

4) La procédure Sub

L'effet d'une procédure Sub n'est pas de renvoyer une valeur comme pour une fonction, mais de modifier l'environnement.

Une procédure peut avoir, ou non, des paramètres. En VBA, les seules procédures pouvant être appelées par l'utilisateur à partir du tableur sont les procédures sans paramètre (qui sont appelées **macros**).

Pour faire exécuter une procédure avec paramètres, il faut l'inclure à l'intérieur d'une macro et transmettre les valeurs des paramètres en utilisant des mécanismes d'entrée.

Procédure Sub sans paramètre

Exemple : Dans l'éditeur VBA, créer un nouveau module et saisir le code ci-dessus.

```
Sub Coloriercell()  
    'Modification : colorie l'intérieur de trois cellules de la feuille  
    Cells(2, 4).Interior.Color = RGB(0, 0, 255)  
    Cells(3, 3).Interior.Color = RGB(0, 255, 0)  
    Cells(4, 2).Interior.Color = RGB(255, 0, 0)  
  
End Sub  
|
```

Revenir à la page de calcul d'Excel, cliquer sur l'onglet Développeur et dans le ruban, fenêtre code, sélectionner Macros. Le nom de la macro que l'on vient de créer apparaît, cliquer sur exécuter.

Voici quelques commentaires concernant la ligne de code n°1 :

```
Cells(2, 4).Interior.Color = RGB(0, 0, 255)
```

L'instruction

```
Cells(2, 4).Interior.Color
```

fait référence à la couleur de l'intérieur de la cellule de la deuxième ligne et quatrième colonne (cellule D2) de la feuille active du classeur.

L'instruction

```
RGB(0, 0, 255)
```

utilise la fonction RGB qui possède trois arguments et qui retourne un entier représentant la couleur utilisée. Cette fonction est importante et voici ci-dessous les compléments utiles s'y rapportant.

RGB(*Red, Green, Blue*) donne une couleur définie par sa quantité de rouge, de vert et de bleu, par la donnée des trois nombres qui sont des entiers compris entre 0 et 255. Voici un tableau qui donne les principales couleurs couramment utilisées.

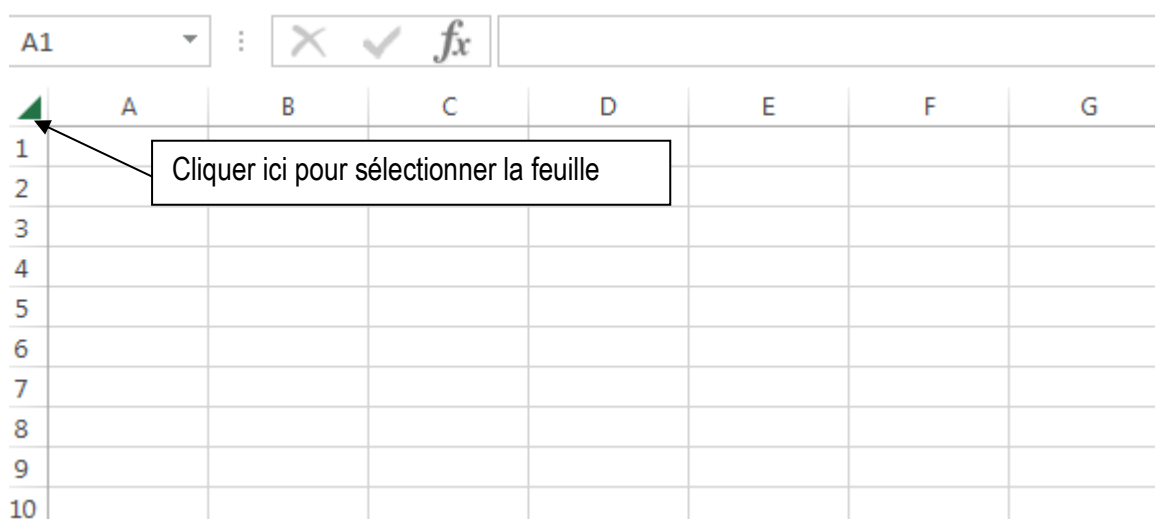
Couleur	Composante de rouge	Composante de vert	Composante de bleu
Noir	0	0	0
Bleu	0	0	255
Vert	0	255	0
Cyan	0	255	255
Rouge	255	0	0
Magenta	255	0	255
Jaune	255	255	0
Blanc	255	255	255

Une fois la procédure **Coloriercell()** a été exécutée, on peut se demander comment revenir à l'état précédent, c'est à dire à une feuille dont aucune cellule n'a été coloriée. Il est nécessaire de construire une nouvelle macro pour faire ce travail. Voici le détail de cette macro appelée **supprimerCouleurFeuille()**.

