

INITIATION AUX ALGORITHMES – LES BOUCLES LE LANGAGE VISUAL BASIC APPLICATIONS D'EXCEL (VBA EXCEL)

Les boucles sont les structures les plus puissantes de la programmation. On dit aussi STRUCTURES REPETITIVES. On utilise les instructions de boucle pour que le programme exécute un bloc d'instructions plusieurs fois.

1. Copier le document Projets_Eleves qui se trouve dans le dossier Ressources_BTS_2\Processus_7 dans votre dossier
2. Ouvrir le document
3. Cliquer sur le bouton « Activer les macros » si besoin

I – La structure WHILE WEND

WHILE Condition	TANT QUE	Tant que la condition est VRAI, les instructions vont être exécutées.
Instructions		
WEND	FIN TANT QUE	

Exemple 1 : Remplir une plage de cellules

1. Ouvrir la feuille Test_1

L'objectif est de remplir la colonne Echéance TANT QUE l'échéance est inférieure à la durée.

L'algorithme a déjà été créé. Comment l'exécuter ?

2. Onglet Développeur
3. Clic sur l'outil Macros
4. Sélectionner la macro « Completer_Echéance »
5. Clic sur Exécuter

La colonne Echéance est complétée.

6. Saisir 8 en B3
7. Exécuter la macro

Visualisation du code

1. Clic sur Visual Basic
2. Clic sur le module « Test_1 »

```

Sub Completer_Echéance()
    Dim Durée As Integer
    Dim i As Integer

    'Récupère le nombre qui est dans la cellule B3 de la feuille Test_1
    Durée = Worksheets("Test_1").Range("B3")

    i = 1

    While i <= Durée

        'Place dans les cellules de la feuille Test_1 la valeur de i
        'La colonne Echéance est placée dans la cellule A6
        'La première échéance qui est égale à 1 sera placée en A7

        Worksheets("Test_1").Range("A" & i + 6) = i

        i = i + 1

    Wend

    MsgBox "Fin "

End Sub

```

Déclaration de variables
La variable i va représenter l'échéance

Initialisation. On donne à i la valeur de 1 car l'échéance va démarrer à 1

Condition d'entrée dans la boucle. Tant que l'échéance est inférieure à la durée

Ce bloc d'instructions sera exécuté tant que la condition est VRAI

La variable i augmente de 1 à chaque fois que la boucle sera exécutée.
C'est une instruction d'affectation

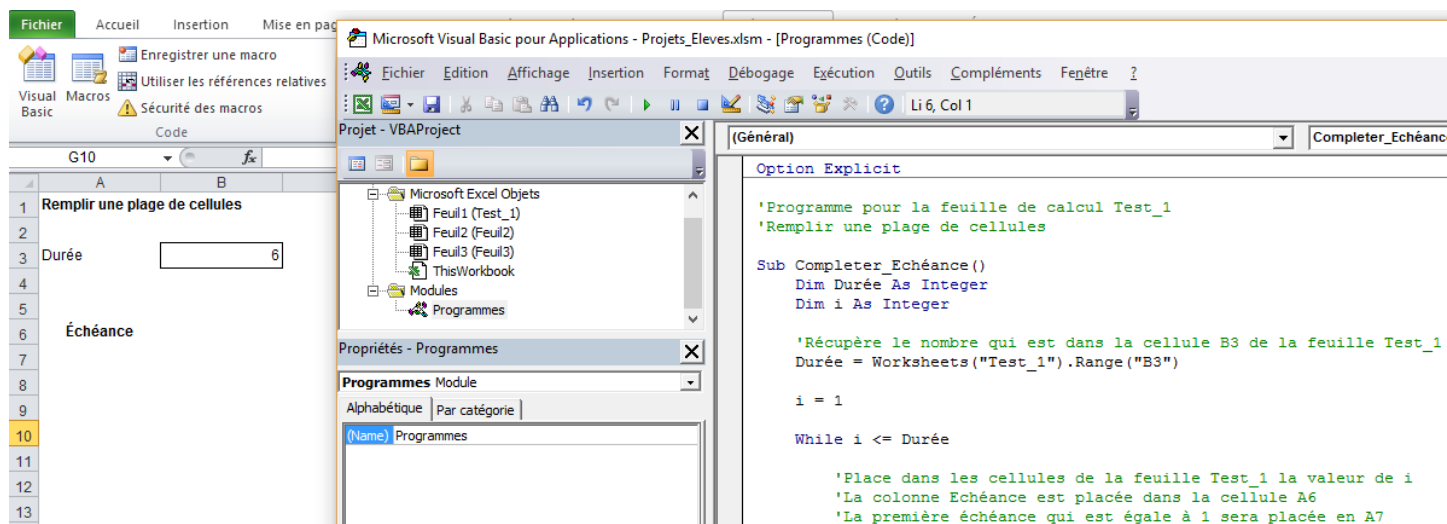
La condition **While i<=Durée** est réévaluée

