

Examen d'informatique

Jeudi 30 janvier 2014

Durée : 1h30

Documents, calculettes et téléphones portables interdits

Le barème est donné à titre indicatif, il est susceptible d'être modifié

Remarques :

- la lisibilité et l'efficacité (optimisation du temps d'exécution par exemple) de vos programmes seront pris en compte pour l'évaluation de vos réponses ;
- les deux dernières pages de ce sujet sont une annexe dans laquelle figure des extraits de votre photocopié de cours qui peuvent vous être utiles pour répondre à certaines questions ;
- à chaque question, vous pouvez supposer que les fonctions et procédures des questions précédentes sont définies correctement, vous pouvez donc les utiliser.

Exercice 1 : Projets d'investissement (14 pts)

Dans cet exercice, on considère le problème de la sélection de projets dans lesquels une société voudrait investir. Pour cela, 199 projets ont été minutieusement étudiés par des experts. Afin de pouvoir les comparer, les experts ont synthétisé leurs évaluations en notant les projets selon trois critères : la *pertinence* des objectifs du projet, la *qualité* du projet et sa *durabilité*. Chaque projet est évalué sur chaque critère par une note allant de 0 (plus mauvaise note possible) à 50 (meilleure note possible). Les experts ont reporté les évaluations des 199 projets dans une feuille Excel selon le format suivant : la première colonne contient le nom des projets, les deuxième, troisième et quatrième colonnes contiennent les notes obtenues par les projets selon le critère pertinence (colonne 2), qualité (colonne 3) et durabilité (colonne 4). Enfin, à la colonne 5 figure le coût d'investissement du projet. La figure 1 est un exemple du fichier Excel dans lequel sont regroupées les évaluations des projets.

	1	2	3	4	5	6
1	Nom Projet	Pertinence	Qualité	Durabilité	Coût	
2	AAA	35	42	18	25 000 €	
3	ABC	25	33	45	17 000 €	
4	DVD	32	40	15	30 000 €	
5	FAQ	46	19	31	8 000 €	

FIGURE 1 – Evaluation des projets effectuée par les experts

On remarque que la première ligne du fichier contient les en-têtes des colonnes, la liste des projets ne commence donc qu'à partir de la deuxième ligne. **Dans cet exercice, on travaille donc sur une feuille Excel dans laquelle 199 projets sont listés de la ligne 2 à la ligne 200 selon le format expliqué ci-dessus.**

Partie 1. Analyse des données (5.5 pts)

Dans un premier temps, la société souhaite analyser les données fournies par les experts.

Q 1.1 (1 pt) Ecrivez en VBA la fonction `controleProjet` qui prend en paramètre un numéro de ligne `li`, et retourne vrai si les notes du projet de la ligne `li` sont bien comprises entre 0 et 50 (inclus) sur chaque critère, et faux sinon. On suppose ici que le numéro de ligne est compris entre 2 et 200.

```
Function controleProjet(ByVal li As Long) As Boolean
Dim j As Long
controleProjet = True
For j = 1 To 3
    controleProjet = controleProjet And Cells(li, j).Value >= 0 And _
        Cells(li, j).Value < 51
Next j
End Function
```

- -0.5pt si type de retour faux ou non précisé (MAX -1pt sur ttes les questions)
- -0.5pt si valeur de retour fausse ou non précisée
- -0.25pt si variables non déclarées et/ou non initialisées (MAX -1pt sur ttes les questions)
- -0.25pt par pte erreur de syntaxe (ByVal, oubli du .value, etc.) (MAX -1pt sur ttes les questions)
- -0.5 pt par erreur de calcul (oubli d'un critère, résultat faux, etc.)

