

Exercice 1

Ecrivez un algorithme qui demande un nombre à l'utilisateur, et qui affiche le carré de ce nombre.

Exercice 2

Soit la suite (v_n) définie par $v_n = 3^n$.

Ecrire un algorithme qui permet à l'utilisateur de rentrer une valeur de n et qui renvoie la valeur v_n .

Exercice 3

Soit la suite (u_n) définie par $\begin{cases} u_{n+1} = 2u_n + 3 \\ u_0 = 0 \end{cases}$.

1. Ecrire un algorithme qui calcule le terme u_7 .
2. Utiliser un repère pour tracer le nuage des points jusqu'à $n = 7$.
3. Ecrire un algorithme qui permet à l'utilisateur de rentrer une valeur de n et qui renvoie
 - la valeur u_n .
 - La liste des valeurs de u_1 jusqu'à u_n (u_1, u_2, \dots, u_n)
4. Que constatez-vous lorsque vous rentrez une grande valeur pour n ?

Exercice 4

Soit la suite (u_n) définie par :

$$u_n = e^{-n+1}.$$

1. En modifiant l'algorithme de l'exercice 4 conjecturer la limite de la suite (u_n) .
2. Ecrire un algorithme renvoyant la valeur de n pour laquelle $|u_n| \leq 10^{-3}$.

Exercice 5

Ecrivez un algorithme qui échange la valeur de deux variables. Prenez par exemple $a = 5$ et $b = 2$. Le programme doit faire en sorte que $a = 2$ et $b = 5$.

Exercice 6

La suite de Fibonacci : 1 – 1 – 2 – 3 – 5 – 8 – 13 – 21 ...

Ecrire un algorithme qui pour une valeur n renvoie le n – ième terme de la suite de Fibonacci.

Exercice 6

Ecrivez un algorithme qui demande deux nombres, puis qui dit si le produit est positif ou négatif. Si le produit est nul, on le comptera positif.

Exercice 7

Ecrivez un algorithme qui demande un nombre, puis qui demande un nouveau nombre jusqu'à ce qu'il concorde avec le premier. A chaque fois, l'algorithme dira si le nombre entré est plus grand ou plus petit que le nombre à trouver.

On peut imaginer améliorer l'algorithme en faisant en sorte qu'il génère aléatoirement le nombre à trouver.