



TP2 :

Partie 1 : Les macros-commandes de Microsoft PowerPoint

Partie 2 : Les macros-commandes de Microsoft Excel

Objectifs et déroulement de la séance :

Partie 1 (2 heures) :

- Introduction des objets et des actions de base du grapheur
- Introduction à la programmation de dessins

Partie 2 (2 heures) :

- Introduction à la programmation sous Excel

1 Les macros-commandes de Microsoft PowerPoint

Préambule :

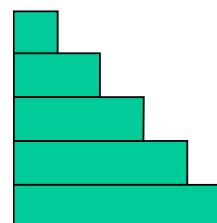
Avant de commencer à utiliser le grapheur, vérifiez, à l'aide d' « *Explorateur Windows* », la présence du répertoire "home sur sarado/INF11/TP2-3 " dans lequel vous allez enregistrer votre travail tout au long de la première partie de cette séance. Créez ce répertoire s'il n'existe pas dans votre hiérarchie (cf TP1).

Exercice 1.1 : Escalier 5 marches (version 1)

On souhaite programmer une macro réalisant la figure ci-contre :

La marche la plus haute est un carré de 40*40 pixels, celle d'en-dessous un rectangle de 40*80 pixels, et la longueur de chaque marche suivante augmente de 40 pixels.

Le premier carré à son coin supérieur gauche positionné aux coordonnées (10,10).



a. Complétez l'algorithme ci-dessous :

```

Algo Escalier1
Début
    Rectangle (position X :          position Y :          côté X :          côté Y:          )
    Rectangle (position X :          position Y :          côté X :          côté Y:          )
    Rectangle (position X :          position Y :          côté X :          côté Y:          )
    Rectangle (position X :          position Y :          côté X :          côté Y:          )
    Rectangle (position X :          position Y :          côté X :          côté Y:          )
Fin
    
```

b. Implémentez la macro en VBA sous PowerPoint

Astuce :

- Enregistrez une macro (cf Guide § 4.3.1) qui trace un rectangle, que vous nommerez « Escalier1 ». Cela doit donner, aux commentaires et valeurs numériques près :

```
Sub Rectangle()
ActiveWindow.Selection.SlideRange.Shapes.AddShape(msoShapeRectangle, 435#, 182#, 49#,
56#).Select
End Sub
```

- Modifiez le code : adaptez les valeurs des paramètres pour que la macro trace la première marche, c'est-à-dire un carré de 40*40 dont le coin supérieur gauche se trouve au point de coordonnées (10,10).
- Sauvegardez et testez le fonctionnement de votre macro (cf Guide § 4.3.2).
- Modifiez le code Visual Basic de la macro afin qu'elle trace également les autres marches : faire des « copier-coller » de la première instruction (le carré) et adaptez les valeurs des paramètres pour que chaque nouvelle instruction trace le rectangle du dessous du rectangle précédent.
- Testez le fonctionnement de votre macro.

Exercice 1.2 : Escalier 5 marches (version 2)

a. Complétez l'algorithme ci-dessous pour réaliser le même escalier avec une itération de type « pour » :

```
Algo Escalier2
Début
  {déclarations}
  positionX, positionY, longueur, hauteur : entier
  i : entier

  {initialisations}
  positionX ← _____
  positionY ← _____
  longueur ← _____
  hauteur ← _____

  Pour i = _____ jusqu'à _____ faire
    Rectangle(_____, _____, _____, _____)
    positionY ← _____
    longueur ← _____
  Fin Pour
Fin
```

b. Implémentez la macro en VBA sous PowerPoint et testez la.