Instruction exécutée quand on sort de la boucle

L'instruction $i=i+1$ est très importante car elle permet de sortir de la boucle. Si elle est absente, on aurait une boucle infinie (qui va se répéter indéfiniment, le programme ne s'arrêtera jamais). En cas de boucle infinie, il faut appuyer sur les touches CTRL+PAUSE pour arrêter le programme.

Exécution du code Pas à pas

1. Rétrécir la fenêtre Visual Basic de façon à voir en même temps la feuille Excel et le code



2. Effacer toutes les échéances
3. Saisir 5 en B3
4. Clic sur l'outil Macros
5. Sélectionner la macro « Compléter_Echéance »
6. Clic sur **Pas à pas détaillé**
7. Appuyer sur la touche F8 pour exécuter pas à pas chaque ligne de code et observer bien ce qui se passe.

Amélioration du code

Le programme précédent fonctionne mais il a un défaut

Démonstration :

1. Saisir 10 en B3
2. Clic sur l'outil Macros
3. Sélectionner la macro « Compléter_Echéance »
4. Clic sur Exécuter
5. Saisir 5 en B3
6. Clic sur l'outil Macros
7. Sélectionner la macro « Compléter_Echéance »
8. Clic sur Exécuter
9. Quel est l'inconvénient ? _____
10. Il faut donc d'abord su_____ les données existantes avant de compléter les échéances
11. Afficher le code Compléter_Echéance_2
12. Entourer les lignes de codes qui permettent d'effacer les données existantes

```
Sub Compléter_Echéance_2()
    Dim Durée As Integer
    Dim i As Integer

    'Récupère le nombre qui est dans la cellule B3 de la feuille Test_1
    Durée = Worksheets("Test_1").Range("B3")

    i = 1

    'Efface les données existantes
    Worksheets("Test_1").Range("A7").Select
    ActiveCell.CurrentRegion.Select
    Selection.ClearContents

    'Sélectionne la cellule A7
    'Sélectionne toutes les cellules non vides autour de la cellule A7
    'Efface le contenu de la sélection

    While i <= Durée

        'Place dans les cellules de la feuille Test_1 la valeur de i
        'La colonne Échéance est placée dans la cellule A6
        'La première échéance qui est égale à 1 sera placée en A7

        Worksheets("Test_1").Range("A" & i + 6) = i

        i = i + 1
    End While

    MsgBox "Fin "
End Sub
```

13. Tester la procédure « Compléter_Echéance_2

Exemple 2 : Table de multiplication

1. Ouvrir la feuille de calcul « Test_2 »

	A	B	C	D	E	F	G	H	I
1	Table de multiplication			Fin	100		Multiplicateur	5	
2									
3									
4	Nombre	Résultat							
5									
6									
7									
8									
9									

L'objectif est de remplir 2 colonnes :

- La colonne Nombre en commençant par 1 jusqu'au nombre situé dans la cellule E1
- La colonne Résultat qui sera la multiplication de Nombre par le multiplicateur (qui se trouve dans la cellule H5)

2. Afficher la procédure « Table Multiplication » qui se trouve dans le module « Test_2 » ; Il manque 2 lignes à compléter dans le code

```
Sub Table_Multiplication()
    Dim Fin As Integer
    Dim Multiplicateur As Integer
    Dim i As Integer

    'Récupère le nombre qui est dans la cellule E1 de la feuille Test_2
    Fin = Worksheets("Test_2").Range("E1")

    'Récupère le nombre qui est dans la cellule H1 de la feuille Test_2

    i = 1

    'Efface les données existantes
    ActiveSheet.Range("A5", ActiveSheet.Range("A5").End(xlDown).End(xlToRight)).Select
    Selection.ClearContents

    While i <= Fin

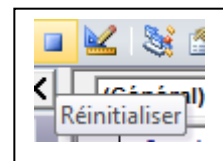
        'Place dans les cellules de la feuille Test_2 la valeur de i
        'La colonne Nombre est placée dans la cellule A4
        'La première valeur qui est égale à 1 sera placée en A5
        Worksheets("Test_2").Range("A" & i + 4) = i
        'Remplit la colonne Résultat

        i = i + 1

    Wend

    MsgBox "Fin"
    Worksheets("Test_2").Range("H1").Select
End Sub
```

