

Créer un classeur Excel vierge, et commencer par enregistrer le fichier (prenant en charge les macros) sous le nom *TP3_nom1_nom2*. Chaque exercice doit être résolu dans une feuille différente portant son nom.

Exercice 1. Prise en main UserForm. Dans VBA, insérer un *Module* ainsi qu'un *UserForm*.

a. Dans le module, écrire une macro qui affiche la UserForm à l'aide de la commande :

```
UserForm1.Show
```

Essayer aussi la commande **UserForm1.Show 0** et comparer. Ajouter deux boutons sur la feuille Excel, un nommé "Afficher" et l'autre "Effacer" qui respectivement affiche et ferme (**Unload UserForm1**) la UserForm.

b. Dans la UserForm1, on ajoutera du code propre à la UserForm (clique-droit puis "Code"). Essayer le code suivant :

```
Private Sub UserForm_Initialize()
```

```
Me.StartUpPosition = 3
```

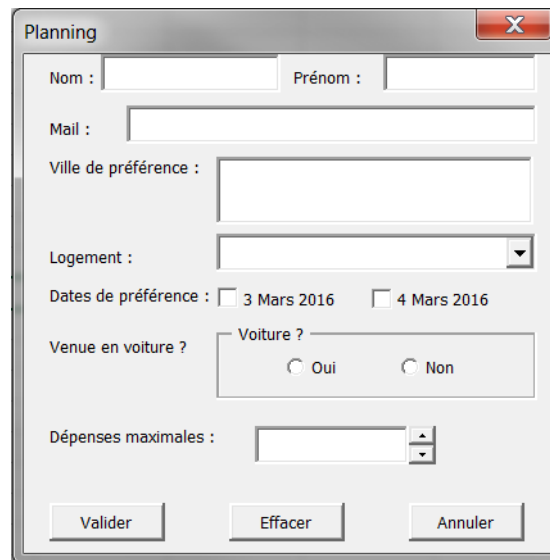
```
End Sub
```

Essayer ensuite les valeurs 0, 1 et 2.

Exercice 2. Planning. Remplir la première ligne d'une nouvelle feuille Excel comme suit :

	A	B	C	D	E	F	G	H
1	Nom	Prénom	Email	Ville préférée	Logement	Date(s)	Voiture	Dépenses max
2								

Créer un UserForm nommé **PlanningUserForm** qui contient les éléments suivants :



Au moment de l'initialisation de l'UserForm, on souhaite :

a. remettre à défaut toutes les valeurs. On utilisera les formules suivantes :

```
TextBox.Value=""      ListBox.Clear      ComboBox.Clear      CheckBox.Value=False
```

b. ajouter les différentes villes "Montpellier, Troyes, Sceaux", les logements "Co-habitation, Hôtel, Appartement". On utilisera la formule :

```
ListBox.AddItem "Montpellier"      ComboBox.AddItem "Co-habitation"
```

Remarque : On peut s'éviter d'écrire trois fois de suite `ListBox.AddItem ...`, en utilisant :
`With ListBox`

```
.AddItem "Montpellier"
```

```
.AddItem ...
```

```
End With
```

On va maintenant assigner des macros pour interagir avec l'UserForm. Pour cela on peut double cliquer sur l'un des boutons :

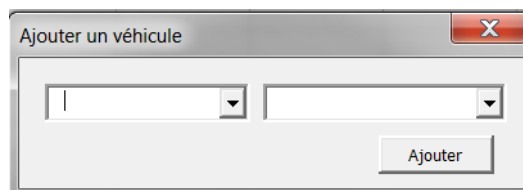
- écrire la macro qui permet d'afficher la valeur du SpinButton (quand on change sa valeur) dans le TextBox juxtaposé,
- écrire la macro qui ferme l'UserForm quand on clique sur "Annuler",
- écrire la macro qui réinitialise l'UserForm quand on clique sur "Effacer",
- écrire la macro qui ajoute à la dernière ligne de la feuille Excel (la première ligne non vide) les données du formulaire. Tester votre programme pour obtenir cette ligne :

	A	B	C	D	E	F	G	H
1	Nom	Prénom	Email	Ville préférée	Logement	Date(s)	Voiture	Dépenses max
2	Dupond	Thibaut	thibaut.dupond@gmail.com	Troyes	Hôtel	3 mars 4 mars	1	50
3								

On peut tester si un CheckBox est coché à l'aide de la commande "CheckBox.Value=True", et la commande "CheckBox.Caption" permet de renvoyer la valeur texte de la CheckBox.

Exercice 3. Remplir une nouvelle feuille Excel et créer un UserForm **VoitureUserForm** comme suit :

	A	B	C
1	Marque	Véhicule	Nombre
2	Citroen	C1	0
3	Citroen	C3	0
4	Citroen	C4	0
5	Citroen	C5	0
6	Citroen	DS5	0
7	Peugeot	208	0
8	Peugeot	308	0
9	Peugeot	508	0
10	Peugeot	2008	0
11	Renault	Twingo	0
12	Renault	Clio	0
13	Renault	Megane	0
14	Renault	Scenic	0
15	Renault	Espace	0



- A l'initialisation de l'UserForm, remplir le ComboBox1 des éléments "Citroen, Peugeot, Renault".
- Ecrire le code, exécuté au moment du *changement* de la ComboBox1, qui remplit la ComboBox2 des éléments suivant : C1, C3, C4, C5, DS5 si on a sélectionné Citroen ; 208, 308, 508, 2008 si on a sélectionné Peugeot ; Twingo, Clio, Megane, Scenic, Espace si on a sélectionné Renault. Pour ce conditionnement on utilisera la commande :

```
Case Is = 0
```

```
With ComboBox2 ...
```

```
End With
```

```
Case Is = 1 ...
```

```
End Select
```

La variable *ind* contient le numéro de la ligne sélectionnée dans la ComboBox1.

- Ecrire le code, exécuté quand on clique sur "Ajouter", qui permet d'ajouter 1 au total correspondant à la ComboBox2 sur la feuille Excel.

AIDE VBA	
WorksheetFunction.CountA(Range("A :A"))	Compte le nombre de lignes non vides de la colonne A
ComboBox1.ListIndex	N° ligne sélectionnée d'une ComboBox (commençant par 0)