Tp n°2 : Fonctions statistiques et VBA (Visual Basic for Applications)

Partie 1: Fonctions statistiques

Excel permet d'effectuer une analyse statistique d'un fichier de données à l'aide d'un utilitaire qu'il faut au préalable charger comme suit:

1. Cliquer sur l'onglet fichier
2. Choisissez Options
3. Sélectionnez le module Compléments
4. Cliquez sur le bouton Atteindre (en bas de page)
5. Cochez la case « Analysis ToolPak », puis cliquez sur OK.

Vous pouvez à présent manipuler aisément des fonctions statistiques prédéfinies telles que

- la **MEDIANE()** qui renvoie la valeur qui se trouve au milieu d'une série ;
- l' **ECARTYPE()** qui mesure la dispersion des valeurs par rapport à la moyenne ;
- la régression linéaire qui permet d'expliquer la corrélation (comportement) d'une des variables avec les autres données.

Exercice 1 : Football

Télécharger le fichier football.xlsx qui présente quelques chiffres statiques de 20 clubs de football. Dans le ruban du menu Données, à droite, cliquez sur le bouton « Utilitaire d'analyse ».

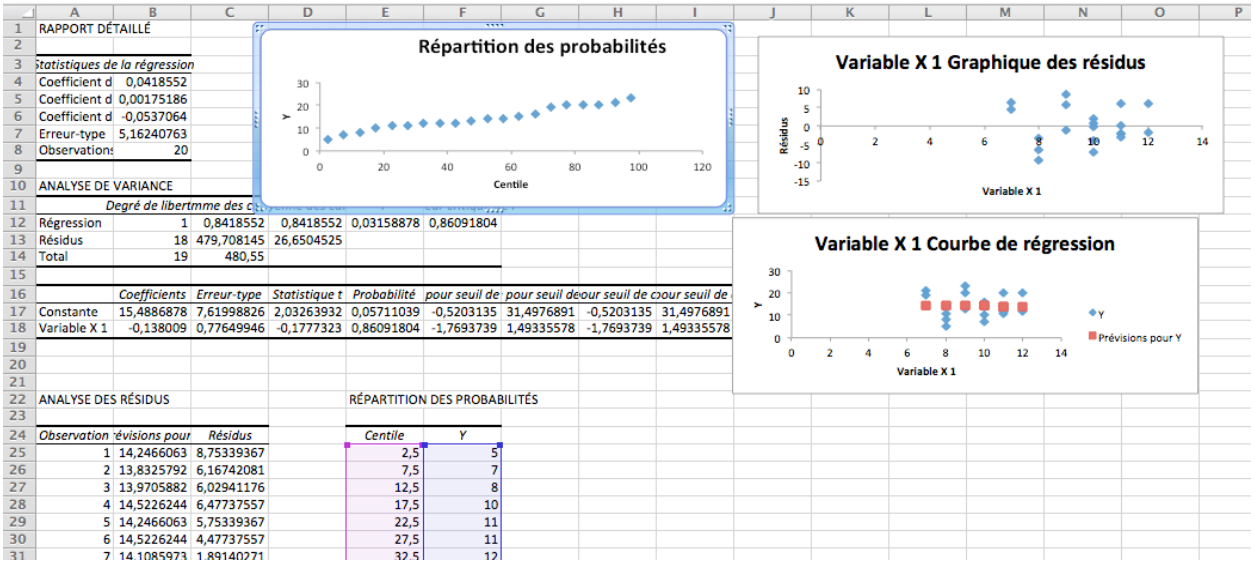
a) Parmi les outils d'analyse, cliquez sur Statistiques descriptives. Remplissez la grille qui s'affiche comme suit : la plage d'entrée contient la donnée $B$1 :$F$21 ; A droite de « Insérer une nouvelle feuille », indiquez RESULTAT. Cochez les cases « Intitulés en première ligne », « Rapport détaillé » et « Niveau de confiance pour la moyenne » (à 95%).