Saisir le code ci-contre de cette macro dans le même module que précédemment et procéder comme ci-dessus pour l'exécuter.

Pour réaliser les exercices suivants et obtenir les figures sous la forme demandée, il faut commencer par mettre les cellules de la feuille de calcul sous forme de petits carrés.

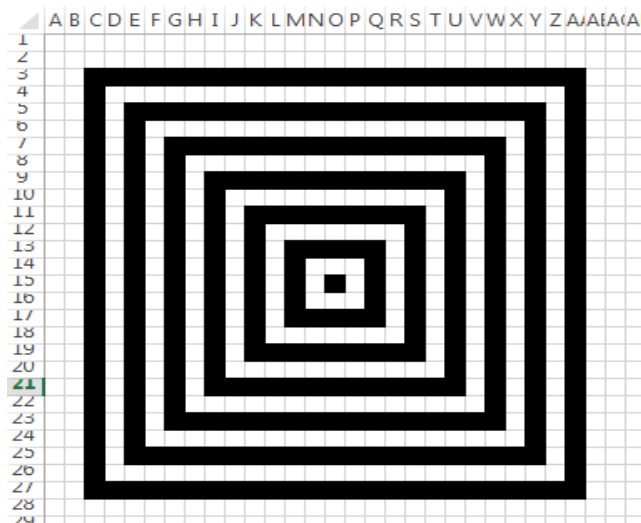
Sélectionner l'onglet Accueil dans le ruban, sélectionner la totalité de la feuille de calcul en cliquant comme indiqué dans la figure ci-dessous,



puis dans le groupe de commandes Cellules,

Exercice 14

On souhaite créer la figure ci-contre formée de 7 carrés emboîtés, centrés sur la cellule O15 de la feuille.



On pourra définir une macro appelée **emboîtés()**, qui appellera 7 fois une procédure appelée **carré(k)** où k est un entier défini dans la macro appelante.

Remarque : La procédure carré est une procédure avec paramètre, de paramètre k . Il n'est pas possible de l'appeler directement depuis la feuille de calcul via la commande Macros. Cette procédure est cependant exécutable lorsqu'elle est appelée dans la macro emboîtés().

Créer une Macro avec l'enregistreur de Macros

On souhaite réaliser une Macro qui va mettre en forme le titre d'un document comme indiqué dans le tableau ci-dessous : la hauteur de ligne est fixée à 30, la police de caractères choisie est Arial, taille 12, caractères Gras, le fond des cellules est colorié en jaune, une bordure épaisse entoure le titre.

Dans une nouvelle feuille de calcul, saisir le titre correspondant à l'état initial.

	A	B	C	D	E	F	G	H	I	J
1	Notes	Maths	Physique	Chimie	Méca	Labo	Anglais	Français	Ora I1	Ora I2

Dans une nouvelle feuille de calcul, saisir le titre correspondant à l'état initial.

Sélectionner la plage A1 : J1.

Cliquer dans l'onglet Développeur, dans le groupe Code sur Enregistrer une macro. Renseigner la boîte de dialogue qui s'est ouverte comme ci-contre.

On clique dans cette boîte sur OK.

Dans le groupe Police de l'onglet Accueil, on choisit la police Arial, la taille 12 et on clique sur l'icône G (caractères gras). Dans ce même groupe, on clique sur l'icône couleur de remplissage, on choisit la couleur Jaune. Il reste à ouvrir le menu Bordures dans ce même groupe et y choisir Bordures épaisses en encadré.

On choisit l'onglet Accueil, groupe Cellules, on clique sur Format, puis sur Hauteur de ligne, on place 30 dans la boîte de dialogue puis on clique sur OK.

On clique alors sur Arrêter l'enregistrement dans le groupe Code, onglet Développeur.

Comment utiliser cette macro ? : La macro est enregistrée, il reste à l'exécuter dans une nouvelle feuille de calcul sur un autre titre qui aura été créé et sélectionné. Créer un nouveau titre en renseignant quatre cellules par exemple comme suit :

Etape 1 : On crée le titre et on le sélectionne

	A	B	C	D	E
1					
2					
3					
4			Janvier	Février	Mars
5					
6					

Etape 2 : On fait exécuter la macro, soit en utilisant le raccourci clavier choisi : Ctrl t, ou bien en cliquant sur l'icône Macro et en cliquant sur la touche Exécuter lorsque la macro a été sélectionnée dans la liste (ici, il n'y en a qu'une qui est la macro Titre).

