

VBA pour EXCEL

Version 2013

I. INTRODUCTION

VBA est l'acronyme de Visual Basic (Simple et Visuelle) pour Application. Le VBA a besoin d'une application hôte pour pouvoir exécuter ses programmes. En d'autres termes, lorsque vous écrivez un programme en VBA pour Excel, vous devez absolument posséder Excel pour faire tourner votre programme. Les applications hôtes de VBA sont essentiellement les applications de la suite Office, mais d'autres programmes comme Autocad, peuvent être programmés à l'aide de VBA.

Les programmes écrits en VBA sont donc interprétés et non compilés.

Les programmes interprétés s'exécutent moins rapidement que les programmes compilés.

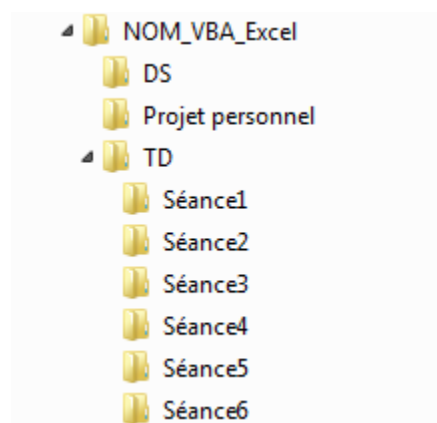
Le VBA permet à celui qui utilise le tableur Excel d'augmenter les potentialités du tableur et de l'adapter à ses besoins et à son environnement de travail.

En effet, il est extrêmement facile de saisir des données dans le tableur, mais dès que l'on veut exercer un contrôle minimal sur les informations saisies, il faut avoir recours à la programmation VBA. De la même manière si vous développez des modèles dans le tableur qui doivent être utilisés par d'autres, la programmation VBA vous aidera à définir des écrans d'aide spécifiques ou bien des formulaires de saisie personnalisés qui faciliteront la tâche de ceux qui doivent entrer les informations. Enfin, vous pouvez écrire vos propres fonctions qui sont absentes du tableur Excel.

Mais pour apprendre le VBA pour Excel, il faut déjà avoir des notions sur Excel et ensuite apprendre le langage de programmation VBA.

Pour les TPs

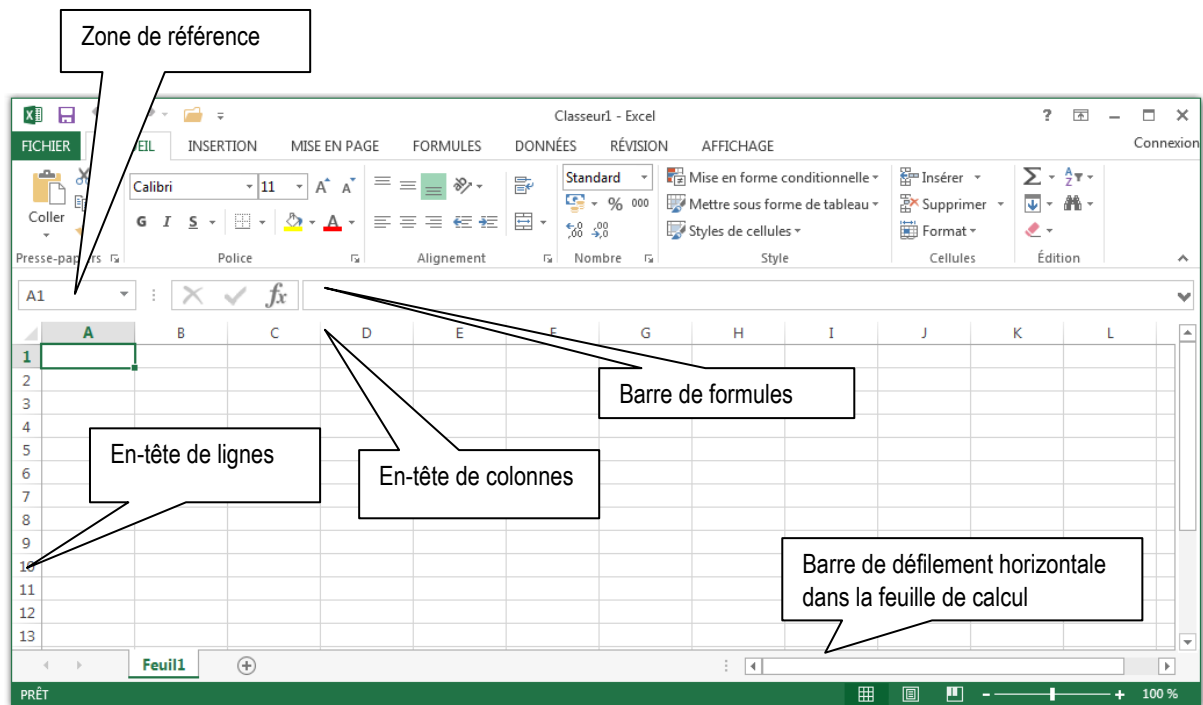
Vous commencerez par créer un répertoire et des sous répertoires comme indiqué ci-dessous.



II. Le classeur et ses feuilles

Lorsqu'on crée un nouveau classeur dans Excel 2013, on obtient un classeur contenant une feuille de calcul nommée *Feuil1*

Voici la page qui s'ouvre après lancement du tableur.



Chaque feuille de calcul se présente sous forme d'un tableau de 2^{20} ($=1048576$) lignes et 2^{14} ($=16384$) colonnes, soit au total : 17179869184 cellules.

Chaque cellule peut contenir un nombre, du texte ou une formule.

Afin de saisir quelque chose dans une cellule, il faut placer le curseur dans cette cellule à l'aide de la souris en cliquant dans la cellule, il suffit alors de saisir le nombre ou le texte souhaité au clavier et d'appuyer sur la touche Entrée à la fin de la saisie.

TP 1 : Somme d'entiers consécutifs

Ouvrir le classeur appelé TP123_Excels qui contient 6 feuilles.

Les feuilles 1, 2 et 3 serviront pour votre travail.

Les feuilles appelées *Somme*, *Fonction* et *Suite* contiennent les solutions des exercices proposés et peuvent être à tout moment utilisées pour vérification ou récupération d'une formule.

Voici la page que l'on souhaite obtenir dans ce premier exemple. (Le corrigé est dans la feuille *Somme*).

	A	B	C	D	E	F
	n =	S= 1 + 2 +...+ n		n =	2n - 1 =	S = 1 + 3 + 5 +...+ (2n -1)
1						
2	1	1		1	1	1
3	2	3		2	3	4
4	3	6		3	5	9
5	4	10		4	7	16
6	5	15		5	9	25
7	6	21		6	11	36
8	7	28		7	13	49
9	8	36		8	15	64
10	9	45		9	17	81
11	10	55		10	19	100
12	11	66		11	21	121
13	12	78		12	23	144
14	13	91		13	25	169
15	14	105		14	27	196
16	15	120		15	29	225
17	16	136		16	31	256
18	17	153		17	33	289
19	18	171		18	35	324
20	19	190		19	37	361
21	20	210		20	39	400

- Après avoir lancé Excel, se placer dans la feuille vierge 1, placer le curseur en A1 , remplir la première colonne.
- Placer le curseur en B1, saisir le titre : $S = 1 + 2 + 3 + \dots + n$.
- Saisir le contenu des deux cellules suivantes : dans la cellule B2 taper = A2, dans la cellule B3, saisir = A 3 + B 2

Il suffit alors de placer le curseur dans la cellule B3, de cliquer sur la croix en bas à droite de la cellule et en laisser le doigt enfoncé de tirer vers le bas jusqu'à la ligne 21 afin de recopier le contenu de la cellule B3, contenu qui va s'actualiser dans chacune des cellules de B4 à B21.

Placer le curseur dans une des cellules de la colonne B afin de contrôler la formule et de comprendre comment l'écriture relative de la formule placée en B3 s'est adaptée durant le déplacement du curseur.

- Compléter de même le tableau 2. On pourra vérifier en ouvrant la feuille *Somme* et en plaçant le curseur dans la cellule dont on souhaite contrôler le contenu.

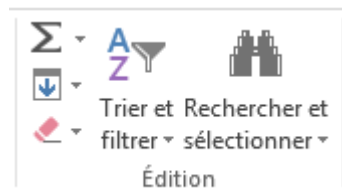
TP 2 : Tableau de valeurs et représentation graphique d'une fonction numérique

On souhaite étudier une fonction numérique en réalisant un tableau de valeurs et l'utiliser pour construire la représentation graphique d'une fonction en utilisant les possibilités graphiques d'Excel.

