Année 2023-2024

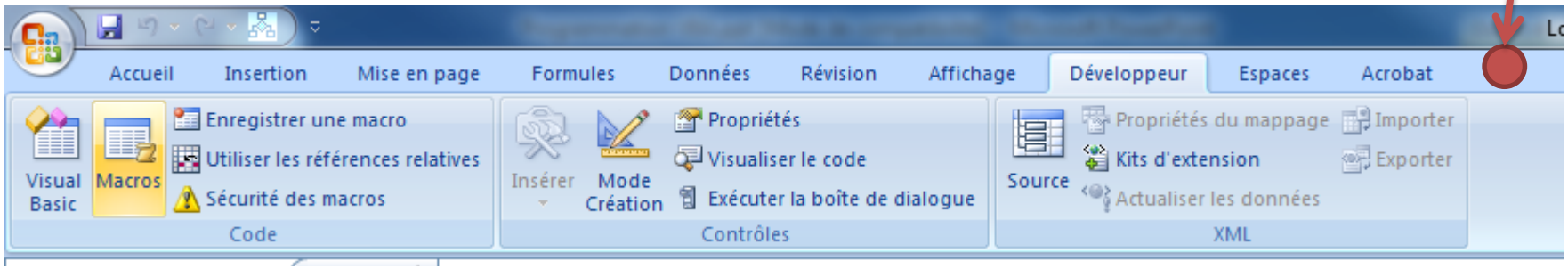
Introduction à la programmation VBA

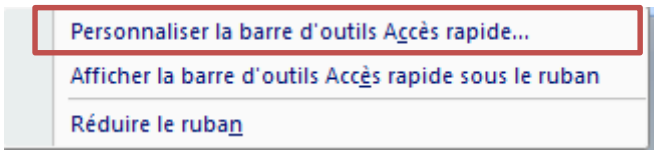
VBA

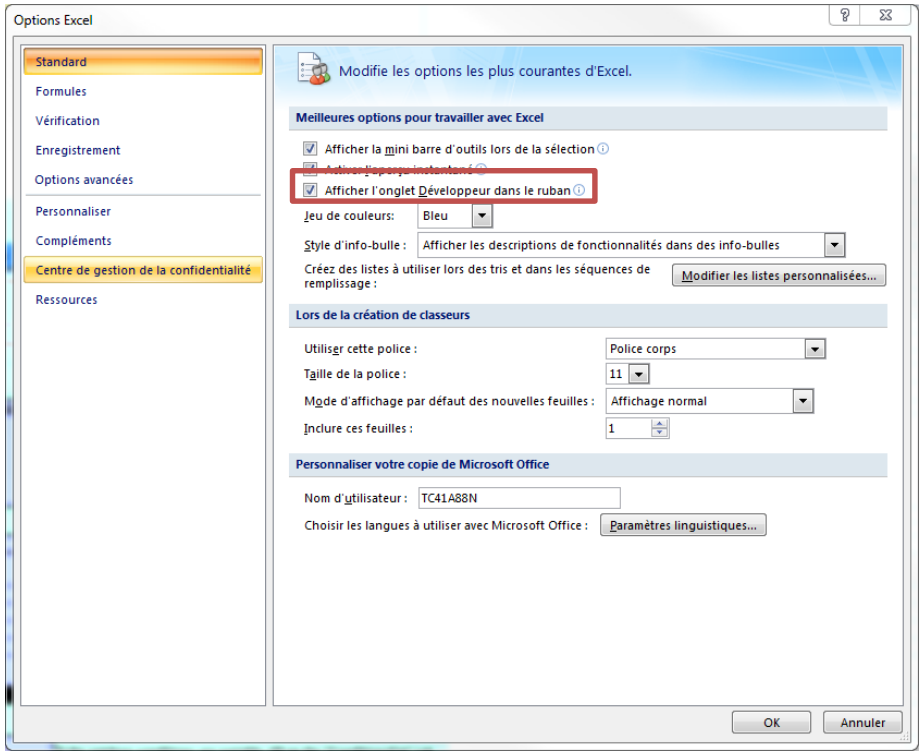
- Visual : Programmation visuelle = éditeur intelligent qui reconnaît les mots clés du language et permet le débogage
- Basic : Beginner's All Purpose Symbolic Instructions Code = code à tout faire pour débutants
- Application : Commun à tous les outils Microsoft

