

# Examen d'informatique

Lundi 28 janvier 2013

Durée : 2h

*Documents, calculatrices 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
- la dernière page de ce sujet est une annexe dans laquelle figure des extraits de votre polycopié de cours qui peuvent vous être utiles pour répondre à certaines questions

## Exercice 1 : Recommandation de films (16 pts)

On s'intéresse à la mise en oeuvre d'un système de recommandation de films. On dispose pour cela d'une liste de films dans une feuille Excel. Plusieurs utilisateurs ont évalué les films qu'ils ont vus parmi ceux de la liste en leur attribuant une note pouvant aller de 1 (s'ils ont détesté le film) à 5 (s'ils ont adoré le film). Ces informations sont stockées dans une feuille Excel où les films sont listés dans la première colonne (un film par ligne) et les notes des différents utilisateurs apparaissent à partir de la deuxième colonne (une colonne par utilisateur) telle que la note figurant à la cellule  $(i, j)$  est la note attribuée par l'utilisateur de la colonne  $j$  pour le film de la ligne  $i$  (la première ligne contient le nom des utilisateurs, et la première colonne le nom des films). Un exemple de feuille Excel stockant ces informations est donné dans la figure 1.

	1	2	3	4	5
1		Antoine	Bob	Claire	Denis
2	<i>Crazy</i>	-1	-1	4	-1
3	<i>Dracula</i>	-1	-1	4	4
4	<i>Intouchables</i>	5	3	2	-1
5	<i>Juno</i>	4	-1	-1	3
6	<i>Scream</i>	1	5	-1	1
7	<i>Titanic</i>	2	-1	-1	3

FIGURE 1 – Evaluation des films par les utilisateurs

On peut ainsi voir dans cet exemple que l'utilisateur *Antoine* a mis la note 1 au film *Scream* et la note 4 au film *Juno*. Lorsque l'utilisateur n'a pas donné de note à un film (il ne l'a peut-être pas vu), la valeur de la cellule correspondante est -1.

On appelle **BaseFilms** la base de notes de films sur laquelle on va travailler. Dans cette base, 49 utilisateurs ont noté 99 films, les données sont stockées dans la feuille Excel courante sous le format expliqué précédemment de la colonne 1 à la colonne 50 et de la ligne 1 à la ligne 100. La ligne 1 contient le nom des utilisateurs, la colonne 1 contient le nom des films. La plage des notes est donc la plage des cellules allant de la cellule (2,2) à la cellule (100,50).

**N.B.** : la base de notes de la figure 1 pourra être parfois utilisée à *titre d'illustration*, mais toutes les fonctions ou procédures qui vous sont demandées dans cet exercice portent sur la base **BaseFilms**.

### Partie 1. Consultation de la base **BaseFilms** (11 pts)

**Q 1.1 (1 pt)** Ecrivez en VBA la fonction `colUt`, qui prend en paramètre un nom d'utilisateur `nomU` et retourne le numéro de la colonne dans laquelle figure les notes de l'utilisateur `nomU`. La fonction `colUt` retourne -1 si `nomU` n'est pas présent dans la feuille Excel. On suppose que le nom de chaque utilisateur est unique.

**Q 1.2 (1 pt)** Ecrivez en VBA une macro d'appel de la fonction `colUt` qui, à l'aide de boîtes de dialogue, demande à l'utilisateur de saisir son nom puis affiche la colonne de l'utilisateur (où -1 si l'utilisateur n'existe pas dans la base **BaseFilms**).

**Q 1.3 (1 pt)** Ecrivez en VBA la macro `coloreJ` qui colore en jaune (`RGB(200,250,0)`) l'intérieur de toutes les cellules de la plage de notes dont la note est manquante (cellules dont la valeur est -1).

**Q 1.4 (3 pts)** Ecrivez en VBA la macro `moyFilms` qui calcule la moyenne des notes de chacun des films et met cette moyenne dans la colonne 51 de la feuille Excel (la moyenne du film à la ligne  $i$  apparaît donc dans la cellule  $(i, 51)$ ). La moyenne des notes d'un film est calculée à partir des notes données par les utilisateurs (sans prendre en compte les -1). Dans l'exemple de la figure 1, la moyenne du film *Intouchables* est  $(5 + 3 + 2)/3 = 3,33$  car les trois notes de ce film sont 5, 3 et 2.