Q 1.2 (1.5 pt) Ecrivez en VBA la macro `controle` qui vérifie que **tous** les projets sont bien notés entre 0 et 50 sur chaque critère. Pour cela la macro `controle` met en gras les noms des projets ayant au moins une note non comprise entre 0 et 50, et affiche à l'aide d'une boîte de dialogue le message « Anomalie! » s'il y a au moins un projet mis en gras. La macro n'affiche rien si toutes les notes de tous les projets sont bien comprises entre 0 et 50.

```
Sub controle()
Dim i As Long, msg As Boolean
msg = False
For i = 2 To 200
    If controleProjet(i) Then
        Cells(i, 1).Font.Bold = True
        msg = True
    End If
Next i
If msg Then
    MsgBox "Anomalie !"
End If
End Sub
```

- -0.25pt si variables non déclarées et/ou non initialisées (MAX -1pt sur ttes les questions)
- -0.25pt par pte erreur de syntaxe (ByVal, oubli du .value, etc.) (MAX -1pt sur ttes les questions)
- /0.5 pt : Font.Bold bien utilisé
- /0.5 pt : affichage correct de "Anomalie"
- /0.5 pt : mise en gras des bons projets

Q 1.3 (1 pt) Ecrivez en VBA la macro **moyenne** qui remplit la colonne 6 en calculant la moyenne des trois notes obtenues (une note par critère) pour chaque projet. Par exemple, l'exécution de la macro **moyenne** sur l'exemple de la figure 1 donnerait la feuille Excel suivante :

	1	2	3	4	5	6
1	Nom Projet	Pertinence	Qualité	Durabilité	Coût	
2	AAA	35	42	18	25 000 €	31,6666667
3	ABC	25	33	45	17 000 €	34,3333333
4	DVD	32	40	15	30 000 €	29
5	FAQ	46	19	31	8 000 €	32

FIGURE 2 – Exécution de moyenne sur l'exemple de la figure 1

```

Sub moyenne()
Dim i As Long
For i = 2 To 200
    Cells(i, 6).Value = (Cells(i, 2).Value + Cells(i, 3).Value + Cells(i, 4).Value) / 3
Next i
End Sub

Sub moyenneBis()
Dim i As Long
For i = 2 To 200
    Cells(i, 6).FormulaR1C1Local = "=MOYENNE(LC(-4):LC(-2))"
Next i
End Sub

```

- -0.25pt si variables non déclarées et/ou non initialisées (MAX -1pt sur ttes les questions)
- -0.25pt par pte erreur de syntaxe (ByVal, oubli du .value, etc.) (MAX -1pt sur ttes les questions)
- /0.5 : boucle correcte
- /0.5 : calcul correct de la moyenne

On considère dorénavant que la macro **moyenne** a été exécutée. La colonne 6 contient donc la moyenne des notes des projets.

Q 1.4 (2 pts) Ecrivez en VBA la macro **recherche** qui demande à un utilisateur le nom d'un projet, puis affiche la note moyenne du projet en question s'il est dans la liste des projets de la feuille Excel ou le message « Projet X absent » sinon (où X est le nom du projet saisi par l'utilisateur).

```
Sub recherche()  
Dim nom As String, i As Long  
i = 2  
nom = Application.InputBox("nom du projet ?")  
Do While Cells(i, 1).Value <> nom And i < 201  
    i = i + 1  
Loop  
If i = 201 Then  
    MsgBox "Projet " & nom & " absent"  
Else  
    MsgBox "la moyenne de " & nom & " est " & Cells(i, 6).Value  
End If  
End Sub
```

- -0.25pt si variables non déclarées et/ou non initialisées (MAX -1pt sur ttes les questions)
- -0.25pt par pte erreur de syntaxe (ByVal, oubli du .value, etc.) (MAX -1pt sur ttes les questions)
- /0.5pt : message final
- /0.5pt : Application.inputbox bien utilisé
- /1pt : boucle (-0.5pt si recherche continuée alors que le projet est trouvé)

Partie 2. Sélection des projets (8.5 pts)

On souhaite maintenant filtrer la liste des projets en supprimant ceux qui sont *Pareto-dominés*. Un projet X est considéré comme Pareto-dominé s'il existe un autre projet Y tel que la note de Y est strictement meilleure que celle de X sur tous les critères. Dans l'exemple de la figure 1, le projet DVD est Pareto-dominé par le projet AAA car il a obtenu une plus mauvaise note que AAA sur les trois critères (pertinence, qualité et durabilité). En revanche, le projet DVD n'est pas Pareto-dominé par le projet FAQ par exemple car DVD a obtenu une meilleure note que FAQ sur le critère qualité.