Cliquez enfin sur OK. Vous devrez obtenir une feuille semblable à cette image :

	A	B	C	D	E	F	G	H	I	J
1	*Victoires*		*Nuls*		*Défaites*		*Buts marqués*		*Buts encaissés*	
2										
3	Moyenne	14,15	Moyenne	9,7	Moyenne	14,15	Moyenne	45,8	Moyenne	45,8
4	Erreur-type	1,124546692	Erreur-type	0,341051007	Erreur-type	1,16138529	Erreur-type	2,93849227	Erreur-type	2,038575354
5	Médiane	13,5	Médiane	10	Médiane	14	Médiane	44	Médiane	44,5
6	Mode	20	Mode	11	Mode	6	Mode	50	Mode	40
7	Écart-type	5,029125696	Écart-type	1,525226472	Écart-type	5,19387291	Écart-type	13,141337	Écart-type	9,116786137
8	Variance de l'échantillon	25,29210526	Variance de l'éch	2,326315789	Variance de l'échan	26,9763158	Variance de l	172,694737	Variance de l'éc	83,11578947
9	Kurstosis (Coefficient d'aplatissement)	-0,825753236	Kurstosis (Coeffic	-0,838065576	Kurstosis (Coefficie	-0,249593	Kurstosis (Co	-0,5273405	Kurstosis (Coeff	-0,624022017
10	Coefficient d'asymétrie	0,095024453	Coefficient d'asy	-0,324359021	Coefficient d'asymé	0,24553743	Coefficient d	0,46390305	Coefficient d'as	0,331674943
11	Plage	18	Plage	5	Plage	19	Plage	45	Plage	33
12	Minimum	5	Minimum	7	Minimum	6	Minimum	27	Minimum	29
13	Maximum	23	Maximum	12	Maximum	25	Maximum	72	Maximum	62
14	Somme	283	Somme	194	Somme	283	Somme	916	Somme	916
15	Nombre d'échantillons	20	Nombre d'échan	20	Nombre d'échantill	20	Nombre d'éc	20	Nombre d'écha	20
16	Niveau de confiance(95,0%)	2,353703277	Niveau de confia	0,713827962	Niveau de confianc	2,43080735	Niveau de co	6,15033501	Niveau de confi	4,266787253

b) Parmi les outils d'analyse, cliquez sur Régression linéaire. Remplissez la plage pour la variable Y par $B$2 :$B$21 et la plage pour la variable X par $C$2 :$C$21. A droite de « Insérer une nouvelle feuille », indiquez REGRESSION. Cochez les cases « Diagramme de répartition des probabilités», « Courbes de régression», « Courbes de résidus» et « Niveau de confiance pour la moyenne » (à 95%).

Cliquez enfin sur OK. Vous devrez obtenir une feuille semblable à cette image :



c) Existe-t-il une corrélation entre les buts marqués et les buts encaissés ? Entre les victoires et les buts marqués? Entre les défaites et les buts encaissés ?

