Lycée Loges

Chapitre 2

Algorithmes classiques
Exercices

Simon Dauguet
simon.dauguet@gmail.com

21 octobre 2024

Exercice 1 :
Proposer une fonction `moyenne(tab: list)-> float`, qui calcule la moyenne des données contenues dans le tableau `tab`.

Exercice 2 :

a) Proposer une fonction `ecartType(tab: list)-> float`, qui calcule l'écart-type des données contenues dans le tableau `tab` (on pourra utiliser la fonction `moyenne()`).

b) Si votre première solution parcourt plus d'une fois le tableau, proposer une solution avec un seul parcours.

Exercice 3 :
On considère un tableau contenant soit des entiers, soit des tableaux d'entiers. Proposer une fonction `somProfonde(tab: list) -> int`, qui calcule la somme de tous les entiers contenus dans le tableau ; *i.e.* `somProfonde( [1,[3,5],2] )` renvoie 11.

Exercice 4 :
On considère de nouveau un tableau contenant soit des entiers, soit des tableaux d'entiers. Proposer une fonction `rechProfonde(tab: list, val: int) -> bool`, qui précise si la valeur `val` est contenue dans le tableau ou l'un de ses sous-tableaux.

Exercice 5 :
De même, on considère encore un tableau contenant des entiers ou des tableaux d'entiers. Proposer une fonction `occProfond(tab: list, val: object) -> int`, qui renvoie le nombre d'occurrence de la valeur `val` dans `tab` et ses sous-tableaux.

Exercice 6 :
Proposer une fonction `factorielle(n: int) -> int`, qui renvoie $n!$.

Exercice 7 :
Proposer une fonction `nextFactorielle(n: int) -> int`, qui renvoie le plus grand entier k tel que $k! < n$.

Exercice 8 :
Proposer une fonction `coeffBinomial(n: int, k: int) -> int`, qui renvoie $\binom{n}{k}$ en limitant le nombre de tours de boucle.

Exercice 9 :
On considère un tableau de tableaux d'entiers `tab2tab` (*i.e.* `[[1,4],[2,1,0]]`). Proposer une fonction `miniMax(tab2tab: list) -> int`, qui renvoie le minimum des maximums des sous-tableaux contenus dans le tableau `tab2tab`. Par exemple, `miniMax( [[1,4],[2,1,0]] )` renvoie 2.

Exercice 10 :
On considère un tableau d'entiers.

a) Proposer une fonction `estPositif(tab: list) -> list`, qui renvoie un tableau de booléen précisant à l'indice i si `tab[i]>0`.

b) Proposer une procédure `rendrePositif(tab: list) -> None`, qui change les valeurs négatives de `tab` par des valeurs nulles. Qu'aurait-il fallu faire si l'on ne voulait pas modifier le tableau initial ?

c) Proposer une fonction `moyennePositif(tab: list) -> tuple`, qui renvoie un couple composé de la moyenne des valeurs positives de `tab` et du pourcentage de nombres utilisés pour le calcul. Par exempel, `moyenne( [-1,3,5,-2,13] )` renvoie `(7, 0.6)`.

Exercice 11 :
Proposer une fonction `estPremier(n: int) -> bool`, qui précise si l'entier naturel n est premier. On fera attention à réduire le nombre de tours de boucle.

Exercice 12 :
On considère un tableau d'entiers compris entre 0 et 100.

a) Proposer une fonction `compte(tab: list, n: int) -> int`, qui compte le nombre de n dans le tableau `tab`.

b) Proposer une fonction `tri(tab: list) -> list`, qui renvoie un tableau trié dans l'ordre croissant des nombres contenus dans `tab`.

c) Si votre fonction `tri()` effectue plusieurs lectures du tableau, proposer une solution avec une seule lecture.

Exercice 13 :
On considère le tableau trié dans l'ordre décroissant `tab=[9,5,5,3,2,1,0,-2,-2]`.

Proposer une fonction `rechDichoDecroissant(tab: list, val: int) -> bool`, qui implémente la recherche dichotomique dans un tableau de nombres triés dans l'ordre décroissant.

Exercice 14 :
On suppose que l'on possède un tableau de mots `tab2mots` ordonnés selon l'ordre alphabétique.

a) Proposer une fonction `recherche(tab2mots: list, mot: str) -> bool`, qui renvoie True ou False selon si `mot` est dans `tab2mots` ou non.

b) Proposer une fonction `rechercheI(tab2mots: list, mot: str) -> int`, qui renvoie l'indice du `mot` dans le `tab2mots` s'il existe, -1 sinon.

c) Proposer une fonction `infAlphabetique(tab2mots : list, mot : str) -> int`, qui renvoie le nombre de mots qui soient classés avant `mot` dans la liste `tab2mots`.

Exercice 15 :

On souhaite construire une pyramide de hauteur h à base carrée composée de petits cubes. Ainsi si $h = 1$, la pyramide se compose de 1 cube. Si $h = 2$, la pyramide se compose de $9 + 1$ cubes.

a) De combien de cubes est formée une pyramide de hauteur 3 ? de hauteur 4 ?

b) Exprimer $n(h)$ le nombre de cubes nécessaire pour une pyramide de hauteur h.

c) Proposer une fonction `hauteurMax(n: int) -> int`, qui renvoie la hauteur maximale possible pour faire une pyramide à base carrée avec n cubes.

Exercice 16 :

On considère une fonction f définie sur $\mathbb{R}$ définie par $x^3 - 4$.

a) Écrire en python la fonction `f(x: float) -> float`, qui définie f.

b) Montrer que $f(x) = 0$ possède une unique solution sur $\mathbb{R}$ comprise entre 1 et 2.

c) Calculer c le milieu de $a = 1$ et $b = 2$, dans quel cas l'antécédent de 0 par f est-il compris dans $[a, c]$?

d) Proposer un code qui permette d'approximer $\sqrt[3]{4}$ à `epsilon=0.001` près.

e) Proposer une fonction `solveF(fct: object, a: float, b: float, epsilon: float) -> float`, qui approxime un antécédent de 0 par `fct()` dans $[a, b]$ à `epsilon` près (l'antécédent est supposé existé). Par exemple, `solveF(f, 1, 2, 0.25)` renvoie 1.5625.