Examen d'informatique

Lundi 18 mai 2015

Durée : 2h

Documents, calculettes et téléphones portables interdits

Remarques :

- vous trouverez à la fin du sujet quelques rappels de classes VBA extraits de votre polycopié de cours qui peuvent vous être utiles pour répondre à certaines questions ;
- la lisibilité et l'efficacité de vos programmes seront pris en compte pour l'évaluation de vos réponses ;
- à chaque question, vous pouvez supposer que les fonctions et procédures des questions précédentes sont définies correctement, vous pouvez donc les utiliser ;
- le barème est donné à titre indicatif, il est susceptible d'être modifié.

Exercice 1 : Sudoku (14 pts)

Le jeu du *Sudoku* consiste à remplir une grille de taille 9×9 avec les chiffres de 1 à 9 de manière à ce que chaque ligne, chaque colonne et chaque « sous-carré » de taille 3×3 contiennent une seule fois les chiffres de 1 à 9. La grille de jeu est ainsi un carré de 9 cases de côté subdivisé en 9 sous-grilles carrées identiques, appelées *sous-carrés*. Dans cet exercice, on considère qu'une grille de Sudoku est représentée dans la feuille Excel sur la plage de cellules allant de la cellule L1C1 à la cellule L9C9, comme le montre la grille de la figure 1. Les 9 sous-carrés de la grille sont délimités par les bordures noires dans la figure.

	1	2	3	4	5	6	7	8	9	10
1	7									
2				3	6		9		8	
3	8	3		4			1	2		
4	2					7				
5	3				8				1	
6				1					4	
7		7	3			1		4	2	
8	4		2		3	6				
9									6	
10										

FIGURE 1 – Grille de Sudoku

	1	2	3	4	5	6	7	8	9	10
1	7	9	5	2	1	8	4	6	3	
2	1	2	4	3	6	5	9	7	8	
3	8	3	6	4	7	9	1	2	5	
4	2	8	1	5	4	7	6	3	9	
5	3	4	9	6	8	2	7	5	1	
6	5	6	7	1	9	3	2	8	4	
7	6	7	3	9	5	1	8	4	2	
8	4	1	2	8	3	6	5	9	7	
9	9	5	8	7	2	4	3	1	6	
10										

FIGURE 2 – Solution de la grille de Sudoku

La solution de la grille de la figure 1 est donnée dans la figure 2. On peut en effet observer que sur chaque ligne, sur chaque colonne et dans chaque sous-carré de la solution, chaque chiffre de 1 à 9 n'apparaît qu'une seule fois. **Dans tout l'exercice, on suppose que, à tout moment, la grille est valide, c'est à dire que les chiffres dans la grille partiellement (ou complètement) remplie ne sont que des chiffres de 1 à 9 et qu'ils n'apparaissent pas plus d'une fois par ligne, colonne ou sous-carré.**

Partie 1. Possibilités par case (8 pts)

Q 1.1 Ecrire en VBA la fonction `nLigne` qui prend en paramètre un nombre `nb` et un numéro de ligne `li`, et retourne vrai si le nombre `nb` n'est pas déjà présent dans la ligne `li` et faux sinon.

La fonction `nColonne` prend en paramètre un nombre `nb` et un numéro de colonne `co`, et retourne vrai si le nombre `nb` n'est pas déjà présent dans la colonne `co` et faux sinon. On suppose que cette fonction est définie, elle peut donc être utilisée par la suite (on ne vous demande pas de l'écrire).

Q 1.2 Ecrire en VBA la fonction `nCarre` qui prend en paramètre un nombre `nb`, un numéro de ligne `li` et un numéro de colonne `co`, et retourne vrai si le nombre `nb` n'est pas déjà présent dans le sous-carré auquel appartient la case `(li,co)`. L'appartenance d'une case à un sous-carré dépend de la position de la case : par exemple la case `(2,4)` appartient au sous-carré allant de la cellule L1C4 à la cellule L3C6, et la case `(8,8)` appartient au sous-carré allant de la cellule L7C7 à la cellule L9C9.

Q 1.3 Ecrire en VBA la fonction `nbPossibilites` qui prend en paramètre une numéro de ligne `li` et un numéro de colonne `co`, et retourne le nombre de chiffres (entre 1 et 9) qu'il est possible de mettre dans la case `(li,co)` (en fonction des nombres déja présents sur la ligne `li`, sur la colonne `co` et dans le sous-carré auquel la case `(li,co)` appartient). On suppose que la case `(li,co)` est vide. Par exemple, l'instruction `nbPossibilites(4,7)` sur la grille de la figure 3 retournerait 4 car les quatre chiffres 3, 5, 6 et 8 sont les seuls chiffres qu'il est possible de placer dans la case `(4,7)`.

	1	2	3	4	5	6	7	8	9	10
1	7									
2				3	6		9	7	8	
3	8	3		4	7		1	2		
4	2					7				
5	3				8		7		1	
6				1					4	
7		7	3			1		4	2	
8	4		2		3	6			7	
9				7					6	
10										

FIGURE 3 – Grille partiellement remplie

Q 1.4 Ecrire en VBA la macro `unePossibilite` qui colore en gris (`RGB(220,220,220)`) l'intérieur des cellules vides de la grille de sudoku pour lesquelles il n'y a qu'une possibilité. On rappelle qu'une cellule vide a pour valeur `""`.

