Série 7 (algo+python)

Exercice 1

Ecrire une fonction qui prend pour argument deux entiers et qui retourne le plus grand.

Exercice 2

Ecrire une fonction qui prend pour argument deux entiers a et b et retourne leur pgcd.

Exercice 3

Ecrire une fonction qui prend pour argument un entier n et retourne le nombre de chiffres qui compose n.

Exemple : si n=1452635 la fonction retourne la valeur 7.

Exercice 4 :

Ecrire une fonction qui calcule la somme des puissances p-ieme des entiers $S_p(n)$:

$S_p(n) = 1^p + 2^p + 3^p + \ldots + n^p$

On sait que :

$S_1(n) = 1 + 2 + 3 + \ldots + n$

$S_2(n) = 1^2 + 2^2 + 3^2 + \ldots + n^2$

$S_3(n) = 1^3 + 2^3 + 3^3 + \ldots + n^3$

une fonction puissance(x,p), auquel l'on donnera deux entiers x et p

Exercice 5

Un entier naturel de trois chiffres est dit cubique s'il égal à la somme des cubes de ses trois chiffres.

Ecrire une fonction qui prend pour argument un nombre n et qui retourne 1 si ce nombre est cubique ou 0 sinon.

Exemple : 153 est cubique car $153 = 1^3 + 5^3 + 3^3$

Exercice 6

Ecrire une fonction qui prend pour argument un entier n et qui retourne le factoriel de n.

Exercice 7

Ecrire un programme qui permet de calculer la somme suivante :

S= $\frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \cdots + \frac{1}{n!}$ **n étant un entier impair >5**

Le programme utilisera une fonction récursive nommée FACT qui permet de calculer le factoriel d'un nombre entier positif.

Exercice 8

La suite de Fibonacci est définie récursivement par la relation $U_n = U_{n-1} + U_{n-2}$. Cette définition doit être complété par une condition d'arrêt, par exemple : u1 = u2 = 1. Écrire une fonction récursive qui calcule et renvoie le n-ième terme de la suite de Fibonacci (n donné en paramètre de la fonction).