

Révision 1 Python

Lundi : on dépasse la moyenne

1. Écrire une fonction `Moyenne(L)` qui renvoie la moyenne des éléments de la liste `L` (liste non vide de nombres).
2. Écrire une fonction `DepasseMoyenne(L)` qui renvoie la liste des éléments de `L` strictement plus grand que la moyenne. La liste des éléments sera créée par des `append` successifs. Attention : la fonction `Moyenne(L)` devra être appelée une et une seule fois.
3. Quelle est la complexité de ces deux fonctions ?
4. Écrire une fonction `DepasseMoyenneC(L)` qui renvoie la liste des éléments de `L` strictement plus grand que la moyenne. La liste des éléments sera créée par compréhension. Attention : la fonction `Moyenne(L)` devra être appelée une et une seule fois.

Mardi : on travaille à la chaîne

1. Écrire une fonction `Occurrences(chaine,x)` qui renvoie le nombre d'occurrences de `x` dans une chaîne de caractères `chaine`.
2. Écrire une fonction `VoyelleOuTEsVoyelle(chaine)` qui renvoie la liste des indices des voyelles dans `chaine`.

Mercredi : on compte les lapins

1. Écrire une fonction **non récursive** nommée `Fib(n)` qui renvoie le terme d'indice n de la suite $(F_n)_{n \in \mathbb{N}}$ définie par :
$$F_0 = F_1 = 1 \quad \text{et} \quad \forall n \in \mathbb{N} \quad F_{n+2} = F_{n+1} + F_n$$
2. (question de maths) Trouver l'expression de F_n en fonction de n .
3. (question de maths) En déduire que $F_n \xrightarrow[n \rightarrow \infty]{} +\infty$.
4. Écrire une fonction `Seuil(s)` qui renvoie le plus petit indice n tel que $F_n > s$.

Jeudi : on cherche à être à côté des meilleurs

1. Écrire une fonction `Maximum(L)` qui renvoie le maximum d'une liste non vide de nombres `L`.
2. Écrire une fonction `Indices(L)` qui renvoie la liste des indices des éléments égaux au maximum de `L`. On créera cette liste par compréhension et on n'appellera qu'une seule fois la fonction `Maximum`.
3. Écrire une fonction `ACotéDuMeilleur(L)` qui renvoie la liste des indices des éléments à côté d'un maximum (à sa droite ou à sa gauche).

Vendredi : on ouvre un dictionnaire pour se cultiver

1. Écrire une fonction `DictionnaireOccurrences(chaine)` qui prend comme paramètre une chaîne de caractères et renvoie un dictionnaire dont les clefs sont les caractères de la chaîne et la valeur associée est le nombre d'occurrences de ce caractère dans `chaine`.
2. Écrire une fonction qui permettent de trouver un des caractères qui apparaît le moins souvent dans `chaine`.
3. Considérons le graphe définie par le dictionnaire d'adjacence :

`Dico={"A":["B","C","D"], "B":["A","C","E"], "C":[], "D":["B","C"], "E":["C","D"]}`

- (a) Faites un dessin du graphe.
- (b) Donner la liste des points qui seront visités si on fait un parcours en largeur en partant de `A`.
- (c) Donner la liste des points qui seront visités si on fait un parcours en profondeur partant de `B`.

Réponses de la feuille de révision