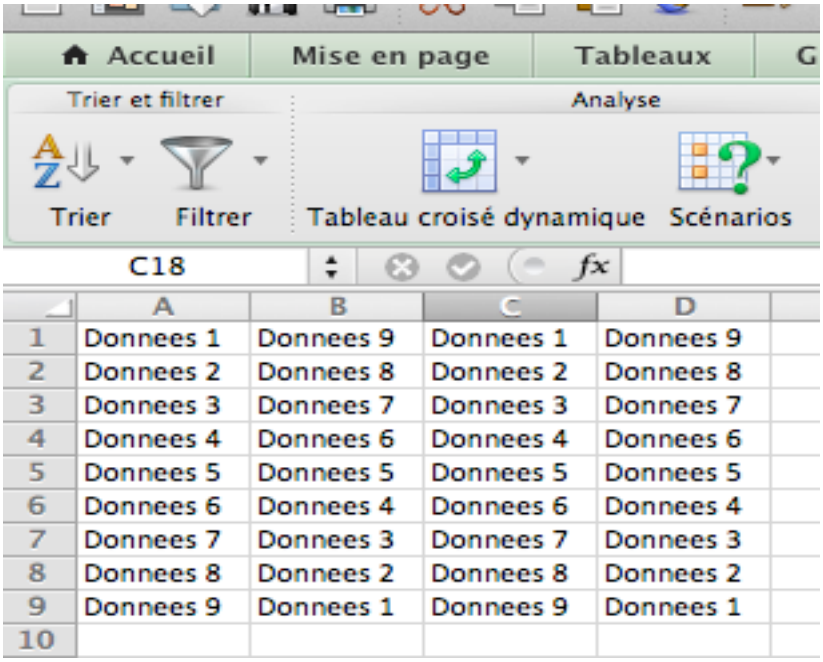
Partie 2 : Macros

Les macros servent essentiellement à automatiser et personnaliser des actions dans le classeur. Vous pouvez ainsi écrire des procédures pour les tâches répétitives, mais aussi adapter l'outil Excel pour qu'il réponde exactement à vos besoins particuliers (interagir avec les manipulations de l'utilisateur, piloter d'autres applications...). En gros, les macros sont pratiques si vous avez régulièrement une longue suite d'actions à effectuer.

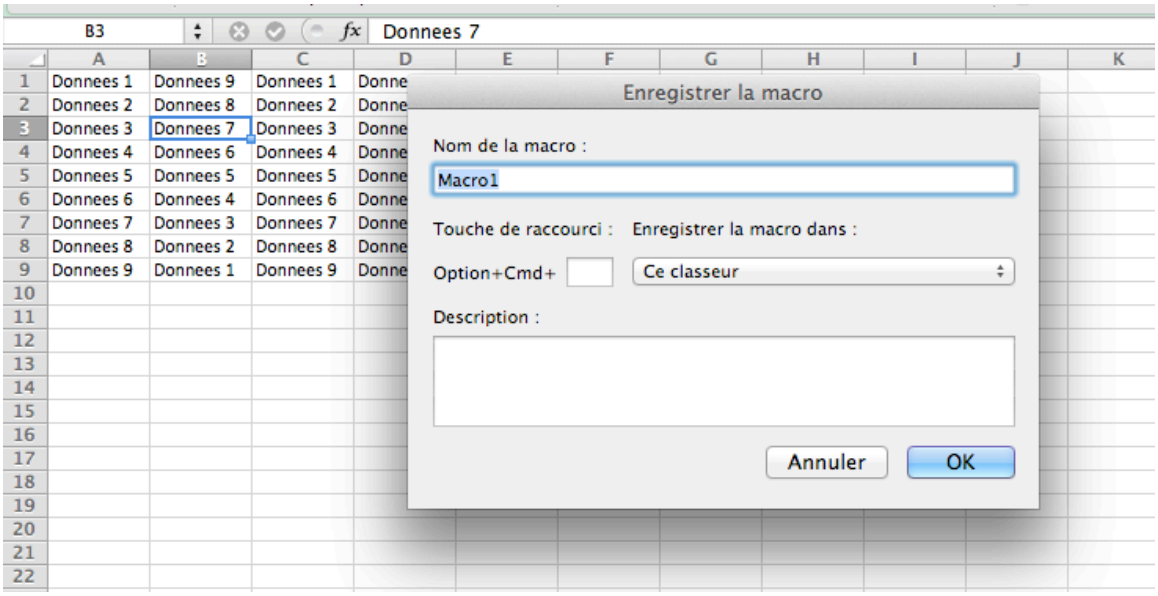
Exercice 2 : Ma première macro.

Dans cet exercice, vous allez pouvoir apprendre à utiliser l'**enregistreur de macro** intégré à Excel et créer ainsi votre première macro qui consiste à mettre en gras et en italique une cellule.

