

VBA – Niveau 1 - Épisode 0

Exercice 1

Écrire un script VBA (Sub sommecarresimpairs()) qui demande à l'utilisateur de rentrer un nombre entier compris entre 1 et 100 puis calcule la somme des carrés des nombres impairs compris entre 1 et ce nombre.

On pourra s'assurer que l'utilisateur rentre bien un nombre dans l'intervalle demandé.

Exercice 2

Écrire un script VBA (Sub devinelancer()) qui va vous opposer à votre programme. L'ordinateur simule le lancer d'un dé à 6 faces ($x = \text{Int}(6 * \text{Rnd}() + 1)$). Vous devez deviner ce nombre en au plus 3 essais. Si vous devinez le nombre cherché le programme vous félicite sinon le programme vous propose de recommencer jusqu'à atteindre les 3 essais maximum autorisés.

Pour obtenir au hasard un entier x compris dans la plage $[a,b]$, on peut utiliser la formule $x = \text{Int}(((b-a)+1) * \text{Rnd}() + a)$.

Exercice 3

Écrire un programme (Sub calculesomme()) qui va vous opposer à votre programme. L'ordinateur vous demande de calculer la somme de deux nombres entiers aléatoires compris entre 1 et 100. Vous devez calculer cette somme de tête en au plus 5 essais. Si vous trouvez le nombre cherché le programme vous félicite sinon le programme vous propose de recommencer jusqu'à atteindre les 5 essais maximum autorisés.

Exercice 4

Écrire un programme (Sub palindrome()) qui indique si un mot donné par l'utilisateur est un palindrome. Un mot est un palindrome si l'ordre de ses lettres reste le même quand on le lit de gauche à droite ou inversement.

Exemple : radar, kayak, Laval, ressasser.

En VBA, la fonction **StrReverse**(uneChaîne) permet d'inverser l'ordre des caractères dans une chaîne passée en argument. Elle renvoie donc la chaîne résultant de cette opération.

La comparaison de chaînes de caractères est par défaut sensible à la casse, car la représentation binaire de chaque caractère est considérée. Par exemple, **Laval** n'est pas un palindrome dans ce contexte. Pour rendre la comparaison insensible à la casse, utiliser la directive « **Option Compare Text** » au début du module de cet exercice ou alors il faut utiliser **LCase**(uneChaîne) pour rendre en minuscule la chaîne de caractères...

VBA – Niveau 1 - Épisode 0

Exercice 5

On suppose un mot et une lettre saisis par l'utilisateur. Écrire le programme (Sub devineunelettre()) qui vérifie si la lettre donnée est une lettre du mot. Le programme s'arrête dès qu'on a vérifié que la lettre est une lettre du mot.

N.B Pour connaître la longueur du mot on utilisera `len("mot")` et pour obtenir la *i*ème lettre `mid("mot",i,1)`.

Exercice 6

Ce programme (Sub motdepasse()) va opposer deux joueurs. Le premier joueur choisi un mot de passe et le second essaie de le deviner.

On pourra limiter le nombre d'essais et déclarer le Joueur 1 vainqueur si le mot de passe n'est pas découvert à l'issue de ces essais autorisés.

Exercice 7

Écrire un programme (Sub nombrevoyelles()) qui renvoie le nombre de voyelles contenues dans une chaîne de caractères entrées par l'utilisateur.

Exercice 8

Écrire un programme (Sub pascal()) qui demande à l'utilisateur de saisir un entier naturel *n* puis affiche sur la Feuil1 du classeur Excel les *n* premières lignes du triangle de Pascal.

Lexique des variables

<i>n</i>	(entier)	le nombre de lignes à afficher	DONNÉE
<i>i</i>	(entier)	numéro de ligne courant	INTERMÉDIAIRE
<i>j</i>	(entier)	numéro de colonne courant	INTERMÉDIAIRE
<i>x</i>	(entier)	le nombre courant dans le triangle	INTERMÉDIAIRE

Algorithme

```
pour i de 0 à n - 1 faire  
    x ← 1  
    écrire x  
    pour j de 1 à i faire  
        x ← (x × (i - j + 1)) / j  
        écrire " ", x  
    fpour  
    écrire "\n"  
fpour
```

VBA – Niveau 1 - Épisode I

Prix TTC

Signature de la fonction :

Public Function MonTTC(pht As Double) As Double

Entrée : pht (réel)

Sortie : prix TTC (réel)

Calcul :

$$\text{Prix ttc} = \text{pht} * 1.2$$

Capital final – Intérêts composés

Signature de la fonction :

Public Function MonKF(KI As Double, i As Double, duree As Long) As Double

Entrée : capital initial (réel)

Intérêt (réel)

Durée (réel)

Sortie : capital final

Calcul :

$$\text{Capital final} = \text{capital initial} * (1 + \text{intérêt})^{\text{durée}}$$

Prix TTC avec condition

Signature de la fonction :

Public Function MonTTCBis(pht As Double, cat As String) As Double

Entrée : prix ht (réel)

Catégorie (chaîne de caractères)

Sortie : prix ttc (réel)

