

# ALGORITHMIQUE ET PROGRAMMATION

## PROGRAMMATION AVEC LE LANGAGE VISUAL BASIC APPLICATION

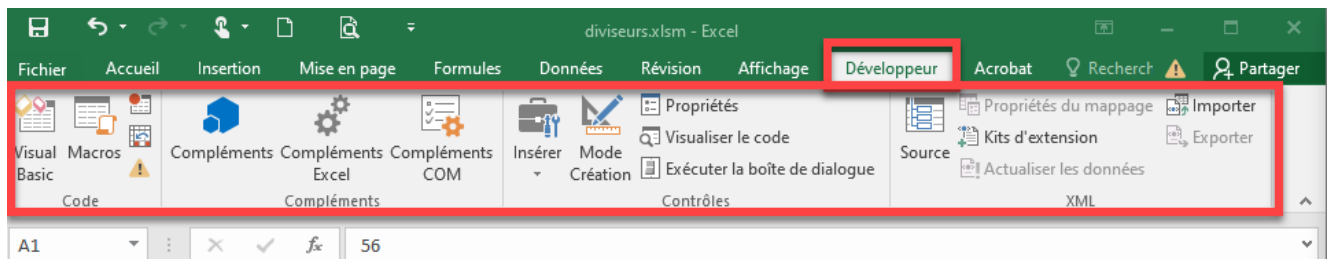
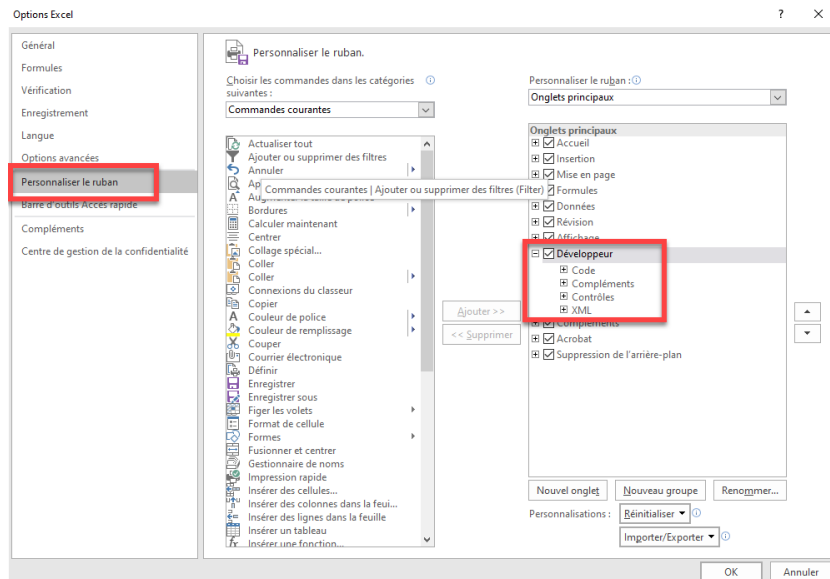
### ACTIVITE : TD1

Le programme suivant a pour but d'afficher la liste des diviseurs d'un nombre situé dans la cellule A1 de la feuille de calcul. Les diviseurs successifs apparaîtront par paires (diviseur et quotient qui est lui-même un diviseur) dans les colonnes B et C.

### PREPARATION DE L'ENVIRONNEMENT DE PROGRAMMATION :

Avant de nous lancer dans la programmation VBA, il faut d'abord activer l'environnement de développement VBA.

Pour le faire, nous allons, d'abord, activer l'onglet Développeur pour faire apparaître le ruban des outils de l'environnement de développement. Dans ces outils, on trouvera notre éditeur pour saisir notre programme, le corriger, le compiler et l'exécuter.



Maintenant que notre espace de travail est prêt nous allons nous lancer dans la découverte de la programmation VBA dans Excel. C'est le même environnement qu'on peut trouver pour Word, PowerPoint par exemple.

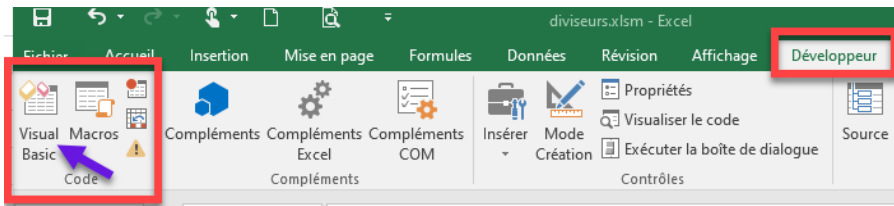
C'est un début qui pourra nous amener à passer facilement à travailler dans d'autres environnements de programmation et langages.