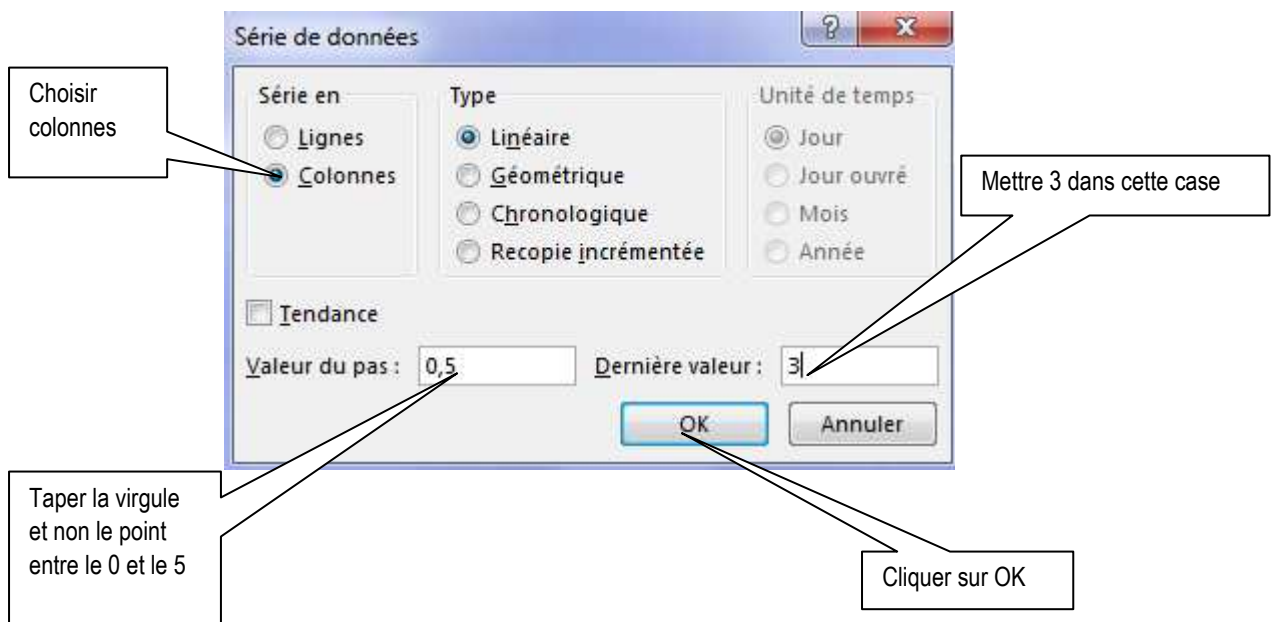
La fonction à étudier est la fonction définie par : $x^2 + 3x - 4$.

Dans la colonne A, on définit la plage de valeurs de la variable x que l'on fera varier de -6 à 3 avec un pas de 0.5 . Dans la colonne B, on affichera la liste des images en écrivant en B2 la formule $= A2^2 + 3 * A2 - 4$, formule que l'on recopiera jusqu'en B20 à l'aide du déplacement du curseur.

- Se placer dans la cellule A1 de la feuille 2, taper la lettre x puis Entrée.
- En cellule A2, saisir -6 puis Entrée. Replacer le curseur dans la cellule A2.
- Dans la barre d'outils, sélectionner l'onglet Accueil, le menu Edition,

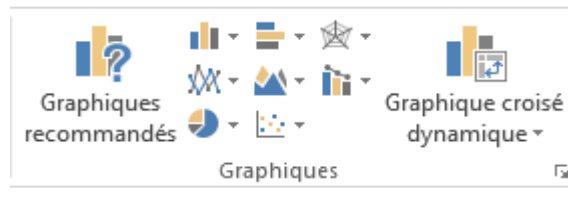


- Sélectionner Remplissage et dans ce sous menu choisir Série. Renseigner alors la boîte de dialogue qui s'ouvre comme indiquée ci-dessous.



- Cliquer dans la cellule B1 et saisir le texte $f(x)$.
- Cliquer dans la cellule B2 et saisir la formule $= A2^2 + 3 * A2 - 4$ et appuyer sur Entrée.
- Replacer le curseur dans la cellule B2 et double cliquer sur le point situé dans le coin inférieur droit de la cellule B2, la colonne B se remplit automatiquement.

- h) Sélectionner les données dans la plage A2 :B20
i) Pour faire apparaître le graphique de f , cliquer sur l'onglet Insertion menu / Graphique



- j) Choisir nuage de points avec courbe lissée par exemple.
k) Utiliser les outils de la barre d'outils pour le modifier ou le compléter.

TP 3 : Une suite numérique

On souhaite étudier les suites récurrentes du type : $u_{n+1} = au_n + bu_{n-1}$

Les nombres a et b seront des nombres réels que l'on pourra modifier, ainsi que les valeurs de u_0 et u_1 des deux premiers termes de la suite.

- a) Ouvrir la feuille 3 et la remplir comme indiqué dans la figure ci-dessous.

	A	B	C	D	E	F
1	Tabulation des termes de la suite : $u_{n+1} = a*u_n + b*u_{n-1}$					
2						
3	a =	1		u0 =	0	
4	b =	1		u1 =	1	
5						
6						
7	n =	un =			=E\$3	
8	0	0				
9	1	1				
10	2	1				
11	3	2				
12	4	3				
13	5	5				
14	6	8				
15	7	13				

Afin de comprendre la syntaxe utilisée dans les cellules B8, B9 et B10, il est nécessaire de comprendre la notion de référence relative et de référence absolue.

TP 4 : Traitement de donnée statistiques à une variable

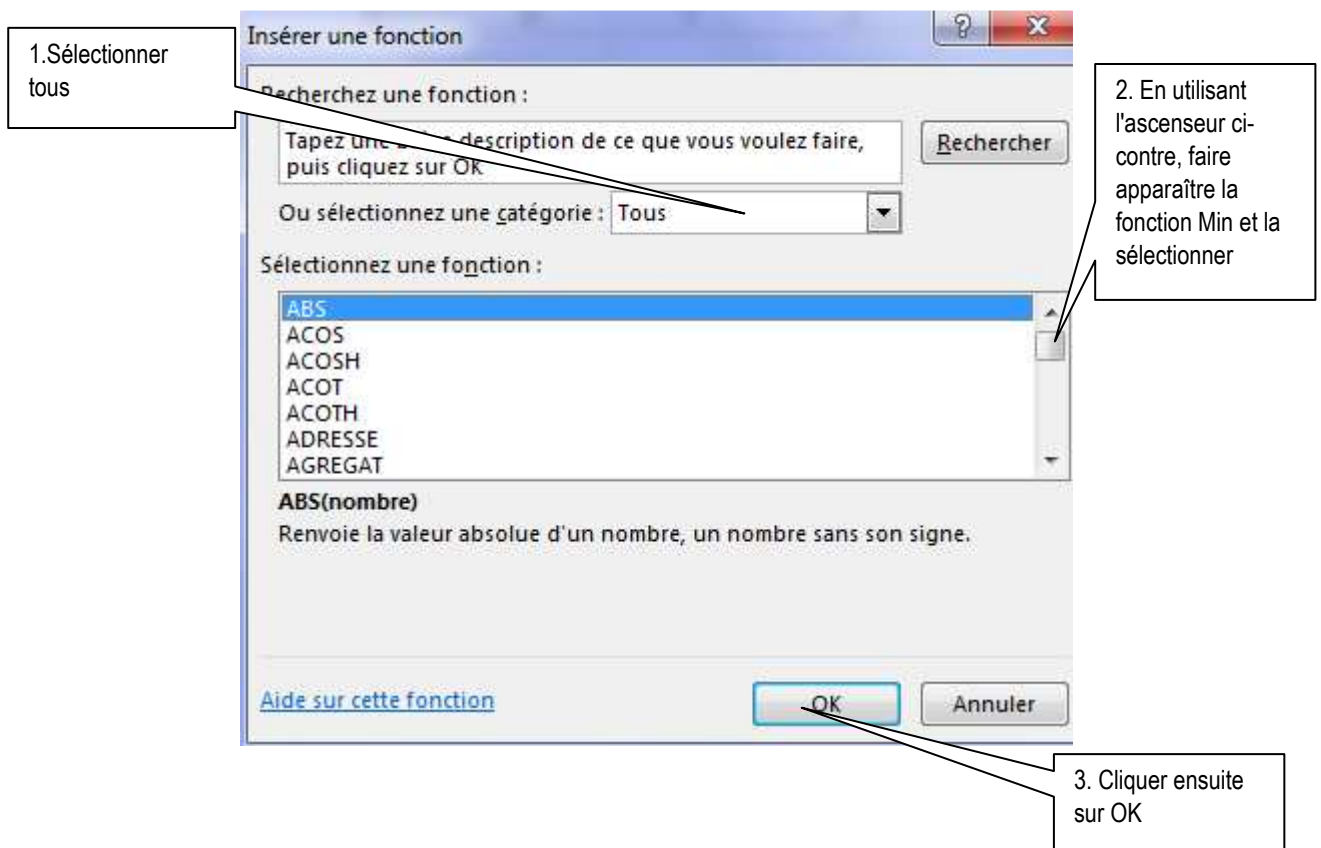
1) Calculer les paramètres statistiques des données brutes à l'aide des fonctions intégrées d'Excel