Calcul :

Si (catégorie = « luxe »)

$$\text{Alors pttc} = \text{pht} * 1.33$$

$$\text{Sinon pttc} = \text{pht} * 1.2$$

VBA – Niveau 1 - Épisode I

Location de véhicules - 1

Signature de la fonction :

Public Function MaLocation1(jour As Long, km As Double, cat As String) As Double

Entrée : Km (réel)
Jour (réel)
Catégorie (chaîne)

Sortie : Prix de la location (réel)

Calcul :

Si (Jour > 30)

Alors prix = 75 * Jour

Sinon

Si (Catégorie = « luxe »)

Alors prix = 80 * Jour + 0.2 * Km

Sinon prix = 70 * Jour + 0.15 * Km

Location de véhicules – 2

Signature de la fonction :

Public Function MaLocation2(jour As Long, km As Double, carburant As String) As Double

Entrée : Km (réel)
Jour (entier)
Carburant (chaîne)

Sortie : Prix de la location (réel)

Calcul :

• Si (Jour > 30)

Alors Prix = 80 * Jour + 0.2 * Km

Sinon Prix = 75 * Jour + 0.1 * Km

• Si (carburant = « diesel »)

Alors Le Prix calculé ci-dessus doit être divisé par 2

VBA – Niveau 1 - Épisode I

Calcul de salaire de commerciaux

Signature de la fonction :

Public Function MonSalaire(ca As Double, statut As String) As Double

Entrée : CA (réel)
Statut (chaîne)

Sortie : Salaire (réel)

Calcul :

- Salaire = $0.75 * CA$
Dans 2 situations :
 1. Statut = « junior »
 2. Statut = « senior » et ($CA > 100$)
- Dans toutes les autres situations.
Salaire = $0.25 * CA$

VBA – Niveau 1 - Épisode II

1. Fonctions SI et SELON

A l'aide d'une part, le branchement conditionnel **IF**, d'autre part, le branchement multiple **SELECT CASE**, écrire deux versions de la fonction personnalisée permettant de réaliser le calcul suivant :

Entrée : prix (réel)
Seuil (réel)

Sortie : demande (réel)

Calcul : $DA = -0.3 \times \text{prix} + 200$
Si $DA \leq 0$ Alors demande = 0
Si $DA \geq \text{seuil}$ Alors demande = seuil
Dans les autres cas, demande = DA

2. Simulation et gestionnaire de scénarios

Construire la feuille de calcul suivante :

Prix (P)	60	P
Quantité (Q)	20	Q
Couts Fixes	50	CF
Cout Unitaire	2	CU
Cout Total	90	$CT=CU*Q+CF$
Recette	1200	$R=P*Q$
Bénéfice	1110	$B=R-CT$

- a. A l'aide de la table de simulation à une entrée, affichez les valeurs du bénéfice pour les prix allant de 50 à 100 avec un pas de 10. Nous devrions obtenir une table ressemblant à ceci :

prix	1110
50	910
60	1110
70	1310
80	1510
90	1710
100	1910

- b. A l'aide de la table de simulation à deux entrées, affichez les valeurs du bénéfice pour le croisement des prix allant de 50 à 100 avec un pas de 10, et de quantité allant de 5 à 15 avec un pas de 5. Soit :

VBA – Niveau 1 - Épisode II

		quantité		
	1110	5	10	15
prix	50	190	430	670
	60	240	530	820
	70	290	630	970
	80	340	730	1120
	90	390	830	1270
	100	440	930	1420

- c. A l'aide du gestionnaire de scénarios, définissez deux variantes portant sur le prix, la quantité et le coût unitaire. La synthèse devra inclure la recette et le bénéfice. Voici un exemple de sortie.

Synthèse de scénarios				
		Valeurs actuelles :		
			s1	s2
Cellules variables :				
Prix		60	100	60
Quantité		20	50	20
Coût_Unitaire		2	1	20
Cellules résultantes :				
Recette		1200	5000	1200
Bénéfice		1110	4900	750

La colonne Valeurs actuelles affiche les valeurs des cellules variables au moment de la création du rapport de synthèse. Les cellules variables de chaque scénario se situent dans les colonnes grisées.

3. Ma petite entreprise

M. EUSEBIO veut créer une entreprise de production de voiturettes sans permis, il vous confie l'étude de faisabilité. Les données du problème sont les suivantes :

- Prix d'une voiturette [P] : 100 K€
- Quantité annuelle à produire [Q] : 130
- Coûts fixes [CF] : il recouvre les frais de location, stockage, électricité,... il n'a de fixe que le nom, il s'agit en réalité d'une fonction à seuil qui dépend de la quantité produite

Quantité (entier)	Coût fixe (K€ - entier)
< 50	1 000
>=50 et <100	2 000
>=100 et <150	8 000
>= 150	40 000

- Coût unitaire (par unité produite) [CU] : 25 K€
- Coût total de la production [CT] : CF+CU*Q