---

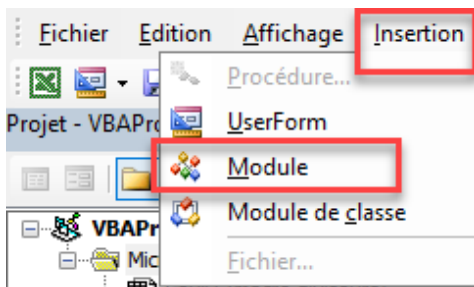
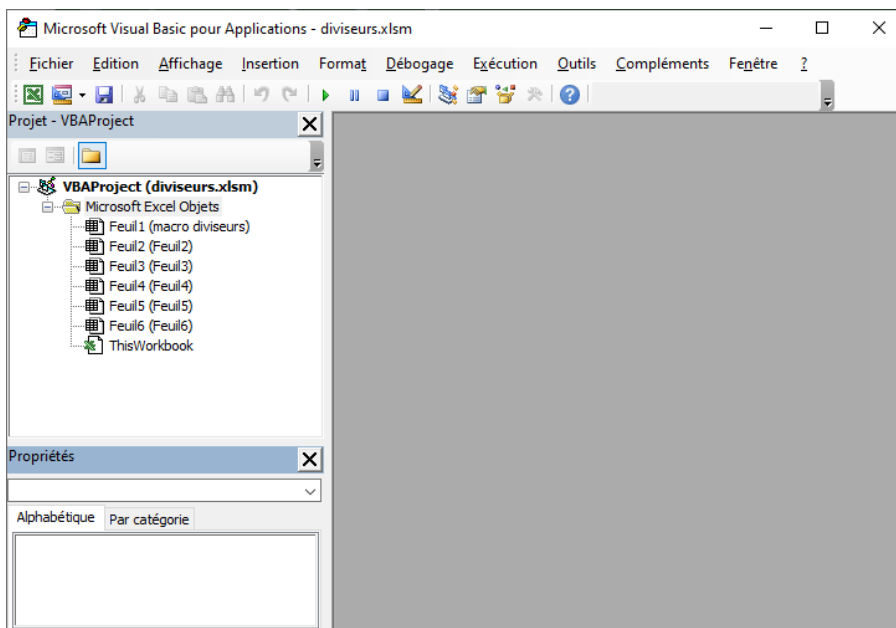
## ECRIRE MON PREMIER PROGRAMME VBA :

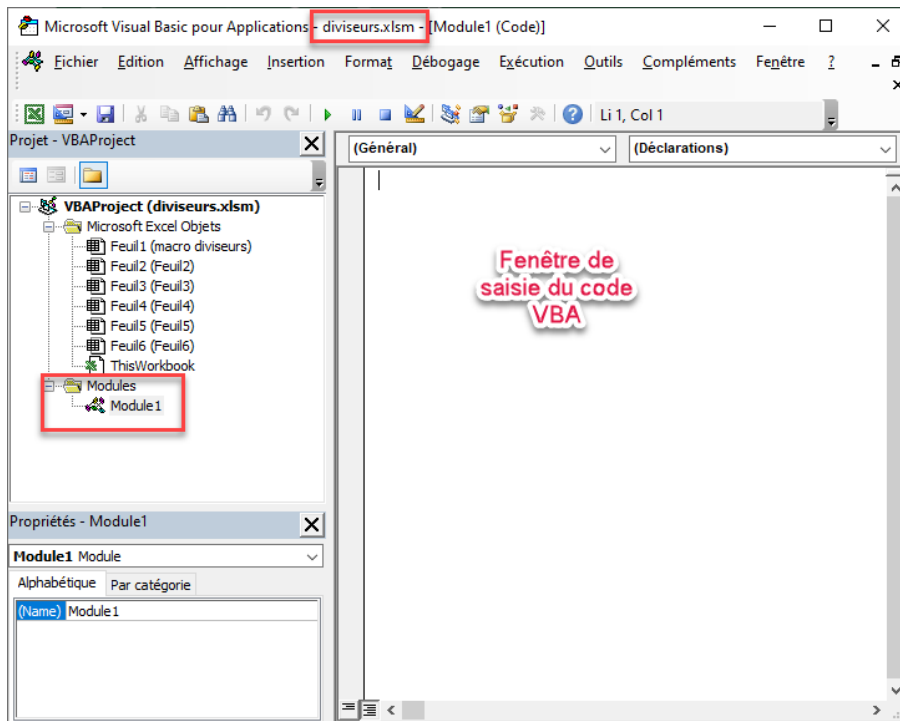
Pour écrire un programme VBA, on doit d'abord ouvrir l'éditeur Visual Basic.

