

Devoir Surveillé N°2

Durée : 1 h 30 min.

N.B :

- La simplicité et la lisibilité des solutions seront tenues en compte lors de la correction et de l'évaluation ;
- Les fonctions prédéfinies sont strictement interdites sauf celle qui retourne le nombre d'éléments d'une liste.

Exercice 1 : Modification d'une chaîne de caractères.

Donner une procédure (*python*) **Modification_chaine_caracteres**(*i,x*) qui nous permet de changer un caractère, d'une chaîne de caractères, d'indice *i* par un caractère stocké dans la variable *x*.

Exercice 2 : Suite de Thue-Morse.

On appelle suite de **Thue-Morse** la suite définie récursivement, $\forall n \in \mathbb{N}$, par :

$$\begin{cases} t_0 = 0 \\ t_{2n} = t_n \\ t_{2n+1} = 1 - t_n \end{cases}$$

On peut montrer que cette suite ne contient que les valeurs 0 et 1. Les premières valeurs de la suite sont : 0,1,1,0,1,0,0,1,1...

Ecrire un seul programme (*python*) composé d'un **programme appelant (principal)** et des **fonctions** suivantes :

- 1) **Thue_Morse(k)** : Retourne une liste qui contient les valeurs de la suite de **Thue-Morse**, avec *k* est le nombre des éléments de cette liste.
- 2) **Thue_Morse_Compressee(T)** : Retourne la liste compressée de la liste *T* :
Soit *T* une liste à laquelle on affecte la fonction **Thue_Morse(k)**. On compresse la liste *T* en une liste *S* définie de la façon suivante :
 - le premier élément de *S* est égal au premier élément de *T* ;
 - les autres éléments de *S* sont les longueurs respectives des séquences d'éléments consécutifs égaux dans *T*.

Exemple : si $T = [0, 1, 1, 0, 1, 0, 0, 1, 1]$, alors $S = [0, 1, 2, 1, 1, 2, 2]$.

- 3) **Thue_Morse_Decompressee_Portion(S,q)** : Retourne une portion décompressée de la liste *S* dont l'indice est *q* ($0 < q < m$ Avec *m* : la longueur de la liste *S*).
- Exemple :** **Thue_Morse_Decompressee_Portion(S,2)** : Retourne la liste [1,1].
- 4) **Thue_Morse_Decompressee(S)** : Retourne la liste *T* (Décompressée) en utilisant la fonction de la question précédente ;
- 5) **Taux_Compression(T)** : Retourne le taux de compression, défini par $\tau = m/k$

Exercice 3 : Mot de Dyck.

On dit que *L* est un mot de Dyck si et seulement si la liste *L* contient autant de 1 que de -1, et si pour tout $k \leq n$, les *k* premiers termes de la liste contiennent au moins autant de 1 que de -1.

Remarque culturelle : Il s'agit en fait de toutes les formules correspondant à des parenthésages corrects : **Par exemple :**

- La formule parenthésée $e = ((()()))$ correspond au mot de Dyck [1,1,-1,1,1,-1,1,-1,-1,-1] ;
- De même, la formule $e = ()()$ correspond à [1,-1,1,-1] ;
- La liste vide est aussi un mot de Dyck ;
- La formule parenthésée $e = =)()()$ ne correspond pas au mot de Dyck [-1,1,1,-1,-1,1].

Ecrire une fonction **Dyck (L)**, en *python*, qui étant donnée une liste *L* dont tous les éléments valent 1 ou -1, renvoie **True** si *L* est un mot de Dyck, et **False** sinon.

Exercice 4 : Transformation d'un tableau à deux dimensions en un tableau à une dimension.

Devoir Surveillé N°2

Durée : 1 h 30 min.

N.B. :

- La simplicité et la lisibilité des solutions seront tenues en compte lors de la correction et de l'évaluation ;
- Les fonctions prédéfinies sont strictement interdites sauf celle qui retourne le nombre d'éléments d'une liste.
Donner une fonction (*python*) qui transfère une liste M à deux dimensions L et C dans une liste V à une dimension, en utilisant **une seule boucle**.