On souhaite dans la feuille appelée *Résumée* commencer par afficher le minimum, le maximum, la moyenne, la variance et l'écart-type des données contenues dans la feuille appelée *Données* comme indiqué dans la figure 1 ci-dessous

Figure 1

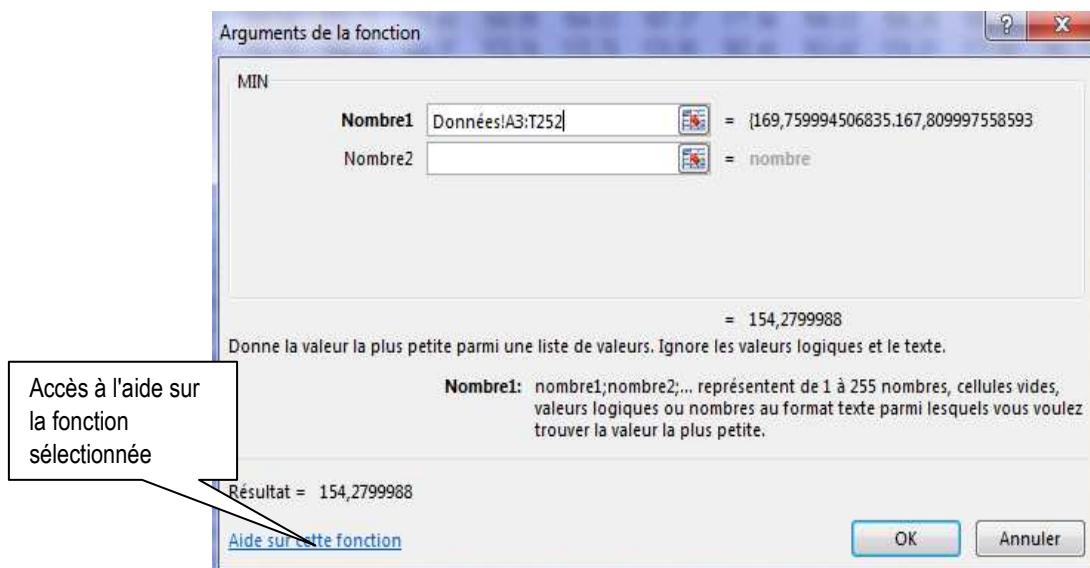
B	C	D	E	F	G
Résumé statistique des données brutes					
Min =	154,28	Moy =	169,96	Var =	16,519
Max =	183,80			Sigma =	4,064

- Se placer sur la feuille *Résumé*
- Placer le curseur dans la cellule B1 et saisir le texte : *Résumé des données brutes* et taper sur la touche Entrée.
- Placer le curseur dans la cellule B2, saisir le texte Min=, taper sur Entrée
- Puis placer le curseur dans la cellule C2 et cliquer sur l'icône fx de la barre d'outils.



- Renseigner alors la boîte de dialogue comme indiqué ci-dessous, les valeurs se trouvent dans la feuille appelée *Données* et dans la plage A3 : T252. La syntaxe est : **Données!A3:T252**

Il suffit alors de cliquer sur OK, la valeur minimale vient se placer dans la cellule C2.



- f) Procéder de même pour remplir le contenu des cellules suivantes. Chacune des fonctions utilisées pouvant être récupérée en utilisant l'icône fx de la barre d'outils.

2) Construction d'un tableau de regroupement en classes d'amplitude 2

Le minimum de la série qui vaut environ 154.28 permet de débiter la première classe à 154 et le maximum permet de débiter la dernière à 182.

En vous basant sur la syntaxe de la troisième feuille appelée *Réponses* en cas de difficultés, compléter le tableau 1.

Quelques commentaires : Pour déterminer l'effectif de chaque classe, voici la méthode qui a été employée.

Il faut étudier avec soin le contenu de la formule placée en D7 formule qui a ensuite été recopiée jusqu'en D21.

En nommant la plage **Données!\$A\$3:\$T\$252** par **donnee**

=NB.SI(donnee;">="&B7)-NB.SI(donnee;">="&C7)

On utilise la fonction NB.SI d'Excel qui demande deux arguments, la plage puis le critère.

L'instruction suivante va comptabiliser le nombre de cellules de la plage contenant les données qui sont supérieures ou égales au contenu de la cellule B7 qui est ici 154, la réponse est donc 5000 puisque toutes les valeurs de la série sont supérieures au minimum (154.28..) calculé précédemment.

=NB.SI(donnee;">="&B7)

En ôtant au résultat précédent le nombre de valeurs supérieures ou égales à 156, on obtiendra le nombre de valeurs appartenant à la classe [154;156[.

Une fois le tableau rempli, il est possible de calculer la moyenne, la variance et l'écart type des données résumées en prenant les centres de classe.

Remarque : Deux fonctions distinctes, **VAR.P** et **VAR** portent le nom de variance dans Excel, voir l'aide d'Excel qui permet de les différencier.

Ici, on utilise l'instruction VAR.P qui considère que les données coïncident avec toute la population pour en calculer la variance. L'instruction VAR estime la variance d'une population en considérant que les données représentent un échantillon de la population à mesurer, c'est un estimateur non biaisé de la variance de la population dans laquelle est puisé l'échantillon.

La somme de chacune des colonnes utiles du tableau a été effectuée en utilisant la fonction somme accessible dans la barre d'outils par l'icône Σ , on place par exemple le curseur en D23 pour calculer la somme des effectifs, on "double clique" sur l'icône Σ , la somme du contenu des cellules figurant au dessus de la cellule D23 est automatiquement calculée et affichée en D23.

3) Construction d'un tableau de regroupement en classes d'amplitude 4

On pourra afin de mesurer l'effet du regroupement en classes sur la valeur des paramètres de la série, reprendre le travail précédent avec des classes d'amplitude 4.

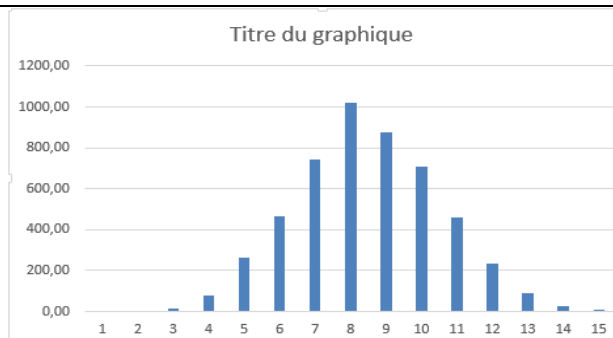
Les résultats sont visibles dans la feuille *Réponses*.

4) Construction d'un diagramme pour chacun des tableaux

- a) On sélectionne la colonne D7:D21 du tableau 1 à l'aide de la souris,
- b) On choisit l'onglet Insertion,
- c) Dans le menu Graphique on choisit dans la fenêtre Histogramme 2D

Voici résumées dans le tableau qui suit les différentes étapes pour obtenir le graphique qui figure dans la feuille *Réponses*.

Voici ce qui s'affiche après avoir choisi histogramme 2D



On clique sur une des barres du graphique afin de la sélectionner puis on effectue un clic droit qui ouvre une boîte de dialogue dans laquelle on choisit Mettre en forme d'une série de données.

On agit ensuite sur le curseur Largeur de l'intervalle que l'on met à 0% afin de supprimer l'intervalle qui sépare les barres du diagramme.

