

Série des exercices N°1

1) En utilisant une boucle while, écrire la fonction logarithme_binaire, qui à un réel strictement positif a associe : $\min \{n \in \mathbb{N}, 2^n > a\}$

2) (Calcul d'une suite de Syracuse)

Soit N un nombre entier strictement positif.

Ecrire une **fonction récursive** : Syracuse(n, N) définie par la suite:

$$\begin{cases} u_0 = N \\ U(n) = \begin{cases} U(n-1)/2 & \text{Si } U(n-1) \text{ est pair} \\ 3*U(n-1)+1 & \text{Si } U(n-1) \text{ est impair} \end{cases} \end{cases}$$

La conjecture affirme que pour tout N , il existe un indice n tel que $u_n = 1$. Ecrire une deuxième fonction indice(N) qui renvoie l'indice n correspondante à un entier N donné en paramètres

3) Ecrire une fonction en python permettant de calculer le factoriel d'un nombre entier n .

4) Ecrire une fonction en python permettant de vérifier si un entier p est premier ou non.

5) Ecrire une fonction en python permettant de vérifier si un entier p est parfait ou non.

Rappel : un nombre est parfait s'il est égal à ses diviseurs stricts.

Exemple : $1+2+3=6$

6) Ecrire une fonction en python permettant de vérifier si un entier p est un carré parfait ou non.

Rappel : un nombre entier x est un carré parfait s'il $\Leftrightarrow \exists y \in \mathbb{N} : y^2 = x : y^2 = x$

Exemples : 9, 16, 100, ...

7) Ecrire une fonction en python permettant de calculer la valeur de U_n , avec U_n est une suite récurrente définie par $U_0=1$; $U_1=2$; $U_{n+2}=U_{n+1}+6*U_n$.

8) Ecrire une fonction en python permettant de calculer la valeur de n^p , avec n et p des entiers positifs.

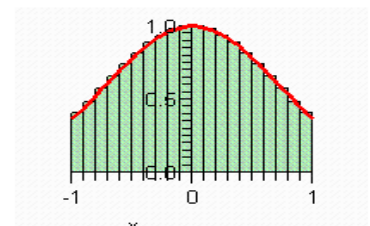
9) Ecrire une fonction en python permettant de lire une valeur entière N (positive ou négative) et qui fournit le nombre de chiffres de N comme résultat.

10) Ecrire une fonction en python permettant de résoudre une équation de type ($f(x)=0$) par la méthode de dichotomie selon le T.V.I.

11) La série $1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$ représente un développement limité de la fonction $f : f(x)=\exp(x)$ au voisinage de 0.

Ecrire une fonction en python permettant de calculer la

valeur de $\exp(x)$ en utilisant la série ci-dessus jusqu'à un ordre n transmis comme argument à cette fonction.



- 12) La fonction $f: f(x)=\exp(-x^2)$ n'est pas intégrable de façon analytique ; cependant l'intégrale $I = \int_a^b f(x)dx$ peut être approché de façon numérique.
Ecrire une fonction en python permettant d'approcher I.
- 13) Ecrire une fonction en python qui calcule la somme $S=\sum_{k=0}^n (2k + 1)^3$
NB : la valeur de n est fournie comme paramètre de la fonction.
- 14) Ecrire une fonction en python permettant de trouver et d'afficher le plus grand diviseur strict p d'un entier n fournie comme paramètre de cette fonction. Cette fonction retourne -1 si p n'est pas trouvé.
- 15) Ecrire une fonction PlusGranDivPrem(n) : permettant de trouver et de retourner le plus grand diviseur premier p de l'entier n passé en paramètres. Cette fonction retourne 1 si n=1.
PlusGranDivPrem(100) =5 ; PlusGranDivPrem(16) =2 ;
PlusGranDivPrem(7) =7
- 16) Ecrire une fonction Factoriser(n) : permettant de factoriser l'entier n en nombres premiers.
Factoriser(30) affiche à l'écran : 5*3*2
Factoriser(60) affiche à l'écran : 5*3*2*2
- 17) Un entier n est dit méchant si et seulement si ses seuls diviseurs premiers sont parmi 2, 3 et 5. Ecrire une fonction en langage Python recevant un entier n en paramètres et permettant d'afficher le message « oui c'est un nombre méchant » ou le message « non, ce n'est pas nombre méchant » selon la valeur de n.

Exemples :

Le nombre 14 n'est pas méchant car il est divisible par le nombre premier 7.

Le nombre 30 est méchant car il est divisible uniquement par les deux nombres premiers 2 et 5.