TP2bis - Master de Physique - M1

PIV, CCP, Phymatech, Nanosciences, Physique Générale

HMPH104

Hervé Wozniak - Université de Montpellier

Année universitaire 2018-2019

Table des matières

Exercice n° I

Un brin d'ADN est faite de bases : adénine (A), cytosine (C), guanine (G) ou thymine (T). Ces brins sont représentés sous forme de chaîne de caractères (`str`) de longueur variable.

A l'intérieur d'un brin d'ADN, on recherche des séquences précises (par exemple, 'ATTGCC') et on souhaite les compter. La séquence recherchée ainsi que brin d'ADN sont stockés sous forme de chaîne de caractères (`str`).

Écrire une fonction `valide()` qui renvoie `True` si son argument (`str`) est valide (donc fait uniquement des 4 lettres de bases, A, C, G ou T), `False` sinon.

Écrire une fonction `entree()` qui permet la saisie au clavier d'une chaîne de caractères, qui la valide avec `valide()`, redemande une entrée en cas de chaîne invalide, et retourne la chaîne de caractères en cas de succès.

Écrire une fonction `proportion()` qui calcule la proportion (en %) de séquences recherchées dans un brin d'ADN.

Élaborer des tests unitaires de chacune des fonctions et procédures, ainsi qu'un test d'intégration.

Toutes les fonctions et procédures doivent être dans un même fichier (`exo_I.1.py`), ainsi que les tests et leurs résultats sous forme de commentaires.

Exercice n° II

Écrire le programme `exo_I.2.py` qui effectue les opérations suivantes : 1) saisir un nombre entier entre 1 et 3999 ; 2) l'afficher en nombre romain.

Tester sur les valeurs données en exemple ci-dessous. Mettre les résultats obtenus en commentaires dans le fichier `exo_I.2.py`

Rappel : Numération romaine

La numération romaine est un système de numération additive utilisé par les Romains de l'Antiquité. Les chiffres romains sont représentés à l'aide de symboles combinés entre eux :

M=1000

D=500

C=100

L=50

X=10

V=5

I=1

Un nombre écrit en chiffres romains se lit de gauche à droite. Sa valeur se détermine en faisant la somme des valeurs individuelles de chaque symbole (du plus grand à gauche, au plus petit à droite), sauf quand l'un des symboles précède un symbole de valeur supérieure ; dans ce cas, on soustrait la valeur du premier symbole au deuxième.

Exemple

3 = III

4 = IV = (5 - 1)

8 = VIII

9 = IX = (10 - 1)

10 = X

42 = XLII = XL + I + I = (50 - 10) + 1 + 1

509 = DIX = D + IX = 500 + (10 - 1))

666 = DCLXVI = D + C+ L + X +V + I = 500 + 100 + 50 + 10 + 5 + 1

Exercice n° III

L'approximant de Padé est une méthode mathématique d'approximation d'une fonction analytique par une fonction rationnelle, c'est-à-dire exprimée sous forme de rapport de deux polynômes $f(x) = \frac{P_n(x)}{Q_m(x)}$. Cette forme d'approximation est très utile en physique lorsque la fonction analytique possède des pôles. En augmentant les degrés n et m des polynômes, l'approximation de $f(x)$ s'améliore.

On étudie deux approximants de Padé de la fonction $f(x) = \frac{\tan^{-1}(x)}{x} = \frac{\arctan(x)}{x}$ (en notation informatique) :

$$\frac{P_2(x)}{Q_2(x)} = \frac{1 + \frac{4}{15}t^2}{1 + \frac{9}{15}t^2}$$

et

$$\frac{P_4(x)}{Q_4(x)} = \frac{1 + \frac{7}{9}t^2 + \frac{64}{945}t^4}{1 + \frac{10}{9}t^2 + \frac{5}{21}t^4}$$

Dans un fichier `pade.py`, écrivez les deux fonctions `pade22()` et `pade44()` qui calculent respectivement $\frac{P_2}{Q_2}$ et $\frac{P_4}{Q_4}$. Tracer la fonction $f(x)$ sur l'intervalle $[-5, +5]$ avec 51 points sous forme de trait continu, et superposez les deux approximants sous forme de symboles non reliés. La légende de la figure doit permettre de reconnaître les approximants.

A l'exécution, le message d'erreur "RuntimeWarning: invalid value encountered in true_divide" devrait apparaître. Expliquer pourquoi et proposez une solution en commentaire dans le fichier `pade.py`.