VBA – Niveau 1 - Épisode II

- Chiffre d'affaire [CA] : $P \cdot Q$
- Bénéfice brut [BB] : $CA - CT$
- Zone de production : défavorisée
- Destination de la production : locale
- Taux de Taxe [Taux] : il dépend à la fois de la **quantité** produite (**valeur entière**), de la **destination** de la production (« exportation » ou « locale ») et de la **zone** dans laquelle sera édifée l'usine (« défavorisée », « industrielle » ou « agglomération »). La détermination se fait comme suit

Quantité	Taux
< 50	0.20
>= 50 et < 80	0.35
>= 80 et < 100	0.75

Au-delà de ces quantités, la situation diffère selon que l'on exporte ou non la production :

Destination « locale »		Destination « exportation »	
Quantité	Taux	Quantité	Taux
>= 100	0.95	>=100 et <150	0.80
		>=150	0.95

Enfin, selon la zone sur laquelle l'usine est construite, ce taux sera réduit des valeurs suivantes (il s'agit ici d'une simple soustraction, ex : défavorisée => $Taux = Taux - 0.10$) :

Zone	Réduction
« défavorisée »	0.10
« industrielle »	0.05
« agglomération »	0.00

- Taxes [Taxes] : $Taux \cdot BB$, si BB est positif, 0 sinon
- Bénéfice net [BN] : $BB - Taxes$

Question 1

Construire un nouveau classeur avec une feuille de calcul (dénommée « Calcul ») permettant de connaître le bénéfice net de M. EUSEBIO. Initialement, la production est destinée au marché local et elle a lieu dans une zone défavorisée. **Attention, le taux de taxe et les coûts fixes doivent être déterminés à l'aide d'une fonction VBA.**

1. Quel serait le bénéfice de l'entreprise avec les données ci-dessus ?

VBA – Niveau 1 - Épisode II

2. Quel serait le bénéfice de l'entreprise si elle désirait produire une quantité de 150 à un prix de 50 KF ?
3. Quel serait le bénéfice de l'entreprise si elle décidait de s'implanter en agglomération et d'exporter toute sa production ?

Question 2

Quel serait le bénéfice de l'entreprise pour les quantités allant de 50 à 150 avec un pas de 10 ?

Question 3

Quel serait le bénéfice de l'entreprise si on croisait les quantités ci-dessus avec les différentes destinations de production ?

Question 4

On vous demande de fournir **le bénéfice** et **le montant des taxes** payés par l'entreprise dans les situations suivantes :

- quantité : 50 ; zone : « défavorisée » ; destination : « locale »
- quantité : 100 ; zone : « agglomération » ; destination : « exportation »
- quantité : 250 ; zone : « agglomération » ; destination : « exportation »

Un rapport récapitulatif des résultats doit être produit.

VBA – Niveau 1 - Épisode III

Somme Valeurs Paires (MaSommePaire)

Entrée : Plage de cellules (range)

Sortie : Total (réel)

Calcul : Dans la plage de cellules, repérez les valeurs paires et faites leur somme.

Indication : pour savoir si une valeur est paire, il suffit de vérifier que le reste de la division par 2 est nulle en utilisation l'opérateur modulo (**mod** en VBA, ex. si `plage.cells(1,1).value mod 2 = 0` alors la valeur contenue dans la première cellule est paire)

Ex. Dans la plage suivante, on aurait Total = $12 + 2 + 8 = 22$

12	3	1
2	5	8

Somme des valeurs sur la diagonale (MaSommeDiag)

Entrée : Plage de cellules (range)

Sortie : Total (réel)

Calcul : (a) Assurez-vous que la plage de cellule est carrée (nombre de lignes identique au nombre de colonnes).

(b) Si ce n'est pas le cas, votre fonction doit renvoyer la valeur 0

(c) Si c'est le cas, votre fonction doit renvoyer la somme des valeurs situées sur la diagonale

Ex. Dans la plage suivante, on aurait Total = $2 + 4 + 7 = 13$

2	3	1
0	4	9
5	8	7

Recherche du minimum (MonMIN)

Entrée : Plage de cellules (range)

Sortie : Total (réel)

Calcul : Nous essayons de reproduire la fonction MIN d'Excel

Stratégie :

- Prendre comme repère la première valeur de la cellule, située en (1,1)
- Parcourir l'ensemble des cellules, si le contenu de la cellule courante est plus petit que le repère, elle devient le nouveau repère (on met à jour la variable repère).
- A la sortie des boucles, le repère constitue la plus petite valeur trouvée dans la plage c.-à-d. le minimum.

VBA – Niveau 1 - Épisode III

Ex. Dans la plage suivante, 0 est la valeur minimum.

2	3	1
0	4	9
5	8	7

Détection du plus grand diviseur (MonGrandDiviseur)

Entrée : n (entier)

Sortie : d (entier)

Calcul : Détecter la valeur d, qui est plus grand diviseur de n (ex. n = 12, alors d = 6, il n'y a pas de plus grand nombre qui permette de diviser n sans qu'il n'y ait de reste (division entière)

Indication : Il suffit de tester toutes les valeurs comprises entre **1** et **(n-1)** et de ne conserver que le plus grand. Remarque : est-il nécessaire de tester jusqu'à (n-1) ?