Ajouter l’onglet développeur à Excel

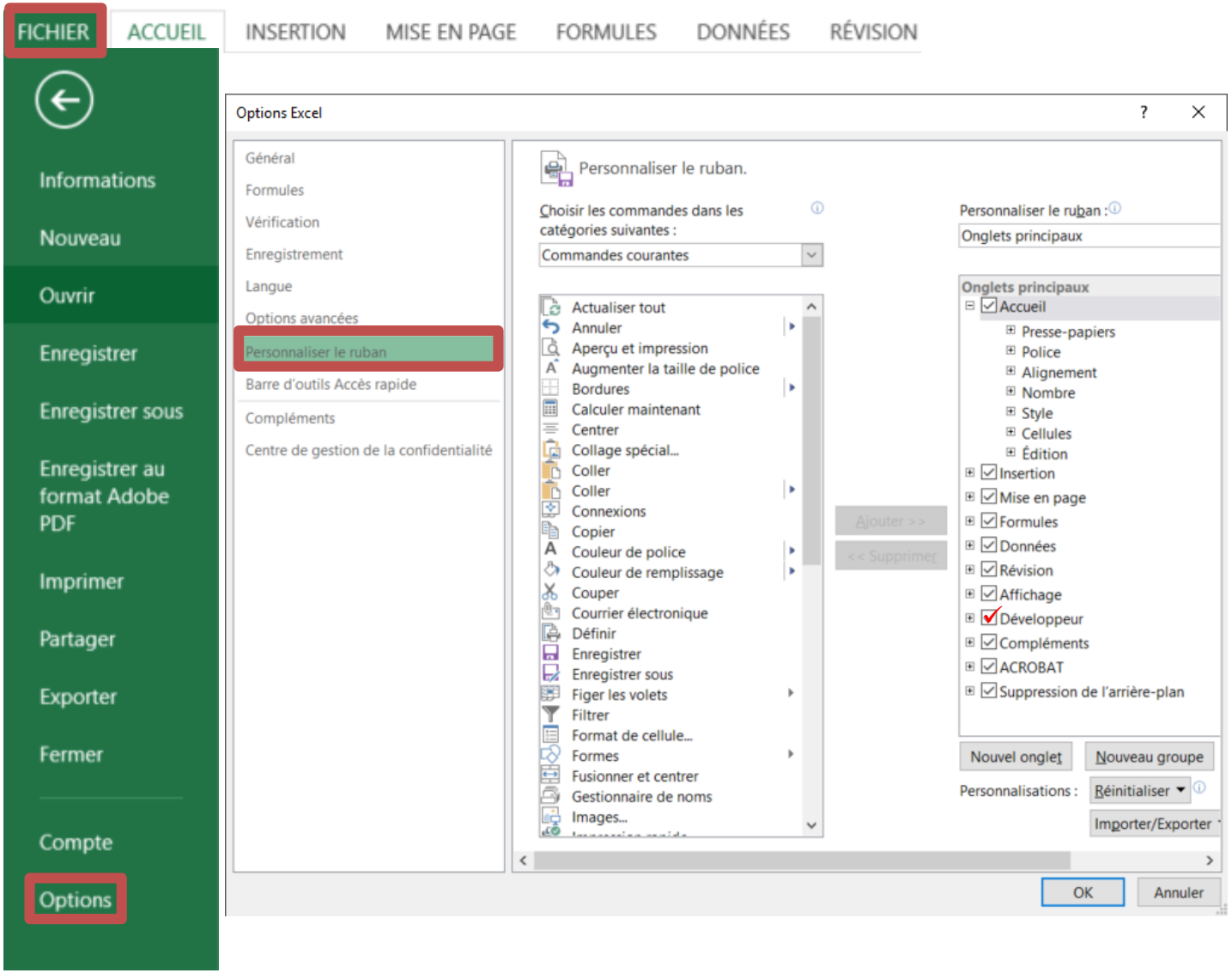
Clic droit







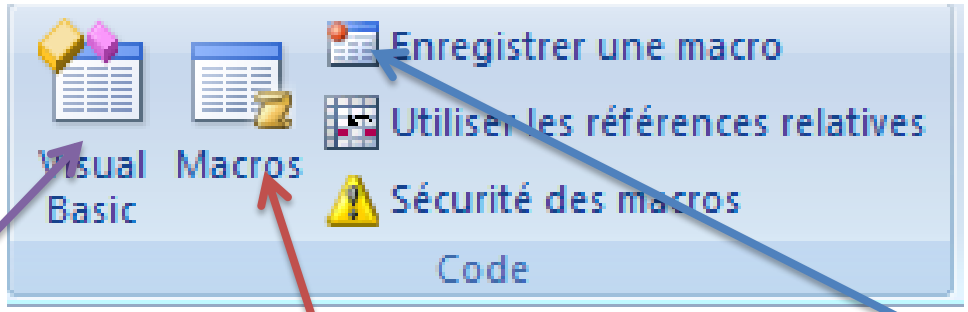
Ajouter l’onglet développeur à Excel



Allez dans « Fichier » puis « Options ».

Sélectionner « Personnaliser le ruban » et cocher la case « Développeur ».

Options de l'onglet Développeur

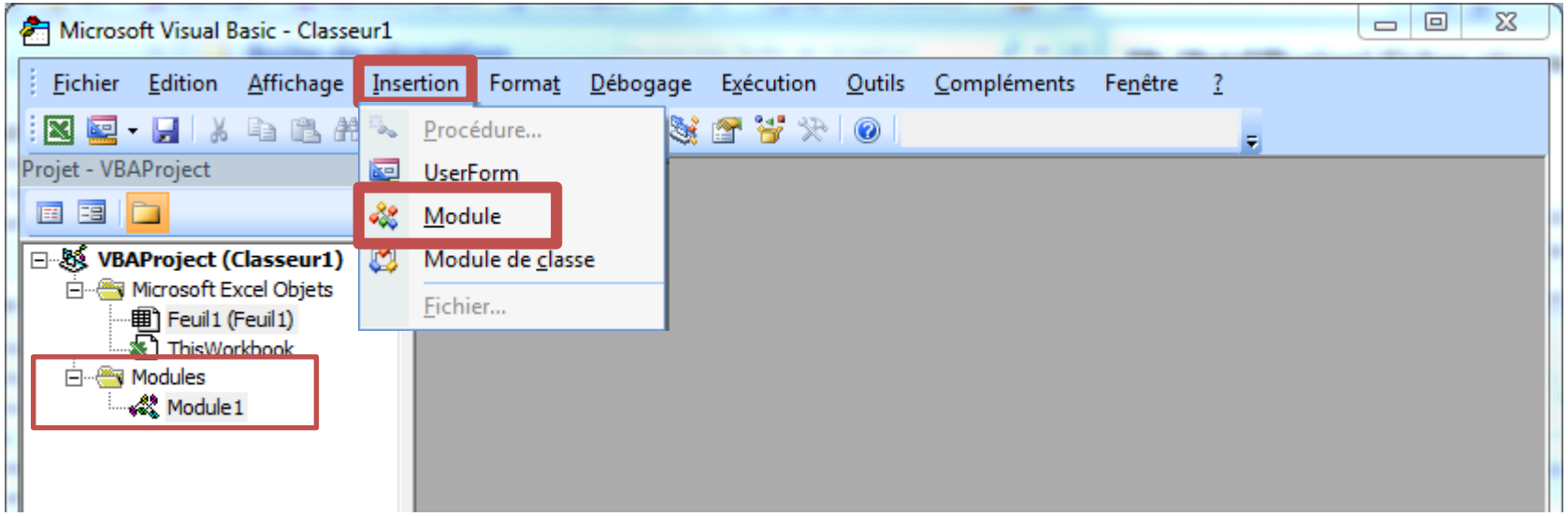


Permet d'ouvrir le script VBA
Cet emplacement nous permet de visualiser les macros.

Permet d'enregistrer une macro
Cela permet d'enregistrer toutes les actions que vous faites lorsque vous modifiez un document actif. Les modifications se retrouvent sous forme de code dans Visual Basic

