VBA : Boucles

Il est possible que lors de l'exécution d'une de vos macros, celle-ci reste bloquée (cela arrive quand le programme ne sort jamais d'une boucle). Dans ce cas, vous pouvez l'arrêter avec les touches CTRL Pause.

Une entreprise gère son parc automobile grâce à un fichier excel. On veut automatiser sa gestion grâce à des macros pour rendre son utilisation plus simple. Le fichier contient :

- Une feuille *Menu* à partir de laquelle toutes les opérations seront effectuées par simple clic sur des boutons. Cette feuille contiendra aussi une liste déroulante pour choisir un des véhicules du parc.
- Une feuille *ListeVéhicules* contenant la liste de tous les véhicules (pour chacun : un numéro, le type de véhicule, l'immatriculation et le nom complet formé du type et de l'immatriculation).
- Pour chaque véhicule, une feuille *VéhiculeN* (où *N* est le numéro du véhicule). Cette feuille contient le type et l'immatriculation ainsi que toutes les dépenses effectuées sur ce véhicule.
- Les feuilles *dépenses essence*, *dépenses par nature* et *dépenses par année* permettront d'obtenir les dépenses en essence, par nature et par année d'un véhicule choisi.
- Une feuille *Véhicule* qui servira de modèle pour créer de nouveaux véhicules.

Récupérez le fichier *TP-VBA-3.xlsm* disponible sur Célène.

1. Le module 1 de l'éditeur Visual Basic contient une macro **Auto_open**. Lisez ses commentaires.
2. La macro **RetourMenu** active la feuille *Menu*. Dans la feuille *ListeVéhicules*, placez un bouton étiqueté *Retour au Menu* qui déclenche cette macro. Testez.
3. Le module 1 contient une macro **compteveh**. Lisez-la et comprenez-la bien. Elle utilise une boucle For Each…
4. Modifiez cette macro pour qu'elle ne compte que le nombre de feuilles dont le nom finit par un chiffre. Pour cela, vous disposez de :
    a. `f.Name` qui donne le nom de la feuille `f`
    b. `right(s,n)` qui donne les `n` derniers caractères de la chaîne de caractère `s`.
    c. `IsNumeric(s)` qui retourne `True` si la chaîne de caractère `s` est un numérique.
    d. `If`…
5. Ecrire une macro **totalv1** qui calcule et affiche (dans une boîte de message) le **montant total des dépenses** de la feuille *Véhicule1*.
6. Attention, cette macro doit fonctionner même si vous l'exécutez depuis la feuille *Menu*. Si ce n'est pas le cas, modifiez-la.
7. En vous inspirant de cette dernière macro et de la macro **compteveh**, créez une macro **total** qui calcule et affiche dans la cellule D12 de *Menu* le total des dépenses du fichier (pour tous les véhicules).
8. Dans la feuille *Menu*, créez une zone de liste déroulante (avec *Insérer* de l'onglet *Développeur*) qui permet de choisir un véhicule (le NOM COMPLET) parmi les véhicules de la feuille *ListeVéhicules*. Notez bien la cellule liée, elle va être très utile pour la suite.

    Pour vérifier, créer une macro **liste** qui affiche dans une boîte de message le nom de la feuille correspondant au véhicule choisi. Associer cette macro à la liste déroulante.
9. Créez une macro **dernierevidange** qui affiche la date de la dernière vidange du véhicule choisi dans la liste dans une cellule (utilisez une boucle !). Appeler cette macro depuis **liste** (utilisez Call).

    Pour les macros suivantes, le calcul devra toujours être fait pour le véhicule sélectionné dans la liste déroulante.

10. Ecrire une macro qui indique la date de la **première vidange** dans la cellule D14. *Résultat pour CLIO RSI : 17/02/2010.* Appeler cette macro à partir de la macro **liste**.

11. Ecrire une macro pour **ajouter une dépense**. Elle doit demander à l'utilisateur, le kilométrage, la nature et le montant de la dépense puis ajoute ces données **à la fin** de la liste des dépenses du véhicule choisi avec la date actuelle (utilisez la fonction *Date*). Associer un bouton à cette macro.

12. Ecrire une macro qui demande un kilométrage de début et un kilométrage de fin à l'utilisateur puis qui calcule le **montant total des dépenses entre ces deux kilométrages**.

    a. Dans un premier temps, on considère que ces deux kilométrages doivent être des valeurs du tableau. Associez un bouton à cette macro.

    b. Ecrire une fonction `minkm` qui retourne le premier kilométrage de la feuille passée en paramètre. Puis une fonction `maxkm` qui retourne le dernier kilométrage.

    c. Reprenez la macro de la question 12a. Modifiez-la de telle façon qu'on redemande un kilométrage de départ à l'utilisateur jusqu'à ce que la valeur saisie soit comprise entre `minkm` et `maxkm`. Idem avec le kilométrage d'arrivée.

