# Examen de rattrapage d'informatique

## Mardi 30 juin 2015

Durée : 1h30

*Documents, calculettes et téléphones portables interdits*

**Remarques :**
- 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 : Nombres heureux (16 pts)**

---

Un nombre entier strictement positif est dit *heureux* lorsque si l'on calcule la somme des carrés des chiffres qui le composent puis la somme des carrés des chiffres du nombre obtenu et ainsi de suite, on aboutit au nombre 1. Par exemple, 10 est un nombre heureux car $1^2+0^2 = 1+0 = 1$. Le nombre 97 est aussi un nombre heureux car $9^2 + 7^2 = 81 + 49 = 130$, et $1^2 + 3^2 + 0^2 = 10$, et $1^2 + 0^2 = 1$.

**Q 1.1 (0.5 pt)** Montrer que 7 est un nombre heureux.

**Q 1.2 (3 pts)** Ecrire en VBA la fonction `sommeCar` qui prend un entier `n` strictement positif en paramètre et retourne la somme des carrés des chiffres de `n`. Si l'entier passé en paramètre n'est pas strictement positif, la fonction `sommeCar` retourne -1.

On souhaite à présent déterminer les nombres heureux entre 1 et 1 000. Pour cela, les nombres de 1 à 1 000 sont stockés dans la première colonne de la feuille Excel à raison d'un nombre par ligne, à partir de la première ligne. Ensuite, le procédé suivant va être appliqué sur chaque ligne $i$ (de la ligne 1 à la ligne 1 000) :
- la somme des carrés des chiffres du nombre de la cellule L$i$C1 est placée en L$i$C2
- si le nombre déterminé en L$i$C2 ne vaut pas 1, on réitère le procédé en plaçant dans la cellule L$i$C3 la somme des carrés des chiffres du nombre placé en L$i$C2
- si le nombre en L$i$C3 ne vaut pas 1, on réitère le procédé dans la cellule suivante, et ainsi de suite...

On s'autorise un maximum de 100 itérations (i.e. on peut aller jusqu'à la colonne 101). On considère que si le nombre 1 n'a jamais été trouvé dans les différentes cellules de la colonne 1 à 101, alors le nombre en L$i$C1 n'est pas un nombre heureux.

**Q 1.3 (0.5 pt)** Ecrire en VBA la macro `initialise` qui remplit la première colonne avec les nombres de 1 à 1 000, à partir de la première ligne.

**Q 1.4 (3 pts)** On suppose que la macro `initialise` a été exécutée. Ecrire en VBA la macro `remplit` qui, pour chaque nombre placé en colonne 1, remplit les cellules des colonnes suivantes sur la même ligne selon le procédé décrit précédemment. A titre d'illustration, la figure 1 présente le remplissage d'une partie de la feuille Excel (plage de cellules L96C1 :L101C10) après exécution des macros `initialise` puis `remplit`.
Dans la suite de l'exercice, on suppose que les macros `initialise` et `remplit` ont été exécutées.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 96 | 96 | 117 | 51 | 26 | 40 | 16 | 37 | 58 | 89 | 145 |
| 97 | 97 | 130 | 10 | 1 | | | | | | |
| 98 | 98 | 145 | 42 | 20 | 4 | 16 | 37 | 58 | 89 | 145 |
| 99 | 99 | 162 | 41 | 17 | 50 | 25 | 29 | 85 | 89 | 145 |
| 100 | 100 | 1 | | | | | | | | |
| 101 | 101 | 2 | 4 | 16 | 37 | 58 | 89 | 145 | 42 | 20 |

FIGURE 1 – Itérations de calcul

**Q 1.5 (2 pts)** Ecrire en VBA la macro `heureux` qui colore en rouge (`RGB (200, 10, 10)`) **l'intérieur** des cellules de la première colonne qui contiennent un nombre heureux, puis affiche le nombre total de nombres heureux entre 1 et 1 000 (inclus).

**Q 1.6 (2 pts)** Ecrire en VBA la procédure `nbIte` qui prend un numéro de colonne `co` en paramètre, et remplit la colonne `co` de la ligne 1 à la ligne 1 000 en indiquant le nombre d'itérations qu'il a fallu pour déterminer si le nombre en colonne 1 à la ligne correspondante était heureux ou non. On suppose que le nombre `co` est strictement plus grand que 1. Par exemple, pour le nombre 97 à la ligne 97, la procédure `nbIte` placerait le chiffre 3 dans la colonne `co` à la ligne 97 car il a fallu 3 itérations pour déterminer que 97 était un nombre heureux.

**Q 1.7 (4 pts)** Ecrire la macro `tri` qui trie les nombres de la colonne 1 (1 à 1 000) en fonction du nombre d'itérations qu'il faut pour déterminer s'ils sont heureux ou non. La macro `tri` commence par demander à l'utilisateur de saisir un numéro de colonne strictement supérieur à 1. Tant que le nombre saisi par l'utilisateur n'est pas strictement supérieur à 1, la macro redemande à l'utilisateur de saisir un numéro de colonne. Soit $c$ le numéro de colonne saisi par l'utilisateur, la macro tri place alors dans la colonne $c$ de la ligne 1 à la ligne 1 000 le nombre d'itérations qu'il a fallu pour montrer que le nombre était heureux ou non, puis elle trie les nombres de la colonne 1 (dans la colonne 1) dans l'ordre croissant de leur nombre d'itérations indiqué dans la colonne $c$. Les seules cellules qui peuvent être modifiées dans cette question sont celles situées en colonne 1 et en colonne $c$. Dans cette question, la notation ne tient pas compte du choix de l'algorithme de tri.

**Q 1.8 (1 pt)** Donner l'ordre de grandeur du nombre d'opérations de la procédure `nbIte` et de la macro `tri`. Justifier brièvement.

## Exercice 2 : Représentation binaire (4 pts)

**Q 2.1** Ecrire en VBA la fonction `decToBin` qui prend en paramètre un nombre entier positif et retourne sa représentation binaire. Par exemple `decToBin(57)` retournerait le nombre 111001.
**NB :** dans cette question, tous les nombres doivent être de **type numérique**.

**Q 2.2** Ecrire en VBA la fonction `divPar8` qui prend en paramètre une chaîne de caractères représentant un nombre binaire et retourne vrai si le nombre est divisible par 8 et faux sinon. Par exemple, `divPar8("111001")` retourne faux.
**NB :** pour cette question, il est possible d'utiliser les fonctions VBA suivantes :
– `Len(s)` : retourne la taille de la chaîne de caractères s
– `Mid(s,i,1)` : retourne le $i$ème caractère de la chaîne s