OPTIONS DES SÉRIES ▼



▲ OPTIONS DES SÉRIES

Tracer la série avec

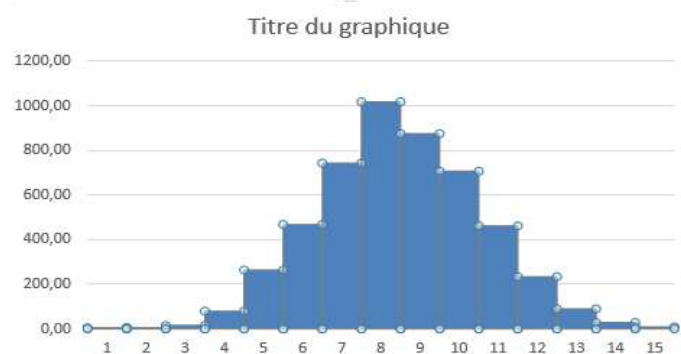
☒ Axe principal

☐ Axe secondaire

Superposition de séries 100 %

Largeur de l'intervalle 0 %

On obtient la figure ci-contre



On choisit Remplissage,

OPTIONS DES SÉRIES ▼



► REMPLISSAGE

► BORDURE

on coche Remplissage dégradé et on ouvre la palette de couleurs

☐ Aucun remplissage

☐ Remplissage uni

☒ Remplissage dégradé

☐ Remplissage avec image ou texture

☐ Motif de remplissage

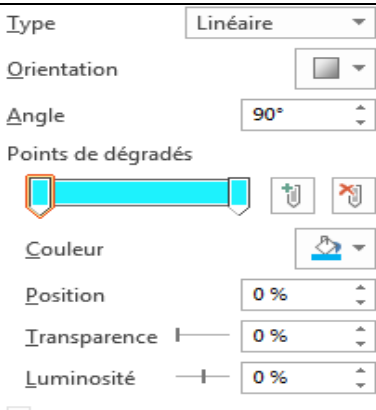


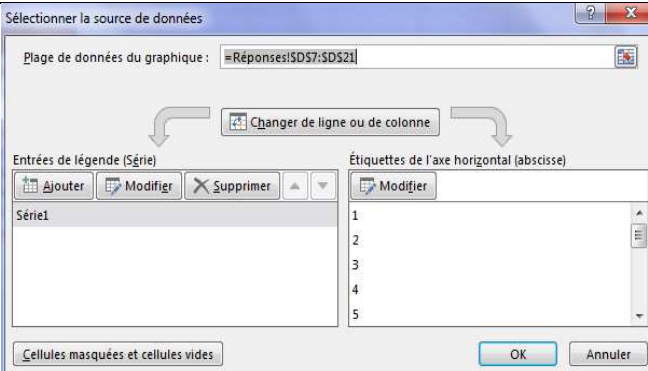
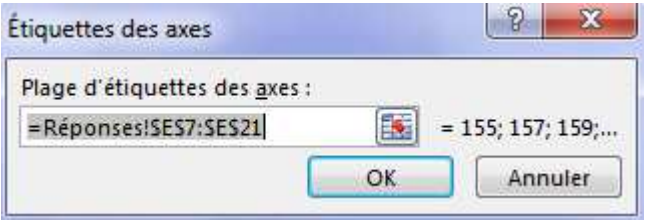
☐ Automatique

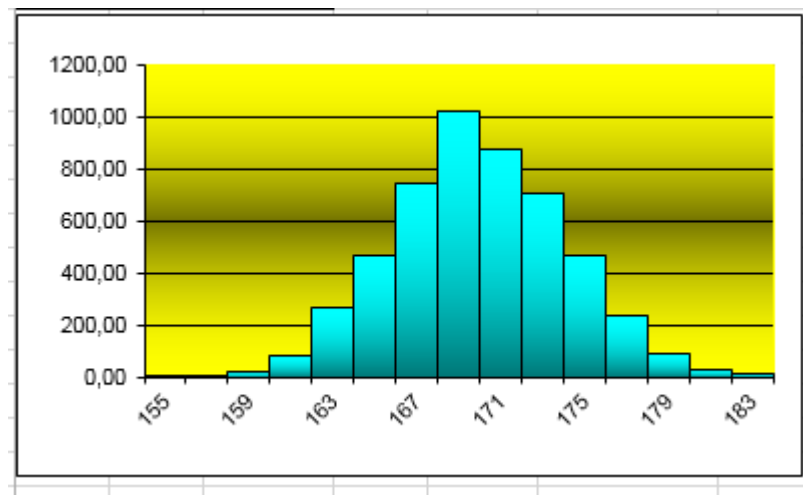
☐ Inverser si négatif

☐ Varier les couleurs par point

Dégradés prédéfinis



	
<p>Il reste à gérer l'axe des abscisses et l'arrière-plan du graphique.</p> <p>On clique dans l'arrière plan du graphique et de nouveau, après un clic droit on sélectionne Mise en forme de la zone de traçage, on coche Remplissage uni et on choisit la couleur Jaune, mais on peut utiliser un dégradé nuancé, il suffit de parcourir les différentes possibilités offertes par le logiciel.</p>	<p>▲ REPLISSAGE</p> <ul style="list-style-type: none"> <input type="radio"/> Aucun remplissage <input checked="" type="radio"/> Remplissage uni <input type="radio"/> Remplissage dégradé <input type="radio"/> Remplissage avec image ou texture <input type="radio"/> Motif de remplissage <input type="radio"/> Automatique <p>Couleur </p> <p>Transparence  0 %</p>
<p>On clique ensuite sur l'axe des abscisses</p> <p>On effectue un clic droit et on choisit Sélectionner des données, la boîte de dialogue ci-contre s'ouvre alors.</p> <p>Dans la partie Etiquettes de l'axe horizontal, on clique sur Modifier et dans notre feuille Excel on sélectionne les données.</p>	 



TP 5 : Lancers de pièces et fluctuation d'échantillonnage

On souhaite simuler le lancer d'une pièce de monnaie n fois et observer, suivant les valeurs de n , la variation de la fréquence de sortie de « Face » ou de « Pile ».

On pourra choisir, par exemple, les valeurs de n : 5, 10, 100 et 1000

A chacune des valeurs de n , on associe une colonne dans laquelle on simulera n fois le lancer d'une pièce de monnaie supposée parfaitement équilibrée.

- 1) Quelle instruction permet de simuler le lancer d'une pièce de monnaie en affichant le mot Pile ou le mot Face avec la probabilité 0.5 ?
- 2) Une fois la colonne remplie des n résultats, comment traiter les données obtenues ?

On souhaite réaliser un petit tableau ayant la forme suivante : pour chacune des colonnes.

Pile	524
Face	476

Quelle formule doit-on placer dans la cellule figurant à droite du mot Pile et qui permettra de comptabiliser le nombre de Pile obtenu ?

- 3) On souhaite faire afficher pour chacun des petits tableaux obtenus, un graphique circulaire qui permettra de visualiser les fréquences pour chaque valeur de n . on pourra une fois la feuille terminée, agir sur la touche F9 pour relancer automatiquement de nouvelles simulations pour les valeurs de n choisies.

Pour obtenir les graphiques circulaires, on procède comme suit : on sélectionne un tableau à l'aide de la souris.

Pile	45
Face	55

Sous l'onglet insertion, sous le groupe graphique, choisir secteur et secteur 3D.

Une fois le graphique affiché et sélectionné, il est possible de modifier ses paramètres par un clic droit et en sélectionnant la rubrique voulue.

TP 6 : Taux d'alcoolémie

On souhaite réaliser une feuille de calcul comme celle qui suit :

	A	B	C	D	E	F	G	H	I	J	K
1	Calcul du taux d'alcoolémie							Elimination			
2											
3	Nbre de verres	Boisson	Volume /verre	% Alcool	Volume Consommé	Volume d'alcool		Temps écoulé (en heure)	Taux résiduel (H)	Taux résiduel (F)	
4											
5	0	Bière	250	5%	0	0		0	0,95	1,11	
6	0	Cognac	20	45%	0	0		1	0,85	1,01	
7	2	Whisky	50	40%	100	40		2	0,75	0,91	
8	0	Champagne	100	12%	0	0		3	0,65	0,81	
9	2	Vin rouge	120	11%	240	26,4		4	0,55	0,71	
10	0	Vin blanc	120	11%	0	0		5	0,45	0,61	
11	Tous les volumes sont exprimés en ml							6	0,00	0,51	
12								7	0,00	0,41	
13				Masse du consommateur :		100 kg		8	0,00	0,31	
14				Quantité d'alcool consommée :		66,4 ml		9	0,00	0,21	
15				Taux d'alcoolémie(H) :		0,95		10	0,00	0,11	
16				Taux d'alcoolémie(F) :		1,11		11	0,00	0,01	
17								12	0,00	0,00	
18		NE PAS PRENDRE LE VOLANT						13	0,00	0,00	
19								14	0,00	0,00	
20		NE PAS PRENDRE LE VOLANT						15	0,00	0,00	
21								16	0,00	0,00	