**Q 1.5 (2 pts)** On suppose que la macro `moyFilms` a été exécutée et que les moyennes de chacun des films figurent donc dans la colonne 51. Ecrivez en VBA la macro `meilleurFilm` qui affiche dans une boîte de dialogue le titre du film ayant la meilleure moyenne (s'il y a plusieurs qui ont la meilleure moyenne, `moyFilms` affiche l'un d'entre eux uniquement).

**Q 1.6 (3 pts)** On suppose maintenant que les films sont listés dans la base **BaseFilms** dans l'ordre **décroissant** de leur moyenne (qui figure dans la colonne 51). Ecrivez en VBA une procédure `filmM` qui prend en paramètre une moyenne  $m$  et affiche dans une boîte de dialogue un film dont la moyenne est  $m$  ou le message "pas de film" s'il n'en existe pas ayant la moyenne  $m$ .

### Partie 2. Système de recommandation (5 pts)

Le système de recommandation s'appuie sur le principe suivant : le film recommandé à un utilisateur  $u_1$  est un film qu'il n'a pas vu (noté -1 par  $u_1$ ) et qui est parmi les films préférés d'un utilisateur  $u_2$  ayant les goûts les plus proches de ceux de  $u_1$ . Pour déterminer le film à recommander à  $u_1$ , on regarde parmi les films les mieux notés par  $u_2$  s'il en existe au moins un que  $u_1$  n'a pas vu. Si c'est le cas, un de ces films est recommandé. Sinon on regarde parmi les films un peu moins bien notés par  $u_2$  s'il en existe au moins un que  $u_1$  n'a pas vu, en quel cas on le recommande. Sinon, on continue ainsi en examinant les films de moins en moins bien notés jusqu'à trouver un film noté par  $u_2$  non vu par  $u_1$  (on suppose qu'il en existe toujours un).

Avant de déterminer le film à recommander à  $u_1$ , il faut donc déterminer l'utilisateur  $u_2$  dont les goûts sont les plus proches. Pour cela on définit le coefficient de proximité de deux utilisateurs comme la somme des produits des notes des films notés par les deux utilisateurs à la fois. Par exemple, dans la base de notes de la figure 1, la proximité de l'utilisateur *Antoine* avec l'utilisateur *Denis* vaut :

$$(4 \times 3) + (1 \times 1) + (2 \times 3) = 19$$

car *Antoine* et *Denis* ont tous deux noté les films *Juno* (4 pour *Antoine* et 3 pour *Denis*), *Scream* (1

pour *Antoine* et 1 pour *Denis*) et *Titanic* (2 pour *Antoine* et 3 pour *Denis*). Les autres films ne sont pas pris en compte dans ce calcul car ils ne sont pas notés par *Antoine* et *Denis* à la fois. Par exemple, la note qu'*Antoine* a donnée au film *Intouchables* n'est pas prise en compte dans ce calcul car *Denis* n'a pas noté ce film.

**Q 1.7 (2 pts)** Ecrivez en VBA la fonction `prox` qui, prend en paramètre deux noms d'utilisateurs de la base `BaseFilms` et retourne le coefficient de proximité de ces deux utilisateurs. La fonction `prox` retourne 0 si les deux utilisateurs passés en paramètre n'ont noté aucun film en commun. On suppose que les deux utilisateurs existent bien dans la base `BaseFilms`.

**Q 1.8 (3 pts)** On suppose qu'il existe en VBA la fonction `plusProche`, qui prend en paramètre le nom d'un utilisateur de la base `BaseFilms` et qui retourne le nom de l'utilisateur de la base `BaseFilms` (autre que lui-même) le plus proche parmi tous les utilisateurs de la base `BaseFilms`. Cette fonction étant déjà définie, vous pouvez l'utiliser sans avoir besoin de l'écrire.