Détection de nombre premier (MonPremier)

Entrée : n (entier)

Sortie : « Premier » ou « Pas premier » (chaîne)

Calcul : Détecter si n est un nombre premier ou pas

Indication :

- Un nombre premier est un entier divisible uniquement par 1 et par lui-même (n)
- A contrario, n n'est donc pas premier si on lui trouve un diviseur parmi **2** et **(n-1)**. C'est ce qu'il faudra exploiter pour cet exercice.

Calcul de factoriel (MonFactoriel)

Entrée : n (entier)

Sortie : Factoriel (réel) (*mettre réel même si la valeur calculée est entière pour dépasser les problèmes de dépassement de capacité*)

Calcul : Factoriel = $1 * 2 * 3 * \dots * n$

VBA – Niveau 1 - Épisode IV

1. Maximiser l'utilité du consommateur (Solveur)

Déterminez la quantité des deux produits x_1 et x_2 qui maximise la fonction d'utilité suivante :

$$U(x_1, x_2) = x_1 \times x_2$$

En tenant compte de la contrainte de budget :

$$2 \times x_1 + 4 \times x_2 \leq 100$$

(Solution : $U = 312.5$, avec $x_1 = 25$ et $x_2 = 12.5$)

2. Optimisation de la production (Solveur)

Une entreprise fabrique des détergents X et Y à partir de 2 matières premières A et B. La matrice de production est la suivante :

Matière Première \ Détergent	X	Y
A	4	2
B	1	3

Cette matrice se lit :

- Il faut 4 unités de A et 1 unité de B pour produire 1 unité X ;
- Il faut 2 unités de A et 3 unités de B pour produire 1 unité de Y.

Question 1 : Construire la feuille de calcul pour calculer les quantités de A et B utilisées en fonction des quantités de X et Y produites. Pour $X = 10$ et $Y = 10$, quelles quantités de A et B seront utilisées ? ($A = 60$ et $B = 40$)

Question 2 : Le coût unitaire de la matière première A est 5 euros, celle de B est de 12 euros. Compléter la feuille de calcul en calculant le coût total de production à partir des quantités de X et Y produites. Quel est le coût total si $X = 10$ et $Y = 10$? (Réponse : Coût total = 780 euros)

Question 3 : La recette unitaire pour X (resp. Y) vendue est 250 euros (resp. 150 euros). Le bénéfice est défini comme la différence entre la recette totale et le coût total. Quelle est la recette totale et le bénéfice total pour $X = 10$ et $Y = 10$? (Recette = 4000 ; Bénéfice = 3220).

Question 4 : Quelle est la combinaison de X et Y qui maximise le bénéfice total ? Sachant que les disponibilités en matière première A (resp. B) est 200 unités (resp. 100 unités). L'autre contrainte bien sûr est que A et B doivent être positifs ou nuls.

(Réponse : $X = 50$ et $Y = 0$; pour un Bénéfice = 10900 euros).

VBA – Niveau 1 - Épisode IV

3. Exercice récapitulatif – Solveur, Simulation et Programmation

La société Nirvana produit les lessives en poudre ALPHA, TATRA, PUMA, qu'elle vend respectivement à 25 euros, 35 euros et 55 euros la boîte.

Pour produire ces lessives, elle utilise trois matières premières (A, B et C). La composition d'une boîte de produit fini en matières premières est (en grammes)¹ :

	A	B	C
ALPHA	150	75	25
TATRA	75	125	50
PUMA	55	65	130

Les ressources sont cependant limitées. Pour une année de production, l'entreprise dispose de 1 tonne de matière première A, 0.5 tonne de B et 0.4 tonne de C.

Les matières premières ont bien entendu un coût, il est fixe pour A (120 euros par kg) et C (250 euros par kg), il dépend des quantités utilisées en ce qui concerne B

Quantité totale utilisée de B	Coût au kg des M.P. (euros)
≤ 200 kg	150
>200 et ≤ 400 kg	130
> 400 kg	60

Question 1 : Construire une feuille EXCEL de manière à obtenir la quantité de matière première utilisée (A, B et C) en fonction de la quantité de lessives produites (ALPHA, TATRA, PUMA) c.-à-d. lors que les quantités de lessives produites sont modifiées, les matières premières utilisées doivent se mettre à jour automatiquement.

- Si l'on produit 1 unité respectivement de ALPHA, TATRA et PUMA, combien de matières premières A, B et C utiliserait-on ? (Réponse : 0.28 kg de A, 0.265 kg de B et 0.205 kg de C)
- Si l'on produit 10 unités respectivement de ALPHA, TATRA et PUMA, combien de matières premières A, B et C utiliserait-on ? (Réponse : 2.8 kg de A, 2.65 kg de B et 2.05 kg de C)

¹ Ce tableau se lit : « pour produire une boîte de ALPHA, nous avons besoin de 150g de A, 75g de B et 25g de C », etc.