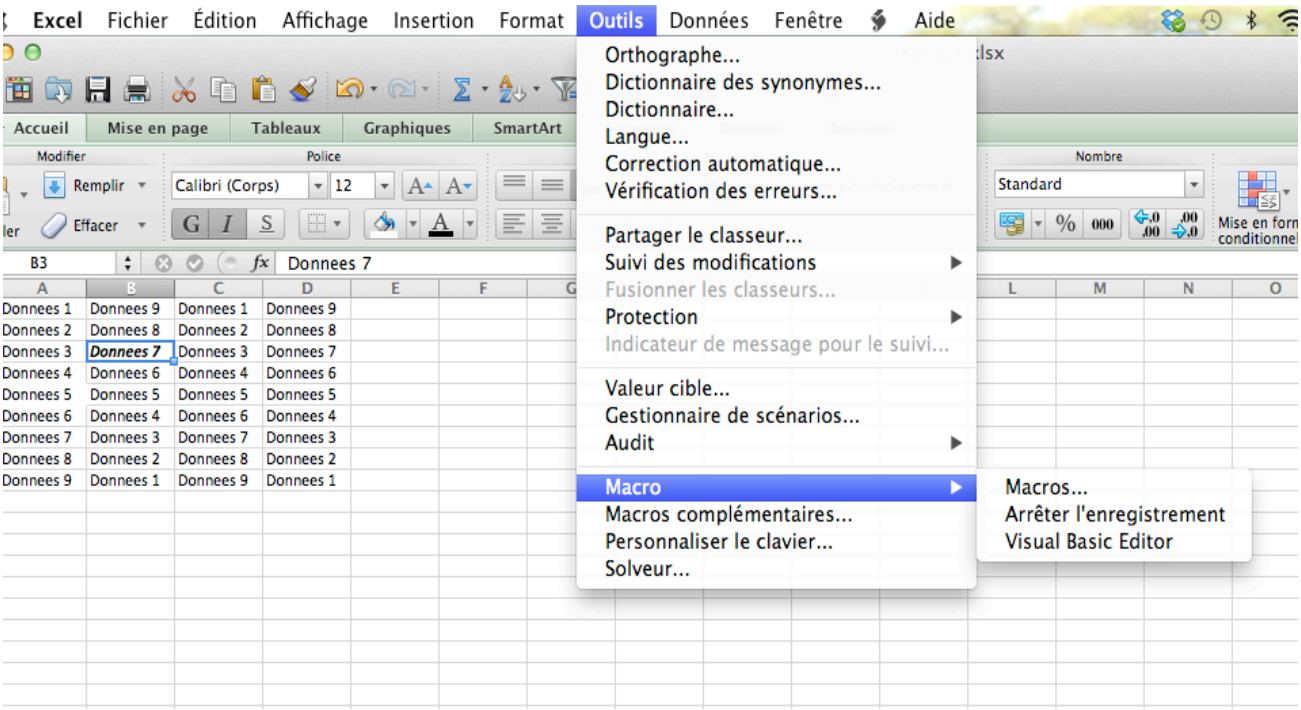
1. Reproduire la feuille Excel suivante :



2. Sélectionner une cellule quelconque, puis dans menu faites Outils -> Macro -> Nouvelle macro. Vous obtiendrez la fenêtre suivante :

