, please install it using the instructions located at www.rmoola.com/pythonlessons.html.

Example.py should help you understand some recursion. Every task will have example code to help you get started. Make sure you read all of **example.py** and try your best to understand.

You should run **example.py** to see the output. The instructions on how to do this are inside the file. Feel free to write and run your own example code before doing this Task to become more comfortable with Python.

Compulsory tasks:

Now that you have read and understood example.py, write recursive functions for the following tasks. Write all of the recursive functions in a file called "recursion.py":

Reverse a string
Find the factorial of an integer (factorial (5) = 5 x 4 x 3 x 2 x 1)
Calculate the nth Fibonacci number (Fibonacci numbers are a sequence where each number is the sum of the previous two - 0 1 1 2 3 5 8 etc.)

Bonus task 1:

Print out the first n Fibonacci numbers (using the previous Fibonacci function is useful. Do this recursively, NO LOOPS)

Bonus task 2:

Implement a search/replace function recursively

Sample I/O
Enter string:
hello world
Enter word to find:
llo
Enter word to replace:
@@
he@@ world

Bonus Challenge:

The flood fill algorithm is commonly used in image processing such as Microsoft Paint. Two good examples are shown in the attached images. Write a recursive program to do this (but on text or numbers, unless you want to work with real images)

Example:

Original grid

[0, 1, 0]
[0, 1, 1]
[0, 0, 0]
Fill (x = 0, y = 0)
[1, 1, 0]
[1, 1, 1]
[1, 1, 1]