

Programmation Python – Algorithmme – Exercices – Devoirs

Exercice 1 corrigé disponible

Donner les lignes d'une fonction Python, la plus simple possible permettant de d'apporter une solution aux problèmes mathématiques suivants :

1. retourner le PGCD et le PPCM de deux nombres entiers A et B
2. Quelle est une valeur approchée de π en utilisant la méthode Monte-Carlo ?
3. Quelle est une valeur approchée de \sqrt{x} en utilisant la méthode de Héron ?
4. Retourner la solution de l'équation $x^3=k$ en utilisant la méthode de dichotomie

Exercice 2 corrigé disponible

On représente un brin d'ADN par une chaîne de caractères qui peut contenir quatre caractères différents : "A" (Adénine), "C" (Cytosine), "G" (Guanine) et "T" (Thymine).

1. Écrire une fonction `estADN` qui prend en argument une chaîne de caractères et renvoie True si cette chaîne correspond à un brin d'ADN et qui renvoie False sinon.

Contrat:

Par exemple, `estADN("TTGAC")` et `estADN("GCAATAG")` renvoient True mais `estADN("AMOG")` et `estADN("CaTg")` renvoient False.

2. Écrire une fonction `masseMolaire` qui calcule la masse molaire d'une séquence ADN passée en argument. Chaque lettre a une masse donnée : "A" (135 g/mol); "T" (126 g/mol); "G" (151 g/mol); "C" (111 g/mol). La masse totale est la somme des masses des lettres de la séquence.

Contrat:

Par exemple, `masseMolaire("AGATC")` renvoie (135 + 151 + 135 + 126 + 111) g/mol, c'est-à-dire 658 g/mol.

3. Chaque base possède une base complémentaire avec laquelle elle peut s'associer : "A" et "T" sont complémentaires, et "C" et "G" sont complémentaires. Écrire une fonction `brinComp` qui étant donné un brin d'ADN `b` calcule et renvoie son brin complémentaire, c'est à dire le brin constitué des bases complémentaires de `b`.

Contrat:

Par exemple, `brinComp("A")` renvoie "T" et `brinComp("AAGT")` renvoie "TTCA".

4. (Bonus, ***) Écrire une fonction `sous_séquence` qui prend en argument deux chaînes de caractères représentant des brins d'ADN et renvoie True si le premier brin est une sous-séquence du deuxième, et qui renvoie False sinon.

Contrat:

Par exemple, `sous_séquence("ATC", "GGTATCG")` renvoie True et `sous_séquence("GC", "AAT")` renvoie False.

Exercice 3 corrigé disponible

En 1202, le mathématicien Leonardo Fibonacci inventa l'énigme suivante : Un homme met un couple de lapins dans un lieu isolé de tous les côtés par un mur. Combien de couples obtient-on en un an si chaque couple engendre tous les mois un nouveau couple à compter du troisième mois de son existence ?

1. Si l'on connaît le nombre de couples de lapins au mois n et au mois $n + 1$, comment connaître le nombre de couples de lapins au mois $n + 2$?
2. Résoudre l'énigme.
3. Plus généralement, écrire une fonction `fibonacci` qui calcule le nombre de couples de lapins au bout d'un nombre n de mois donné en argument.

Exercice 4

- 1.

Écrire une fonction `factorielle` qui renvoie le produit des entiers jusqu'à l'entier entré en paramètre. Ainsi, `factorielle(5) = 1 × 2 × 3 × 4 × 5 = 120`.

On rappelle que par définition, `factorielle(0) = 1`.

2. Écrire une fonction Python qui retourne : $\binom{n}{k}$

Exercice 5

Ecrire une fonction Python qui permette de retourner un couple d'entiers (u, v) tels que $a \cdot u + b \cdot v = \text{PGCD}(a, b)$ (solution particulière de l'équation de Bézout)

Exercice 6

On considère la suite (u_n) définie par $u_0 = 1$ et pour tout entier naturel n par $u_{n+1} = \frac{n+3}{3n+5} \cdot u_n$.

On admet que cette suite est positive et tend vers 0.

Ecrire un programme Python qui affiche la plus petite valeur de n pour laquelle $0 \leq u_n \leq 10^{-3}$.

Exercice 7

On dispose d'un dé équilibré à 6 faces et de deux urnes : l'urne U_1 contient deux boules vertes et 3 rouges, et l'urne U_2 contient 1 boule verte et deux rouges. On lance le dé et si le résultat est 1 ou 2 alors on tire une boule dans l'urne U_1 , sinon on tire dans l'urne U_2 . On considère que la partie est gagnante si on tire une boule verte.

1. Écrire un algorithme en Python permettant de simuler cette partie.
2. Modifier cet algorithme pour qu'il simule n parties et compte le nombre de parties gagnantes.