

CONTRÔLE CONTINU 2

Partie I

Nombres binaires

- a) Combien de bits faut-il, au minimum, pour représenter le nombre six en écriture binaire ? Combien en faut-il, au minimum, pour le représenter en complément à 2 ? Combien en faut-il pour représenter le nombre moins treize en complément à deux ?
- b) Représentez six et moins treize en complément à 2
- c) Calculez leur somme (calcul posé).
- d) Que peut-on faire pour vérifier le résultat ? Faites-le.

Partie II

Tableaux : Happy numbers

Un nombre entier positif (strict) est dit *happy number* si la somme des carrés des chiffres qui le composent est égale à 1. Par exemple, 10 est un *happy number*, puisque $1^2 + 0^2 = 1 + 0 = 1$.

Si le résultat obtenu par cette somme n'est pas 1, mais qu'elle le devient en itérant ce procédé, le nombre est aussi dit *happy number*. Par exemple, 97 est *happy*, puisque : $9^2 + 7^2 = 81 + 49 = 130$; puis, en itérant le procédé, on obtient $1^2 + 3^2 + 0^2 = 1 + 9 + 0 = 10$, qui est *happy*.

On appellera $h : \mathbb{N} \rightarrow \mathbb{N}$, la fonction qui calcule la somme des carrés des chiffres qui composent l'écriture en base dix d'un nombre (ci-dessus appelée *le procédé*).

On veut étudier ces nombres en exploitant la puissance de calcul de la machine.

Exercice 1 Premiers pas

- a) Vérifiez que 7 est *happy*. Combien d'itérations du procédé sont-elles nécessaires ?
- b) Ecrivez une fonction VBA qui calcule h , à partir d'un nombre entier positif.
- c) Ecrivez une macro VBA qui teste si 7 est bien un nombre *happy* pour le nombre d'itération de h que vous avez prévu à la question 1.a. La macro affichera deux messages différents selon qu'elle confirme ou non ce que vous aviez prévu.

Exercice 2 Première exploration

On veut chercher les *happy numbers* entre 1 et 999 avec VBA. Pour cela, on va tenter d'itérer le calcul de h sur tout ces nombres, mais en se limitant à un maximum de 100 itérations.

- a) Ecrivez une procédure VBA qui :
 - Stockera dans un tableau l'avancée des calculs pour chaque nombre entre 1 et 999
 - Au bout de 100 itérations, affichera dans la première ligne de la page Excel, tout les nombres *happy* découverts

Ainsi, le tableau doit débiter son évolution comme indiqué ci dessous :

1		1		1	
⋮		⋮		⋮	
129		86		100	
130	\rightarrow_h	10	\rightarrow_h	1	$\rightarrow_h \dots$
131		11		2	
⋮		⋮		⋮	
999		$h(999)$		$h(h(999))$	

- b) Modifiez la procédure précédente pour qu'elle fournisse le numéro de l'itération à laquelle on découvre qu'un nombre est *happy*. Vous procéderez de la façon suivante :
- Lorsque h abouti *pour première fois* à la valeur 1, vous stockerez, non plus la valeur de h , mais le numéro de l'itération, en négatif, afin de le distinguer des calculs de h usuellement mémorisés
 - Au cours des étapes qui restent pour arriver à 100, vous conserverez cette information
 - Au moment de l'affichage final, sous chaque nombre *happy*, dans la deuxième ligne du tableau Excel, vous indiquerez le nombre d'itérations qui ont été nécessaires

NB Vous pourrez marquer sur l'exercice précédent le/les bloc à remplacer/modifier

Exercice 3 Exploration Complète

On remarque que si $x \leq 999$, alors $h(x) \leq h(999) = 9^2 + 9^2 + 9^2 = 3 \times 81 = 243$. On est donc sûr que si l'on utilise l'exploration précédente sur au moins 243 itérations, tout les nombres *happy* atteindront la valeur 1.

De leur coté, les nombres qui ne sont pas *happy*, on dira *sad*, atteindront, lors de la 243-ème itération, une valeur par laquelle ils sont déjà passés au moins une fois. Par exemple, 89 est *sad*, et :

$h(89) = 145$; $h(145) = 42$; $h(42) = 20$; $h(20) = 4$; $h(4) = 16$; $h(16) = 37$; $h(37) = 58$; $h(58) = 89$; $h(89) = 145$; ... On dit que 89 a atteint une *boucle de spleen* de longueur 8.

On veut trouver les boucles qui caractérisent les nombres *sad* entre 1 et 999.

- a) Modifiez la procédure VBA du 2.a de façon à utiliser un tableau à deux dimensions qui mémorise toutes les étapes du calcul. (on ne se souci plus de l'affichage, ni de la question 2.b)

Ainsi, les premières colonnes du tableau ressembleront à :

1	1	1	
⋮	⋮	⋮	
129	86	100	
130	1	1	...
131	11	2	
⋮	⋮	⋮	
999	h(999)	h(h(999))	

- b) Ecrivez à présent une procédure VBA qui affiche les boucles de *spleen*. Pour cela :
- Appelons *sad ending* le dernier nombre disponible dans l'historique pour un nombre *sad*.
 - Supposez que vous disposez, en variable globale, d'un tableau nommé *historique*, qui contient l'information collectée lors de la question précédente
 - Pour chaque nombre *sad*, affichez sur une nouvelle ligne de la feuille Excel, tout les nombres rencontrés, en remontant de son *sad ending*, jusqu'à la précédente apparition de ce *sad ending* dans l'historique de ce nombre *sad*

Exercice 4 Optimisation

On sait trouver les nombres *happy*. On sait expliquer le *spleen* des nombres *sad*. On veut à présent faire le travail sans avoir à utiliser un énorme tableau d'historique, et éventuellement en diminuant le nombre de calculs.

Voici les limitations imposées :

- Vous devez construire un tableau tel qu'à la question 2.a, mais uniquement pour une itération de h
- Vous pouvez utiliser un tableau pour stocker qui est *happy* et qui est *sad*
- Vous pouvez utiliser un tableau de travail pour l'historique, mais à une seule dimension

Vous décrirez d'abord toutes vos idées, en français. Vous pourrez ensuite formaliser vos idée par du pseudo-code, ou des morceaux de VBA. Seulement après cela, vous proposerez une procédure VBA.