Q 1.5 (2.5 pts) Ecrivez en VBA la macro `pareto` qui indique les projets Pareto-dominés en colorant en rouge (RGB (230,0,0)) l'intérieur des cellules des lignes des projets Pareto-dominés, de la colonne 1 à la colonne 6. Par exemple, l'appel de la macro `pareto` sur l'exemple de la figure 1 colorerait en rouge l'intérieur des cellules de la ligne 4 (de la colonne 1 à la colonne 6) puisque le projet DVD est Pareto-dominé par le projet AAA. L'intérieur des autres cellules ne serait pas modifié car les projets AAA, ABC et FAQ ne sont Pareto-dominés par aucun autre projet.

```

Sub pareto()
Dim i As Long, i2 As Long, j As Long
For i = 2 To 200
For i2 = 2 To 200
If Cells(i, 2).Value < Cells(i2, 2).Value And _
Cells(i, 3).Value < Cells(i2,3).Value And _
Cells(i, 4).Value < Cells(i2, 4).Value Then
    'projet à la ligne i Pareto-dominé par celui de la ligne i2
    For j = 1 To 6
        Cells(i, j).Interior.Color = RGB(230, 0, 0)
    Next j
    'Autre possibilité :
    'Range(Cells(i,1),Cells(i,6)).Interior.Color = RGB(230, 0, 0)
End If
Next i2
Next i
End Sub

```

- -0.25pt si variables non déclarées et/ou non initialisées (MAX -1pt sur ttes les questions)
- -0.25pt par pte erreur de syntaxe (ByVal, oubli du .value, etc.) (MAX -1pt sur ttes les questions)
- /0.5 : cas du projet Pareto-dominé bien déterminé (3 tests)
- /1 : tous les projets sont bien comparés 2 à 2
- /1 : mise en rouge correcte (0.5 pt pour bonne utilisation de Interior.Color et RGB, et 0.5 pour mise en rouge des 6 cellules)

On considère maintenant que les projets non Pareto-dominés ont été listés dans la Feuille 2 du fichier Excel en reportant leur nom en colonne 1, leur coût en colonne 2 et leur moyenne en colonne 3. Pour l'exemple de la figure 1, la liste des projets non Pareto-dominés de la Feuille 2 d'Excel serait la suivante :

	1	2	3	
1	Nom Projet	Coût		
2	AAA	25 000 €	31,6666667	
3	ABC	17 000 €	34,3333333	
4	FAQ	8 000 €	32	

FIGURE 3 – Liste des projets non Pareto-dominés (Feuille 2 d'Excel)

Dorénavant, on ne sait donc pas combien de projets sont listés dans la Feuille 2 d'Excel. Si 3 projets non Pareto-dominés sont listés dans la Feuille 2 d'Excel par exemple, alors la ligne 5 et les suivantes sont vides. On peut remarquer qu'une cellule vide a la valeur "".

Q 1.6 (1.5 pt) Ecrivez en VBA la fonction `meilleur` qui prend en paramètre un numéro de ligne `li` et retourne la ligne du projet qui a obtenu la meilleure moyenne (on suppose qu'il n'y en a qu'un) parmi les projets listés à partir de la ligne `li`. Sur l'exemple de la figure 3, si l'on suppose qu'il n'y a

que 3 projets non Pareto-dominés, meilleur(2) retournerait 3 et meilleur(4) retournerait 4.

```

Function meilleur(ByVal li As Long) As Long
Dim i As Long, max As Long
max = li
i = li + 1
Do While Cells(i, 1).Value <> ""
If Cells(i, 3).Value > Cells(max, 3).Value Then
    max = i
End If
i = i + 1
Loop
meilleur = max
End Function

```

- -0.5pt si type de retour faux ou non précisé (MAX -1pt sur ttes les questions)
- -0.25pt si variables non déclarées et/ou non initialisées (MAX -1pt sur ttes les questions)
- -0.25pt par pte erreur de syntaxe (ByVal, oubli du .value, etc.) (MAX -1pt sur ttes les questions)
- /0.5pt : boucle correcte
- /1pt : gestion du max (initialisation à li /0.5, mise à jour /0.5)