Exercice 1.3 : Escalier 5 marches (version 3)

Dans l'algorithme précédent, les valeurs de `positionY` et `longueur` étaient mises à jour à chaque itération. Il est possible de produire un algorithme similaire sans les 2 dernières affectations dans l'itération. On peut exprimer la longueur en fonction de la hauteur et de `i`. De même, on peut exprimer `positionY` en fonction de `positionX` et `i`.

a. Complétez l'algorithme ci-après pour réaliser le même escalier en conservant une itération de type « pour ».

b. Implémentez la macro en VBA sous PowerPoint et testez la.

```

Algo Escalier3
Début
  {déclarations}
  positionX, hauteur : entier
  i : entier

  {initialisations}
  positionX ← ____
  hauteur ← ____

  Pour i = ____ jusqu'à ____ faire
  | Rectangle(____,____,____,____)
  Fin Pour
Fin

```

Exercice 1.4 : Escalier 5 marches (version 4)

- Complétez l'algorithme ci-dessous pour réaliser le même escalier avec une itération de type « tant que »
- Implémentez la macro en VBA sous PowerPoint et testez la

```

Algo Escalier5
Début
  {déclarations}
  positionX, hauteur : entier
  i : entier

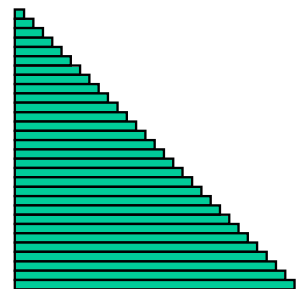
  {initialisations}
  positionX ← ____
  hauteur ← ____
  i ← ____

  tant que (____) faire
  | Rectangle(____,____,____,____)
  | i ← ____
  Fin Tant Que
Fin

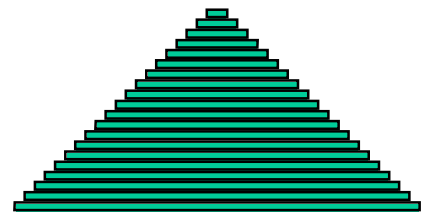
```

Exercice 1.5 : Escalier 30 marches

- Produire l'algorithme pour réaliser escalier30a avec une itération de type « pour »
- Produire l'algorithme pour réaliser escalier30b avec une itération de type « tant que »
- Implémenter les deux macros en VBA sous PowerPoint et les tester

**Exercice 1.6 : La pyramide**

- Produire un algorithme pour réaliser Pyramide
- Traduire l'algorithme en VBA sous PowerPoint et le tester.



Exercice 1.7 : Frise de 25 cercles

Produire l'algorithme de cette frise de 25 cercles de 20 pixels de diamètre.



Traduire l'algorithme en VBA puis tester.

Exercice 1.8 : Frise de 12 carrés

Produire l'algorithme de cette frise de 12 carrés de 30 pixels de coté espacés de 30 pixels :



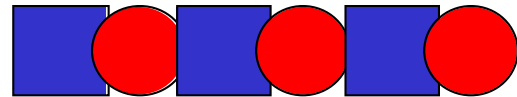
Traduire l'algorithme en VBA puis tester.

Exercice 1.9 : Carrés bleus et rond rouge

a. Produire un algorithme pour réaliser cette frise.
Il s'agit d'un carré bleu (de 30 pixels de coté) et d'un rond rouge (de 30 pixels de diamètre), qui se chevauchent sur 8 pixels. Ceci est répété 3 fois.

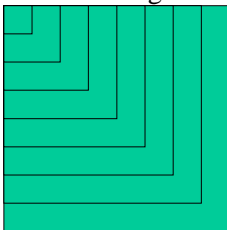
- Utilisez une itération
- Respectez les couleurs

b. Traduire l'algorithme en VBA sous PowerPoint et le tester.



Exercice 1.10 : Carrés emboîtés

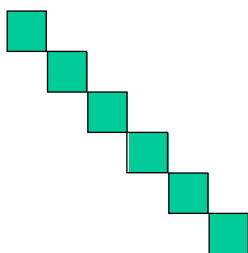
Produire l'algorithme réalisant les carrés emboîtés. Traduire en VBA. Tester.



Exercice 1.11 : Carrés en escalier

Produire l'algorithme réalisant les carrés en escalier.

Traduire en VBA. Tester.

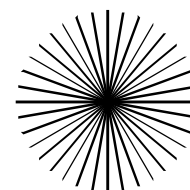


Exercice 1.12 : Etoiles et rotation

On s'intéresse à présent à la réalisation de la figure ci-contre :

1^{ère} étape : analyse de la figure

Si on analyse la structure de cette figure, on voit que l'objet de base est un simple trait. L'ensemble de la figure est obtenu par une répétition de traits identiques ayant subi des rotations incrémentées de 10° .