- 1) Renseigner les libellés et remplir les colonnes A, B, C et D du premier tableau.
- 2) A partir du contenu des cellules des colonnes A, C, saisir en E5 la formule permettant le calcul du volume consommé, recopier cette formule jusqu'en E10.
Faire de même pour la colonne F.
- 3) Saisir les formules en F14 et F15 : en F14 =somme(F5 :F10) et en F15 = F14/(F13*C) où C=0.6 pour une femme et 0.7 pour un homme.
Afin de colorier l'intérieur de la cellule F15 si le contenu de celle-ci dépasse 0.5g/l, utilisez la mise en forme conditionnelle. Cliquer sur la cellule F15, puis, sous l'onglet **Accueil**, dans le groupe **Style**, cliquez sur la flèche en regard de **Mise en forme conditionnelle**, choisir **Règles de mise en surbrillance des cellules**, supérieur à et renseigner la boîte de dialogue.
- 4) Placer le curseur en B25 et saisir la formule :
`=SI(F15>3.5;"RISQUE DE MORT";SI(F15>2.5;"RISQUE DE COMA";SI(F15>0.5;"NE PAS PRENDRE LE VOLANT";""))`

Analyser cette formule et imaginer l'affichage en fonction du résultat qui sera retourné dans la cellule F15

- 5) Remplir la première colonne du tableau Elimination : saisir 0 en H5 puis cliquer dans la barre d'outils, onglet Accueil, menu Edition/Icône Remplissage puis Série. Dans la boîte de dialogue qui s'ouvre choisir série en colonne, Type linéaire, Valeur du pas 1, Dernière valeur 24 puis cliquer sur OK.
- 6) En I5, taper =F15 (on recopie la valeur du taux calculé en F15).

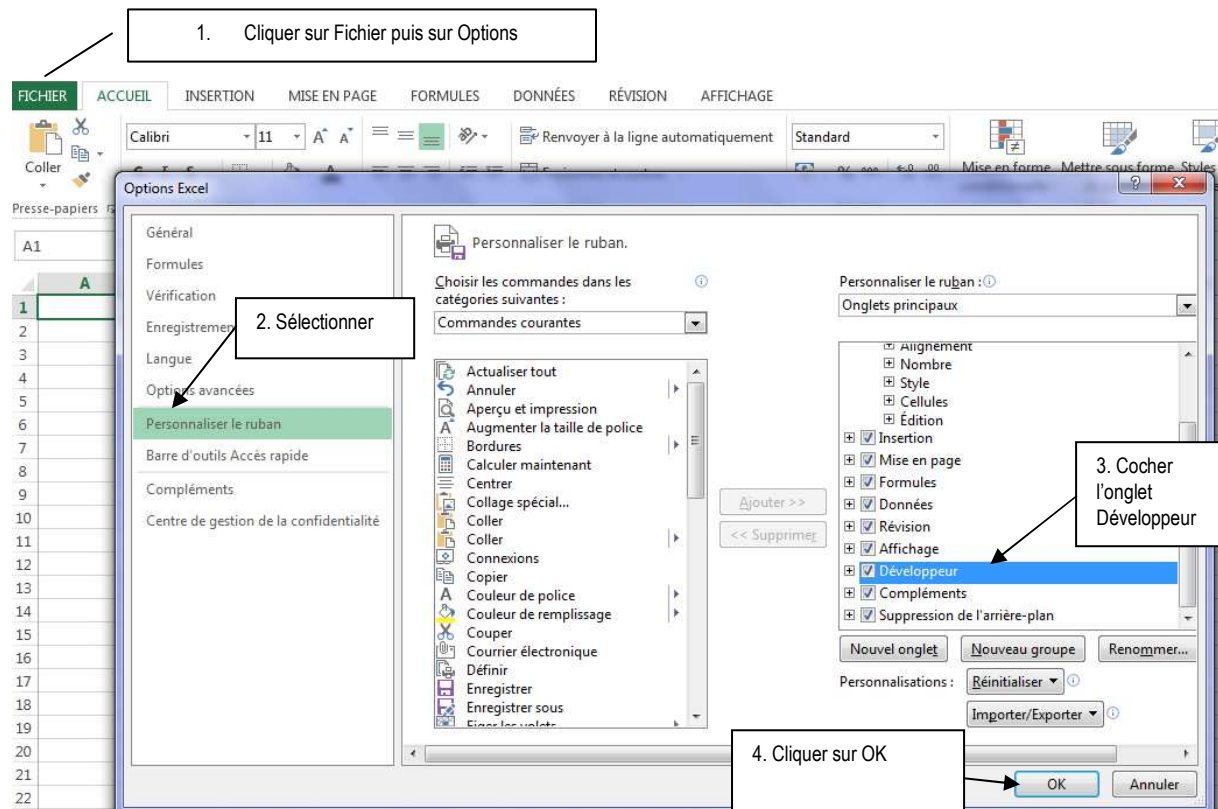
On suppose que l'élimination de l'alcool s'effectue de manière linéaire à raison de 0.1g par heure. Quelle formule doit-on saisir en I6 et recopier jusqu'en I29 afin de calculer le taux d'alcool restant dans l'organisme après chaque heure ?

Sélectionner la plage de cellules I5 :I29 et utiliser la mise en forme conditionnelle vue au 3) pour obtenir un affichage en rouge des cellules dans lesquelles le taux résiduel est supérieur à 0.5.

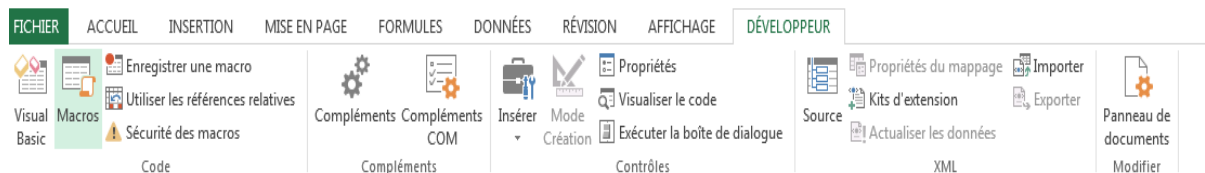
III. L'éditeur VBA

Pour commencer, il faut afficher l'onglet Développeur dans le ruban afin d'accéder à l'environnement de programmation.

Cliquer sur l'onglet fichier du ruban, puis sur la commande Options. Dans la fenêtre Options d'Excel, sélectionner Personnaliser le ruban, cocher la case Développeur puis valider. L'onglet Développeur apparaît dans le ruban.

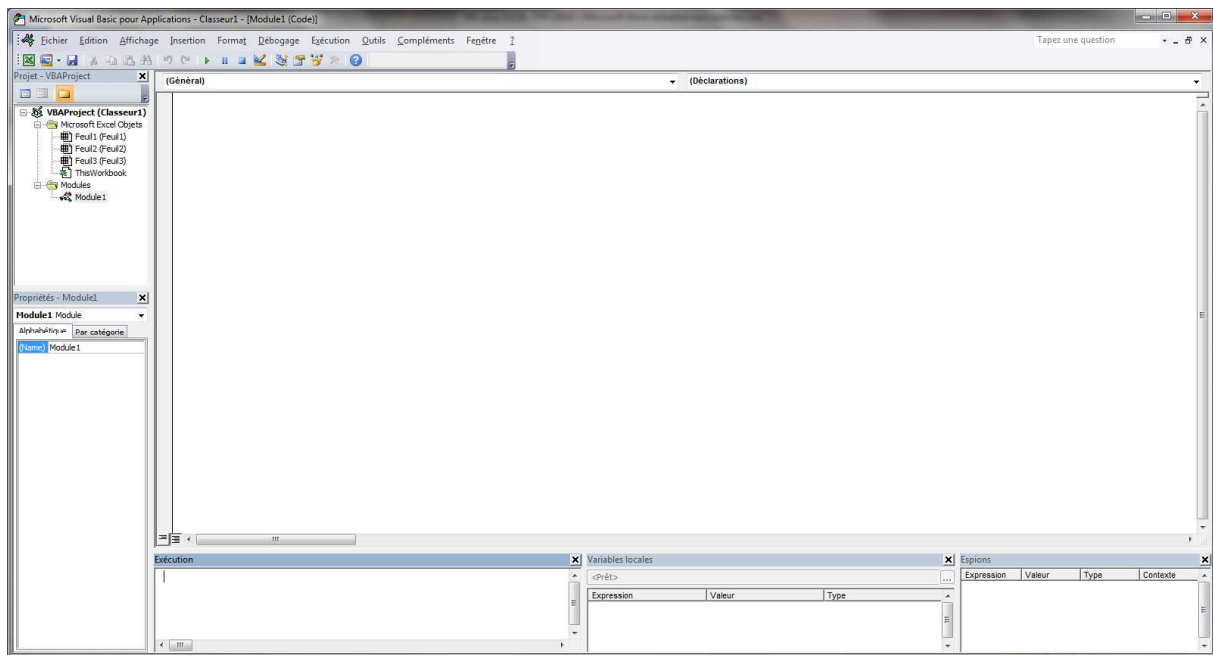


Cliquer alors sur l'onglet Développeur, on voit apparaître une nouvelle barre des tâches.