Q 1.7 (3 pts) Ecrivez en VBA la macro **tri** qui trie les projets dans l'ordre décroissant de leur moyenne. Dans cette question, on ne cherche pas à optimiser le nombre d'opérations effectuées par votre procédure, vous pouvez donc utiliser l'algorithme de tri que vous voulez. L'exécution de la macro **tri** sur l'exemple de la figure 3 donnerait le résultat suivant (si l'on suppose qu'il n'y a que 3 projets non Pareto-dominés) :

	1	2	3
1	Nom Projet	Coût	
2	ABC	17 000 €	34,3333333
3	FAQ	8 000 €	32
4	AAA	25 000 €	31,6666667

FIGURE 4 – Liste des projets non Pareto-dominés triés

```

Sub tri()
Dim i As Long, max As Long, n As String, c As Double, m As Double
i = 2
Do While Cells(i, 1).Value <> ""
    'recherche du max
    max = meilleur(i)
    'echange du i et du max
    n = Cells(max, 1).Value: c = Cells(max, 2).Value: m = Cells(max, 3).Value
    Cells(max, 1).Value = Cells(i, 1).Value
    Cells(max, 2).Value = Cells(i, 2).Value
    Cells(max, 3).Value = Cells(i, 3).Value
    Cells(i, 1).Value = n: Cells(i, 2).Value = c: Cells(i, 3).Value = m
    i = i + 1
Loop
End Sub

```

- -0.5pt si variables non déclarées et/ou non initialisées (MAX -1pt sur ttes les questions)
- -0.25pt par pte erreur de syntaxe (ByVal, oubli du .value, etc.) (MAX -1pt sur ttes les questions)
- /1 pt : recherche correcte du max à chq étape
- /1 pt : mise du max courant à la bonne position sans perte de données
- /1 pt : boucle "globale" correcte

Q 1.8 (1.5 pt) La société dispose d'un budget b . Elle décide de financer tous les meilleurs projets (selon leur moyenne) dans la limite de son budget b . Ainsi si la société a un budget de 11 k€ et que les trois meilleurs projets ont des coûts de 2k€ pour le premier, 7k€ pour le deuxième et 4 k€ pour le troisième, alors la société financera les deux meilleurs projets. Si la société disposait d'un budget de 8 k€ par exemple, elle ne financerait que le meilleur projet. Ecrivez la procédure `ProjetAcceptes` qui prend en paramètre un budget b et affiche successivement le nom de tous les projets qui seront financés par la société. Pour cela, la procédure `ProjetAcceptes` trie les projets dans l'ordre décroissant de leur moyenne puis affiche successivement les noms des projets acceptés (en fonction du budget).

```

Sub projetsAcceptes(ByVal b As Double)
Dim som As Double, i As Long
tri
som = Cells(2, 2).Value: i = 2
Do While som <= b
    MsgBox Cells(i, 1).Value & " accepté"
    i = i + 1
    som = som + Cells(i, 2).Value
Loop
End Sub

```

- -0.5pt si variables non déclarées et/ou non initialisées (MAX -1pt sur ttes les questions)
- -0.25pt par pte erreur de syntaxe (ByVal, oubli du .value, etc.) (MAX -1pt sur ttes les questions)
- /0.5 pt : affichage correct des projets acceptés
- /1 pt : calcul de la somme courante et test d'arrêt

Exercice 2 : Calcul sur des entiers (6 pts)

On considère dans cet exercice le problème du calcul sur des grands nombres entiers. On appelle grand nombre entier un entier pouvant contenir jusqu'à 40 chiffres. Des calculs avec de tels grands nombres ne peuvent pas être effectués avec des calculettes simples car on dépasse les capacités de représentation des calculettes. On cherche donc ici à effectuer les calculs avec de grands nombres entiers à l'aide de VBA. On considère qu'un grand nombre entier est représenté dans la feuille Excel sur une ligne à raison d'un chiffre par colonne. On considère que les grands nombres entiers n'ont pas plus de 40 chiffres et on utilise donc les 40 premières colonnes de la feuille Excel pour représenter les nombres. Ainsi le nombre 45 123 sera représenté dans la feuille Excel en mettant le chiffre 3 dans la colonne 40, le chiffre 2 dans la colonne 39, le chiffre 1 dans la colonne 38, le chiffre 5 dans la colonne 37, et le chiffre 4 dans la colonne 36. Les colonnes 1 à 35 seront vides ou remplies de 0.