VBA – Niveau 1 - Épisode IV

Question 2 : Elaborer une fonction personnalisée **VBA** qui permet de calculer le coût unitaire de la matière première B (réel) en fonction de la quantité totale utilisée (réel).

Question 3 : Compléter la feuille de calcul de manière à

- Obtenir le coût total des matières premières utilisées en fonction de boîtes de lessive produites ;
- Obtenir la recette totale en fonction des boîtes de lessive produites ;
- Obtenir le bénéfice total de l'entreprise en fonction des boîtes de lessive produites.

Quel serait par exemple le bénéfice de l'entreprise s'il décidait de produire 1 boîte de chaque lessive ? 10 boîtes de lessive C uniquement ?

(Solution : 1 boîte de chaque → bénéfice = -9.6 euros ; 10 boîtes de PUMA seulement → bénéfice = 61.5 euros)

Question 4 : A partir de la feuille de calcul que vous avez préparé, calculez la combinaison optimale de lessive à produire de manière à maximiser le bénéfice de l'entreprise. Attention, la solution doit tenir compte des disponibilités en matières premières, les quantités à produire doivent être positives ou nulles.

(Solution : PUMA#3077 maximisera le bénéfice)

Question 5 : A la suite d'une analyse approfondie du marché, l'entreprise décide de se spécialiser et de produire exclusivement 3000 boîtes de la lessive PUMA par an. A priori, il s'agit d'une excellente stratégie, sauf que les marchés des matières premières A et C sont soumis à de fortes variations. Afin de vérifier la robustesse de son choix, l'entreprise vous demande de construire une table de simulation à double entrée, où le prix de A varierait de 90 à 180 euros, avec un pas de 5 euros ; et le prix de C, de 250 euros à 400 euros, avec un pas de 10 euros.

(Solution : Table de simulation à 2 entrées)

Question 6 : Ecrire une fonction VBA qui prend en entrée une plage de cellules (contenant exclusivement des réels) et une valeur (réelle) qui servira de référence. La fonction doit alors renvoyer un nombre réel correspondant au ratio entre le nombre de fois où le contenu des cellules est supérieur à la valeur de référence et le nombre total de cellules. *(Ex. Appliquer votre fonction sur la table de simulation à deux entrées en prenant comme référence le bénéfice net de l'entreprise avec les données initiales du tableau.)*

VBA – Niveau 1 - Épisode V

Exercice 1 : **Function MonPlusPetit(v As Double) As Long**

Ecrire une fonction qui renvoie le plus petit entier n tel que $(2^n \geq v)$

Stratégie : commencer avec $n = 0$, augmenter n tant que $2^n < v$

Exercice 2 : **Function MonPgcd(a As Long, b As Long) As Long**

Ecrire une fonction qui renvoie le PGCD (plus grand commun diviseur) de deux entiers a et b

Stratégie : *Tant que a est différent de b* , calculer la valeur absolue de l'écart entre a et b , et affecter le résultat au plus grand de deux. Etc. Le résultat à la sortie de la boucle est a (ou b , qu'importe puisque $a = b$).

Exercice 3 : **Function MonPpcm(a As Long, b As Long) As Long**

Ecrire une fonction qui renvoie le PPCM (plus petit commun multiple) de deux entiers a et b

Stratégie : $\text{PPCM}(a,b) = a * b$ divisé par $\text{PGCD}(a,b)$, attention on passe par une division entière

Exercice 4 : **Public Function MonMin(plage As Range) As Double**

Ecrire une fonction qui renvoie la plus petite valeur dans une plage de cellule.

Stratégie : Prendre en repère la première cellule. Parcourir les cellules, si le contenu de la cellule courante est plus petit que le repère, il devient le nouveau repère. Attention, vous devez utiliser la boucle *For Each*.

Exercice 5 : **Public Function MonMax(plage As Range) As Double**

Ecrire une fonction qui renvoie la plus grande valeur dans une plage de cellule.

Stratégie : Même schéma que minimum, mais en inversant le sens des comparaisons.

Exercice 6 : **Function MaDivision(a As Double, b As Double) As Variant**

Ecrire une fonction qui renvoie la division a / b . Attention, si $(b = 0)$, la fonction doit renvoyer la chaîne « division par zéro ».

Exercice 7 : **Public Function MaMoyenneRobuste(plage As Range) As Variant**

Cette fonction doit calculer la moyenne des valeurs d'une plage de cellules, après avoir retranché la plus petite et la plus grande valeur.

VBA – Niveau 1 - Épisode V

14	2	15
95	13	18
16	17	12

Exemple : Dans la plage `[14, 2, 15, 95, 13, 18, 16, 17, 12]`, la fonction doit calculer la moyenne des valeurs (14, 15, 13, 18, 16, 17, 12), en excluant donc 2 et 95. *Attention*, si le nombre de cellules de la plage est inférieur ou égal à 2, on renvoie la chaîne de caractères « calcul impossible ».

Exercice 8 : Public Function MaMoyenneNonVide(plage as Range) as Variant

Ecrire une fonction VBA qui prend en entrée une plage de cellules. Elle doit calculer la moyenne des valeurs en prenant en compte uniquement les cellules non vides. Attention, dans le cas où toutes les cellules sont vides, la fonction doit renvoyer la chaîne « pas de cellules valides ».