Q 1.5 On s'intéresse dans cette question à l'estimation du niveau de difficulté de la grille. On considère que le niveau de difficulté de la grille dépend du nombre moyen de possibilités par case vide. La grille est considérée comme *facile* si le nombre moyen de possibilités par case vide est inférieur à 2, *moyen* s'il est compris entre 2 et 4, *difficile* s'il est entre 4 et 6, et *diabolique* sinon. Ecrire en VBA la macro `difficulte` qui détermine le nombre moyen de possibilités par case vide et affiche par `MsgBox` le niveau de difficulté de la grille.

Partie 2. Indications par chiffre (5 pts)

Dans cette partie de l'exercice, on s'intéresse à l'affichage d'un indice pour l'utilisateur qui cherche à résoudre la grille de sudoku. Puisque chaque chiffre de 1 à 9 apparaît exactement 9 fois dans une grille de sudoku résolue, l'indice consiste à déterminer si un chiffre x est déjà placé dans 8 cases de la grille, car dans ce cas la place du 9e x est évidente. Par exemple, dans la figure 3 le chiffre 7 est déjà présent 8 fois dans la grille, le 9e 7 de cette grille doit donc être placé dans la case `(6,3)`.

Q 1.6 Ecrire en VBA la fonction `verifieParChiffre` qui retourne un chiffre de 1 à 9 qui est déjà présent 8 fois dans la grille. De plus, la fonction donne aussi le numéro de ligne et le numéro de colonne de la case dans laquelle le 9e chiffre doit être placé (précisons qu'il n'est pas demandé ici d'utiliser de boîtes de dialogues). Si aucun des chiffres de 1 à 9 n'est déjà présent 8 fois dans la grille, la fonction `verifieParChiffre` retourne 0.
NB : la notation de cette question tient compte de l'efficacité de votre algorithme.

Q 1.7 On appelle n la taille de la grille de dimension $n \times n$ (dans cet exercice, on a donc $n = 9$). Donner l'ordre de grandeur du nombre d'opérations de votre fonction `verifieParChiffre` dans le pire cas et dans le meilleur cas en fonction de n. Préciser brièvement ce que sont le meilleur cas et le pire cas.

Q 1.8 Ecrire en VBA la macro `testVPC` qui affiche le message « Le nombre x peut être placé dans la case (l,c) » lorsqu'un nombre x (entre 1 et 9) est déjà présent 8 fois dans la grille et que la seule place possible pour le 9e x est la case (l,c), et n'affiche rien sinon.

Partie 3. Evènements (1 pt)

Q 1.9 On suppose qu'en plus de la grille, la feuille Excel est dotée d'un bouton nommé « Arreter » et d'un bouton nommé « Lancer ». Expliquer brièvement et le plus simplement possible ce que fait le programme suivant.

```
Dim tps As Date

Private Sub Arreter_Click()
    End
End Sub

Private Sub Lancer_Click()
    tps = Now()
    Do
        Cells(2, 11).Value = Now() - tps
        DoEvents
    Loop
End Sub
```

Exercice 2 : Représentation binaire (6 pts)

Q 2.1 (2 pts) Ecrire en VBA la fonction `binToDec` qui prend en paramètre un nombre représentant un entier positif en binaire, et retourne sa valeur en base 10. Par exemple `binToDec(111001)` retournerait 57.

Q 2.2 (1 pt) Ecrire en VBA la macro `base10` qui parcourt des nombres binaires listés dans la colonne 1 de la feuille Excel en inscrivant leur valeur en base 10 dans la colonne 2 à la ligne correspondante. On ne sait pas à l'avance combien de nombres binaires sont présents dans la colonne 1. On arrête le parcours des nombres binaires dès que l'on rencontre une cellule vide. On rappelle qu'une cellule vide en VBA a la valeur `""`.

Q 2.3 (3 pts) On suppose que la macro `base10` a été exécutée. Ecrire en VBA la macro `tri` qui trie en ordre croissant les nombres listés dans les colonnes 1 et 2 de la feuille Excel. Dans cette question, la notation ne tient pas compte du choix de l'algorithme de tri.

Rappels VBA

Ces éléments de cours sont donnés à titre indicatif

Classe `Range`

La classe `Range` est la classe des plages de cellules. Entre autres, elle possède les propriétés suivantes :

```
Property Cells(i As Long, j As Long) As Range
' Lecture
' Cellule de la ligne i et de la colonne j relativement à l'objet Range

Property Value As Variant
' Lecture-écriture
' Valeur de la cellule

Property Interior As Interior
' Lecture-écriture
' Objet qui représente le fond des cellules de la plage

Property Font As Font
' Lecture
' Style des caractères affichés dans les cellules de la plage
```

Classe `Interior`

La classe `Interior` est la classe des fonds de cellules. Entre autres, elle possède la propriété suivante :

```
Property Color As Long
' Lecture-écriture
' Couleur (codée en RGB) du fond.
```