Dans l'onglet Développeur, Groupe Code, cliquer sur le bouton Visual Basic. Vous pouvez aussi, utiliser le raccourci clavier **Alt F11** pour afficher l'éditeur VBA.

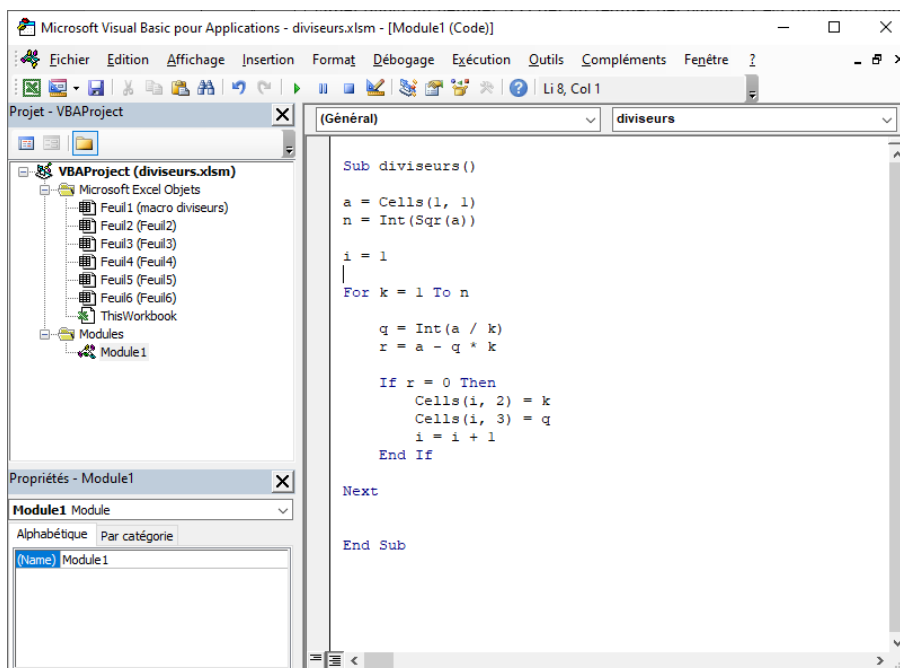


La fenêtre qui s'ouvre offre une nouvelle barre d'outils qui comprend le menu **Insertion**. On choisit alors **Module** :





C'est dans cette fenêtre que l'on saisit le programme suivant :



On retrouve dans ce programme la boucle **For...Next**, l'instruction conditionnelle **If...Then... End If**.

L'exécution du programme avec 56 en cellule A1 (repérée par le couple (1,1)) donne

	A	B	C
1	56	1	56
2		2	28
3		4	14
4		7	8

Mais comme nous l'avons déjà signalé, l'exécution n'est pas ici une fin en soi. On obtiendrait des résultats équivalents avec une calculatrice. Ce qu'Excel offre en plus est de pouvoir suivre le programme, ce qui en fait un véritable outil d'apprentissage de la programmation.

Pour cela il suffit de faire apparaître simultanément la feuille de calcul et l'écran de programmation.

