Lab 04 - Recursion

Before you come to the lab

1. Read this document carefully to properly prepare for the lab and turn in your lab solution (i.e., your lab report as per the instructions presented here).
2. Read Sections 4.1-4.7.

Prelude

You will be creating test cases for this lab. Therefore, as per the guidelines provided in the testing in Python website (https://realpython.com/python-testing/), create the following folder for your lab04:

```
lab04/
│
├── mySolution/
│   └── __init__.py
├── test.py
├── report.ipynb
└── tree.png
```

- `__init__.py` should contain the Python code you have developed as solutions for the exercises in this lab assignment.
- `test.py` will contain your tests.
- `report.ipynb` is your report (a jupyter notebook file).
- `tree.png` is an image file part of your solution for Exercise 1 below.

Exercise 1

Nota Bene For this exercise I suggest you don't use Jupyter Notebook to run your code, because it does not interface well with the turtle library. I suggest you run it from the command line or from your favorite IDE.

1. Given the code below, from Active Code 2 of Section 4.7 of the textbook.

```
import turtle

def tree(branchLen,t):
    if branchLen > 5:
        t.forward(branchLen)
        t.right(20)
        tree(branchLen-15,t)
        t.left(40)
        tree(branchLen-15,t)
        t.right(20)
        t.backward(branchLen)

def main():
    t = turtle.Turtle()
    myWin = turtle.Screen()
    t.left(90)
    t.up()
    t.backward(100)
    t.down()
    t.color("green")
    tree(75,t)
    myWin.exitonclick()

main()
```

- Modify de code using all of the following ideas:
    1. Modify the thickness of the branches so that as the `branchLen` gets smaller, the line gets thinner.
    2. Modify the color of the branches so that as the `branchLen` gets very short it is colored like a leaf.
    3. Modify the angle used in turning the turtle so that at each branch point the angle is selected at random in some range. For example choose the angle between 15 and 45 degrees. Play around to see what looks good.
    4. Modify the branchLen recursively so that instead of always subtracting the same amount you subtract a random amount in some range.

1. In your report, display your code and the image of the tree generated when you run your program.

Exercise 2

In this exercise you are asked to develop some recursive functions in Python, and create unit tests for these functions, as per the guidelines provided in the testing in Python website (https://realpython.com/python-testing/).

1. Write a tail recursive function that computes x^n, called $power(x, n, acc)$, where acc is the accumulator, $x, n \geq 0$
2. Write an improved recursive (not necessarily tail recursive) version of $power(x, n)$, called $powerH(x, n)$, $x, n \geq 0$, that works by breaking n down into halves (where half of $n = n/2$, squaring $x^{n/2}$, and multiplying by x again if n was odd. For example, $x^{11} = (x^5) * (x^5) * x$, whereas $x^{10} = (x^5) * (x^5)$
3. The binomial coefficients
$$\binom{n}{k}$$
can be defined recursively. Letting
$$C(n, k) = \binom{n}{k},$$
we can use the following relationships:
$$\begin{array}{ll} C(n, 0) = 1 \text{ and } C(n, n) = 1 & \text{for } n \geq 0 \\ C(n, k) = C(n-1, k) + C(n-1, k-1) & \text{for } n > k > 0 \end{array}$$
Develop a Python function to compute $C(n, k)$.

Preparing to submit your report

1. Ensure you have structured your `lab04` folder as indicated in Section Prelude above.
2. Ensure you have properly created your unit tests in `test.py` in your `lab04` folder.
3. Ensure you have inserted your jupyter notebook report in your `lab04` folder, as required in Exercise 1 above.
4. Create a zip file of your `lab04` folder.

What to submit

At the Lab web page in D2L, click on `Lab Solution Submission`, then attach and submit **only the zip** file you have created as per the instructions above.