Etape 3 : Le résultat apparaît dans la feuille

	A	B	C	D	E
1					
2					
3					
4			Janvier	Février	Mars
5					
6					

On peut maintenant visualiser le code de cette macro, enregistré automatiquement pendant sa création, en sélectionnant Visual Basic dans la feuille de calcul.

```

Option Explicit
Sub Titre ()
'
' Titre Macro
' Met en forme le titre sélectionné : Police Arial, taille 12, caractère gras, fond jaune, bord
'
' Touche de raccourci du clavier: Ctrl+t
'
With Selection.Font
.Name = "Calibri"
.Size = 12
.Strikethrough = False
.Superscript = False
.Subscript = False
.OutlineFont = False
.Shadow = False
.Underline = xlUnderlineStyleNone
.ThemeColor = xlThemeColorLight1
.TintAndShade = 0
.ThemeFont = xlThemeFontMinor
End With
With Selection.Font
.Name = "Arial"
.Size = 12
.Strikethrough = False
.Superscript = False
.Subscript = False
.OutlineFont = False
.Shadow = False

```

Après avoir examiné ce code, on peut enlever les parties inutiles et renseigner différemment les paragraphes.

Remarque : Ce code généré par Excel peut bien sûr être récupéré et inséré dans un autre projet, on voit ici toute la puissance de la méthode, qui permet sans avoir à apprendre les instructions, de construire des procédures performantes.

Créer une procédure Sub avec paramètres

Un exemple

Examiner la procédure ci-contre qui calcule et affiche le pgcd et le ppcm de deux entiers a et b .

```
Sub pgcd_ppcm(a As Long, b As Long)
    'On crée les variables utiles à la réalisation du programme
    Dim c As Long, d As Long, e As Long
    'Modification : affiche la valeur de l'entier a en cellule B1
    Cells(1, 2).Value = a
    'Modification : affiche la valeur de l'entier b en cellule B2
    Cells(2, 2).Value = b
    'Calcule et stocke le produit a*b dans d
    d = a * b
    'Boucle de calcul de a et b
    Do Until b = 0
        c = b
        b = a Mod b
        a = c
    Loop
    'Calcul du ppcm de a et b
    e = d / a
    'Modification : affichage du pgcd de a et b en cellule B3
    Cells(3, 2).Value = a
    'Modification : affichage du ppcm de a et b en cellule B4
    Cells(4, 2).Value = e
End Sub
```

Il n'est pas possible d'appeler directement cette procédure depuis une feuille de calcul comme on le fait pour une fonction ou une macro classique.

Deux possibilités s'offrent à nous :

- Méthode 1 : On appelle cette procédure dans une autre procédure sans paramètre.
- Méthode 2 : On utilise des boîtes de dialogue pour la saisie des valeurs de a et b .

Méthode 1, on crée une procédure **appel()** dans laquelle les deux valeurs données à a et b seront transmises à la procédure **pgcd_ppcm**.

```
Sub appel()
    pgcd_ppcm 462, 420
End Sub
```

On a procédé à la mise en page au préalable en remplissant la colonne A. On appelle depuis la feuille de calcul la macro **appel()** qui lancera la procédure **pgcd_ppcm** et remplira automatiquement la colonne B.

	A	B
1	a=	462
2	b=	420
3	PGCD=	42
4	PPCM=	4620
5		

On voit bien sûr tout de suite les inconvénients de cette méthode, il faut modifier la procédure d'appel lorsqu'on change les valeurs de *a* et *b*.

Remarque : On notera que la procédure `pgcd_ppcm` est utilisée sans parenthèses.

Cette méthode pourra cependant être utilisée lorsqu'on veut tester une procédure Sub avec paramètres.

Méthode 2 (Utilisation des boîtes de dialogues d'entrée et sortie des données)

Méthode **InputBox**, dont voici la syntaxe

Appllication.InputBox(prompt, title, default, left, top, helpFile, helpContextID, type)

prompt : C'est le message d'invite affiché dans la boîte (sous forme de chaîne de caractères)

title : Est le titre (la valeur par défaut est « Entrée »)

default : Le cadre de saisie peut être initialisé avec Default

left, top : La boîte de dialogue est positionnée dans l'écran aux coordonnées left et top (par défaut elle est positionnée au centre de l'écran)

helpFile, helpContextID : Les arguments servent à associer des fichiers d'aide à l'utilisation de la boîte de dialogue.

type : Précise le type de données qui sera renvoyé par la méthode InputBox comme indiqué dans le tableau ci-dessous.