Remarque : La fonction `IsEmpty()` permet de savoir si une cellule est vide (elle renvoie TRUE) ou non (elle renvoie FALSE).

VBA – Niveau 1 - Épisode VI

1. Travailler sur une matrice à structure et taille fixées

Dans cette première partie, nous programmons différentes macros qui manipulent une matrice dont nous connaissons à l'avance la structure et la taille (Feuille « **test.1** ») c.-à-d. nous savons à l'avance que la première ligne correspond à un en-tête, qu'il y a 5 pays, 3 produits, et que la dernière colonne correspond à un total.

Pays	Continent	Produit.1	Produit.2	Produit.3	Total
Allemagne	Europe	36	58	63	157
Belgique	Europe	47	68	47	162
Chili	Ailleurs	45	39	45	129
France	Europe	12	46	78	136
Inde	Ailleurs	59	41	78	178

La colonne « Total » contient la somme des valeurs en ligne.

a. Proportions en ligne

A l'aide d'une macro VBA, calculer les pourcentages des ventes de produits pour chaque pays. Le résultat doit apparaître dans la feuille « **test.3 (résultat-0)** ». Il devrait ressembler à ceci.

Pays	Continent	Produit.1	Produit.2	Produit.3
Allemagne	Europe	0.22929936	0.36942675	0.40127389
Belgique	Europe	0.29012346	0.41975309	0.29012346
Chili	Ailleurs	0.34883721	0.30232558	0.34883721
France	Europe	0.08823529	0.33823529	0.57352941
Inde	Ailleurs	0.33146067	0.23033708	0.43820225

Signature de la macro : `Public Sub MonPourcentage()`

b. Proportions en ligne avec conditions

Programmez une macro qui réalise la même opération, mais uniquement pour les pays d'Europe. Pour les autres, le terme « non concerné » doit être inséré dans la première cellule. Le résultat doit apparaître dans la feuille « **test.3 (résultat-1)** ».

Pays	Continent	Produit.1	Produit.2	Produit.3
Allemagne	Europe	0.229299363	0.36942675	0.40127389
Belgique	Europe	0.290123457	0.41975309	0.29012346
Chili	Ailleurs	pas concerné		
France	Europe	0.088235294	0.33823529	0.57352941
Inde	Ailleurs	pas concerné		

Signature de la macro : `Public Sub MonPourcentagePlus()`

VBA – Niveau 1 - Épisode VI

c. Proportions en ligne avec mise en forme

Même question qu'en (a), sauf que la cellule doit être mise en fond vert lorsque le ratio est supérieur à 0.4. Le résultat doit apparaître dans la feuille « test.3 (résultat-2) ».

Pays	Continent	Produit.1	Produit.2	Produit.3
Allemagne	Europe	0.22929936	0.36942675	0.40127389
Belgique	Europe	0.29012346	0.41975309	0.29012346
Chili	Ailleurs	0.34883721	0.30232558	0.34883721
France	Europe	0.08823529	0.33823529	0.57352941
Inde	Ailleurs	0.33146067	0.23033708	0.43820225

Signature de la macro : `Public Sub MonPourcentagePlusEncore()`

2. Travailler sur des plages de données à taille non définie

Dans cette seconde partie, nous connaissons l'organisation des données, mais pas le nombre de valeurs à manipuler. La macro VBA doit donc s'adapter et arrêter les traitements lorsqu'elle accède à une cellule vide, marquant la fin de la liste à traiter.

Valeurs	a. Liste sur une colonne
10	
25	Dans la feuille « test.4 », la macro doit parcourir une liste de valeurs située
81	forcément sur colonne A et commençant à la seconde ligne.
32	
41	Elle doit mettre en fond vert chaque cellule comportant une valeur supérieure à
91	50. On considère que les valeurs contenues dans les cellules sont toujours
47	numériques, il n'est donc pas nécessaire de procéder à une vérification.
91	
57	Dès qu'elle rencontre une cellule vide, la macro doit s'arrêter et afficher dans
18	une boîte de dialogue le message « Recherche terminée ».
12	
80	<u>Remarque 1</u> : Utilisez la procédure standard de VBA Excel <code>MsgBox()</code> pour afficher
40	la boîte de dialogue.
45	
78	<u>Remarque 2</u> : La commande <code>IsEmpty()</code> permet de savoir si une cellule est vide.
93	Voir les corrigés des TD précédents.
44	
69	
10	Signature de la macro : <code>Public Sub MaColonneEnVert()</code>
70	
23	

VBA – Niveau 1 - Épisode VI

b. Liste sur plusieurs colonnes

Dans la feuille « test.5 », la macro doit parcourir une matrice de valeurs contigües commençant à la seconde ligne, mais dont on ne connaît ni le nombre de lignes, ni le nombre de colonnes pour chaque ligne.

