

Algorithmique T.D. n° 2

Pour chaque exercice, il est demandé de décrire soigneusement le lexique et l'algorithme.

Exercice 1 : signe d'un produit de deux nombres

Ecrire un algorithme qui saisit deux nombres et affiche l'un des deux messages suivants : "le produit des deux nombres est positif ou nul" ou "le produit des deux nombres est négatif" **sans** calculer le produit de ces deux nombres.

Exemple : *données saisies :* -12, -3
 résultat affiché : le produit des deux nombres est positif ou nul

Exercice 2 : note finale

La note finale d'un étudiant dans une matière est calculée à partir des trois notes qu'il a obtenues dans cette matière, en prenant la meilleure parmi les deux notes intermédiaires suivantes :

- 1 - la moyenne de ces trois notes initiales,
- 2 - la moyenne de sa meilleure et de sa plus mauvaise note.

Ecrire un algorithme qui lit le nom et les trois notes d'un étudiant, calcule les deux notes intermédiaires et affiche le nom de l'étudiant et sa note finale.

Exercice 3 : consonne ou voyelle ?

Ecrire un algorithme qui saisit une lettre de l'alphabet et affiche un message indiquant si la lettre est une voyelle ou une consonne.

Exercice 4 : équation du second degré à une inconnue

On veut écrire un algorithme qui calcule et affiche la ou les solutions réelles d'une équation du second degré à une inconnue, c'est-à-dire : **$ax^2 + bx + c = 0$**

Pour vous éviter de rechercher dans vos vieux souvenirs de mathématiques de 3^{ème}, la méthode de résolution de ce type d'équation est rappelée ici :

a) on calcule le discriminant delta (souvent noté Δ) qui est égal à : **$b^2 - 4ac$**

b) selon le signe de delta, on a zéro, une ou deux solutions réelles :

delta < 0 : pas de solution réelle

delta = 0 : une solution : $x_0 = -b / 2a$

delta > 0 : deux solutions : $x_1 = \frac{-b + \sqrt{\text{delta}}}{2a}$ et $x_2 = \frac{-b - \sqrt{\text{delta}}}{2a}$

Ecrire l'algorithme qui lit les données a, b et c et affiche la ou les solutions, ou un message indiquant qu'il n'y a pas de solutions réelles. Bien entendu, on dispose de la fonction *racine* qui calcule la racine carrée d'un nombre.

Exercice 5 : date

Ecrire un algorithme qui décode une date du 21^{ème} siècle : la donnée saisie est un entier compris entre 10100 et 311299. L'algorithme construit puis affiche une chaîne de caractères comprenant : les numéros du jour dans le mois, du mois dans l'année, et l'année sur quatre chiffres, séparés par des '-'.
Exemple : donnée : 30916 valeur de la chaîne construite : "3-9-2016"

On utilisera les opérations sur les entiers et sur les chaînes pour construire le résultat. On dispose également d'une fonction Conv qui convertit une valeur entière en la chaîne des chiffres le composant. Par exemple, Conv(96) produit la chaîne "96", Conv(3) produit la chaîne "3".

Exercice 6 : binaire

Ecrire un algorithme qui pour un nombre binaire de quatre chiffres 0 ou 1 affiche sa valeur en base 10. On étudiera trois solutions au problème selon la forme des données :

- forme 1 : les données sont quatre entiers (0 ou 1), par exemple : 1, 1, 0, 1.
- forme 2 : la donnée est un entier formé de 0 et de 1 (en base 10), par exemple : 1101
- forme 3 : la donnée est une chaîne non vide de quatre caractères au maximum, ne comportant que des '0' et des '1', par exemple : "1101"

Exemple : données saisies : forme 1 : 0, 1, 0, 1 forme 2 : 101 forme 3 : "101"
résultat affiché : 5

Exercice 7 : Verbe du premier groupe ?

- a) Réaliser la fonction **estDuPremierGroupe** qui détermine si un verbe est du premier groupe, c'est-à-dire s'il se termine par « er » (sauf « aller »):

fonction **estDuPremierGroupe** (V : chaîne) —→ booléen
// estduPremierGroupe(V) renvoie vrai si le verbe V est du premier groupe, et renvoie faux sinon

- b) Ecrire un algorithme principal qui saisit un verbe et affiche un message indiquant si le verbe lu est du premier groupe ou non. Le verbe est donné à l'infinitif. L'algorithme utilisera la fonction estduPremierGroupe.