Valeur	Type de données renvoyé par InputBox
0	Formule
1	Valeur numérique
2	Chaîne de caractères
4	Valeur booléenne (False ou True)
8	Référence de cellules
16	Valeur d'erreur
64	Tableau de valeurs

Pour l'affichage de résultats, on utilise la boîte de message **MsgBox** dont la syntaxe est :

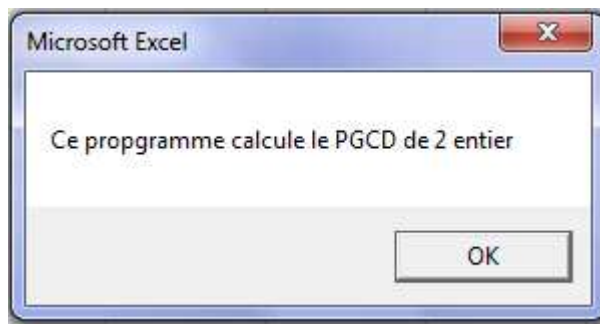
MsgBox(Prompt, Buttons, Title)

MsgBox permet d'afficher une boîte de dialogue dans laquelle se trouvent un message et des boutons de commande. Le seul paramètre obligatoire ici est prompt.

Pour avoir des informations supplémentaires sur l'option Buttons, utiliser l'aide de Visual Basic.

Voici les syntaxes et les affichages correspondants de trois utilisations différentes de MsgBox

MsgBox ("Ce programme calcule le PGCD de 2 entiers")



MsgBox "Le PGCD de a et b est " & a, Title:="Réponse pgcd"



MsgBox "Autre recherche ?", vbYesNo + vbQuestion, "Recherche"



Voici le détail du code du programme de calcul du pgcd et du ppcm de deux entiers en utilisant les boîtes de dialogue pour l'entrée des données et l'affichage des résultats.

```
Sub pgcd_ppcm2()  
    'Modif : ce programme demande de saisir deux entiers a et b puis calcule et affiche via _  
    des boîtes de dialogue, le pgcd et le ppcm de ces deux entiers  
    'On crée les variables utiles à la réalisation du programme  
    Dim a As Long, b As Long, c As Long, d As Long, e As Long  
    'Saisie de la valeur de a  
    a = Application.InputBox("Entrer la valeur de l'entier a ", "PGCD et PPCM de 2 entiers", Type:=1)  
    'Saisie de la valeur de b  
    b = Application.InputBox("Entrer la valeur de l'entier b ", "PGCD et PPCM de 2 entiers", Type:=1)  
    'On calcule et stocke le produit a*b  
    d = a * b  
    Do Until b = 0  
        c = b  
        b = a Mod b  
        a = c  
    Loop  
  
    'Calcule du PPCM de a et b  
    e = d / a  
    'Affichage du PGCD de a et b par la boîte de message  
    MsgBox "Le PGCD de a et b est " & a, Title:="Réponse PGCD"  
    'Affichage du PPCM de a et b par la boîte de message  
    MsgBox "Le PPCM de a et b est " & e, Title:="Réponse PPCM"  
End Sub
```

Exercise 15

Ecrire une procédure qui permet de résoudre une équation du second degré.

B	C	D	E	F	G
Résolution de l'équation de second degré					
Entrée des donnée					
	a	1			
	b	1			
	c	-2			
Les solutions					
Il y a deux racines réelles distinctes					
	x_1	1,00			
	x_2	-2,00			

Exercice 16

Un enfant collectionne des images de son équipe de football préférée, images qu'il trouve dans des boîtes de céréales. Il y a 11 images différentes à gagner.

Combien faut-il acheter de boîtes de céréales en moyenne pour avoir la collection complète des joueurs ? (on, suppose bien entendu que l'on a la même probabilité d'obtenir chacune des images dans chaque paquet de céréales).

Exercice 17 (Fluctuation d'échantillonnage et simulation)

La feuille 1 du classeur Fluctuation d'échantillonnage contient dans la plage A100 :R299, 3600 mesures effectuées lors de la fabrication de composants électroniques d'une unité de production. L'objectif de ce TP est l'étude de la variable \bar{X} qui prend pour valeur la moyenne d'échantillons de taille n tirés au hasard dans l'ensemble des mesures précédentes.