La macro doit :

- Démarrer à la cellule A2 de la feuille (ligne 2 et colonne 1) ;
- Pour chaque ligne, elle doit parcourir les cellules d'une colonne à l'autre. Elle met chaque cellule comportant une valeur supérieure à 50 en fond vert. La recherche en ligne est terminée lorsque que l'on rencontre une cellule vide (ex. C2 dans la copie d'écran ci-dessous lorsqu'on explore la ligne 2) ;
- Lorsqu'une ligne a été parcourue, la macro doit passer à la ligne suivante et réitérer l'exploration horizontale. Puis passer à la ligne, etc.
- Dès que l'on rencontre une cellule vide dans la première colonne d'une ligne, on arrête la recherche et on affiche dans une boîte de dialogue le message « recherche terminée » avec le nombre de cellules passées en fond vert.

Voici un exemple d'exécution du programme :

	A	B	C	D	E	F
1	Valeurs					
2	10	72				
3	25	86	86	67		17
4	81	76	25	9		
5	32	3	38			
6	41					
7	91				86	
8	47					
9	91					
10	57					
11	18				99	
12	12					
13	80					
14	40				39	
15	45	92				
16	78	8	52	87		41
17	93	70				
18	44	55	85	28		80
19	69	97	85	28		
20						
21						
22						

VBA – Niveau 1 - Épisode VI

Signature de la macro : `Public Sub MaColonneEnVertPlus()`

3. Répétition de l'exécution d'une feuille et récupération des résultats

a. Simulation de jets de pièces de monnaie

La feuille « **simulation pièce** » permet de simuler le jet de 3 pièces de monnaie. Nous obtenons en **C5** le nombre d'apparition de « FACE ».

Notre objectif est de réitérer 1000 fois l'expérience et de stocker au fur et à mesure les résultats obtenus dans la **colonne F** (en commençant par la case F2, puis F3, etc.). En **H4** doit s'inscrire la moyenne empirique (l'estimation de l'espérance) d'obtention du côté FACE dans ce type d'expérimentation (la formule est déjà insérée dans la cellule). Comme il s'agit d'une simulation, les résultats peuvent varier légèrement d'une session à l'autre.

Voici la configuration de la feuille avant exécution de la macro. Toutes les formules sont prêtes, elle doit se charger de répéter l'expérience, de collecter le résultat en **C5** et de l'inscrire dans la colonne F.

	A	B	C	D	E	F	G	H
1		Aléatoire	Pièce		Essai	Nb.Face		
2	Jet 1	0.71497146	pile		1			
3	Jet 2	0.70250637	pile		2			Moyenne
4	Jet 3	0.38416707	face		3			#DIV/0!
5		Nb.Face	1		4			
6					5			
7					6			

Voici ce que l'on devrait obtenir à l'issue de l'exécution de la macro.

	A	B	C	D	E	F	G	H
1		Aléatoire	Pièce		Essai	Nb.Face		
2	Jet 1	0.14947798	face		1	2		
3	Jet 2	0.21521484	face		2	3		Moyenne
4	Jet 3	0.01784918	face		3	1		1.52
5		Nb.Face	3		4	2		
6					5	1		
7					6	2		

A la première expérience, FACE est apparue 2 fois ; à la seconde, 3 fois ; etc.

L'estimation issue de la simulation est plutôt pas mal puisque l'espérance mathématique est égale à $n \times p = 3 \times 0.5 = 1.5$ (où n est le nombre de pièces lancées à chaque expérience, p est la probabilité d'obtenir FACE sur une pièce).

Signature de la macro : `Public Sub MaSimulationPiece()`

VBA – Niveau 1 - Épisode VI

b. Scénarios multiples

Une feuille de calcul « **scénarios multiples** » permet de calculer le bénéfice des entreprises à partir, entre autres du prix unitaire du produit mis en vente, la quantité produite et le coût unitaire. La feuille se présente comme suit :

	A	B
1	Prix	120
2	Quantité	60
3	Cout Unitaire	2.4
4	Cout Fixe	1000
5	Cout total	1144
6	Recette	7200
7	Bénéfice	6056

Pour les valeurs : prix = 120, quantité = 60, coût unitaire = 2.4, le bénéfice sera de 6056.

On se propose d'évaluer l'impact sur les bénéfices de différentes combinaisons de prix, quantité et coût unitaire. Le [gestionnaire de scénarios](#) d'Excel devrait faire l'affaire dans ce cas figure. Sauf que nous voulons combiner les valeurs de prix allant de 60 à 120 avec un pas de 10, de quantité allant de 40 à 60 avec un pas de 1, de coût unitaire allant 1.5 à 2.5 avec un pas de 0.1. Rentrer ces multiples combinaisons unes à unes (*il y en a 1617 en tout*) dans le gestionnaire de scénarios n'est pas envisageable. Il nous faut passer par une macro.

Elle va se charger de rentrer les combinaisons de valeurs en **B1**, **B2** et **B3**, de les reporter en **colonnes D**, **E** et **F**, de récupérer le bénéfice en **B7**, pour l'inscrire dans la **colonne G**. Voici une copie d'écran de la feuille après exécution de la macro.