3. Compléter les rectangles
4. Exécuter le programme
5. Que constatez-vous ? La colonne _____ n'est _____ ; Donc il ma _____ des ins _____
6. Compléter les 2 lignes manquantes
7. Enregistrer les modifications
8. Exécuter le programme
 - en cas d'erreur de code, cliquer sur débogage, l'instruction sera surlignée en jaune
 - Corriger l'erreur
 - Cliquer ensuite sur l'outil « Réinitialiser »



Créer un bouton de commande pour exécuter un programme

1. Revoir la vidéo « Créer un bouton de commande »
2. Créer un bouton de commande pour le programme « Table Multiplication »
3. Tester

Exemple 3 : Calcul de ristournes

1. Ouvrir la feuille de calcul « Test_3 »

	A	B	C	D	E
1					
2		CALCUL DE LA RISTOURNE ANNUELLE CLIENTS 2015			
3					
4		Nom du client	CA NET HT	RISTOURNE	
5		BARON Mélanie	4 200,00		
6		BAVENT Jean	25 000,00		
7		GANON Tancrede	40 458,33		
8		LEFER Soidic	56 600,00		
9		MARTIAL Aline	83 221,67		
10		MICHAUD Luc	52 743,85		

Règles de calcul de la ristourne

CA Net HT	Taux Ristourne
[80 000 € et +	8%
[50 000 € - 80 000 € [6%
[30 000 € - 50 000 € [4%
[5 000 € - 30 000 € [2%
[0 € - 5000 € [1%

CA Net HT		Taux Ristourne	Montant ristourne	Cumul Ristourne
De	A			
0 €	5 000,00 €	1%	(5000-0)*0,01=50	50,00 €
5 000,00 €	30 000,00 €	2%	(30000-5000)*0,02=500	550,00 €
30 000,00 €	50 000,00 €	4%	(50000-30000)*0,04=800	1 350,00 €
50 000,00 €	80 000,00 €	6%	(80000-50000)*0,06=1800	3 150,00 €
80 000,00 €	et +	8%	(CA Net HT - 80000)*8%	3150+(CA Net HT - 80000)*8%

L'objectif est de compléter par VBA le calcul de la ristourne mais on ne veut pas que les autres utilisateurs modifier la formule.

2. Ouvrir le module « Test_3 »

```
'Fonction qui calcule la ristourne
Function Ristourne(CANet As Single) As Single

    Select Case CANet
        Case Is >= 80000
            Ristourne = 3150 + (CANet - 80000) * 0.08
        Case Is >= 50000
            Ristourne = 1350 + (CANet - 50000) * 0.06
        Case Is >= 30000
            Ristourne = 550 + (CANet - 30000) * 0.04
        Case Is >= 5000
            Ristourne = 50 + (CANet - 5000) * 0.02
        Case Else
            Ristourne = CANet * 0.01
    End Select
End Function

'Programme qui calcule et complète la colonne Ristourne pour chaque client
Sub Calcule_Ristourne()
    Dim i As Integer

    i = 0
    'Le 1er client est en B5
    Worksheets("Test_3").Range("B5").Select

    While ActiveCell <> ""
        'Tant que la cellule active n'est pas vide
        'Place dans la colonne D Ristourne le montant de la ristourne
        'Pour calculer la ristourne, on fait appel à la fonction Ristourne
        'La fonction fait le calcul à partir du CANet qui se trouve dans la colonne C

        Worksheets("Test_3").Range("D" & i + 5) = Ristourne(Worksheets("Test_3").Range("C" & i + 5))

        i = i + 1
        'i augmente de 1, permet de passer à la cellule suivante

        Worksheets("Test_3").Range("B" & i + 5).Select 'Selectionne la cellule suivante
    Wend
End Sub
```

Il y a une fonction Ristourne qui permet de calculer la ristourne. Cette fonction est ensuite utilisée dans la procédure « Calcule_Ristourne »