Cliquer sur l'icône Visual Basic pour accéder à l'éditeur, dans la nouvelle fenêtre qui s'est ouverte, ouvrir le menu **Insertion** du ruban et cliquer sur **Module**.

C'est dans ce module que l'on va saisir le code de nos différents programmes. Vous pouvez utiliser l'onglet Affichage et sélectionner les différentes options afin de personnaliser la fenêtre.



Remarque : Depuis un classeur Excel, on arrive à l'écran de l'éditeur VBA par la commande **Alt+F11**

Cette commande permet de passer du classeur Excel à l'éditeur VBA, puis de l'éditeur VBA au classeur Excel.

Remarque importante : Lorsqu'on souhaite enregistrer un classeur contenant des macros ou des procédures contenant du code Visual basic, il faut l'enregistrer en sélectionnant le type de classeur qui prend en charge les macros.

Le classeur enregistré portera l'extension **.xlsm**, alors qu'un classeur ne contenant ni macro ni code VBA possède l'extension **.xlsx**.

Paramétrage

Pour accéder aux paramètres de l'éditeur VBA, cliquer sur Outils puis Options

Règles de présentation

- Une seule instruction par ligne. Sinon, on peut mettre plusieurs instructions sur une ligne à condition de le séparer par le caractère :
- Une instruction peut déborder sur la ligne suivante, dans ce cas la ligne sauf la dernière se termine par le caractère _ (<espace><signe souligné>)
- Le VBA n'est pas sensible à la casse. Conseil : mieux vaut tout taper en minuscules

IV. Procédures et fonctions

Un projet correspond au fichier Excel et permet au programmeur d'y insérer son code.

A l'intérieur d'un projet, le code est stocké dans Module1 (insérer automatiquement par VBA). Vous pouvez modifier le nom de ce module en affichant la fenêtre Propriété (touche de fonction F4) dans l'éditeur.

Vous pouvez également insérer un module en faisant un clic droit dans l'Explorateur de projets puis en choisissant la commande Insertion, Module.

La touche de fonction F2, permet de voir toutes les bibliothèques de VBA.

Dans un module, le code s'écrit dans une procédure. Il existe trois types de procédures : les procédures de type **Sub**, les procédures de type **Function** et le procédure de type **Property**

La grande différence entre les procédures et les fonctions est que les procédures ne renvoient pas de résultat. En général, les procédures n'ont pas de paramètres.

Nous allons dans un premier temps, utiliser uniquement des procédures de type Function.

1) Les procédures de type Function

La syntaxe est la suivante :

```
[Private|Public|[Static] Function nom[(arguments)] [As type]

[Instructions]

End Function
```

- Private (paramètre optionnel) : indique que cette procédure est accessible uniquement aux procédures définies dans le même module que cette procédure.
- Public (paramètre optionnel) : indique que cette procédure est accessible à toutes les autres procédures de tous les autres modules.
- Static (paramètre optionnel) : indique que les variables locales à cette procédure gardent leur valeur d'un appel à l'autre.
- nom : indique le nom de la procédure function.
- arguments : indique la liste des variables et leur type à passer à function lors d'un appel de cette fonction.
- type : indique le type de retour de la fonction function.

Voici le détail du code d'une procédure de type Function qui renvoie le volume d'un cylindre de base un cercle de rayon r et de hauteur h .

```
Function volume_cylindre(ByVal r As Single, ByVal h As Single) As Single
'Hypothèse : r et h sont des nombres réels positifs
'Renvoie le volume d'un cylindre de base le cercle de rayon r et de
'hauteur h

Dim v As Single
v = WorksheetFunction.Pi * r * r * h

volume_cylindre = v

End Function
```

Pour saisir cette fonction dans VBA, vous devez ouvrir Visual Basic Editor, Insérer un module : Onglet Insertion/Module), placer le curseur dans la fenêtre de code et saisir le code de la fonction.

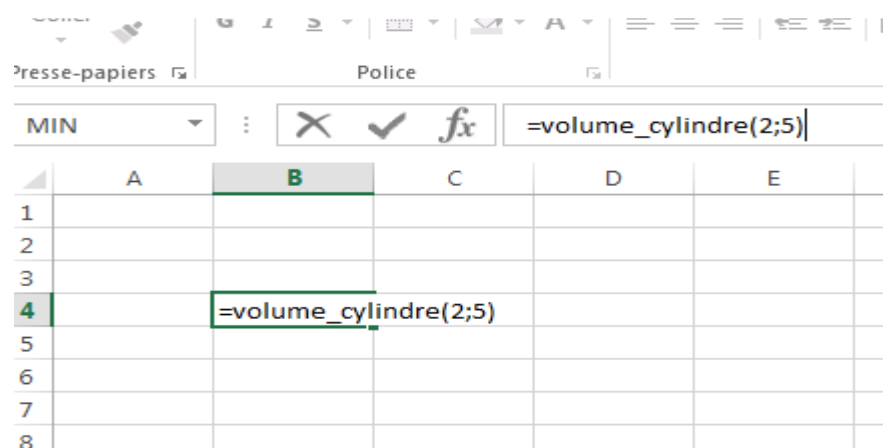
Le nom de cette fonction est volume_cylindre, deux arguments sont nécessaires à son utilisation, r déclaré comme un nombre réel (la variable r est de type Single), h défini comme un nombre réel (la variable h est de type Single). On précise dans le titre le type de résultat retourné par la fonction, ici un nombre réel.

En vert, deux lignes : ce sont des commentaires, qui ne seront pas exécutés, mais qui renseignent l'utilisateur et qui sont indispensables lorsque l'on veut comprendre ou utiliser une fonction dont on lit le code.

Dans le corps de la fonction, on a déclaré une variable v par l'instruction Dim en précisant le type de variable.

La ligne suivante représente le calcul du volume, la suivante l'affichage du résultat.

On retourne dans la feuille de calcul en cliquant sur l'icône Excel ou avec la commande Alt+F11. On peut maintenant utiliser directement la fonction créée par un appel dans une cellule quelconque de l'une des feuilles du classeur ouvert, comme indiqué dans l'écran ci-après.



Remarque : On notera le point-virgule comme séparateur entre les deux valeurs de r et de h respectivement égales à 2 et 5. Un appui sur Entrée affiche le résultat dans la cellule B4.

Enregistrer votre classeur, après lui avoir donné un nom et avoir sélectionné le type (classeur prenant en charge les macros).

Exercice 1 :

Dans le même module que celui créé précédemment, construire la fonction volume_sphere qui affiche le volume d'une sphère de rayon connu r .

Utiliser votre fonction dans une des pages du classeur.

2) Les variables

a) Les types de variables

Il existe différents types de variables acceptés par Excel : les valeurs numériques, les chaînes de caractères, les valeurs booléennes, les dates, les constantes, les tableaux (ou matrices) sont les types le plus utilisés.

Une variable se déclare avec l'instruction **Dim**, selon la syntaxe suivante : **Dim Nom As Type** où le nom de la variable obéit à certaines règles :

- Il doit commencer par une lettre,
- Il ne peut contenir plus de 255 caractères,
- Le point, le point d'exclamation, les espaces et les caractères @, &, \$ et # ne sont pas autorisés.
- Le nom ne doit pas être un nom réservé, c'est-à-dire un mot reconnu comme un élément du langage Visual Basic (nom de fonctions mathématiques ou autres ...).

Visual basic accepte les variables déclarées implicitement mais il est fortement recommandé de déclarer explicitement les variables avant de les utiliser.

Visual Basic permet de forcer la déclaration des variables avant leur utilisation (comme fait dans le programme page précédente) grâce à l'instruction Option Explicit.