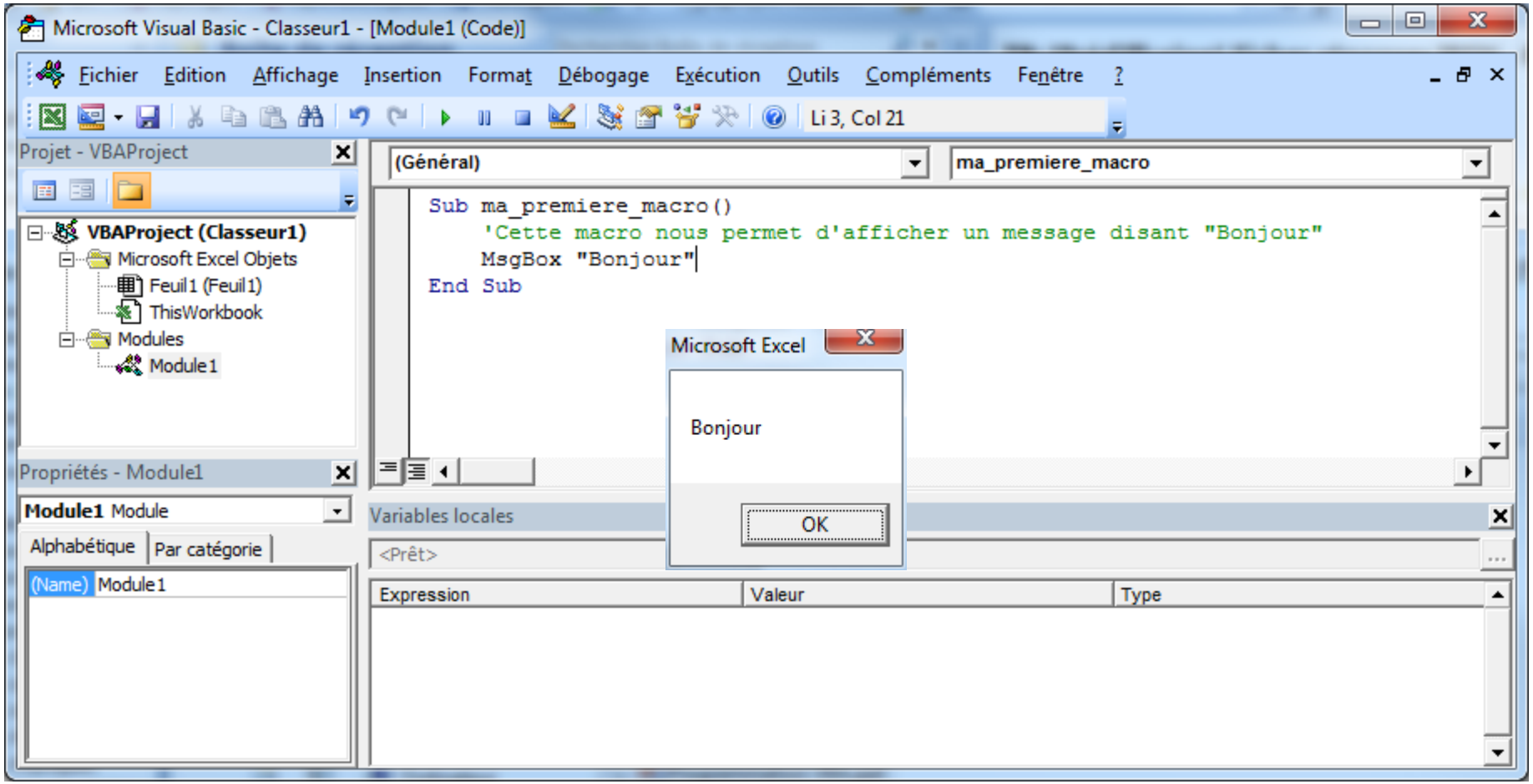
Permet d'exécuter une macro
La fenêtre vous permet d'exécuter votre code si vous n'avez pas créé de boutons particuliers.

Créer un premier script



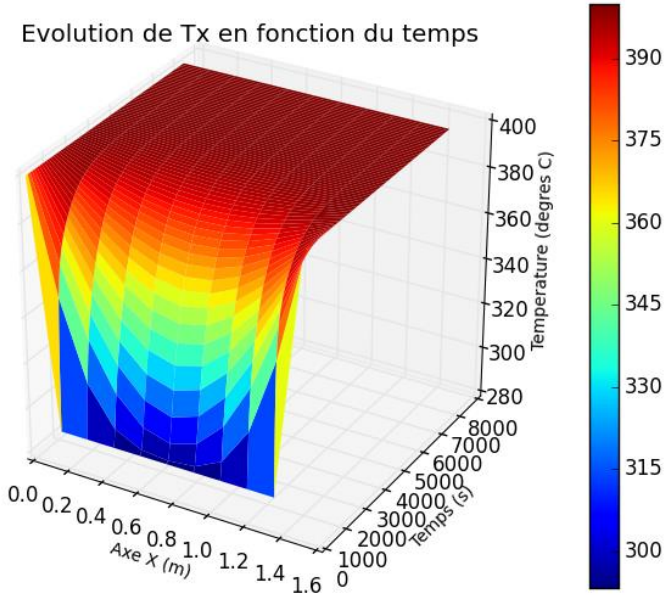
Pour créer votre macro, il faut commencer par créer un module.

Créer un premier script



Fil rouge

Le cours sera structuré autour de la modélisation du transfert de la chaleur dans une plaque.



Un peu de math

La résolution de ce problème demande de petites connaissances en physique:

Conduction	Convection
$\rho . C_p . \frac{\partial T}{\partial t} - \lambda \nabla^2 \mathrm{T} = 0$	$\rho . C_p . \frac{\partial T}{\partial t} - \mathrm{h . A} . \nabla^2 \mathrm{T} = 0$

Et en mathématiques :

Conduction	Convection
$\frac{\partial T}{\partial t} = \frac{\lambda}{\rho . C_p} \frac{\partial^2 T}{\partial x^2}$	$\frac{\partial T}{\partial t} = \frac{\mathrm{h . A}}{\rho . C_p} \frac{\partial^2 T}{\partial x^2} - \frac{\mathrm{h_i}}{\rho . C_p} (T - T_i) - \frac{\mathrm{h_e}}{\rho . C_p} (T - T_e)$

Un peu de math

Pour transformer des différentielles, nous utilisons la méthodes des différences finies:

$$\frac{\partial T}{\partial t} = \frac{T_x^{t+\Delta t} - T_x^t}{\Delta t}$$

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_{x-1}^t - 2.T_x^t + T_{x+1}^t}{(\Delta x)^2}$$

Un peu de math

Les équations finales sont les suivantes:

Conduction

$$T_x^{t+\Delta t} = \mu_m . T_{x-1}^t + (1 - 2.\mu_m).T_x^t + \mu_m . T_{x+1}^t$$

Avec convection

$$T_x^{t+\Delta t} = \mu_m . T_{x-1}^t + (1 - 2.\mu_m).T_x^t + \mu_m . T_{x+1}^t - \mu_i (T_x^t - T_i^\infty) - \mu_e (T_x^t - T_e^\infty)$$

$$\mu_m = \frac{\lambda . \Delta t}{\rho . C_p . (\Delta x)^2} \quad \mu_i = \frac{h_i . \Delta t}{\rho . C_p} \quad \mu_e = \frac{h_i . \Delta t}{\rho . C_p}$$

Un peu de math

Pour obtenir la matrice suivante:

$$\begin{Bmatrix} T_1^{t+\Delta t} \\ T_2^{t+\Delta t} \\ T_3^{t+\Delta t} \\ T_4^{t+\Delta t} \\ T_5^{t+\Delta t} \end{Bmatrix} = \begin{bmatrix} 1-(\mu_1+\mu_i) & \mu_1 & 0 & 0 & 0 \\ \mu_1 & 1-2.\mu_1 & \mu_1 & 0 & 0 \\ 0 & \mu_1 & 1-2.\mu_1 & \mu_1 & 0 \\ 0 & 0 & \mu_1 & 1-2.\mu_1 & \mu_1 \\ 0 & 0 & 0 & \mu_1 & 1-(\mu_1+\mu_e) \end{bmatrix} . \begin{Bmatrix} T_1^t \\ T_2^t \\ T_3^t \\ T_4^t \\ T_5^t \end{Bmatrix} + \begin{Bmatrix} \mu_i . T_i^\infty \\ 0 \\ 0 \\ 0 \\ \mu_e . T_e^\infty \end{Bmatrix}$$

Il nous reste 5 cours pour faire le programme correspondant.

Les variables

Qu'est ce qu'une variable?

Une variable est un espace de mémoire dans créé par l'ordinateur pour y stocker des informations.

Analogie:

Vous pouvez imaginer cela comme une boite à laquelle vous donnez un nom et dans laquelle vous stockez des informations.