3. Rétrécir la fenêtre Visual Basic de façon à voir en même temps la feuille Excel et le code
4. Exécuter la macro « Calcule_Ristourne » en mode Pas à Pas détaillé (Voir page 2) pour bien comprendre ce que fait le code

Exemple 4: Calcul du salaire saisi

Barème						A	B	C
Rémunération mensuelle nette		Quotité saisissable	Montant saisi	Cumul Saisi		Salarié	Salaire Net	Salaire Saisi
De	A				1			
0 €	310,83 €	1/20	15,54 €	15,54 €	2	Alavi Martine	300,00 €	15,00 €
310,83 €	606,67 €	1/10	29,58 €	45,12 €	3	Becaude Serge	600,00 €	44,46 €
606,67 €	904,17 €	1/5	59,50 €	104,62 €	4	Dupont Marc	900,00 €	103,79 €
904,17 €	1 200,83 €	1/4	74,17 €	178,79 €	5	Flavier Micheline	1 300,00 €	211,85 €
1 200,83 €	1 497,50 €	1/3	98,89 €	277,68 €	6	Gruder Marcel	1 400,00 €	245,18 €
1 497,50 €	1 799,17 €	2/3	201,11 €	478,79 €	7	Leroy Paul	2 000,00 €	679,62 €
1 799,17 €	Totalité		478,79 €	totalité au delà de 1 799,17 €			

'Fonction qui calcule le salaire saisissable

Function SalaireSaisi(SalNet As Single) As Single

Select Case SalNet

Case Is >= 1799.17

SalaireSaisi = 478.79 + (SalNet - 1799.17)

Case Is >= 1497.5

SalaireSaisi = 277.68 + (SalNet - 1497.5) * 2 / 3

Case Is >= 1200.83

SalaireSaisi = 178.79 + (SalNet - 1200.83) * 1 / 3

Case Is >= 904.17

SalaireSaisi = 104.62 + (SalNet - 904.17) * 1 / 4

Case Is > 606.67

SalaireSaisi = 45.12 + (SalNet - 606.67) * 1 / 5

Case Is > 310.83

SalaireSaisi = 15.54 + (SalNet - 310.83) * 1 / 10

Case Else

'SalNet >=0

SalaireSaisi = SalNet * 1 / 20

End Select

End Function

'Programme qui calcule et complète la colonne Salaire Saisi pour chaque salarié

Sub Calcule_Salaire_Saisi()

Dim i As Integer

i = 0

'Le 1er salarié est en A2

Worksheets("Test_4").Range("A2").Select

While ActiveCell <> ""

'Tant que la cellule active n'est pas vide

'Place dans la colonne C (Salaire saisi) le montant du salaire saisi

'Pour calculer le salaire saisi, on fait appel à la fonction SalaireSaisi

'La fonction fait le calcul à partir du Salaire net qui se trouve dans la colonne B

i = i + 1 'i augmente de 1, permet de passer à la cellule suivante

Wend

End Sub

1. Compléter les 2 lignes de code manquantes
2. Enregistrer
3. Exécuter la macro « Calcule_Salaire_Saisi » ; Contrôler les résultats ; Corriger en cas d'erreur
4. Expliquer le rôle de l'instruction `i=i+1` : Cela signifie que i au _____ de _____ après chaque bo_____. Elle est très im_____ car elle permet de sortir de la _____. Le programme va s'exécuter tant qu'il y aura des sa_____

Exemple 5: Tableau amortissement d'un emprunt

1. Voir feuille « Test_5 »
2. Exécuter le programme en changeant la durée
3. Constat : _____
4. Compléter les 2 lignes manquantes
5. Enregistrer
6. Exécuter

8. Encadrer toutes les instructions de la boucle

End Sub

[illegible]

Exemple 6: Calcul de la prime d'ancienneté

1. Ouvrir le module « Test_6 »

On désire calculer la prime d'ancienneté tant qu'il y aura des salariés.