Partie 1 : Tirage de 400 échantillons de taille n dans la population précédente

Vous devez réaliser une procédure codée en VBA qui va tirer 400 échantillons de taille n dans les données précédente de la plage A100 :R299, calculer la moyenne de chaque échantillon de taille n et afficher cette moyenne dans un tableau qui occupera la plage A1 :J40 de la feuille 1 et pourra donc ensuite être modifiée à souhait.

Dans la feuille 1 du classeur placer la taille n de l'échantillon en cellule M2. (On peut commencer avec la valeur 50 par exemple)

Les 400 cellules de la plage A1 :J40 vont recevoir les 400 moyennes des échantillons de taille n .

Ouvrir un module de code dans Visual Basic et coder le programme demandé n suivant l'algorithme ci-dessous.

Algorithme	Commentaires
Lire la valeur de n dans la cellule M2	n désigne la taille de l'échantillon
Pour i allant de 1 à 40	i désigne le rang de la ligne du tableau des résultats
Pour j allant de 1 à 10	j désigne le rang de la colonne du tableau des résultats
Initialiser la variable s à 0	s est la variable qui va contenir la somme des mesures
Pour k allant de 1 à n	
Choisir un nombre entier a au hasard entre 100 et 299	a est le rang de la ligne choisie au hasard
Choisir un nombre entier b au hasard entre 1 et 18	b est le rang de la colonne choisie au hasard
Remplacer s par la somme de s et la valeur de la donnée qui occupe la ligne a colonne b	
Fin de la boucle pour k	
Calculer la moyenne m des mesures	
Arrondir m à 10^{-1} près.	
Afficher la moyenne m de l'échantillon dans la cellule ligne i et colonne j du tableau des résultats	On copie la moyenne de l'échantillon dans la cellule (i, j)

	du tableau des résultats
Fin pour la boucle j	
Fin pour la boucle i	

Une fois la procédure saisie, on automatisera son lancement par la création d'un bouton sur lequel on portera le texte **Tirage des échantillons**.

L'appui sur ce bouton déclenchera donc un nouveau tirage de 400 échantillons de taille n et affichera les 400 moyennes dans la plage A1 :J40.

Partie 2 : Analyse des données des 400 échantillons.

- 1) Calculer la valeur de l'écart type σ' des moyennes des 400 échantillons.
Quelle formule relie l'écart-type σ de la population des 3600 données et l'écart-type de la variable \bar{X} ? (ce résultat provient du cours de probabilité).
Vérifier cette formule en utilisant 5 fois le programme précédent, à chaque fois pour une valeur différente de n .
On notera les résultats des 5 écarts-types et les 5 valeurs de n dans un tableau.
- 2) On fixera, pour cette question et la partie 3 la valeur de n à 100.
Résumer la série des 400 valeurs dans un tableau de 10 classes environ, on centrera le tableau sur la valeur moyenne μ des 3600 données.
Afficher l'histogramme lié au tableau précédent et comparer par superposition des graphiques, cette distribution à celle issue de la loi normale de moyenne μ et d'écart-type $\frac{\sigma}{\sqrt{n}}$ (on suppose que σ est connu et égal à 8).

Partie 3 : Intervalle de confiance de la moyenne

Le cours relatif à l'estimation des paramètres donne un résultat concernant l'intervalle de confiance de la moyenne μ d'une variable aléatoire en utilisant la moyenne \bar{x}_e d'un échantillon de taille n avec un niveau de confiance $1 - \alpha$ ou de risque α , cet intervalle peut s'écrire :

$$\left[\bar{x}_e - t_{\alpha/2} \frac{\sigma}{\sqrt{n}} ; \bar{x}_e + t_{\alpha/2} \frac{\sigma}{\sqrt{n}} \right]$$

Signification des paramètres de l'intervalle de confiance :

\bar{x}_e	Moyenne de l'échantillon prélevé
$t_{\alpha/2}$	Valeur réelle liée au risque α qui vérifie la condition $P(Z \leq t_{\alpha/2}) = 1 - \alpha$ où Z suit la loi normale centrée réduite.
σ	Ecart-type de la population dans laquelle est prélevée l'échantillon
n	Taille de l'échantillon

Avant de tirer un échantillon de taille n dans la population, la probabilité que la variable aléatoire \bar{X} qui prend pour valeur la moyenne d'un échantillon de taille n , vérifie la condition :

$$P\left(\mu - t_{\alpha/2} \frac{\sigma}{\sqrt{n}} \leq \bar{X} \leq \mu + t_{\alpha/2} \frac{\sigma}{\sqrt{n}}\right) = 1 - \alpha$$

Après tirage de l'échantillon, on a donc deux possibilités : l'intervalle de confiance contient la moyenne μ de la population ou l'intervalle de confiance ne contient pas cette moyenne.