Les variables

Nom	Type	Détails	Mémoire
Byte	Numérique	Nombre entier (0 => 255) [2^8]	1 octet
Integer	Numérique	Nombre entier (± 32 767) [2^{15}]	2 octets
Long	Numérique	Nombre entier (± 2 147 483 647) [2^{31}]	4 octets
Currency	Numérique	Nombre à décimale fixe (± 922 337 203 685 477.5807) [$2^{63}/1000$]	8 octets
Single	Numérique	Nombre à virgule flottante ($\pm 3,4.10^{38}$) [$2^{128}=2^{2^7}$]	16 octets
Double	Numérique	Nombre à virgule flottante ($\pm 3,7.10^{308}$) [$2^{1023}=2^{2^{10}}$]	128 octets
String	Texte	Texte	
Date	Date	Date / Heure	
Boolean	Boolean	True / False	1 bite
Object	Objet	Objet Microsoft	
Variant	Tous	Tout type de données	

Déclarer des variables

En programmation, il est préférable de déclarer les variables. Cela économise du temps de calcul.

- Déclarer une variable à la fois

```
Dim a As String
Dim b As Integer
Dim variable As Boolean
```

- Déclaration multiple

```
Dim a As String, b As Integer, variable As Boolean
```

Opération sur les variables (nombres)

Les opérations sur les nombres sont les suivantes:

```
Dim a As Integer, b As Integer
a = 4
b = 9

c = a + b         'Addition
d = a - b         'Soustraction
e = a * b         'Multiplication
f = a / b         'Division
g = a Mod b       'Reste de la division
```

Opération sur les variables (String)

Les opérations sur les chaines de caractères utiles sont les suivantes:

```
Dim a As String, b As String
a = "je m'appel Thomas"
b = "j'ai 28ans"

c = a + b           'Addition de chaines
d = a & b           'Addidion de chaines (méthode alternative)
e = a + " et " + b  'Addition de chaines

f = Len(a)          'Nombre de caractères
g = Left(b, 3)      'Récupère les 3 premiers caractères en partant de la gauche
h = Right(a, 5)     'Récupère les 5 derniers caractères de la chaine
i = Mid(a, 5, 2)    'Récupère les 2 caractères à partir du 5ème caractère de la chaine

j = InStr(a, "e")   'Renvoie le nombre de fois où se trouve "e" dans la chaîne

k = UCase(a)        'Toutes les lettres en majuscules
l = LCase(k)        'Toutes les lettres en minuscules

m = Replace(a, "m'appel", "me nomme")   'Remplace un morceau de la chaine par un autre
```

a	"je m'appel Thomas"
b	"j'ai 28ans"
c	"je m'appel Thomasj'ai 28ans"
d	"je m'appel Thomasj'ai 28ans"
e	"je m'appel Thomas et j'ai 28ans"
f	17
g	"j'a"
h	"homas"
i	"'a"
j	2
k	"JE M'APPEL THOMAS"
l	"je m'appel thomas"
m	"je me nomme Thomas"

Opérateurs de comparaison

Opérateur	Signification
=	Est égal à
<>	Est différent de
<	Est plus petit que
<=	Est plus petit ou égal à
>	Est plus grand que
>=	Est plus grand ou égal à

Autres opérateurs utiles	
AND	[Condition 1] And [condition 2] Les deux conditions doivent être vraies
OR	[Condition 1] Or[condition 2] Au moins 1 des 2 conditions doit être vraie
NOT	Not [Condition 1] La condition doit être fausse

Opérateur de comparaison

Lorsque nous utilisons un opérateur de comparaison, le résultat sera « True » ou « False ».

```
'Opérateurs de comparaison
a = 1
b = 2
test1 = a = b    'False
test2 = a <> b   'True
test3 = a > b    'False
test4 = a < b    'True

test5 = True And False      'False
test6 = True And True       'True
test7 = False And False     'False

test8 = True Or False       'True
test9 = False Or False      'False

test10 = Not False          'True
test11 = Not True           'False
```

Utilisation des données du classeur

Pour aller chercher une donnée dans une cellule du classeur, nous devons renseigner son emplacement:

```
Workbooks("nom_fichier.xlsm").Sheets("Nom_feuille").Range("A1").Value
```

Workbooks = Classeur

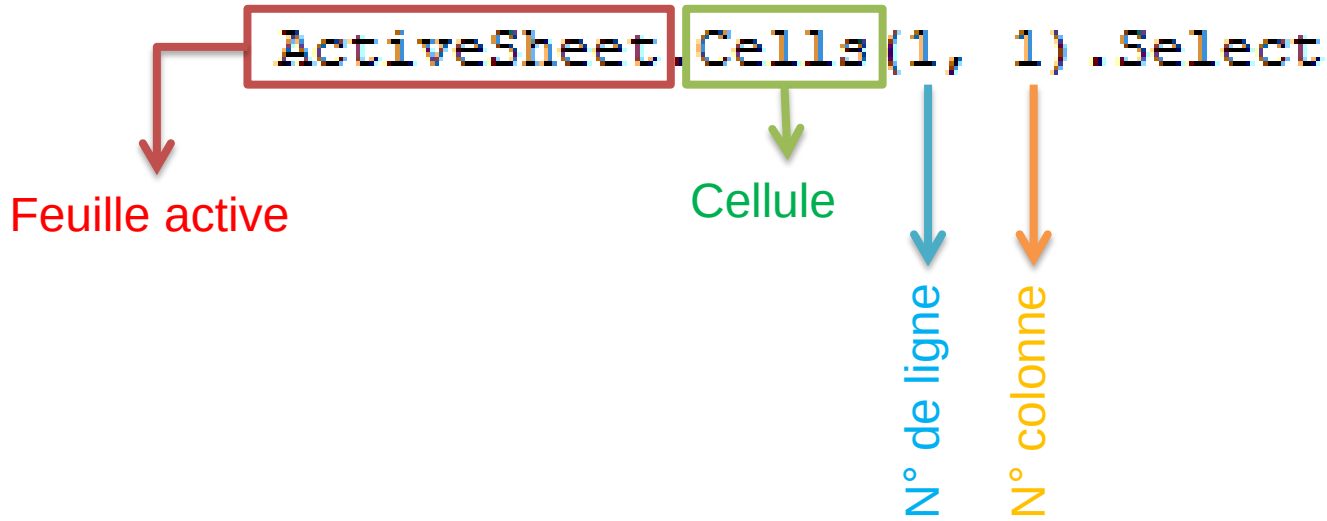
Sheets = Feuille

Range/Cells = Champ/Cellule

* Si vous ne renseignez que les coordonnées de la cellule, le programme ira chercher les informations dans le classeur actif, dans la feuille active.

Utilisation des données du classeur

Il est possible de renseigner une cellule par ses coordonnées:



Ecriture équivalente:

```
ActiveSheet.Range("A1").Select
```

Attributs de « Range »

Voici quelques attributs que vous pouvez utiliser:

```
a = Range("B1").Value              'Valeur (retourne un tableau si plage)
b = Range("A1").Formula            'Formule (ex:"=sum(A2:A10)")
c = Range("A1").FormulaLocal       'Formule en français (ex:"=somme(A2:A10)")
d = Range("A1").Left     'Espacement par rapport à la bordure gauche (pixels)
e = Range("A1").Top      'Espacement par rapport à la bordure haute (pixels)
f = Range("A1").MergeCells         'True si la cellule est fusionnée à un autre
g = Range("A1").Name     'Nom feuille + coordonées ("=Feuil1!$A$1")
h = Range("A1").Name.Name          'Nom de la case si vous lui en avez donné un
```

Vous pouvez parcourir les différentes options à l'aide de la fenêtre des variables locales.

