TD 4 - Procédures, fonctions et macros

Exercice 1 : Paramètres et portée des variables

Soit le programme VBA suivant :

```
Dim c As Single

Sub testAB()
    Dim a As Single, b As Single
    a = 1: b = 2: c = 10
    MsgBox "Avant, a = " & a & ", b = " & b & " et c = " & c
    procA
    MsgBox "Après procA, a = " & a & ", b = " & b & " et c = " & c
    procB a, b
    MsgBox "Après procB, a = " & a & ", b = " & b & " et c = " & c
    b = funA(a)
    c = funA(30)
    MsgBox "Après funA, a = " & a & ", b = " & b & " et c = " & c
End Sub

Sub procA()
    Dim a As Single, b As Single
    a = 5: b = c
End Sub

Sub procB(b As Single, a As Single)
    MsgBox "Dans procB, a = " & a & "et b = " & b
    c = 50
End Sub

Function funA(x As Single) As Single
    funA = 10 * x + 5
End Function
```

Quels sont les différents affichages effectués par **MsgBox** lors de l'exécution de la macro `testAB` ?

Exercice 2 : Aire d'un triangle

La formule suivante permet de calculer l'aire A d'un triangle dont on ne connaît pas la hauteur, à partir de la longueur de ses trois côtés a, b et c :

$$A = \sqrt{p(p-a)(p-b)(p-c)}$$

où le demi-périmètre p vaut :

$$p = \frac{a+b+c}{2}$$

Q2.1 Écrire une fonction VBA qui prend en paramètre trois réels `a`, `b` et `c` et retourne la valeur de l'aire d'un triangle de côtés de longueurs `a`, `b` et `c`. On pourra utiliser la fonction `Sqr` de VBA qui retourne la racine carrée du nombre réel passé en paramètre.

Q2.2 Écrire en VBA une macro qui demande à l'utilisateur de saisir les valeurs des trois côtés d'un triangle a, b et c puis qui affiche « Plus grande » si l'aire du triangle de côtés de longueurs a, b et c est plus grande que l'aire du triangle de côtés de longueurs 10, 15 et 20, et « Plus petite » sinon. La macro vérifie que les données saisies par l'utilisateur sont bien numériques.

Exercice 3 : Nombre parfait

Un nombre parfait est un entier naturel égal à la somme de ses diviseurs stricts (distincts de lui-même). Par exemple $6 = 1 + 2 + 3$ est un nombre parfait.

Q3.1 Écrire en VBA une fonction `parfait` qui prend en paramètre un entier `n` et retourne vrai si `n` est parfait, et faux sinon.

Q3.2 Écrire en VBA une procédure `nbParfaits` qui prend un entier `n` en paramètre et qui affiche successivement tous les nombres parfaits entre 1 et `n`.

Q3.3 Écrire en VBA une macro qui demande à l'utilisateur de saisir un entier `n` et qui affiche tous les nombres parfaits entre 1 et `n`.

Exercice 4 : Palindrome

Q4.1 Écrire en VBA une fonction `palin` qui prend en paramètre un mot et retourne vrai si ce mot est un palindrome (une séquence de lettres qui se lit indifféremment dans les deux sens, comme « laval »), et faux sinon.
On pourra utiliser les fonctions VBA sur les chaînes de caractère suivantes :

- `Len(s)` : retourne la taille de la chaîne de caractères s
- `Mid(s,i,1)` : retourne le ième caractère de la chaîne s

Par exemple, `Len("bonjour")` retourne 7 et `Mid("bonjour",2,1)` retourne le caractère `o`.

Q4.2 Écrire une un macro de test de `palin`. Cette macro commence par demander à l'utilisateur de saisir un mot puis affiche le message « C'est un palindrome. » si le mot saisi est un palindrome, et « Ce n'est pas un palindrome. » sinon. La macro vérifie aussi que l'utilisateur saisit bien une chaîne de caractères.