

## List 02 – Data Structures - Recursion

- 1) Create a recursive function that receives a positive integer N and prints all natural numbers from 0 to N in ascending order.
- 2) Create a recursive function that sums up the elements of an integer array.
- 3) Create a recursive function that receives a positive integer N and calculates the sum of the numbers from 1 to N.
- 4) Create a recursive function that reverses a string. For example, if the input is "hello", the output should be "olleh".
- 5) Create a recursive function that receives a positive integer N and returns the factorial of that number.
- 6) The superfactorial of a number N is defined by the product of the first N factorials of N. Thus, the superfactorial of 4 is **sf (4)** =  $1! * 2! * 3! * 4! = 288$   
Use previous recursive factorial function to implement this one.
- 7) Tetranacci numbers start with four predetermined terms and from there all other numbers are obtained by summing the previous four numbers. The first tetranacci numbers are: 0, 0, 0, 1, 1, 2, 4, 8, 15, 29, 56, 108, 208... Create a recursive function that receives a number N and returns the Nth term in the tetranacci sequence.
- 8) Write a recursive function that calculates the sum of the first n cubes:  
 $S(n) = 1^3 + 2^3 + \dots + n^3$
- 9) Write a recursive function that calculates the sum of the digits of an integer. For example, if the input is 123, the output should be  $1+2+3 = 6$ .
- 10) Pell numbers are defined by the following recursion:

$$P_n = \begin{cases} P_0 = 0, \\ P_1 = 1 \\ P_n = 2P_{n-1} + P_{n-2} \end{cases}$$

Some numbers in this sequence are: 0, 1, 2, 5, 12, 29, 70, 169, 408, 985...

Create a recursive function that receives a number N and returns the Nth Pell number.