2^{ème} étape : macro de la figure de base

Enregistrez une macro que vous nommerez « RayonOblique » et qui fait les actions suivantes :

- trait horizontal¹
- rotation² de 10°

Où se trouve le centre de rotation du trait ? La rotation se fait-elle dans le sens horaire ou trigonométrique ?

3^{ème} étape : étude du code obtenu

A l'aide de l'éditeur Visual Basic, observez le code ainsi produit. Cela doit donner, aux commentaires et valeurs numériques près :

```
Sub rayonOblique()
    ActiveWindow.Selection.SlideRange.Shapes.AddLine(222#, 222#, 456#, 222#).Select
    ActiveWindow.Selection.ShapeRange.Rotation = 10#
End Sub
```

Les 2 premiers nombres dans la () de l'action AddLine sont les coordonnées (X,Y) de l'extrémité gauche du trait, les 2 suivants celles de l'extrémité droite. En langage algorithmique, nous noterons ces deux instructions :

Trait (X1, Y1, X2, Y2)

Rotation (angle)

Remplacez les coordonnées par : (150#, 300#, 400#, 300#)

4^{ème} étape : introduction de l'itération

Dans l'éditeur Visual Basic, faites une copie de la macro « rayonOblique() » que vous placerez à la suite des autres et que vous nommerez « Etoile ». De manière similaire à la macro Escalier2, introduisez une boucle « pour » avec les valeurs adaptées au nombre d'itérations nécessaires pour réaliser la figure. Pour avoir une rotation incrémentée de 10 à chaque pas de l'itération, il faudra que la valeur 10# de l'instruction :

```
ActiveWindow.Selection.ShapeRange.Rotation = 10#
```

soit remplacée par une expression calculant cette valeur à partir de l'indice de l'itération. A vous de la trouver !

¹ Cliquez sur le bouton « trait » de la barre de dessin, puis tracez le trait à la souris en maintenant la touche « MAJ » enfoncée.

² Faites un double-clic sur le trait obtenu. Dans la boîte de dialogue « option de forme » qui apparaît, sélectionnez l'onglet « taille », et écrivez 10 dans la rubrique « angle ». Voir Guide §4.2.

Algorithme

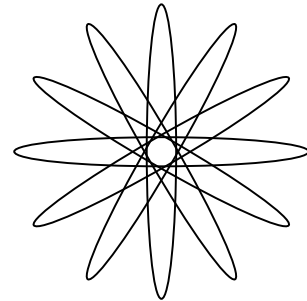
Lorsque la macro fonctionne, notez ci-dessous l'algorithme correspondant :

```
algo Etoile
début
```

```
fin
```

Exercice 1.13 : Rosace

Sur le modèle de l'étoile, produire un algorithme dessinant une rosace.
La forme de base est un ovale transparent sur lequel on exerce une rotation.
Implémenter l'algorithme en VBA puis tester.

**2 Les macros-commandes de Excel**

Le principe des macro-commandes est aussi utilisable dans Excel. Il est aussi possible de capturer des macros.

Exercice 2.1 : Capture de macro en Excel

- Créez un nouveau document sous Excel.
- Débutez l'enregistrement d'une macro (Menu outil, Macro, Nouvelle Macro).
- Pendant l'enregistrement,
 - sélectionnez la cellule A1, entrez la valeur 1
 - sélectionnez la cellule A2, entrez la valeur 2
 - sélectionnez la cellule A3.
- Arrêtez l'enregistrement de la macro.
- Observez le code (Menu outil, Macro, Macros. Sélectionnez la macro nouvellement créée. Cliquez sur le bouton modifier). Vous devez obtenir un code similaire à celui-là.

```
Sub Macro1()
'
' Macro Macro1
' Macro enregistrée le 17/09/2007 par
ldb
'
    Range("A1").Select
    ActiveCell.FormulaR1C1 = "1"
    Range("A2").Select
    ActiveCell.FormulaR1C1 = "2"
    Range("A3").Select
End Sub
```

Modifiez le code précédent de la façon suivante :

```
Sub Macro1()
'
    Cells(1,1) = 1
    Cells(2,1) = 2
End Sub
```

Effacez les valeurs sur la feuille Excel, puis exécutez la macro modifiée. Vérifiez que vous obtenez bien les valeurs attendues.

