

## Interrogation 2 d'Informatique

Jeudi 12 décembre 2013

Durée : 1h30

*Le barème n'est pas définitif, il est donné à titre indicatif.*

### Exercice 1 : Vote par approbation (10 pts)

On considère dans cet exercice le problème de sélection d'un (ou plusieurs) candidat(s) à l'aide de la méthode de *vote par approbation* : chaque votant indique quels candidats il approuve parmi la liste des candidats, et le candidat sélectionné est celui qui a obtenu le plus d'approbations. En cas d'égalité, on considère que tous les candidats ayant obtenu le plus grand nombre d'approbations sont sélectionnés. Les votants peuvent approuver autant de candidats qu'ils veulent.

Supposons par exemple qu'il y ait 4 candidats  $a$ ,  $b$ ,  $c$  et  $d$  et 4 votants  $v_1$ ,  $v_2$ ,  $v_3$ ,  $v_4$ . Si le votant  $v_1$  approuve les candidats  $b$  et  $c$ , le votant  $v_2$  approuve les candidats  $a$  et  $c$ , le votant  $v_3$  approuve le candidat  $d$  uniquement et le votant  $v_4$  approuve les candidats  $c$  et  $d$ , alors  $c$  sera le candidat vainqueur car il a été approuvé par trois des votants, alors que  $d$  n'a été approuvé que par deux votants et  $a$  et  $b$  par un votant seulement chacun. Les données de l'élection sont stockées dans une feuille Excel sous le format suivant : la première ligne contient le nom des candidats, la première colonne contient le nom des votants et la cellule  $(i, j)$  contient un 1 si le votant de la ligne  $i$  approuve le candidat de la colonne  $j$  et un 0 sinon. Pour l'exemple précédent, cela donnerait la feuille Excel suivante :

	1	2	3	4	5	6
1		a	b	c	d	
2	v1	0	1	1	0	
3	v2	1	0	1	0	
4	v3	0	0	0	1	
5	v4	0	0	1	1	
6						
7						

Dans cet exercice, nous allons travailler sur les données d'une élection dans laquelle figurent 15 candidats et 99 votants. Les noms des candidats sont donc dans la plage de cellules de L1C2 à L1C16, les noms des votants dans la plage de cellules de L2C1 à L100C1, et les votes (0 ou 1) dans la plage de cellules allant de L2C2 à L100C16.

**Q 1.1 (2 pts)** Ecrivez en VBA la fonction `nbAppro` qui étant donné un numéro de colonne `col` retourne le nombre de votants qui ont approuvé le candidat de la colonne `col`. (On suppose ici que le numéro de colonne passé en paramètre est compris entre 2 et 16 inclus).

**Q 1.2 (1 pt)** Ecrivez en VBA la macro `score` qui, pour chaque candidat, inscrit dans la ligne 102 le nombre de votants ayant approuvé le candidat.

**Q 1.3 (1 pt)** En utilisant les formules Excel, proposez une autre version de la macro `score` que vous appellerez `scoreBis`.

**Q 1.4 (2 pts)** Ecrivez en VBA la macro `vainqueur` qui affiche le nom du candidat vainqueur selon la méthode de vote par approbation. S'il y a plusieurs candidats vainqueurs, la macro `vainqueur` affiche le nom de tous les vainqueurs dans des boîtes de dialogue successives (une boîte de dialogue par candidat vainqueur).

**Q 1.5 (4 pts)** Ecrivez en VBA la macro `resultats` qui affiche les candidats dans la colonne 20 dans l'ordre décroissant du nombre d'approbations qu'ils ont obtenues. Vous écrivez votre macro `resultats` pour qu'elle s'applique à l'élection considérée dans cet exercice (avec 15 candidats et 99 votants), mais si on l'appliquait au petit exemple donné au début de l'exercice, la colonne 20 devrait contenir le candidat *c* sur la première ligne, le candidat *d* sur la deuxième ligne, le candidat *a* (ou *b*) sur la troisième ligne et le candidat *b* (ou *a*) sur la quatrième ligne.

**Exercice 2 : Traitement de nombres binaires (10 pts)**

On considère dans cet exercice que différents nombres entiers positifs sont stockés en binaire sur 16 bits dans la feuille Excel à raison d'un nombre par ligne. Chaque cellule contient un bit. Par exemple, les nombres 1011100, 1111111100000000 et 101 seraient stockés dans la feuille Excel de la manière suivante :

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	
2	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	
4																	
5																	
6																	

**Q 2.1 (1 pt)** Ecrivez en VBA une fonction `divPar4` qui étant donné un numéro de ligne *li* retourne vrai si le nombre représenté à la ligne *li* est divisible par 4, et faux sinon.

**Q 2.2 (4 pts)** Ecrivez en VBA une fonction `nbPremier` qui étant donné un numéro de ligne *li* retourne vrai si le nombre représenté à la ligne *li* est premier, et faux sinon. On rappelle qu'un nombre est premier s'il n'est divisible que par 1 et lui-même.

**Q 2.3 (3 pts)** Ecrivez une macro `premOuDiv4` qui met en rouge les nombres de la feuille Excel qui sont divisibles par 4 et en gras ceux qui sont premiers. Pour mettre en rouge (ou en gras) un nombre, vous modifiez le style de caractères de chaque bit du nombre. Notons que nous ne savons pas ici combien de nombre binaires sont stockés dans la feuille Excel. On suppose donc que l'on peut arrêter le parcours des nombres à la première ligne dont la première cellule est vide (i.e. de valeur " ").

**Q 2.4 (1 pt)** Donnez les représentations binaires sur 16 bits du nombre entier relatif -263 selon la représentation en valeur absolue et selon la représentation en complément à 2.

**Q 2.5 (1 pt)** On suppose ici que la représentation d'un nombre décimal s'effectue sous le format mantisse-exposant sur 32 bits selon la convention suivante : le premier bit est le bit de signe (0 pour positif et 1 pour négatif), les 8 bits suivants permettent de représenter l'exposant, et les 23 derniers bits sont utilisés pour représenter la mantisse. Donnez la représentation du nombre 344,2109375 selon cette convention.

## ANNEXE

**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 Range(Cell1 As Range, Cell2 As Range) As Range**

' *Lecture*

' *Plage dont les coins (supérieur gauche et inférieur droit) sont Cell1*

' *et Cell2 relativement à l'objet Range*

**Property Value As Variant**

' *Lecture-écriture*

' *Valeur de la cellule*

**Property FormulaR1C1Local As String**

' *Lecture-écriture*

' *Formule de la cellule (affichée dans la barre de formule)*

**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 Font**

La classe **Font** est la classe des styles de caractères. Entre autres, elle possède les propriétés suivantes :

---

**Property Color As Long**

' *Lecture-écriture*

' *Couleur (codée en RGB) des caractères.*

**Property Bold As Boolean**

' *Lecture-écriture*

' *Graisse des caractères.*

**Property Italic As Boolean**

' *Lecture-écriture*

' *Inclinaison des caractères.*

**Property Size As Integer**

' *Lecture-écriture*

' *Taille (en points) des caractères.*

---