Attributs de « Cells »

Voici quelques attributs que vous pouvez utiliser:

```
a = Cells(1, 1).Value              'Valeur (retourne un tableau si plage)
b = Cells(1, 1).Formula            'Formule (ex:"=sum(A2:A10)")
c = Cells(1, 1).FormulaLocal       'Formule en français (ex:"=somme(A2:A10)")
d = Cells(1, 1).Left     'Espacement par rapport à la bordure gauche (pixels)
e = Cells(1, 1).Top      'Espacement par rapport à la bordure haute (pixels)
f = Cells(1, 1).MergeCells         'True si la cellule est fusionnée à un autre
g = Cells(1, 1).Name     'Nom feuille + coordonées ("=Feuil1!$A$1")
h = Cells(1, 1).Name.Name          'Nom de la case si vous lui en avez donné un
i = Cells(1, 1).Address            'Retourne l'adresse de la cellule ("$A$1")
```

Lire / écrire dans une cellule

Je veux prendre la valeur de la cellule « A1 »

```
variable = Range("A1").Value
```

Et la mettre dans la cellule « A2 »

```
Range("A2").Value = variable
```

Exercice

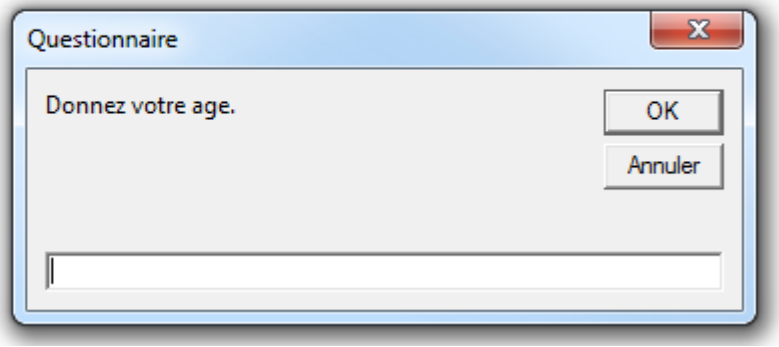
Créer un programme qui vous demande votre nom, taille et poids pour vous donner votre IMC

Pour créer une fenêtre vous permettant de donner une information, vous pouvez utiliser « InputBox ».

La donnée rentrée va être stockée dans « variable ».

$$IMC = \frac{Masse(kg)}{(Taille(m))^2}$$

```
Sub importer_information()
    'Cette fenêtre permet d'importer une information
    variable = InputBox("Donnez votre age.", "Questionnaire")
End Sub
```



Les conditions

If(eng) = Si

Else if(eng) = D'autre si

Else(eng) = Autre

End(enf) = Fin

Les conditions servent à effectuer des actions en fonction de critères précis.

```
If [condition 1] Then   'Si la condition 1 est validée alors
    'Instructions 1

ElseIf [condition 2] Then   'Si la condition 2 est validée alors [optionnel]
    'Instructions 2

Else    'Sinon (si aucune des conditions n'est validée) [optionnel]
    'Instruction 3

End If  'Fin des conditions
```

Exemple

```
Sub commentaire_sur_la_note()
    'Les élèves sont notés sur 6
    'La note se trouve dans la cellule A1
    note = Range("A1").Value

    If note = 6 Then
        commentaire = "Parfait"
    ElseIf note > 3 Then
        commentaire = "Bien"
    ElseIf note = 3 Then
        commentaire = "Vous avez la moyenne"
    ElseIf note >= 1 Then
        commentaire = "Mauvais résultat!"
    ElseIf note = 0 Then
        commentaire = "Très mauvais résultat!"
    Else
        commentaire = "Erreur, cette note n'est pas permise!"
    End If

    'Le commentaire est ajouté dans la cellule B1
    Range("B1").Value = commentaire

End Sub
```

Les conditions (« Case »)

Case(eng) = Cas

Lorsque vous avez plusieurs conditions, vous pouvez utiliser l'outil suivant:

```
Sub commentaire_sur_la_note()
    'Les élèves sont notés sur 6
    'La note se trouve dans la cellule A1
    note = Range("A1")

    Select Case note     ' <= la valeur à tester (ici, la note)
    Case Is = 6          ' <= si la valeur = 6
       commentaire = "Excellent résultat !"
    Case Is = 5          ' <= si la valeur = 5
       commentaire = "Bon résultat"
    Case Is = 4          ' <= si la valeur = 4
       commentaire = "Résultat satisfaisant"
    Case Is = 3          ' <= si la valeur = 3
       commentaire = "Résultat insatisfaisant"
    Case Is = 2          ' <= si la valeur = 2
       commentaire = "Mauvais résultat"
    Case Is = 1          ' <= si la valeur = 1
       commentaire = "Résultat exécrable"
    Case Else            ' <= si la valeur n'est égale à aucune des valeurs ci-dessus
       commentaire = "Erreur, cette note n'est pas permise!"
    End Select

    'Commentaire en B1
    Range("B1") = commentaire
End Sub
```

Exercices sur les conditions

Sélection de tarif EDF. Vous devez renseigner la puissance voulue dans la cellule A2 et:

- La macro affiche le tarif de l'abonnement dans B2
- La macro affiche le prix du kWh dans C2

Puissance souscrite (kVA)	Abonnement (€TTC/mois)	Prix du kWh (€TTC/mois)
3	5,74	15,55
6	8,92	14,67
9	10,42	14,83
12	11,96	14,83
15	13,50	14,83

Conditions : Exercices

Prix des places de ciné:

- Moins de 14 ans : 5,00€
- Moins de 18 ans : 7,70€
- Etudiant : 8,70€
- Adulte : 12,10€

- 3D +2€
- Lunettes +1€

Faire un programme qui vous donne le prix de votre place de ciné.

Les boucles

Une boucle permet de gérer les répétitions:

```
Sub boucle_while()

    Cells(1, 1) = 1
    Cells(2, 1) = 2
    Cells(3, 1) = 3
    Cells(4, 1) = 4
    Cells(5, 1) = 5
    Cells(6, 1) = 6
    Cells(7, 1) = 7
    Cells(8, 1) = 8
    Cells(9, 1) = 9
    Cells(10, 1) = 10

End Sub
```

=

```
Sub boucle_while()

    i = 1   'Initialisation

    While i <= 10   'Condition pour rester dans la boucle

        Cells(i, 1) = i
        i = i + 1

    Wend

End Sub
```

Boucle « while »

While (eng) = tant que

Les instructions comprises dans la boucle vont se répéter tant que la condition d'entrée est vraie

```
While [Condition]

    'instructions

Wend 'Indique la fin de la boucle
```

Boucle « Do »

Do (eng) = fait

Cette boucle fonctionne de la même manière que la boucle while. Cependant, la condition peut se placer en fin de boucle (Loop = Boucle).

```
Do While [Condition]
    'Instructions
Loop
```

```
Do
    'instructions
Loop While [Condition]
```

Il est possible de quitter la boucle lorsque la condition devient vrai en utilisant « until » (jusqu'à)

```
Do Until [Condition]
    'Instructions
Loop
```

Boucle « For »

For (eng) = depuis

To (eng) = à

La boucle for permet de déterminer un nombre de répétition:

```
For i = 1 To 6
    'instructions
Next
```

Ici, la variable « i » est augmenter de 1 depuis la valeur 1 à la valeur 6.