Nom Salarié	Ancienneté	Salaire
Fred	5	2500
Marc	12	1500
Bon	2	1450

Ancienneté	Prime
>10	10% du salaire
Entre 5 et 10	5% du salaire
Sinon	2% du salaire

```
Sub Calcul_Prime_Anciennete()  
    Dim NomSalarie As String  
    Dim Anciennete As Single  
    Dim Salaire As Single  
    Dim PrimeAnciennete As Single  
    Dim Reponse As String  
  
    Reponse = "O"  
  
    While Reponse = "O"  
        NomSalarie = InputBox("Entrer le nom du salarié")  
        Anciennete = InputBox("Entrer l'ancienneté")  
        Salaire = InputBox("Entrer le salaire")  
  
        If Anciennete > ____ Then  
            PrimeAnciennete = Salaire * ____  
        Else  
            If Anciennete >= ____ Then  
                PrimeAnciennete = Salaire * ____  
            Else  
                PrimeAnciennete = Salaire * ____  
            End If  
        End If  
  
        MsgBox "La prime d'ancienneté du salarié : " & NomSalarie & " est de : " & PrimeAnciennete  
  
        Reponse = InputBox("Y a t-il un autre salarié ? ")  
    Wend  
  
    MsgBox "Terminé"  
  
End Sub
```

1. Compléter le code
2. Quelle est la condition qui permet d'entrer dans la boucle ? _____
3. Encadrer le bloc d'instructions de la boucle
4. Quel est le rôle de la variable Reponse ? _Tant que Reponse sera égale à _____, les ins_____ de la _____ vont être exécutées
5. Comment va-t-on sortir de la boucle ? _____
6. Cette instruction est-elle importante ? Reponse = InputBox("Y a t-il un autre salarié ? ")
Pourquoi ? Parce qu'elle permet de saisir _____ et donc de so_____ de la b_____.
7. Quelle est l'instruction qui s'affiche après la sortie de la boucle ? _____
8. Exécuter le programme pas à pas avec les données du tableau

Exemple 7: Calcul de la prime d'ancienneté

On désire calculer la prime d'ancienneté tant qu'il y aura des salariés. On aimerait que le programme affiche à la fin :

- Le nombre de salariés saisi
- Le total de la prime d'ancienneté pour tous les salariés

NbSalarie	0	TotalPrimeAnciennete	0		
Nom Salarié	Ancienneté	Salaire	Prime anciennete	NbSalarie	TotalPrimeAnciennete
Fred	5	2500	125		
Marc	12	1500	150		
Bon	2	1450	29		

```
Sub Calcul_Prime_Anciennete_2()  
    Dim NomSalarie As String  
    Dim Anciennete As Single  
    Dim Salaire As Single  
    Dim PrimeAnciennete As Single  
    Dim Reponse As String  
    Dim i As Integer  
    Dim TotalPrimeAnciennete As Single  
    'Compter le nombre de salariés  
    'Cumuler la prime d'ancienneté  
  
    Reponse = "O"  
    i = 0  
    TotalPrimeAnciennete = 0  
    'Initialisation des données  
  
    While Reponse = "O"  
        i = i + 1  
        'Le nombre de salariés augmente de 1  
        NomSalarie = InputBox("Entrer le nom du salarié")  
        Anciennete = InputBox("Entrer l'ancienneté")  
        Salaire = InputBox("Entrer le salaire")  
  
        If Anciennete > 10 Then  
            PrimeAnciennete = Salaire * 0.1  
        Else  
            If Anciennete >= 5 Then  
                PrimeAnciennete = Salaire * 0.05  
            Else  
                PrimeAnciennete = Salaire * 0.02  
            End If  
        End If  
  
        TotalPrimeAnciennete = TotalPrimeAnciennete + _____  
        MsgBox "La prime d'ancienneté du salarié : " & NomSalarie & " est de : " & PrimeAnciennete  
  
        Reponse = InputBox("Y a t-il un autre salarié ? ")  
    Wend  
  
    MsgBox "Le nombre de salariés saisi : " & _____  
    MsgBox "Le total de la prime d'ancienneté est de : " & _____  
    MsgBox "Terminé"  
  
End Sub
```

1. Compléter le code
2. Quelle est la condition qui permet d'entrer dans la boucle ? _____
3. Quel est le rôle de la variable Reponse ? _Tant que Reponse sera égale à _____, les ins_____ de la _____ vont être exécutées
4. Cette instruction est-elle importante ? `Reponse = InputBox("Y a t-il un autre salarié ? ")`
Pourquoi ? Parce qu'elle permet de saisir _____ et donc de so_____ de la b_____.
5. Quel est le rôle de cette instruction ? `TotalPrimeAnciennete = TotalPrimeAnciennete + PrimeAnciennete`
Elle permet de cum_____ la prime pour _____ les s_____.
6. Exécuter le programme pas à pas avec les données du tableau