

Épisode III : Notions de programmation

EXERCICE 1

1. Écrire un programme qui permet de saisir une note comprise en 0 et 20 (on redemandera à l'utilisateur de saisir une note jusqu'à ce que celle-ci soit bien comprise entre 0 et 20).
2. Ajouter au programme précédent l'affichage de la mention relative à la note obtenue (on affichera "Ajourné(e)" si la note est strictement inférieure à 10).

EXERCICE 2

1. Écrire un programme qui permet de saisir n notes comprises en 0 et 20 ($0 < n \leq 10$ est connu et fixé c'est à dire qu'on demande de saisir cette valeur à l'utilisateur).
2. Ajouter au programme précédent le calcul de la moyenne arrondie à deux décimales des notes, le minimum, le maximum et le nombre d'occurrences du maximum.
3. Ajouter au programme l'affichage des notes saisies sur la première colonne de la Feuille 1 ainsi que la moyenne arrondie à deux décimales des notes, le minimum, le maximum et le nombre d'occurrences du maximum.
4. On recommence l'exercice mais cette fois-ci on ne connaît pas d'avance le nombre n , le programme continue tant qu'une note incorrecte n'a pas été saisie.

EXERCICE 3

Écrire une fonction "Prix TTC".

Signature de la fonction : Public Function MonTTC(pht As Double) As Double Entrée : pht (réel)

Sortie : Prix TTC (réel)

Calcul : Prix TTC = pht * 1.2

EXERCICE 4

Écrire une fonction "Capital final".

Signature de la fonction :

Public Function MonKF(KI As Double, i As Double, duree As Long) As Double

Entrée : Capital initial (réel), Intérêt (réel), Durée (réel)

Sortie : Capital final

Calcul : Capital final = Capital initial * (1 + intérêt)^{durée}

EXERCICE 5

Écrire une fonction "Prix TTC avec condition".

Signature de la fonction :

Public Function MonTTCBis(pht As Double, cat As String) As Double

Entrée : prix ht (réel), Catégorie (chaîne de caractères)

Sortie : prix ttc (réel)

Calcul :

Si (catégorie = " luxe ")

Alors pttc = pht * 1.33

Sinon pttc = pht * 1.2

EXERCICE 6

Écrire une fonction "Location de véhicules-version 1".

Signature de la fonction :

Public Function MaLocation1(jour As Long, km As Double, cat As String) As Double

Entrée : Km (réel), Jour (réel), Catégorie (chaîne)

Sortie : Prix de la location (réel)

Calcul :

Si (Jour > 30)

Alors prix = 75 * Jour

Sinon

Si (Catégorie = "luxe")

Alors prix = 80 * Jour + 0.2 * Km

Sinon prix = 70 * Jour + 0.15 * Km

EXERCICE 7

Écrire une fonction "Location de véhicules-version 2".

Signature de la fonction :

Public Function MaLocation2(jour As Long, km As Double, carburant As String) As Double

Entrée : Km (réel), Jour (entier), Carburant (chaîne)

Sortie : Prix de la location (réel)

Calcul :

1. Si (Jour > 30)

Alors Prix = 80 * Jour + 0.2 * Km

Sinon Prix = 75 * Jour + 0.1 * Km

2. Si (carburant = "diesel")

Alors Le Prix calculé ci-dessus doit être divisé par 2.

EXERCICE 8

Écrire une fonction "Calcul de salaire de commerciaux".

Signature de la fonction :

Public Function MonSalaire(ca As Double, statut As String) As Double

Entrée : CA (réel), Statut (chaîne)

Sortie : Salaire (réel)

Calcul :

1. Salaire = 0.75 * CA

Dans 2 situations :

(a) Statut = "junior"

(b) Statut = "senior" et (CA > 100)

2. Dans toutes les autres situations : Salaire = 0.25 * CA.

EXERCICE 9

Écrire un programme qui permet à son utilisateur de saisir une valeur entière et qui, en retour lui indique si c'est un nombre premier ou pas.

EXERCICE 10

Écrire un programme qui affiche tous les nombres parfaits compris entre 2 et 10000.

Un nombre parfait est un entier égal à la somme de ses diviseurs, lui exclu.

Par exemple, 28 est un nombre parfait ($28 = 1 + 2 + 4 + 7 + 14$).