Exercice

La conjoncture de syracuse est une suite d'entiers naturels définie de la manière suivante:

- On part d'un entier plus grand que zéro.
- Si l'entier est pair, on le divise par 2
- Si l'entier est impaire, on le multiplie par 3 et on ajoute 1
- La suite devient infinie une fois qu'elle a atteint 1 (1,4,2,1,4,2,1…)

Créer un programme qui donne cette suite dans une même colonne.

```
reste = nombre1 Mod nombre2
```

$$\frac{5}{2} = 2 \quad reste = 1$$

$$\frac{11}{3} = 3 \quad reste = 2$$

Exercices

Trouver le plus petit diviseur commun de deux nombres.

La macro devra:

- Aller chercher les nombres dans des cellules que vous choisissez
- Donner le résultat dans une cellule que vous choisissez

Pour trouver le reste d'une division, vous pouvez utiliser:

```
reste = nombre1 Mod nombre2
```

$$\frac{5}{2} = 2 \quad reste = 1$$

$$\frac{11}{3} = 3 \quad reste = 2$$

Exercices

Trouver le plus grand diviseur commun de deux nombres.

La macro devra:

- Aller chercher les nombres dans des cellules que vous choisissez
- Donner le résultat dans une cellule que vous choisissez

Pour trouver le reste d'une division, vous pouvez utiliser:

```
reste = nombre1 Mod nombre2
```

$$\frac{5}{2} = 2 \quad reste = 1$$

$$\frac{11}{3} = 3 \quad reste = 2$$

Exercice

Déterminer si deux nombres sont premiers entre eux:

La macro devra:

- Aller chercher les nombres dans les cellules de votre choix
- Afficher dans la cellule de votre choix:
  - « Premiers » s'ils sont premiers entre eux
  - « Non premiers » s'ils ne le sont pas

Deux nombre sont premiers entre eux si le seul diviseur commun est 1.

Exercice

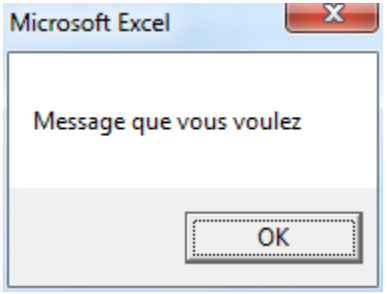
Déterminer si un nombre est premier

La macro devra afficher le message:

- « Premier » si le nombre est premier
- « Non premier » si le nombre n'est pas premier

Pour afficher un message, vous devez utiliser:

```
MsgBox "Message que vous voulez"
```



Exercice

La suite de Fibonacci est une suite d'entiers dans laquelle chaque terme est la somme des deux précédents.

Créez un programme qui:

- Vous demande le premier chiffre
- Vous affiche les n nombres suivants (que vous devez définir).

Exercices

Deux nombres sont dits amicaux si la somme des diviseurs propres de l'un (diviseurs autre que lui-même) égale l'autre.

Ex:

220 et 284

Diviseurs de 220: 1+2+4+5+10+11+20+22+44+55+110=284

Diviseurs de 284: 1+2+4+71+142=220

Ecrivez un programme qui vous demande deux nombres et vous renvoie si ils sont amicaux ou non.

Exercices

La légende raconte qu'un jour, un roi découvrit un nouveau jeu: le jeu d'échec. Le concepteur du jeu voulu que le roi pose un grain de riz sur la première case et double le nombre de grains de riz à chaque case.

Un grain de riz ≈ 0.02g (équivalent à 2.10^{-8} tonnes)

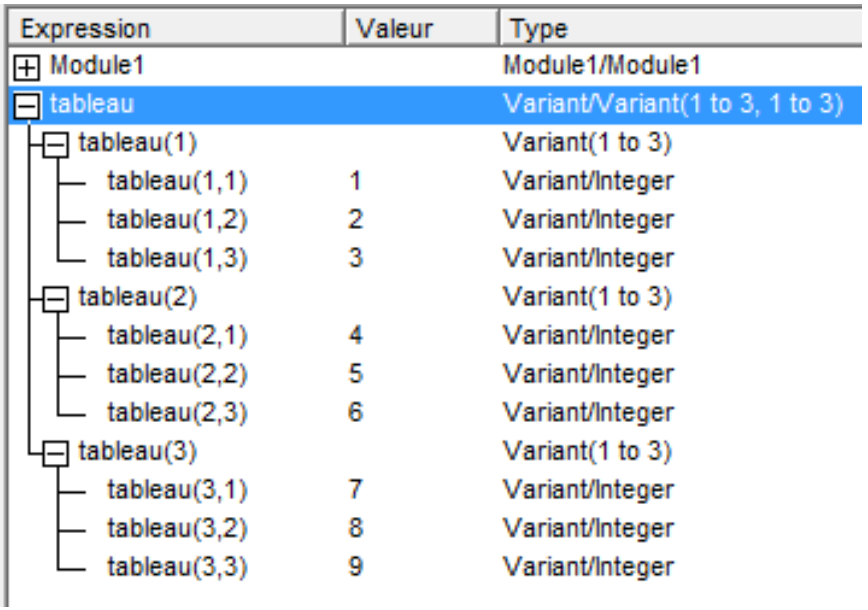
Vous devez créer un programme qui:

- Donne le nombre de grains de riz par case (jeu d'échec = 64 cases)
- Déterminer le nombre de tonnes de riz que le concepteur a gagné.

Les tableaux

L'avantage d'une variable tableau est que vous pouvez y stocker plusieurs valeurs. Pour retrouver une valeur, il suffit de donner ses coordonnées.

```
'Soit la variable tableau suivante
'Tableau à 2 dimensions (3x3)
tableau(1, 1) = 1
tableau(1, 2) = 2
tableau(1, 3) = 3
tableau(2, 1) = 4
tableau(2, 2) = 5
tableau(2, 3) = 6
tableau(3, 1) = 7
tableau(3, 2) = 8
tableau(3, 3) = 9
```



Les tableaux

Un tableau peut avoir plusieurs dimensions:

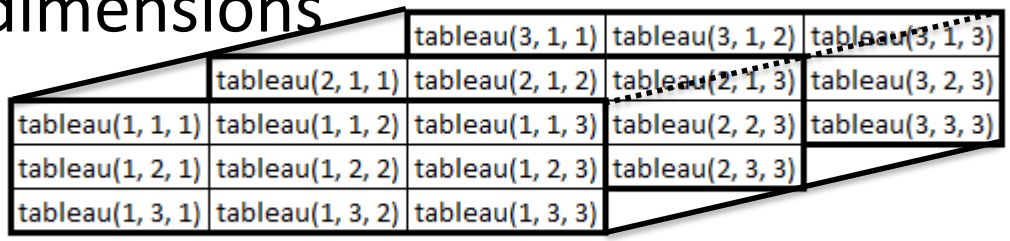
- Une dimension

tableau(1)	tableau(2)	tableau(3)	tableau(4)

- Deux dimensions

tableau(1, 1)	tableau(1, 2)
tableau(2, 1)	tableau(2, 2)
tableau(3, 1)	tableau(3, 2)
tableau(4, 1)	tableau(4, 2)

- Trois dimensions



- Ou plus…

Les tableaux

Pour créer une variable tableau, il est nécessaire de la déclarer.