13. Ecrire une macro qui recopie dans le tableau de la feuille *dépenses essence* toutes les dépenses en essence du véhicule choisi. Attention, on ne veut voir que les dépenses d'essence et ces dépenses doivent être recopiées à la suite les unes des autres (sans lignes vides). La dernière instruction de la macro doit basculer sur la feuille « *dépenses essence* ». Associer un bouton à la macro.

14. Créer une macro **nettoyage** « nettoyant » la feuille *dépenses essence*, c'est-à-dire effaçant le véhicule, son immatriculation et toutes les lignes à partir de la ligne 5.

    Transformez cette macro en procédure afin qu'elle fonctionne avec un nom de feuille passé en paramètre. Appeler cette procédure au début de la macro précédente.

15. Ecrire une macro qui, à partir des dates de dépense, va reporter toutes les **années** dans la table *dépenses par année*. On pourra utiliser la fonction *year* qui renvoie l'année d'une date. Par exemple : `year (Cells(2,1))` renvoie 2009. N'oubliez pas d'appeler la procédure **nettoyage** au début de vote macro. La dernière instruction doit basculer sur la feuille « *dépenses par année* ».

Année
2009
2010
2011

    Modifier cette macro pour faire apparaître le total dépensé chaque année dans la colonne B. Associer cette macro à un bouton.

Année	Dépenses
2009	748,00 €
2010	2 204,00 €
2011	2 901,00 €

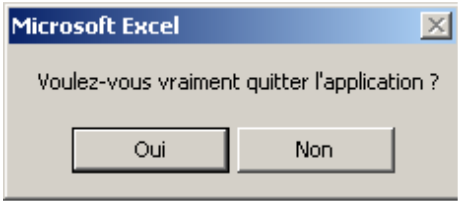
16. Ecrire une macro qui crée le tableau récapitulatif des **natures des dépenses** (sur la feuille *dépenses par nature*). **Attention :** chaque type de dépenses ne doit apparaître **qu'une fois**.

    Modifier cette macro pour afficher dans la colonne B le montant total des dépenses pour chaque nature d'opération. Associer un bouton à cette macro.

Nature	Total
assurance	1940 €
essence	3602 €
vidange	185 €
…	…

17. Reprenez la macro **quitter** (au début du *Module1*). Associer lui un bouton nommé « Quitter » sur la feuille *Menu*. Enregistrez puis testez votre bouton.

    Modifier cette macro pour qu'elle demande confirmation avant de quitter l'application (comme ci-contre).



Pour les questions à l'utilisateur pour lesquelles vous attendez comme réponse *Oui* ou *Non*, vous pouvez remplacer la boîte de saisie par une boîte de message sous la forme suivante :

```
reponse = MsgBox ("Voulez-vous vraiment quitter l'application?", vbYesNo)
```

Vous obtiendrez alors la boîte de message ci-dessus avec les boutons *Oui* et *Non*. Pour connaître la réponse : le bouton *Oui* correspond à la valeur `vbYes` (ou 6), le bouton *Non* à la valeur `vbNo` (ou 7).
Attention : la variable `reponse` est de type entier !
NB : il existe d'autres boîtes de message, vous trouverez les détails dans l'aide sur MsgBox.

18. Créer une macro qui permet de copier la feuille d'un véhicule dans un nouveau fichier Excel nommé par le nom du véhicule. Faites cela en enregistrant une macro à la souris puis en la modifiant. Associez un bouton à cette macro.
19. Créer une macro qui ajoute un véhicule. La macro doit réaliser 4 actions :
    a. Demander les informations sur le nouveau véhicule (type de véhicule et immatriculation)
    b. Ajouter le véhicule dans la feuille *ListeVéhicules* (en modifiant aussi le nombre de véhicules)
    c. Mettre à jour la zone de liste déroulante de la feuille *Menu*
    d. Créer une feuille pour le nouveau véhicule, par copier-coller de la feuille modèle *Véhicule*.

    Pour effectuer c et d, enregistrer une macro à la souris que vous modifierez ensuite.

    Associer un bouton à cette macro.
20. En prenant un prix moyen de 1€ par litre d'essence, écrire une macro qui calcule la **consommation moyenne** d'un véhicule (en litres pour 100 km). Pour le kilométrage, on prendra le kilométrage de la dernière dépense moins celui de la première. *Résultat pour CLIO RSI : 6,13 L/100km*. Afficher le résultat en D22.

> Le type `Integer` est limité aux entiers inférieurs à 32768. Pour pouvoir utiliser des entiers de plus grande taille, on utilise le type `Long`.
>
> Pour afficher le résultat avec 2 décimales, utilisez la fonction `Round(expression, nb)` qui arrondi l'`expression` au nombre de décimales donné par `nb`).

21. On veut remplir automatiquement la feuille *ListeVéhicules* en récupérant les informations dans les feuilles *VéhiculeN* où *N* est un nombre.