	A	B	C	D	E	F	G
1	Prix	120		Prix	Quantité	Cout unitaire	Bénéfice
2	Quantité	60		60	40	1.5	1340
3	Cout Unitaire	2.4		60	40	1.6	1336
4	Cout Fixe	1000		60	40	1.7	1332
5	Cout total	1144		60	40	1.8	1328
6	Recette	7200		60	40	1.9	1324
7	Bénéfice	6056		60	40	2	1320
8				60	40	2.1	1316
9				60	40	2.2	1312

1^{ère} combinaison
2^{ème} combinaison
Etc.

Les valeurs vont au-delà de la ligne 9 bien sûr.

Signature de la macro : `Public Sub MesScenarios()`

VBA – Niveau 1 - Épisode VII

Programmation de macros VBA qui manipule les plages de cellules sélectionnées par l'utilisateur. La sélection peut être simple (une seule zone) ou multiple (plusieurs zones).

1. Sélection simple

a. Recherche de minimum

Ecrire une macro VBA qui, à partir d'une plage de cellules sélectionnée par l'utilisateur (voir le mot clé **Selection** en VBA), vérifie qu'il y a bien une seule zone (dans le cas contraire, elle affiche le message ERREUR dans une boîte de dialogue), puis va repérer la cellule contenant la valeur minimale et la met en fond vert.

Variante 1 : Si le minimum apparaît plusieurs fois, la mise en évidence de la première cellule concernée suffit.

12	15	63	26	36
5	58	34	5	52
12	36	14	87	5

Signature de la fonction : `Public Sub MonMinSelectionSimpleSolution()`

Variante 2 : Si le minimum apparaît plusieurs fois, il faut mettre en évidence toutes les cellules concernées.

12	15	63	26	36
5	58	34	5	52
12	36	14	87	5

b. Recherche de maximum et minimum

Ecrire une macro VBA qui à partir d'une plage de cellules sélectionnée par l'utilisateur, vérifie qu'il y a bien une seule zone (dans le cas contraire, elle affiche le message ERREUR dans une boîte de dialogue), puis va repérer les cellules contenant respectivement la valeur minimale et la valeur maximale : la première sera mise en fond vert, la seconde en fond bleu.

Variante 1 : Si le minimum (maximum) apparaît plusieurs fois, la mise en évidence de la première cellule concernée suffit.

12	15	63	26	36
5	58	34	5	52
12	36	14	87	5

Signature de la fonction : `Public Sub MonMinMaxSelectionSimple()`

VBA – Niveau 1 - Épisode VII

Variante 2 : Si le minimum (maximum) apparaît plusieurs fois, il faut mettre en évidence toutes les cellules concernées.

2. Sélection multiple

a. Manipulation des zones entières

Ecrire une macro VBA qui, à partir d'une sélection multiple de l'utilisateur, met en fond vert chaque zone.

Exemple : il y a 4 zones dans la sélection multiple.

	A	B	C	D	E	F	G
1							
2		45	4				
3		6	3				
4		14	26				
5							
6				12	2		
7				36	24		
8							
9							
10							
11							
12		14	25	14	41		
13		12	2	3	15		
14							
15							
16							
17					12	15	1
18					47	69	57
19							

Signature de la fonction : `Public Sub MonVertToutesZones()`

b. Repérage de la plus grande zone

Ecrire une macro VBA qui, à partir d'une sélection de l'utilisateur :

1. Vérifie qu'il s'agit bien une sélection multiple, dans le cas contraire elle affiche dans une boîte de dialogue le message "sélection incorrecte" (voir MessageBox) ;
2. Si c'est bien une sélection multiple, elle va repérer la zone la plus grande (en nombre de cellules) et la met en fond vert.

Exemple : la zone la plus grande contient $2 \times 4 = 8$ cellules.

VBA – Niveau 1 - Épisode VII

	A	B	C	D	E	F	G
1							
2		45	4				
3		6	3				
4		14	26				
5							
6				12	2		
7				36	24		
8							
9							
10							
11							
12		14	25	14	41		
13		12	2	3	15		
14							
15							
16							
17					12	15	1
18					47	69	57
19							

Signature de la fonction : `Public Sub MaPlusGrandeZone()`

c. Travail à l'intérieur des zones

Ecrire une macro VBA qui va réaliser les opérations suivantes pour chaque zone de la sélection de l'utilisateur :

1. Repère la cellule contenant la valeur minimale ;
2. Met cette cellule en fond vert.

Exemple : Dans les 4 zones de la sélection multiple ci-dessous, la cellule contenant la valeur minimale (à l'intérieur de la zone) est mise en évidence.

	A	B	C	D	E	F	G
1							
2		45	4				
3		6	3				
4		14	26				
5							
6				12	2		
7				36	24		
8							
9							
10							
11							
12		14	25	14	41		
13		12	2	3	15		
14							
15							
16							
17					12	15	1
18					47	69	57
19							

Signature de la fonction : `Public Sub MonMinDansChaqueZone()`

Pour aller plus loin : Dans chaque zone, si le minimum apparaît plusieurs fois, il faudra tous les mettre en évidence.