Q 2.1 (2 pts) Ecrivez en VBA la procédure **additionne** qui prend en paramètre trois numéros de lignes 11, 12 et 13 et remplit la ligne 13 par la somme des deux nombres représentés aux lignes 11 et 12. Ainsi si la ligne 1 contient le nombre 234 et la ligne 2 contient le nombre 862, alors l'exécution de : **additionne 1, 2, 3**

place à la ligne 3 le nombre 1096. Si la somme des deux grands nombres est un grand nombre de plus de 40 chiffres, alors la procédure **additionne** effectue la somme sur les 40 chiffres de droite et affiche le message « Dépassement de capacité ».

```
Sub additionne(ByVal l1 As Long, l2 As Long, l3 As Long)
Dim ret As Byte, som As Byte, i As Long
ret = 0
For i = 40 To 1 Step -1
    som = Cells(l1, i).Value + Cells(l2, i).Value + ret
    If som > 9 Then
        som = som - 10
        ret = 1
    Else
        ret = 0
    End If
    Cells(l3, i).Value = som
Next i
If ret > 0 Then MsgBox "dépassement de capacité"
End Sub
```

- -0.25pt si variables non déclarées et/ou non initialisées (MAX -1pt sur ttes les questions)
- -0.25pt par pte erreur de syntaxe (ByVal, oubli du .value, etc.) (MAX -1pt sur ttes les questions)
- -0.5pt si erreur sur les lignes à utiliser (11, 12 et 13)
- /1.5 pt : gestion de la retenue et de la somme
- /0.5 pt : bonne gestion du dépassement de capacité

Q 2.2 (2 pts) Ecrivez en VBA la fonction **plusGrand** qui prend en paramètre deux numéros de ligne 11 et 12 et retourne le numéro de ligne qui contient le plus grand des deux nombres entre celui de la ligne 11 et celui de la ligne 12. On suppose que les deux nombres sont différents.

```

Function plusGrand(ByVal l1 As Long, ByVal l2 As Long) As Long
Dim i As Long: i = 1
Do While Cells(l1, i).Value = Cells(l2, i).Value And i < 40
    i = i + 1
Loop
If Cells(l1, i).Value < Cells(l2, i).Value Then
    plusGrand = l2
Else
    plusGrand = l1
End If
End Function

```

- -0.5pt si type de retour faux ou non précisé (MAX -1pt sur ttes les questions)
- -0.25pt si variables non déclarées et/ou non initialisées (MAX -1pt sur ttes les questions)
- -0.25pt par pte erreur de syntaxe (ByVal, oubli du .value, etc.) (MAX -1pt sur ttes les questions)
- -0.5pt si erreur sur les lignes à utiliser (l1, l2)
- /2pts : algorithme correct pour la détermination du plus grand

Q 2.3 (2 pts) On suppose dans cette question (uniquement) que les grands nombres entiers sont binaires. Les chiffres de la feuille Excel ne sont donc que des 0 ou des 1. Les nombres représentés dans la feuille sont donc des nombres binaires représentés sur 40 bits. Ecrivez en VBA la fonction `binToDec` qui prend un numéro de ligne `li` en paramètre et retourne la valeur décimale (i.e. en base 10) du nombre binaire représenté à la ligne `li`. Par exemple, si la ligne 1 contient le nombre 100110, alors `binToDec(1)` retourne 38.

NB : on peut remarquer que $2^{40} = 1099511627776$.

```

Function binToDec(ByVal li As Long) As Double
Dim i As Long, res As Double
res = 0
For i = 40 To 1 Step -1
res = res + Cells(li, i).Value * 2 ^ (40 - i)
Next i
binToDec = res
End Function

```

- -0.5pt si type de retour faux ou non précisé (MAX -1pt sur ttes les questions)
- -0.25pt si variables non déclarées et/ou non initialisées (MAX -1pt sur ttes les questions)
- -0.25pt par pte erreur de syntaxe (ByVal, oubli du .value, etc.) (MAX -1pt sur ttes les questions)
- /1.5pt : calcul correct
- /0.5pt : res est bien de type Double ou Single (les autres types numériques ne permettent pas de représenter 2^{40} .)