Épisode I : Branchements conditionnels et stuctures itératives

Exercice 1

Écrire un script VBA (Sub sommecarresimpairs()) qui demande à l'utilisateur de rentrer un nombre entier compris entre 1 et 100 puis calcule la somme des carrés des nombres impairs compris entre 1 et ce nombre. On pourra s'assurer que l'utilisateur rentre bien un nombre dans l'intervalle demandé.

Exercice 2

Écrire un script VBA (Sub devinelancer()) qui va vous opposer à votre programme. L'ordinateur simule le lancer d'un dé à 6 faces (x=Int(6*Rnd()+1)). Vous devez deviner ce nombre en au plus 3 essais. Si vous devinez le nombre cherché le programme vous félicite sinon le programme vous propose de recommencer jusqu'à atteindre les 3 essais maximum autorisés.

Remarque

Pour obtenir au hasard un entier x compris dans la plage [a,b], on peut utiliser la formule x=Int(((b-a)+1)*Rnd()+a)).

Exercice 3

Écrire un programme (Sub calculesomme()) qui va vous opposer à votre programme. L'ordinateur vous demande de calculer la somme de deux nombres entiers aléatoires compris entre 1 et 100. Vous devez calculer cette somme de tête en au plus 5 essais. Si vous trouvez le nombre cherché le programme vous félicite sinon le programme vous propose de recommencer jusqu'à atteindre les 5 essais maximum autorisés.

Exercice 4

Écrire un programme (Sub palindrome()) qui indique si un mot donné par l'utilisateur est un palindrome. Un mot est un palindrome si l'ordre de ses lettres reste le même quand on le lit de gauche à droite ou inversement. Exemple : radar, kayak, Laval, ressasser.

Indication

En VBA, la fonction StrReverse(uneChaîne) permet d'inverser l'ordre des caractères dans une chaîne passée en argument. Elle renvoie donc la chaîne résultant de cette opération. La comparaison de chaînes de caractères est par défaut sensible à la casse, car la représentation binaire de chaque caractère est considérée. Par exemple, Laval n'est pas un palindrome dans ce contexte. Pour rendre la comparaison insensible à la casse, utiliser l'instruction « Option Compare Text » au début du module de cet exercice ou alors il faut utiliser LCase(uneChaîne) pour rendre en minuscule la chaîne de caractères …

Exercice 5

On suppose un mot et une lettre saisis par l'utilisateur. Écrire le programme (Sub devineunelettre()) qui vérifie si la lettre donnée est une lettre du mot. Le programme s'arrête dès qu'on a vérifié que la lettre est une lettre du mot.

Indication

Pour connaitre la longueur du mot on utilisera len(”mot”) et pour obtenir la ième lettre mid(”mot”,i,1).

Exercice 6

Ce programme (Sub motdepasse()) va opposer deux joueurs. Le premier joueur choisit un mot de passe et le second essaie de le deviner. On pourra limiter le nombre d'essais et déclarer le Joueur 1 vainqueur si le mot de passe n'est pas découvert à l'issue de ces essais autorisés.

Exercice 7

Écrire un programme (Sub nombrevoyelles()) qui renvoie le nombre de voyelles contenues dans une chaîne de caractères entrées par l'utilisateur.

Exercice 8

Écrire un programme (Sub pascal()) qui demande à l'utilisateur de saisir un entier naturel n puis affiche sur la Feuil1 du classeur Excel les n premières lignes du triangle de Pascal.

Lexique des variables

n	(entier)	le nombre de lignes à afficher	DONNÉE
i	(entier)	numéro de ligne courant	INTERMÉDIAIRE
j	(entier)	numéro de colonne courant	INTERMÉDIAIRE
x	(entier)	le nombre courant dans le triangle	INTERMÉDIAIRE

Algorithme

pour i **de** 0 **à** $n-1$ **faire**
  $x \leftarrow 1$
  écrire x
  pour j **de** 1 **à** i **faire**
    $x \leftarrow (x \times (i-j+1))/j$
    écrire “␣”, x
  fpour
  écrire “\n”
fpour

Exercice 9

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 10

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

Exercice 11

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 ($28 = 1 + 2 + 4 + 7 + 14$).

Exercice 12

Deux nombres N et M sont amis si la somme des diviseurs de M (en excluant M lui-même) est égale à N et la somme des diviseurs de N (en excluant N lui-même) est égale à M.
Écrire un programme qui affiche tous les couples (N, M) de nombres amis tels que $0 < N < M \leq MAX$, MAX étant lu au clavier.

Indication

Les nombres amis compris entre 1 et 100000 sont $(220, 284)$, $(1184, 1210)$, $(2620, 2924)$, $(5020, 5564)$, $(6232, 6368)$, $(10744, 10856)$, $(12285, 14595)$, $(17296, 18416)$, $(66928, 66992)$, $(67095, 71145)$, $(63020, 76084)$, $(69615, 87633)$ et $(79750, 88730)$.