Dans la fenêtre Visual Basic, dans la barre d'outils sélectionner Outils/Options/Editeur, puis cocher la case Déclaration de variables obligatoires.

b) Valeurs numériques

Type	Valeurs possibles	Mémoire occupée
Byte	Nombre entier compris entre 0 et 255	1 octet
Integer	Nombre entier compris entre -32768 et 32767	2 octets
Long	Nombre entier compris entre -2 147 483 648 et 2 147 483 647	4 octets
Single	Nombre décimal avec virgule flottante appartenant à $[-3,402823.10^{38}; -1,401298.10^{-45}] \cup [1,401298.10^{-45}; 3,402823.10^{38}]$	4 octets
Double	Nombre décimal donné en double précision appartenant à $[-1,79769373486223.10^{308}; -4,94065645841247.10^{-324}]$ ou à $[4,94065645841247.10^{-324}; 1,79769373486223.10^{308}]$	8 octets
Currency	Format monétaire, nombre à virgule fixe, avec 15 chiffres pour la partie entière et quatre chiffres pour la partie décimale, situé dans $[-922\,337\,203\,685\,477,5808; 922\,337\,203\,685\,477,5807]$	8 octets

c) Chaînes de caractères

Ces variables de type String permettent le stockage de données reconnues comme chaîne de caractères. Elles sont principalement utilisées pour stocker du texte. Leur saisie se fait obligatoirement en plaçant le contenu entre guillemets.

```
Dim Message As String  
Message = "Bonjour à tous"
```

d) Valeurs booléennes

Les variables de type Boolean servent à stocker le résultat de tests logiques. Elles renvoient la valeur True ou False et occupent deux octets en mémoire.

e) Dates

Les dates peuvent être affectées à des variables de type Date qui sont stockées sous la forme de nombres à virgule flottante et occupent 8 octets en mémoire.

Pour plus d'information sur les fonctions de date et d'heure, reportez-vous à l'aide de Visual Basic.

f) Les constantes

Les constantes permettent d'attribuer à une variable une valeur fixe qui ne pourra pas être modifiée durant le programme.

La déclaration d'une constante se fait de la manière suivante :

```
Const TVA As Single = 19.6
```

Une variable constante n'est accessible qu'à l'intérieur de la procédure à laquelle elle appartient.

g) Les tableaux, variables indicées (ou matrice)

Les tableaux sont des variables de type Array. L'instruction suivante déclare un tableau nommé Tab1 de 7 nombres réels et un tableau nommé Tab2 de 7 chaînes de caractères :

```
Dim Tab1(7) As Single: Dim Tab2(7) As String
```

Les indices varient ici de 0 à 6.

L'instruction Tab1(3) fait donc référence à la quatrième valeur rangée dans le tableau nommé Tab1.

Il est possible de créer des tableaux à plusieurs dimensions, l'instruction suivante déclare un tableau nommé Tab3 à deux dimensions de 3 lignes et 5 colonnes contenant des nombres entiers :

```
Dim Tab3(1 To 3, 1 To 5) As Integer
```

L'instruction

```
Tab3(2, 3)
```

fait référence au nombre contenu dans la ligne 2, colonne 3 du tableau.

Dimensionnement dynamique

Il est possible de ne pas donner en début de module la dimension exacte du tableau et donner cette dimension au cours du module lorsque celle-ci est alors connue.

```
Dim Table() As Integer
```

Cette instruction déclare la création d'un tableau de nombres entiers, Table dont la dimension est inconnue.

```
ReDim Table(15)
```

fixe au cours du programme la taille de Table à 15 éléments.

h) Portée et durée de vie des variables

Les variables et les constantes ont une portée. La portée désigne son accessibilité pour les procédures ou les modules du projet.

Elles peuvent être accessibles à une procédure, à l'ensemble des procédures d'un module ou à l'ensemble des modules du projet.

Si l'on souhaite qu'une variable ou une constante soit accessible à **l'ensemble des procédures d'un module** (on dit qu'elle a une **portée privée**), elle doit être déclarée dans la section Déclarations du module, c'est-à-dire à l'extérieur de toute procédure avec l'instruction Private selon la syntaxe :

```
Private mavar As Single
```

Elle est de **portée public** lorsqu'elle est accessible à **l'ensemble des procédures du projet**, quel que soit le module de stockage. On doit pour cela la déclarer dans la section déclaration du module, à l'aide de l'instruction Public selon la syntaxe :

3) Les structures de contrôle : boucles et tests

a) Boucle Tant que (ou Do While)

Il existe différentes manières d'utiliser cette boucle, résumées dans le tableau ci-dessous.

Syntaxe	Commentaires
Do While condition Instructions Loop	On continue la boucle tant que la condition est vraie
Do Instructions Loop While condition	On recommence la boucle tant que la condition est vraie
Do Until condition Instructions Loop	On continue la boucle jusqu'à ce que la condition soit vraie
Do Instructions Loop Until condition	On recommence la boucle jusqu'à ce que la condition soit vraie
While condition Instructions Wend	On continue la boucle tant que la condition est vraie

Remarque : Lorsque la condition est en début de boucle, il peut y avoir 0 tour de boucle et lorsque la condition est en fin de boucle il y a au moins un tour de boucle.

b) Boucle For

On utilise l'instruction For lorsque le nombre de tours dans la boucle est connu *a priori*.

Syntaxe	Commentaires
For i = 1 To n Instructions Next i	Pour i allant de 1 jusqu'à n (sans dépasser n) exécuter Instructions Fin du pour i
For k = 1 To n Step 2 Instructions Next k	Pour k allant de 1 jusqu'à n de 2 en 2 (sans dépasser n) Exécuter Instructions Fin de pour k
For Each élément In Collection Instructions Next élément	Permet de généraliser un traitement à tous les objets d'une collection

c) Les tests

Syntaxe	Commentaires
<pre>If condition Then Instructions_1 Else Instructions_2 EndIf</pre>	<p>Si la condition est réalisée, alors Exécuter les instructions 1</p> <p>Sinon Exécuter les instructions 2</p> <p>Fin du si</p>
<pre>If condition Then Instructions EndIf</pre>	<p>Si la condition est réalisée, alors Exécuter les instructions</p> <p>Fin du si</p>
<pre>If condition1 Then Instructions_1 Elseif condition2 Then Instructions_2 Elseif condition3 Then Instructions_3 ... Else Instructions EndIf</pre>	<p>Si la condition 1 est réalisée, alors Exécuter les instructions 1</p> <p>Sinon si la condition 2 est réalisée, alors Exécuter les instructions 2</p> <p>Sinon si la condition 3 est réalisée, alors Exécuter les instructions 3</p> <p>....</p> <p>Sinon Exécuter les instructions</p> <p>Fin du si</p>

d) La structure Select Case

Syntaxe	Commentaires
<pre>Select Case Expression Case valeur 1 Instructions 1 Case valeur 2 Instructions 2 Case valeur 3 Instructions 3 ... Case Else Instructions End Select</pre>	<p>Cette instruction permet d'envisager plusieurs valeurs pour une même expression et de déterminer les instructions à exécuter pour chacun des cas.</p>

e) Opérateurs logiques

Opérateur	Commentaires
Or	Renvoie True si l'une au moins des conditions est vérifiée, False sinon
And	Renvoie True si toutes les conditions sont vérifiées, False sinon
Xor	C'est le ou exclusif. Renvoie True si l'une ou l'autre des conditions est respectée
Not	Nie l'expression devant laquelle il est situé.

Exercice 2

Dans un module du classeur ouvert, saisir la fonction suivante :

```
Function Addition(ByVal a As Integer, ByVal b As Integer) As Integer
    'Renvoie la somme a+b des deux entiers a et b
    Addition = a + b
End Function
```

Sans saisir la seconde fonction suivante, donner le résultat qui sera retourné par celle-ci.

```
Function essai() As Integer
    Dim a As Integer, b As Integer
    a = 3: b = 4
    a = Addition(a, b) + b: b = Addition(a, b) + b: a = Addition(a + b, b)
    essai = a
End Function
```

Contrôler votre réponse en saisissant la fonction dans le même module que celui utilisé pour la fonction Addition.

Exercice 3

Dans un nouveau module, écrire une fonction qui affiche le plus grand de deux nombres réels.

En utilisant la fonction précédente, écrire une fonction qui, étant donnés trois nombres réels a, b et c, renvoie le plus grand des trois.

Exercice 4

Dans cet exercice, un rectangle est représenté par les coordonnées (x_1, y_1) et (x_2, y_2) de ses coins opposés, avec $x_1 < x_2$ et $y_1 < y_2$.

La fonction booléenne **estDansRectangle** a pour paramètres les quatre coordonnées d'un rectangle R et les coordonnées d'un point P et elle renvoie la valeur vraie si et seulement si P appartient au rectangle R (intérieur ou contour).