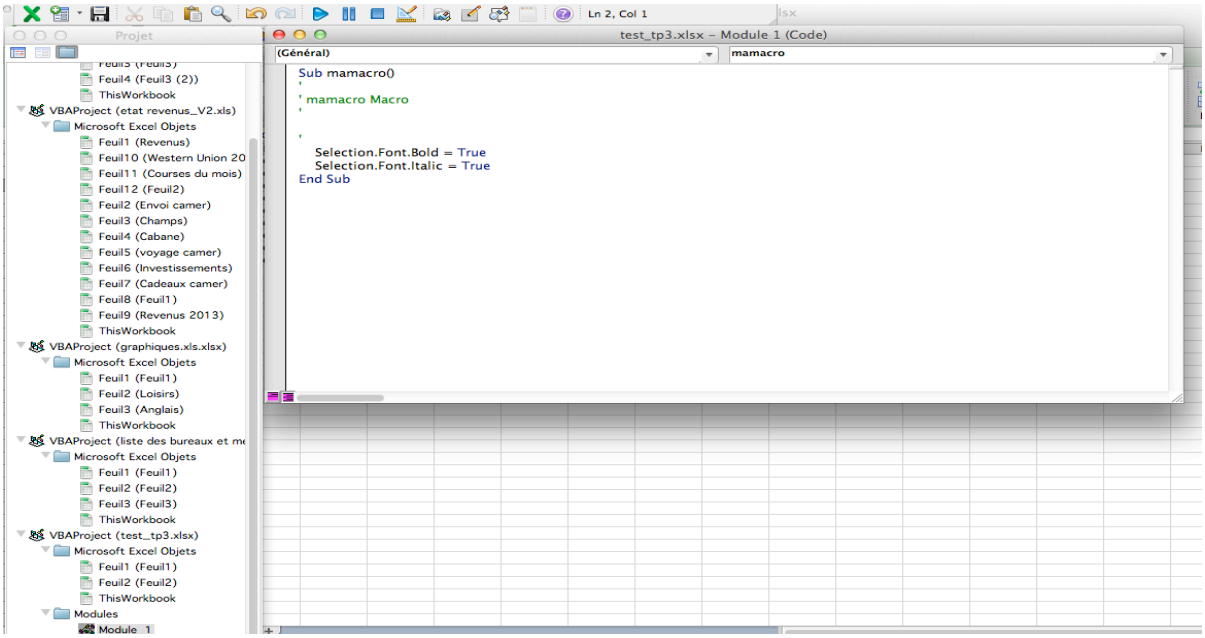


3. Donner un nom à cette macro. Ce nom doit toujours commencer par une lettre et les espaces ne sont pas autorisés. Définissez un raccourci clavier (par exemple Ctrl+m) pour rappeler ultérieurement cette macro et vérifier que l'enregistrement se fait bien dans « Ce classeur ».
4. Cliquez sur OK. L'enregistrement commence à présent. Mettre en gras et en italique la cellule sélectionnée et cliquez sur arrêter l'enregistrement comme ci-dessous (pour certaines versions d'Excel, une barre d'outils flottante permet de stopper l'enregistrement)



5. Sélectionner une autre cellule et exécuter votre macro en faisant dans menu Outils -> Macro -> macro.
6. Pouvez-vous l'exécuter sur une plage de plusieurs cellules sélectionnées en même temps?
7. Nous venons de créer notre première macro grâce à l'enregistreur de macro et nous savons également la rappeler pour reproduire "mécaniquement" une suite d'actions préenregistrées.

Ouvrez à nouveau la boite de dialogue de votre macro et cliquez sur le bouton « Modifier » . Vous devriez obtenir ceci :



Vous êtes alors projetés instantanément dans l'Editeur Visual Basic (VBE) et à l'endroit même où est stocké le code VBA associé à votre macro enregistrée. La macro complémentaire, dans Excel, est un code écrit en VBA.

Exercice 3 :

Vous recevez chaque semaine la liste des nouveaux clients par email. Cette liste est toujours saisie en minuscule. Afin de vous simplifier la tâche, créez une macro de transformation de la colonne A en majuscules (vous pourrez utilisez la fonction prédéfinie d'Excel **MAJUSCULE**). Le fichier tel que reçu est :

	A	B	C
1	**Nouveaux clients**	**Ventes**	
2	angelli		
3	beaud		
4	bernard		
5	blanes		
6	bouisseau		
7	boussy		
8	bremont		
9	chauvet		
10	chneider		
11	claire		
12	clement		
13	cluny		
14	damien		
15			
16			
17			
18			
19			

Partie 3 : Introduction à VBA

Pour une gestion pratique des macros et de VBA, il faut activer l'onglet « *Développeur* ». Pour cela, cliquez sur Fichier > Options > Personnaliser le Ruban puis cochez "Développeur".

Le VBA (Visual Basic for Applications) est un langage proche du Visual Basic qui nécessite une application hôte pour s'exécuter (Excel dans notre cas).

Une procédure est une suite d'instructions effectuant des actions. Elle commence par **Sub** + NomDeLaProcédure et se termine par **End Sub**. Le nom des procédures doit commencer par une lettre et ne doit pas contenir d'espaces. Utilisez le caractère de soulignement pour séparer les mots. Pour déclarer une procédure, taper Sub et son nom puis taper Entrée. VBE ajoute automatiquement les parenthèses et la ligne **End Sub**.

Une fonction est une procédure qui renvoie une valeur. Elle se déclare de la même façon qu'une procédure. En général, on écrit une instruction par ligne. Il est possible d'ajouter des lignes de commentaire entre les lignes d'instruction ou au bout de celles-ci. Les commentaires sont précédés d'une apostrophe et prennent une couleur différente (définie dans les options de VBE).

Chaque procédure Sub ou Function peut être appelée de n'importe qu'elle autre procédure du projet. Pour restreindre la portée d'une procédure au module, déclarez-la en private.