A propos de « Planètes ».

Au TP1, nous avons travaillé sur le fichier TP1-planete.xls. Nous avons rempli la feuille de calcul via l'interface. On se propose de reprendre le fichier et d'effectuer les différents calculs à l'aide d'une macro.

- Copiez TP1-planete.xls et renommez le fichier en TP2-planete.xls
- Déplacez le fichier dans "home sur sarado/INF112/TP2-3"
- Ouvrez TP2-planete.xls

Exercice 2.2 : Calcul du Volume pour le premier Astre (Callisto)

- Créez une nouvelle macro, dans laquelle vous allez calculer le volume de Callisto. Le volume de Callisto se trouve dans la cellule de coordonnées (3,5). On calcule son volume à partir du diamètre situé dans la cellule (3,2).

- Complétez l'algorithme pour mesurer le volume de Callisto
- Traduisez l'algorithme en VBA et testez le.

```
Algo Planetel
Début
    {déclarations}
    Cellule(____, ____) $\leftarrow$ (3.1415926 * (Cellule(____, ____)) * 1000) ^ 3) / 6
Fin
```

```
Sub Planetel ()
    {déclarations}

    Cells(____, ____) = (3.1415926 * (Cells(____, ____)) * 1000) ^ 3) / 6
End Sub
```

Exercice 2.3: Calcul de la densité et de la densité entière pour le premier astre (Callisto)

Complétez l'algorithme et la macro précédente pour calculer la densité et la densité entière.

En VBA, pour calculer la valeur entière d'une cellule, on utilise la fonction `Int (Cells(____, ____))`

Exercice 2.4: Calcul des trois premières formules pour tous les astres

- Complétez l'algorithme pour mesurer le volume et les densités des autres astres en introduisant une itération « pour ».

Algo Planete2

```

Début
  {déclarations}
  i : entier

  Pour i = _____ jusqu'à _____ faire
    Cellule(____ , ____ ) ← (3.1415926 * (Cellule(____ , ____ ) * 1000) ^ 3) / 6
    Cellule(____ , ____ ) ← Cellule(____ , ____ ) / Cellule(____ , ____ )
    Cellule(____ , ____ ) ← Int (Cellule(____ , ____ ))
  Fin Pour
Fin

```

b. Traduisez l'algorithme en VBA et testez-le.

Exercice 2.5: Densités comparées pour tous les astres

Introduisez le calcul de la densité comparée.

Pour rappel, la densité comparée est le rapport entre la densité de l'astre concerné et celle de la terre.

Complétez l'algorithme (noter les modifications apportées dans l'encadré ci-dessous), traduisez le en VBA, et testez le.

Exercice 2.6: Catégorie des Astres

a. Complétez l'algorithme pour inscrire dans la colonne 9 si l'astre est une planète ou un satellite.

b. Traduisez l'algorithme en VBA et testez le.

Algo Planete3

```

Début
  {déclarations}
  i : entier

  Pour i = _____ jusqu'à _____ faire
    [déjà écrit précédemment]
    Si (Cellule(____, ____)=0) alors
      Cellule(____ , 9) ← "satellite"
    Sinon
      Cellule(____ , 9) ← "Planète"
    Fin Si
  Fin Pour
Fin

```

```

Sub Planete3 ()
  If (Cells(____, ____ ) = 0) Then
    Cells(____, 9) = "satellite"
  Else
    Cells(____, 9) = "planète"
  End If
End Sub

```

Exercice 2.7: Planètes tellurique ou jovienne pour tous les astres

a. Complétez l'algorithme pour inscrire dans la colonne 10 si l'astre est une planète jovienne, tellurique ou un satellite.

b. Traduisez l'algorithme en VBA et testez le.