- 1) Ecrire la fonction **estDansRectangle** en utilisant que des instructions conditionnelles (If Then ...) sans utiliser d'opérateur booléen.
- 2) Ecrire la fonction **estDansRectangle** en utilisant les opérateurs \geq , And
- 3) Ecrire la fonction **estDansRectangle** en utilisant les opérateurs \geq , Or, Not.

Exercice 5

- 1) Ecrire une fonction qui calcule de manière itérative la somme $1 + x + x^2 + \dots + x^n$, où n est un entier naturel et x un nombre réel.
- 2) Ecrire la fonction précédente en utilisant une méthode récursive.

Exercice 6

Ecrire une fonction qui calcule de manière récursive le PGCD de deux entiers a et b .

On utilisera le résultat suivant :

$$\begin{cases} \text{PGCD}(a, 0) = a \\ \text{PGCD}(a, b) = \text{PGCD}(b, a \bmod b) \end{cases}$$

Exercice 7

Le miroir d'une chaîne de caractères S est la chaîne formée des mêmes caractères que S , mais dans l'ordre inverse. Par exemple, « dcba » est le miroir de « abcd ».

- 1) Ecrire une définition itérative de la fonction *miroir*.
- 2) Ecrire une définition récursive de la fonction *miroir*.
- 3) Ecrire une définition itérative de la fonction *miroir_nombre* qui, étant donné un entier naturel n , renvoie le miroir de n .

Remarque : On utilisera l'aide d'Excel pour recueillir les informations nécessaires au traitement des chaînes de caractères, en particulier pour les instructions : Mid, Len et Left

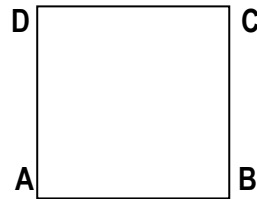
Exercice 8

Simulation de variables aléatoires. On utilisera le même module pour chacune des fonctions qui suivent.

- 1) Ecrire une fonction qui simule le comportement d'une variable aléatoire X qui suit la loi de Bernoulli de paramètre p où p est un nombre réel de $]0,1[$.
- 2) Ecrire une fonction qui simule le comportement d'une variable aléatoire Y qui suit la loi Binomiale de paramètres n et p , où n est un entier naturel strictement positif et p est un nombre réel de $]0,1[$.

Remarque : On utilisera l'aide d'Excel pour la fonction Rnd qui génère des nombres pseudo-aléatoires.

Exercice 9



Une fourmi se déplace sur les arêtes d'un carré. Elle part de A et choisit sa direction au hasard, elle met une minute pour parcourir une arête.

Lorsqu'elle arrive sur un nouveau sommet, elle choisit à chaque fois sa direction au hasard.

On appelle X la variable aléatoire qui prend pour valeur le temps mis par la fourmi pour atteindre le sommet C.

- 1) Construire une fonction qui simule la variable aléatoire X .
- 2) Utiliser votre fonction pour estimer $E(X)$ et $V(X)$.

Exercice 10

La chasse aux canards.

Dix chasseurs, tous excellents, ils ne ratent jamais leur cible, tirent simultanément sur dix canards sauvages, chaque chasseur choisissant son canard au hasard. On appelle X la variable aléatoire qui prend pour valeur le nombre de canards survivants à l'issue d'une salve. On cherche à estimer l'espérance mathématique de X .

- 1) En utilisant l'algorithme suivant, construire une fonction qui permet de répondre à la question

Créer une liste de 10 éléments qui seront tous initialisés à 1 (ce sont les canards vivants avant le tir)

Pour j allant de 0 à 9

 Tirer un numéro x au hasard dans $\{0, 1, 2, \dots, 9\}$ (le chasseur n° j choisit son canard)

 Remplacer le numéro x de la liste des canards par 0 (le canard n° x a été tué)

Fin du pour j

Calculer la somme s des termes de la liste (on compte les survivants)

Afficher la valeur de la somme s .

- 2) Après avoir saisi votre fonction dans un module, la faire exécuter 1000 fois dans une feuille de calcul puis, calculer la moyenne des résultats obtenus. En déduire une estimation de l'espérance mathématique.

Exercice 11

Les nombres persistants.

On considère un nombre entier naturel n .

Afin de calculer la persistance de n , on calcule le produit de ses chiffres puis on recommence jusqu'à obtention d'un nombre à un seul chiffre. La persistance de n est le nombre d'étapes pour obtenir un nombre à un seul chiffre.

Exemple : La persistance de 25 est 2 ; $25 \rightarrow 10 \rightarrow 0$

La persistance de 729 est 3 ; $729 \rightarrow 126 \rightarrow 12 \rightarrow 2$

- 1) Ecrire une fonction (pers(n)) qui affiche la persistance d'un entier n .
- 2) Ecrire une fonction qui affiche le plus petit entier de persistance 4.
- 3) Ecrire une fonction qui, utilisée plusieurs fois, permet l'affichage des 10 premiers entiers de persistance 4.

Exercice 12 (Cryptographie)

Introduction

Depuis l'invention de l'écriture et les premières guerres, il a toujours été important de pouvoir transmettre des messages protégés, c'est-à-dire des messages qui ne puissent être compris par l'ennemi même en cas d'interception.

Le chiffrement est la capacité à rendre un message illisible et le déchiffrement, la capacité à transformer le message illisible en un message clair.

Ne pas confondre le mot chiffrement et le mot codage. Le codage est une méthode permettant de passer d'une représentation des données vers une autre. Par exemple, dans un ordinateur chaque lettre est représentée par un nombre (code ASCII).

Le chiffrement affine

Chaque caractère utilisé dans notre écriture possède un code ASCII. Voici le code ASCII des 26 lettres de notre alphabet écrites en minuscules.

a	b	c	d	...	x	y	z
97	98	99	100	...	120	121	122

Pour simplifier ici, nous n'utiliserons que des lettres minuscules non accentuées et nous n'utiliserons pas d'espace dans les messages que nous allons crypter.

On appelle rang d'une lettre sa position dans l'alphabet en comptant à partir de zéro.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

La méthode de chiffrement affine.

On substitue à chaque lettre composant le message une autre lettre suivant le principe suivant :

La lettre de rang x est chiffrée par la lettre de rang, le reste de la division euclidienne de $ax + b$ par 26, où a et b sont des entiers inférieurs à 26. (Le nombre a doit être premier avec 26).

Le couple (a, b) est la clé de chiffrement.

Voici un exemple de chiffrement affine, la clé de chiffrement utilisée ici est (3,7)

Lettre	e	s	s	a	i
Rang	4	18	18	0	8
Rang chiffré	19	9	9	7	5
Lettre chiffrée	t	j	j	h	f

Ainsi le mot **essai** est chiffré **tjjhf**.

Fonction de cryptage

Construire le code VBA d'une fonction de cryptage de clé (a, b) , la première ligne de code sera :

```
Function cryptage(ByVal phrase As String, ByVal a As Integer, ByVal b As Integer) As String
```

L'appel de la fonction se fera dans une cellule de la feuille de calcul comme suit :

=cryptage("essai";3;7)

La réponse affichée sera donc **tjjhf**.

Déchiffrement affine

On considère un chiffrement affine de clé (a, b) . Le déchiffrement affine s'effectue avec un chiffrement affine de clé (a', b') vérifiant les deux conditions : le produit $a \times a'$ est congru à 1 modulo 26 et b' est le reste de la division du produit $a' \times (26 - b)$ par 26.

Trouver la clé de déchiffrement pour la clé de chiffrement (3, 7).

Vérifier en utilisant votre fonction de cryptage réalisée précédemment avec le mot **tjjhf**.

Ecrire une fonction de déchiffrement connaissant la clé de chiffrement, la première ligne de code sera :

```
Function decryptage(ByVal phrase As String, ByVal a As Integer, ByVal b As Integer) As String
```

Où a et b correspondent à la clé de chiffrement.

Remarques : Voici une liste de quelques fonctions VBA qui pourront être utilisées dans vos fonctions de cryptage et de décryptage :

Len : donne le nombre de caractères d'une chaîne

Exemple : Len (essai) retourne 5

Mid : extrait un groupe de lettre dans une chaîne de caractères

Exemple : Mid(essai,3,2) retourne sa, le groupe de 2 lettres à partir de la 3^{ième} lettre du mot essai

Asc : retourne le code ASCII d'un caractère

Exemple : Asc(s) retourne 115

Chr : retourne le caractère associé au nombre

Exemple : Chr(115) retourne s

Mod : donne le reste de la division de deux entiers

Exemple : 52 Mod 9 retourne 7