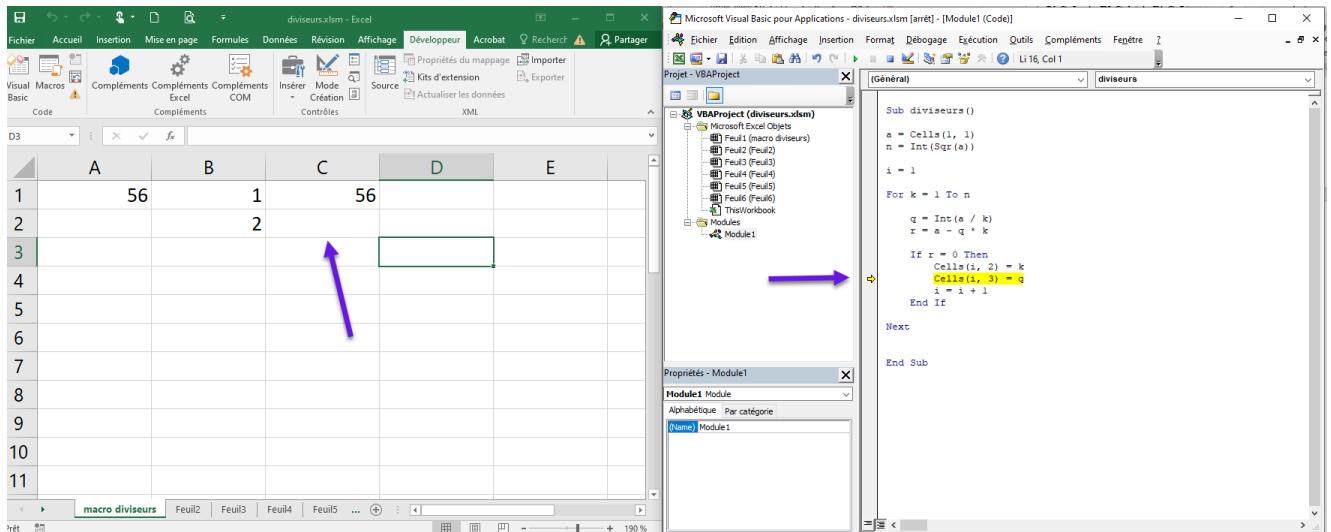
## EXECUTION DU PROGRAMME

The screenshot displays two windows side-by-side. On the left is the Microsoft Excel window titled 'diviseurs.xlsm - Excel'. The 'Développeur' ribbon is active, and the 'Code' window shows the VBA project. The spreadsheet shows cell A1 containing the value 56. On the right is the Microsoft Visual Basic pour Applications window titled 'Microsoft Visual Basic pour Applications - diviseurs.xlsm - [Module1 (Code)]'. The 'Débogage' (Debug) tab is selected, and the code for the 'diviseurs' subroutine is visible. A red annotation 'Alt + F11' is placed over the VBA editor window, with a sub-note 'Pour afficher rapidement l'éditeur VBA'.

L'onglet **Débogage** de la fenêtre de programmation permet d'effectuer le programme pas à pas en appuyant sur la touche F8.

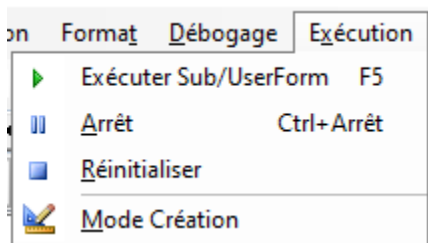
The screenshot shows the 'Débogage' (Debug) menu in the VBA editor. The 'Pas à pas détaillé' option is highlighted with a red box. Other options in the menu include 'Pas à pas principal', 'Pas à pas sortant', 'Exécuter jusqu'au curseur', 'Ajouter un espion...', 'Modifier un espion...', 'Espion express...', 'Basculer le point d'arrêt', 'Effacer tous les points d'arrêt', 'Définir l'instruction suivante', and 'Afficher l'instruction suivante'. The background shows the VBA code for the 'diviseurs' subroutine.

On obtiendra par exemple en cours d'exécution :



En dehors de l'instruction spécifique **cells**, qui permet de placer dans la cellule repérée par sa ligne et sa colonne un contenu choisi, les autres instructions de boucle ou conditionnelles sont communes à la plupart des langages de programmation implantés classiquement sur calculatrice ou dans des logiciels de calcul formel comme Maple. La syntaxe de telle ou telle instruction peut bien entendu différer d'un langage à l'autre mais il s'agit globalement des mêmes objets, utilisés d'une façon identique.

On peut remarquer que, bien que Visual Basic soit un langage objet, On peut s'en servir comme s'il s'agissait d'un langage procédural classique.



## CODES SOURCES :

```
Sub diviseurs()
```

```
a = Cells(1, 1)
```

```
n = Int(Sqr(a))
```

```
i = 1
```

```
For k = 1 To n
```

```
  q = Int(a / k)
```

```
  r = a - q * k
```

```
  If r = 0 Then
```

```
    Cells(i, 2) = k
```

```
    Cells(i, 3) = q
```

```
    i = i + 1
```

```
  End If
```

```
Next
```

```
End Sub
```