Un tableau est dit de taille fixe lorsque ses dimensions sont définies lorsque l'on déclare la variable.

```
'Le tableau a une taille fixe. Elle ne peut pas être modifiée.
Dim tableau1(1 To 10) As Variant
```

Un tableau est dit de taille variable lorsque la taille du tableau est définie après sa déclaration

```
'Le tableau a une taille variable. Elle peut être modifiée.
Dim tableau2() As Variant
ReDim tableau2(1 To 10)
ReDim tableau2(1 To 5)
```

Les tableaux

Un tableau peut être défini par les valeurs contenues dans une plage de cellule:

	A	B
1	1	2
2	3	4
3	5	6
4	7	8
5	9	10
6		

```
'Les données sont contenues dans la plage A1:B5.
Dim tableau As Variant
tableau = Range("A1:B5")
```

Expression	Valeur	Type
tableau		Variant/Variant(1 to 5, 1 to 2)
tableau(1)		Variant(1 to 2)
tableau(1,1)	1	Variant/Double
tableau(1,2)	2	Variant/Double
tableau(2)		Variant(1 to 2)
tableau(2,1)	3	Variant/Double
tableau(2,2)	4	Variant/Double
tableau(3)		Variant(1 to 2)
tableau(3,1)	5	Variant/Double
tableau(3,2)	6	Variant/Double
tableau(4)		Variant(1 to 2)
tableau(4,1)	7	Variant/Double
tableau(4,2)	8	Variant/Double
tableau(5)		Variant(1 to 2)
tableau(5,1)	9	Variant/Double
tableau(5,2)	10	Variant/Double

Les tableaux

Fonctions utiles:

```
Dim tableau As Variant
tableau = Range("A1:B5")

'ReDim : ReDimension, permet de redimensionner un tableau
'   tableau de départ:  première dimension = 5
'                       deuxième dimension = 2
'Preserve permet de préserver les données
ReDim Preserve tableau(1 To 5, 1 To 3)
'   tableau d'arrivée:  première dimension = 5
'                       deuxième dimension = 3
'L'outil "ReDim" ne permet d'agir que sur la dernière dimension

'Ubound renvoie la coordonnée max
'Lbound renvoie la coordonnée min
a = UBound(tableau, 1)   'Max dimension 1 (=5)
b = UBound(tableau, 2)   'Max dimension 2 (=3)
c = LBound(tableau, 1)   'Min dimension 1 (=1)
d = LBound(tableau, 2)   'Min dimension 2 (=1)

'Efface le contenu et les dimensions du tableau
Erase tableau
```

Exercice

Créer un tableau à 2 dimensions dont les valeurs vont être insérées dans la plage A1:F10.

- 10 éléments (1 à 10) dans la dimension n°1
- 6 éléments (1 à 6) dans la dimension n°2

Le tableau devra avoir les valeurs suivantes:

Dimension n°2 / Dimension n°1	1	2	3	4	5	6
1	=	= *	= *	= *	= *	= *
2	=	= *	= *	= *	= *	= *
3	=	= *	= *	= *	= *	= *
4	=	= *	= *	= *	= *	= *
5	=	= *	= *	= *	= *	= *
6	=	= *	= *	= *	= *	= *
7	=	= *	= *	= *	= *	= *
8	=	= *	= *	= *	= *	= *
9	=	= *	= *	= *	= *	= *
10	=	= *	= *	= *	= *	= *

Exercice

Vous devez:

- Extraire les informations de la plage A1:B10 sous la forme de tableau
- Créer un tableau de dimension 2 (1ère dimension=10, deuxième dimension=1) variant dont les valeurs seront écrit sous la forme de String comme des formules:

« =R[0]C[-2]*R[0]C[-1] » si la valeur dans la colonne A < 10

« =R[0]C[-2]+R[0]C[-1] » si la valeur dans la colonne A >= 10

- Entrer ces valeurs dans la plage C1:C10.

```
Range("C1:C10").FormulaR1C1 = tableau
```

Subroutine et fonctions

Plus un programme est complexe, plus il est difficile de s'y retrouver.

Pour plus de clarté, il est conseillé d'utiliser des subroutines ou des fonctions

Subroutine

Une subroutine est une partie de code pouvant être appelée par un programme amont. Les données d'entrée sont également des données de sortie.

```
Sub programme_amont()
    a = 2
    b = 3
    'La routine va additionner a et b et retourner le résultat dans c
    addition a, b, c            'c = 5
    'La routine va multiplier a et b et retourner le résultat dans c
    multiplication a, b, c      'c = 6
End Sub
```

```
Sub addition(x1, x2, x3)
    x3 = x1 + x2
End Sub
```

```
Sub multiplication(x1, x2, x3)
    x3 = x1 * x2
End Sub
```

Subroutine

Une subroutine peut être « privée » ou « publique ».

- Lorsqu'elle est privée, elle ne peut pas être appelée.
- Lorsqu'elle est publique, elle peut être appelée par des éléments du fichier.
- Une subroutine ayant des attributs ne peut être appelée que par d'autres subroutines ou fonctions.

Subroutine

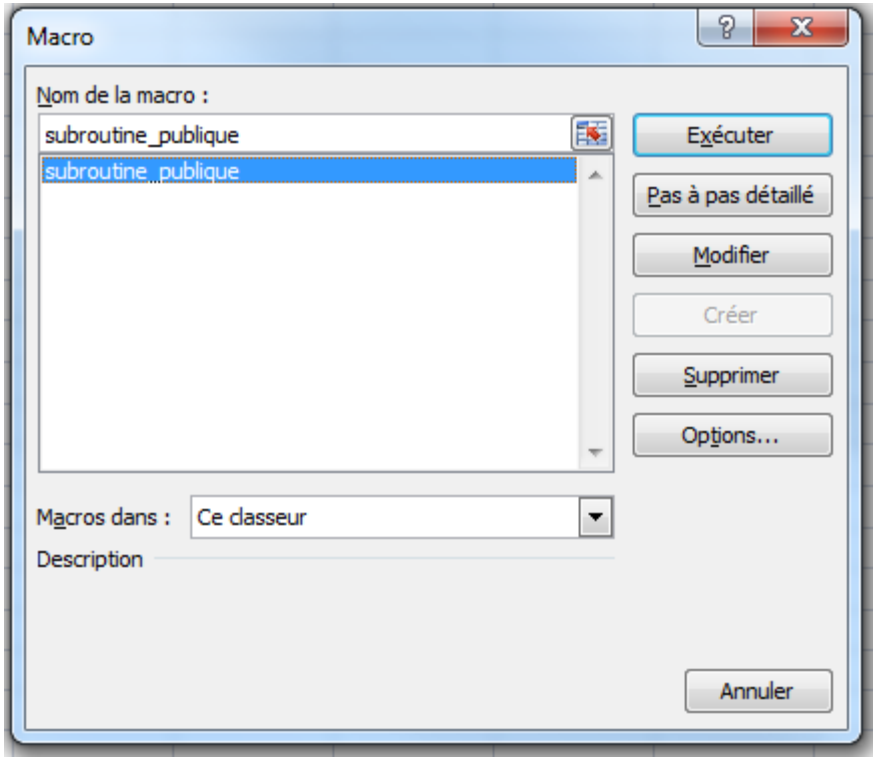
```
Public Sub subroutine_publique()
    a = 1
    b = 2

    'Message "Hello!"
    subroutine_privee

    'c=a+b
    addition a, b, c
End Sub
```

```
Private Sub subroutine_privee()
    MsgBox "Hello!"
End Sub
```

```
Sub addition(x1, x2, x3)
    x3 = x1 + x2
End Sub
```