La déclaration explicite d'une variable se fait par le mot **Dim** (abréviation de Dimension). La syntaxe est "**Dim** NomDeLaVariable **As** Type", Type pouvant être Integer (entier), string (chaîne de caractère), Single ou double (pour les nombre réels)… Une fois une variable déclarée, on peut lui affecter une valeur à l'aide du signe « = ». La valeur peut être une constante ou le résultat d'une expression. Exemple : `Dim Invite As String.`

Les constantes sont définies en utilisant le mot-clé **`Const`** à la place de **`Dim`**, en faisant suivre le type de la variable par sa valeur derrière le symbole =. Exemple : `Const PI As Double = 3.14159`

Les conditions sont très utiles en programmation, elles nous serviront à effectuer des actions en fonction de critères précis (même principe que la fonction SI). La principale fonction est **If**, voici comment elle fonctionne :

```
If [CONDITION ICI] Then ' => SI condition validée ALORS
   'Instructions si vrai
Else ' => SINON
   'Instructions si faux
End If
```

Exercice 4 :

1. Dans l'éditeur VBE rentrer le code suivant

```
Sub Essai()
 Dim Invite as String 'Nom de l'utilisateur
 Invite = "Toto"
'Message bonjour à l'utilisateur
 MsgBox "Bonjour " & Invite
End Sub
```

Que constatez-vous en exécutant la macro `Essai`?

`MsgBox` est une fonction qui affiche un message dans une boîte de dialogue.

2. Dans l'éditeur VBE rentrer le code suivant (nous n'analyserons pas ce code):

```
Sub manipulations_des_colonnes()
'
'Mon premier commentaire !
'
    Columns("A:A").Select
    Selection.ClearContents
    Columns("C:C").Select
    Selection.ClearContents
    Columns("B:B").Select
    Selection.Cut Destination:=Columns("A:A")
    Columns("D:D").Select
    Selection.Cut Destination:=Columns("C:C")
    Columns("C:C").Select
End Sub
```

Que constatez-vous en exécutant cette nouvelle macro dénommée `manipulations_des_colonnes`?

L'instruction `Columns("A:A").Select` permet de sélectionner la colonne A
L'instruction `Columns("A:F").Select` permet de sélectionner les colonnes contigues de A à F

Partie 4 : Les Entrées et sorties en VBA

L'utilisation de boîtes de dialogue au cours de l'exécution d'un programme permet d'interagir avec l'utilisateur, soit en lui donnant des informations (valeurs dites sorties du programme) soit en lui demandant des informations (valeurs dites entrées du programme). Deux fonctions sont définies en VBA afin de permettre l'affichage de boîtes de dialogue :

- **`Application.InputBox`** : affiche une boîte de dialogue présentant une zone de texte dans laquelle l'utilisateur est invité à entrer des informations. Voici les 3 arguments que nous allons renseigner pour utiliser la fonction

`Application.InputBox([TEXTE], [TITRE], [TEXTE 2])` avec

- **Texte** : correspond au message d'invite affiché dans la boite de dialogue
- **Titre**: titre de la boîte de dialogue
- **Texte 2** : texte apparaissant par défaut dans la zone de texte lors de l'affichage de la boite de dialogue.

- **`MsgBox`** : affiche un message à l'intention de l'utilisateur dans une boîte de dialogue et lui propose éventuellement de choisir entre différentes possibilités en cliquant sur l'un des boutons de commande affichés. Voici les 3 arguments que nous allons renseigner pour utiliser la fonction

`MsgBox([TEXTE], [BOUTONS], [TITRE])` avec

- **Texte** : texte de la boîte de dialogue
- **Boutons** : choix des boutons (oui, non, annuler, etc.) + autres options
- **Titre** : titre de la boîte de dialogue

Les valeurs prises par l'option `[BOUTONS]` sont données par le tableau suivant :

Valeur	Description
0	OK
1	OK Annuler
2	Abandonner Recommencer Ignorer
3	Oui Non Annuler
4	Oui Non
5	Recommencer Annuler

Les valeurs renvoyées par **`MsgBox`** sont données par le tableau suivant :

Valeur	Bouton correspondant à la valeur
1	OK
2	Annuler
3	Abandonner
4	Recommencer
5	Ignorer
6	Oui
7	Non

Exercice 5 :

1. Exécutez le code VBA suivant :

```
Sub age()
Dim age As Integer
' Retourne l'age de l'utilisateur
age = Application.InputBox("Quel est votre age?", "Age", "0")
End Sub
```

2. Sélectionner la cellule B2 et Exécutez le code VBA suivant :

```
Sub effacer_B2()
    If MsgBox("Etes-vous certain de vouloir supprimer le contenu de B2 ?", 3, "Demande de confirmation") = 6 Then
```

```
        Range("B2").ClearContents
        MsgBox "Le contenu de B2 a été effacé !"
    End If
End Sub
```

Que constatez-vous ? A quoi correspondent les chiffres 3 et 6 dans ce code ?

