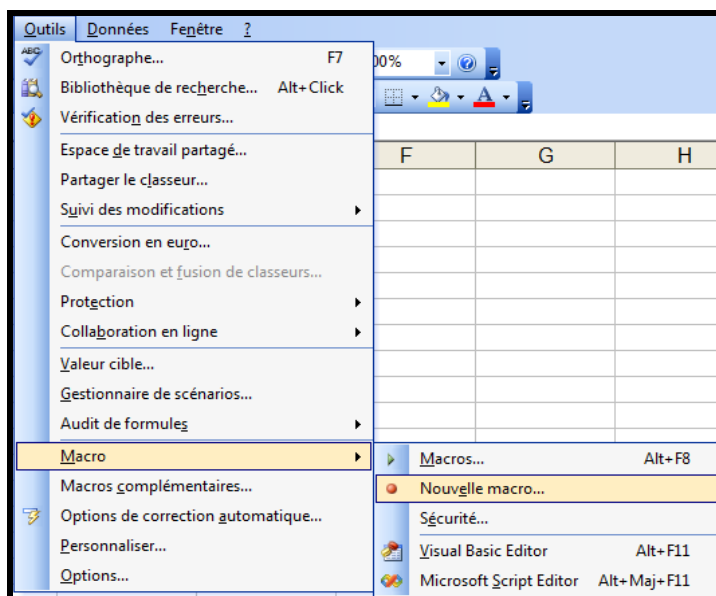


Sommaire

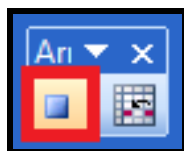
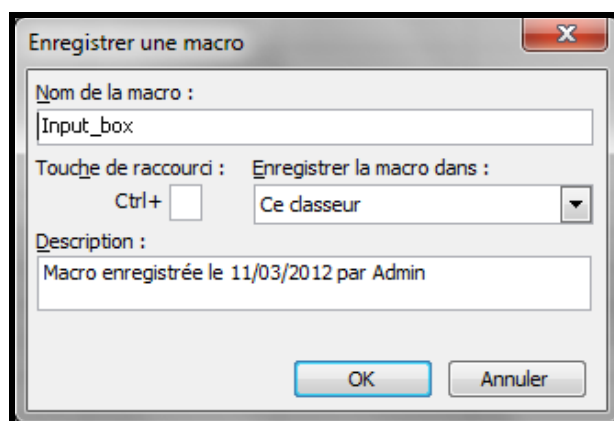
A) - CRÉER MACRO par SAISIE Code VISUAL BASIC	2
1) - Chargement d'une variable avec InputBox	2
2) - Chargement d'une variable par InputBox et Sauvegarde avec celle-ci	5
3) - Charger une variable Nombre avec InputBox et utiliser cette variable	5
4) - Fonction If Then..Else..End If	6
5) - Fonction If Then..Else..End If et Traitement des Erreurs	8

A) - CRÉER MACRO par SAISIE Code VISUAL BASIC**1) - Chargement d'une variable avec InputBox**

Pour accéder à la création de macro, cliquez dans la barre des menus sur "**Outils**", puis sur "**Macro**", et enfin sur "**Nouvelle macro...**".

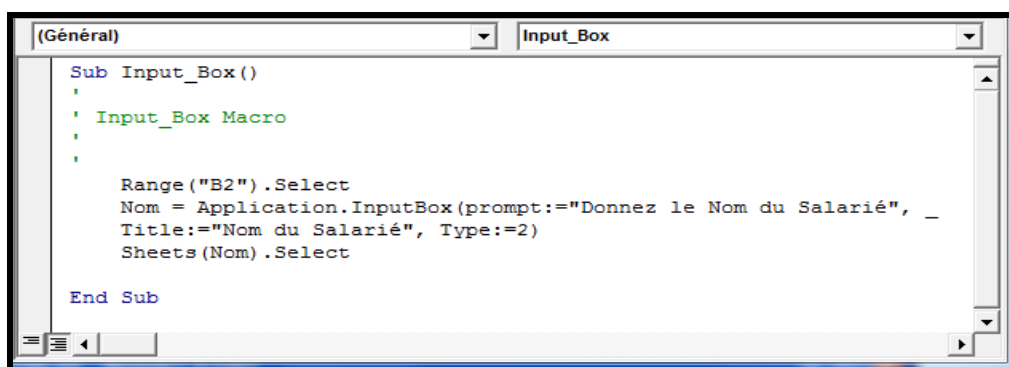


Dans la boîte de dialogue qui apparaît, entrez dans la zone "**Nom de la macro**", le titre que vous allez donner à votre macro (par exemple **Input_Box**), puis dans la zone "**Enregistrer la macro dans**", l'endroit d'où sera accessible cette macro. Soit dans le *Classeur personnel*, soit dans un *nouveau classeur*, ou bien dans *ce classeur*. Nous choisissons cette dernière option et nous cliquons sur le bouton "**OK**" pour confirmer.