Fonctions

Les fonctions ont le même rôle que les subroutines mais peuvent ressortir un seul résultat:

```
Sub programme_amont()
    a = 2
    b = 3
    'La fonction va additionner a et b
    c = addition(a, b)          'c = 5
    'La fonction va multiplier a et b
    d = multiplication(a, b)    'd = 6
End Sub
```

```
Function addition(x1, x2)
    addition = x1 + x2
End Function
```

```
Function multiplication(x1, x2)
    multiplication = x1 * x2
End Function
```

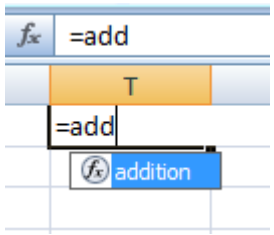
Le fonctionnement des fonctions est le même que le fonctionnement des fonctions que vous utilisez dans les cellules Excel.

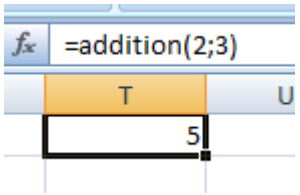
Fonctions

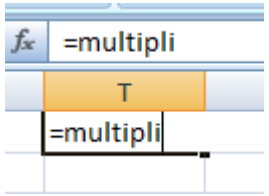
De même que les subroutines, une fonction peut être privée ou publique. Si une fonction est privée, elle ne peut être utilisée que dans les scripts. Si une fonction est publique, elle peut être utilisée dans les cases d'Excel.

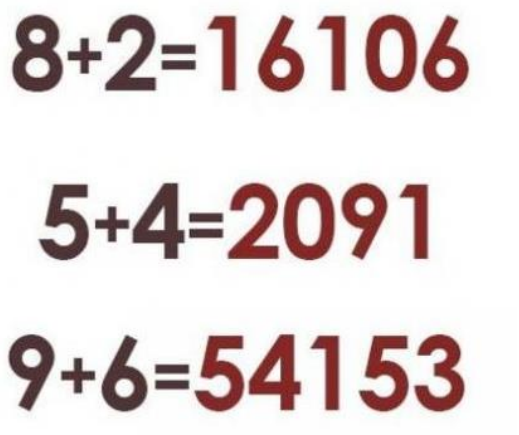
```
Public Function addition(x1, x2)
    addition = x1 + x2
End Function

Private Function multiplication(x1, x2)
    multiplication = x1 * x2
End Function
```









Exercice

Créer un programme qui donne les réponses à ce casse tête avec 3 fonctions ou subroutines.

- Une fonction devant retourner le résultat de la multiplication sous forme de texte
- Une fonction devant retourner le résultat de la soustraction sous forme de texte
- Une fonction devant retourner le résultat de l'addition sous forme de texte

Le programme principale est chargé de compiler le tout.

```
a = 48      'La variable a est un nombre
b = CStr(a) 'La variable b est une chaîne de caractère

'Cstr(x) permet de transformer un nombre en chaîne de caractère
```

Exercice : Triangle de Sierpiński

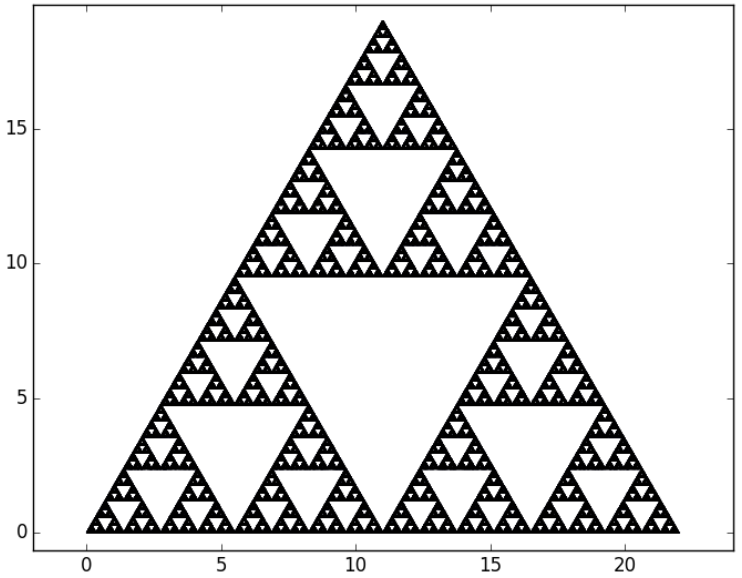
Les règles pour la création de ce triangle sont simples:

- On initialise la position d'un point au hasard (ou choisi)
- On créé 3 point (A, B, C) n'étant pas sur le même ligne.
- Pour trouver la position du point suivant, on prend un nombre aléatoire (entre 1 et 3 inclus):

➢Si n=1, alors le nouveau point se trouve à mi-distance entre le point précédent et le point A

➢Si n=2, alors le nouveau point se trouve à mi-distance entre le point précédent et le point B

➢Si n=3, alors le nouveau point se trouve à mi-distance entre le point précédent et le point C



Exercice : La fougère de Barnsley

Les règles pour la création de cette forme sont simples:

- On initialise la position d'un point au hasard (ex: P_i [0, 0])
- Pour trouver la position du point suivant, on prend un nombre aléatoire n entre 0 et 1 et on suit le système suivant:

$$\begin{cases} \forall n < 0.01 & x_i = 0 & y_i = 0.16 * y_{i-1} \\ \forall 0.01 \leq n < 0.86 & x_i = 0.85 * x_{i-1} + 0.04 y_{i-1} & y_i = 1.6 - 0.04 * x_{i-1} + 0.85 * y_{i-1} \\ \forall 0.86 \leq n < 0.93 & x_i = 0.2 * x_{i-1} - 0.26 * y_{i-1} & y_i = 1.6 + 0.23 * x_{i-1} + 0.22 * y_{i-1} \\ \forall n \geq 0.93 & x_i = -0.15 * x_{i-1} + 0.28 * y_{i-1} & y_i = 0.44 + 0.26 * x_{i-1} + 0.24 * y_{i-1} \end{cases}$$