1) Recherche de différentes valeurs de t_α

Reporter dans votre feuille de calcul le tableau suivant qui sera complété en utilisant le tableur.

Risque α	10%	5%	1%
Valeur du nombre t_α			

2) Test sur 100 échantillons

L'objectif de cette question est de tester les 100 premiers échantillons parmi les 400 prélevés dans la partie 1. Ces échantillons sont placés dans la plage A1 :J10 de votre feuille de calcul.

Vous devez réaliser une procédure qui va étudier chaque échantillon de la plage A1 :J10 pour savoir si cet échantillon donne un intervalle de confiance qui contient ou non la moyenne μ .

Si l'échantillon donne un intervalle ne contenant pas la moyenne, on affichera en rouge sa moyenne.

Coder cette procédure et l'essayer sur les 100 échantillons pour les trois risques du tableau ci-dessus.

Automatiser le lancement de votre procédure par un bouton sur lequel on portera le mot « **Test** ».

Reporter et compléter le tableau ci-dessous.

Risque α	10%	5%	1%
Nombre d'échantillons conduisant à un « mauvais » intervalle de confiance :			
Essai 1			
Nombre d'échantillons conduisant à un « mauvais » intervalle de confiance :			
Essai 2			

Nombre d'échantillons conduisant à un « mauvais » intervalle de confiance : Essai 3			
---	--	--	--

Aide : Pour la procédure de test des échantillons, il suffit, une fois la moyenne de l'échantillon sélectionnée, de calculer les deux bornes de l'intervalle de confiance et de tester si la moyenne μ est située entre ces deux bornes.

Dans la négative, on affiche la valeur de la moyenne correspondante en rouge afin, une fois le test terminé, de voir les différents échantillons sur le 100 étudiés qui ne conduisent pas à un intervalle de confiance contenant la moyenne.

Il faudra, pour un nouveau test, remettre en noir les moyennes qui ont été colorées lors du test précédent. Ceci peut être fait dans la procédure, ou manuellement une fois le test effectué.

Exercice 18 (Test de conformité du Khi-deux)

On fait l'hypothèse que le générateur des nombres pseudo aléatoires d'Excel est parfait et qu'il permet de simuler correctement le lancer d'un dé qui sera donc considéré comme parfait dans ce TP.

Partie 1

Dans la feuille 1 de votre classeur, et en utilisant les fonctions disponibles dans Excel, réaliser la simulation suivante :

On lance un dé cubique parfait 120 fois.

On classe les effectifs de sortie de chaque face dans un tableau.

On calcule la distance dite du Khi-deux pour cette série, qui sera comparée à la distribution théorique uniforme.

Les résultats obtenus permettent-ils de conclure que le dé est parfait ?

Partie 2

Dans la feuille 2 de votre classeur, on souhaite réaliser le travail précédent 100 fois en utilisant la programmation VBA.

Algorithme

Définir une variable appelée table qui contiendra 6 valeurs (les effectifs de sorties des faces).

Pour p allant de 1 à 100

Initialiser les 6 valeurs de la table à zéro

Pour k allant de 1 à 120

Lancer le dé

Ajouter 1 dans la table à effectif de la face du dé qui vient de sortir.

Fin du pour k

Calculer la distance du Khi-deux associée au contenu de la table des effectifs construite à l'aide de la boucle précédente.

Placer le résultat de cette distance ligne p+1, colonne 9 de la feuille 2

Fin du pour p

Partie 3

Afficher le nombre N de distance parmi les 100 qui sont supérieur à 11.07, valeur de la table du Khi-deux à 5 degrés de liberté associée à un risque de 5%.

Afficher le nuage de points associé aux 100 distances calculées comme dans la figure ci-dessous.

Automatiser la procédure VBA précédente à l'aide d'un bouton de lancement automatique et observer la fluctuation de N et sa valeur moyenne sur plusieurs lancements de la procédure.

Sur les 100 simulations réalisées, combien conduisent à la conclusion que le dé est parfait ?