Exercice 6 :

Ecrire une fonction VBA qui demande à l'utilisateur s'il est mineur, et si tel n'est pas le cas, une autre boite de dialogue s'affiche pour lui demander de rentrer son vrai âge.

Partie 5: Les fonctions en VBA

Une fonction est une suite d'instructions qui renvoie une et une seule valeur (résultat d'une expression).La déclaration d'une fonction VBA s'e_ectue à l'aide du mot clé **Function** selon le format suivant :

```
Function nomFonc (par As typePar , . . . ) As typeRes
'Commentaires
instructions
nomFonc = expression
End Function
```

où `nomFonc` désigne le nom de la fonction et `par` le nom d'un paramètre de type `typePar`. Une fonction peut être déclarée avec un, plusieurs ou sans paramètres. `As typeRes` permet de spécifier le type de valeur renvoyée par la fonction. `Instructions` désignent les instructions de la fonction à définir.

Pour calculer la somme des n premiers entiers naturels par exemple, on peut définir la fonction `sommeEnt` suivante:

```
Function sommeEnt (n As Long) As Long
' Fonction retournant la somme des n premiers entiers
sommeEnt = (n *(n + 1))/2
End Function
```

Pour exécuter une fonction, il suffit de faire appel à elle en écrivant son nom (en respectant la casse) suivi de parenthèses contenant les arguments éventuels. L'appel d'une fonction sans paramètres consiste donc à écrire son nom suivi de parenthèses vides.

Pour afficher la somme des 10 premiers entiers naturels par exemple, on peut écrire la macro `afficheSom10` qui affiche le résultat de l'appel de la fonction `sommeEnt` avec 10 comme paramètre :

```
Sub afficheSom10()
' Aff iche la somme des 10 premiers entiers
MsgBox sommeEnt(10)
End Sub
```

Si on souhaite afficher le résultat de `afficheSom10()` dans la cellule `A11,` on remplace alors `MsgBox sommeEnt(10)` par `Range("A11") = sommeEnt(10).`

Exercice 6:

1. Ecrire une fonction qui permet de calculer la circonférence d'un cercle dont on connaît le diamètre d, et la tester pour d=10
2. Ecrire une fonction qui permet de calculer la surface d'un rectangle et la tester avec les valeurs de votre choix.

Exercice 7 :

Le but de cet exercice est de calculer le montant de l'impôt que l'utilisateur de votre programme va payer. Le calcul de l'impôt est progressif. Le revenu de la personne est considéré en trois tranches : de 0 à 1000, de 1001 à 5000 et plus de 5000. La personne va donc payer :
– 10% de la tranche de 0 à 1000 euros
– 30% de la tranche de 1001 à 5000 euros
– 50% de la tranche au-dessus de 5000 euros.

Par exemple l'impôt correspondant au revenu 3000 est (0,10*1000+0,3* (3000-1000)).

On considère que le revenu de l'utilisateur est déjà saisie dans la cellule G15. Ecrire un programme qui doit afficher l'impôt à payer dans la cellule J15. (Exécuter votre macro avec différents revenus que vous allez saisir dans la cellule G15).

Exercice 8 :

Soit le programme VBA suivant :

```
Option Explicit

Dim a As Single, b As Single

Sub testAB()
    Dim a As Single
    a=1 : b=2

    MsgBox "Avant, a=" & a & " et b=" & b
    procA
    MsgBox "Après procA, a=" & a & " et b=" & b
    procB
    MsgBox "Après procB, a=" & a & " et b=" & b

End Sub

Sub procA()
    Dim b As Single : b=4
End Sub

Sub procB()
    b=3 : a=10
End Sub
```

1. Quel est l'affichage effectué par MsgBox lors de l'exécution de la macro testAB ?
2. Soient a et b deux variables de même type, proposez un algorithme échangeant leur contenu ?