Ecrivez en VBA la procédure `recom` qui prend en paramètre un utilisateur  $u$  de la base `BaseFilms` et affiche dans une boîte de dialogue le film qui lui est recommandé. Pour déterminer le film à recommander, `recom` détermine dans un premier temps l'utilisateur  $u_p$  le plus proche de  $u$  (on suppose qu'il en existe toujours un), puis détermine un film préféré de  $u_p$  parmi les films notés par  $u_p$  non vus par  $u$ . Pour déterminer ce film préféré, on peut utiliser l'algorithme suivant : on regarde parmi tous les films notés 5 par  $u_p$  s'il en existe au moins un non vu par  $u$ . Si c'est le cas on le recommande à  $u$  (s'il en existe plusieurs, il suffit d'en recommander un). Sinon, on regarde parmi tous les films notés 4 par  $u_p$  s'il en existe au moins un non vu par  $u$ . Si c'est le cas on en recommande un à  $u$ . Sinon, on continue avec les films notés 3, puis 2, etc.

---

## Exercice 2 : Binaire et tableau (4 pts)

---

On suppose dans cet exercice qu'un nombre binaire  $b$  sur 16 bits est stocké dans un tableau VBA unidimensionnel, appelé `tabBin`, de taille 16, tel que le bit à la  $i$ -ème position (en partant de la gauche) de  $b$  est dans la case  $i$  de `tabBin`. Le tableau `tabBin` est déclaré comme variable globale du programme par l'instruction suivante :

```
Dim tabBin (1 To 16) As Byte
```

Par exemple, si le nombre 1100011010101010 est stocké dans `tabBin`, alors la valeur de `tabBin(1)` est 1, celle de `tabBin(2)` est 1, celle de `tabBin(3)` est 0, celle de `tabBin(4)` est 0, ..., celle de `tabBin(15)` est 1 et celle de `tabBin(16)` est 0.

**Q 2.1 (1 pt)** On suppose que `tabBin` a été rempli de manière à représenter un nombre binaire (que l'on ne connaît pas). Ecrivez en VBA la fonction `div4` qui retourne vrai si le nombre stocké dans le tableau `tabBin` est divisible par 4 et faux sinon.

**Q 2.2 (3 pts)** Ecrivez en VBA la macro `saisirNb` qui permet à un utilisateur de stocker un nombre binaire  $b$  dans `tabBin`. Pour cela, la macro `saisirNb` demande à l'utilisateur de saisir successivement les 16 chiffres de  $b$  (du chiffre de droite à celui de gauche) et remplit `tabBin` de ces 16 chiffres. Chaque chiffre saisi par l'utilisateur doit être un 0 ou un 1. Si ce n'est pas le cas, la macro `saisirNb` affiche un message d'erreur qui redemande à l'utilisateur de saisir un chiffre jusqu'à ce que l'utilisateur saisisse un 0 ou un 1. Les messages à afficher dans les boîtes de dialogue sont les suivants :

- message de demande d'un nouveau chiffre : “Saisir le chiffre à la position 1”, “Saisir le chiffre à la position 2”, “Saisir le chiffre à la position 3”, etc.
- message d'erreur lors de la saisie du  $i$ -ème chiffre : “Resaisir le chiffre  $i$ ” (où  $i$  est un nombre entre 1 et 16)

# 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** `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*

---

## 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.*

---

## Fonction `Application.InputBox`

---

**Function** `Application.InputBox (Prompt As String, _  
Optional Title As String, _  
Optional Default As String, _  
Optional Left As Integer, _  
Optional Top As Integer, _  
Optional HelpFile As String, _  
Optional HelpContextId As Variant, _  
Optional Type As Integer) As Variant`

' *Affiche une boîte de dialogue qui comporte un cadre de saisie et des*

' *boutons OK et Annuler et renvoie les informations saisies.*

---

Valeur	Type de données renvoyé
0	Formule
1	Valeur numérique
2	Chaîne de caractères
4	Valeur booléenne ( <b>True</b> ou <b>False</b> )
8	Référence de cellule (objet <b>range</b> )
16	Valeur d'erreur

---

TABLE 1 – Valeurs du paramètre `Type` de `Application.InputBox`