

Algorithmique T.D. n° 7

Exercice 1 : Le dernier

On considère une séquence de caractères représentée dans un fichier. Ecrire un algorithme qui affiche la valeur du dernier caractère de la séquence.

Exercice 2 : Interclassement

On considère **deux séquences d'entiers positifs triées en ordre croissant** représentées dans des fichiers. Un même nombre peut apparaître plusieurs fois dans une séquence. Ecrire un algorithme qui réalise **l'interclassement** des deux séquences : le résultat est un fichier contenant une troisième séquence en ordre croissant comportant tous les éléments présents dans les deux séquences lues. Si un même élément apparaît **x** fois dans la première séquence et **y** fois dans la deuxième, il doit apparaître **x+y** fois dans la séquence résultat. Les séquences lues peuvent être vides.

Exemple : séquence 1 : 1 1 3 4 11 11 25 28 30 30 32 40
 séquence 2 : 1 2 3 5 5 11 24 25 29
 séquence résultat : 1 1 1 2 3 3 4 5 5 11 11 11 24 25 25 28 29 30 30 32 40

Exercice 3 : Séquence de durées

On considère une séquence de **durées** représentée dans un fichier. Ecrire un algorithme qui calcule et affiche la durée moyenne des durées de la séquence.

Le type Durée est défini comme suit dans le lexique partagé :

Durée : type agrégat
 h : entier ≥ 0 ; // heures
 m, s : entiers entre 0 et 59 // minutes et secondes
faagrégat

Exercice 4 : Racine carrée

Ecrire la fonction **Racine** qui calcule la racine carrée d'un réel **X** par la méthode de Newton : on considère la suite a définie par récurrence :

$$a_0 = X/2$$

$$a_{i+1} = (a_i + X/a_i) / 2$$

Le résultat est a_n tel que : $|(a_{n+1} - a_n) / a_n| \leq \epsilon$, pour $\epsilon = 10^{-6}$.

Racine : fonction (**X** : réel ≥ 0) \longrightarrow réel ≥ 0
 { *Racine(X) désigne la racine carrée de X* }

Exemple : Pour le calcul de Racine(9) on a :
 $a_0 = 4.5$, $a_1 = (4.5 + 9/4.5) / 2 = 3.25$, $a_2 = (3.25 + 9/3.25) / 2 = 3.00961$
 $a_4 = 3.0000153$, $a_5 = 3$, $a_6 = 3$

Exercice 5 : Combien de LE ?

On considère un fichier de caractères comportant un texte formé de lettres majuscules et d'espaces. Un mot est une suite de lettres délimitée par des espaces ou par les extrémités du texte. Ecrire un algorithme qui affiche un message indiquant le nombre de LE présents dans le texte.

Exemple : fichier lu : LA BELLE BALLE ROULE DANS LES BOIS
 résultat affiché : Nombre de LE : 4

Exercice 6 : Les chaînes à A

On considère un fichier comportant une séquence de chaînes de caractères. Ecrire un algorithme qui affiche et compte toutes les chaînes qui contiennent au moins trois occurrences du caractère 'A'.
Exemple :

Contenu du fichier :

"ABRACADABRA "	"ET VOILA"	"MA CAMERA"	"HAHAHA"	"JOLI TOUR"	"N'EST-CE PAS"	
----------------	------------	-------------	----------	-------------	----------------	--

Résultat affiché :

ABRACADABRA

MA CAMERA

HAHAHA

Nombre de chaînes comportant au moins trois A : 3

Exercice 7 : Union

On considère deux ensembles d'entiers représentés dans des fichiers. Chaque entier n'apparaît qu'en un exemplaire dans une séquence. Ecrire un algorithme qui produit un fichier contenant l'**union** des deux ensembles.

Exemple : *fichier 1* : 10 5 12 7 19 28 6
 fichier 2 : 5 14 19 2 10 6 29 3 7
 fichier résultat : 10 5 12 7 19 28 6 14 2 29 3

Exercice 8 : Trouver le dernier

Ecrire la fonction **trouverDerCar** qui détermine lequel parmi deux caractères est le dernier à apparaître dans une chaîne donnée :

fonction trouverDerCar (**c1**, **c2** : caractère, **x** : chaîne) —> caractère
// trouverDerCar(c1, c2, x) renvoie le caractère (c1 ou c2) qui apparaît le dernier dans x
// si x ne comporte aucune occurrence de c1 ni de c2, on renvoie ' ' (un espace).

Exemples : trouverDerCar('b','a',"charabia") renvoie 'a'
 trouverDerCar('a','h',"architecte") renvoie 'h'
 trouverDerCar('d','e',"charabia") renvoie ' '

Exercice 9 : Nicomaque de Gérase (mathématicien et philosophe néo-pythagoricien)

Le mathématicien Nicomaque de Gérase (1^{er} siècle après JC) a écrit dans son introduction à l'arithmétique, que tout cube est égal à la somme de nombres impairs consécutifs ; par exemple :

$$1^3 = 1, 2^3 = 3 + 5, 3^3 = 7 + 9 + 11, 4^3 = 13 + 15 + 17 + 19$$

Ecrire l'action décomposerCube qui calcule et affiche une décomposition du cube d'un entier en une somme d'entiers impairs consécutifs :

action décomposerCube (consulté **x** : entier > 0)
{ Effet : calcule et affiche X^3 en une somme d'entiers impairs consécutifs }

Exemple : décomposerCube(3) a pour effet d'afficher : $27 = 7 + 9 + 11$