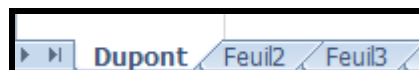
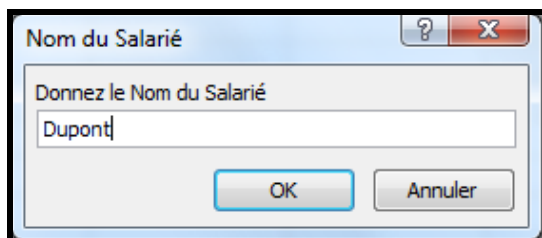


Une nouvelle barre d'outils apparaît sur votre feuille Excel avec le bouton "**Arrêter l'enregistrement**". Cliquez dessus, pour mettre fin à l'enregistrement des commandes dans la macro

Pour effectuer l'insertion d'une variable dans une cellule, il faut utiliser la commande **Application.InputBox** (**prompt** := "*Texte se rapportant à la variable*", **Title** := "*Nom de la boîte de dialogue*", **Type**:= x).



Dans cet exemple nous chargeons dans la variable **Nom** la valeur saisie (Format *Texte*) et nous sélectionnons, avec cette valeur, la feuille portant le même nom. Si vous lancez cette macro, vous obtenez la boîte de dialogue ci-dessous, dans laquelle vous devez saisir le texte ou la valeur demandée par le texte mis dans la zone grisée (correspond à l'argument *Prompt*). Dans le bandeau bleu, apparaît le texte correspondant à l'argument **Title**. Si vous saisissez le nom **Dupont** et que vous cliquez sur le bouton "OK", la macro sélectionne la feuille portant le même nom que celui saisi.



Les explications concernant cette fonction **InputBox**, vous sont données dans la table ci-dessous.

Référence du développeur Excel			
Application.InputBox, méthode			
Afficher tout			
Cette méthode affiche une boîte de dialogue dans laquelle l'utilisateur peut saisir des données. Elle renvoie les informations saisies dans la boîte de dialogue.			
Syntaxe			
expression.InputBox(Prompt, Title, Default, Left, Top, HelpFile, HelpContextID, Type)			
<i>expression</i> Variable qui représente un objet Application .			
Paramètres			
Nom	Obligatoire/Facultatif	Type de données	Description
<i>Prompt</i>	Obligatoire	Chaîne	Message à afficher dans la boîte de dialogue. Il peut s'agir d'une chaîne de caractères, d'un nombre, d'une date ou d'une valeur booléenne (Microsoft Excel contraint la valeur à avoir le type de données String avant de l'afficher).
<i>Title</i>	Facultatif	Variante	Titre de la zone d'entrée. Si vous ne spécifiez pas cet argument, le titre par défaut est « Entrée ».
<i>Default</i>	Facultatif	Variante	Spécifie une valeur qui apparaît dans la zone de texte au moment de l'affichage initial de la boîte de dialogue. Si vous ne spécifiez pas cet argument, la zone de texte reste vide. Cette valeur peut être un objet Range .
<i>Left</i>	Facultatif	Variante	Spécifie une position x pour la boîte de dialogue, exprimée en points et calculée à partir du coin supérieur gauche de l'écran.
<i>Top</i>	Facultatif	Variante	Spécifie une position y pour la boîte de dialogue, exprimée en points et calculée à partir du coin supérieur gauche de l'écran.
<i>HelpFile</i>	Facultatif	Variante	Nom du fichier d'aide pour cette zone d'entrée. Si les arguments <i>FichierAide</i> et <i>HelpContextID</i> sont présents, un bouton ? apparaît dans la boîte de dialogue.
<i>HelpContextID</i>	Facultatif	Variante	Numéro d'identificateur contextuel de la rubrique d'aide dans <i>FichierAide</i> .
<i>Type</i>	Facultatif	Variante	Spécifie le type des données renvoyées. Si vous ne spécifiez pas cet argument, la boîte de dialogue renvoie du texte.

L'argument **Type** peut avoir les valeurs du tableau ci-après, selon que nous saisissons, une *Formule*, un *Nombre*, du *Texte*, une valeur *Logique*, Etc...

Les tableaux suivants répertorient les valeurs pouvant être passées dans l'argument Type. Il peut s'agir d'une seule valeur ou d'une somme de valeurs. Par exemple, pour une zone d'entrée qui peut accepter du texte et des nombres, affectez la valeur 1 + 2 à <i>Type</i> .	
Valeur	Signification
0	Une formule.
1	Un nombre.
2	Texte (une chaîne).
4	Une valeur logique (True ou False).
8	Une référence de cellule, sous la forme d'un objet Range .
16	Une valeur d'erreur, telle que #N/A.
64	Un tableau de valeurs.

Remarques concernant INPUTBOX**Remarques**

Utilisez la méthode **InputBox** pour afficher une simple boîte de dialogue afin d'entrer des informations qui seront utilisées dans une macro. La boîte de dialogue possède un bouton **OK** et un bouton **Annuler**. Si vous cliquez sur le bouton **OK**, la méthode **InputBox** renvoie la valeur entrée dans la boîte de dialogue. Si vous cliquez sur le bouton **Annuler**, la méthode **InputBox** renvoie la valeur **False**.

Si **Type = 0**, **InputBox** renvoie la formule sous la forme de texte, par exemple, " $=2*PI()/360$ ". S'il existe des références dans cette formule, elles sont renvoyées sous forme de références de style A1. (Utilisez la méthode **ConvertFormula** pour changer de style de références.)

Si **Type = 8**, **InputBox** renvoie un objet **Range**. Vous devez utiliser l'instruction **Set** pour affecter le résultat à un objet **Range**, comme illustré ci-dessous :

Visual Basic pour Applications

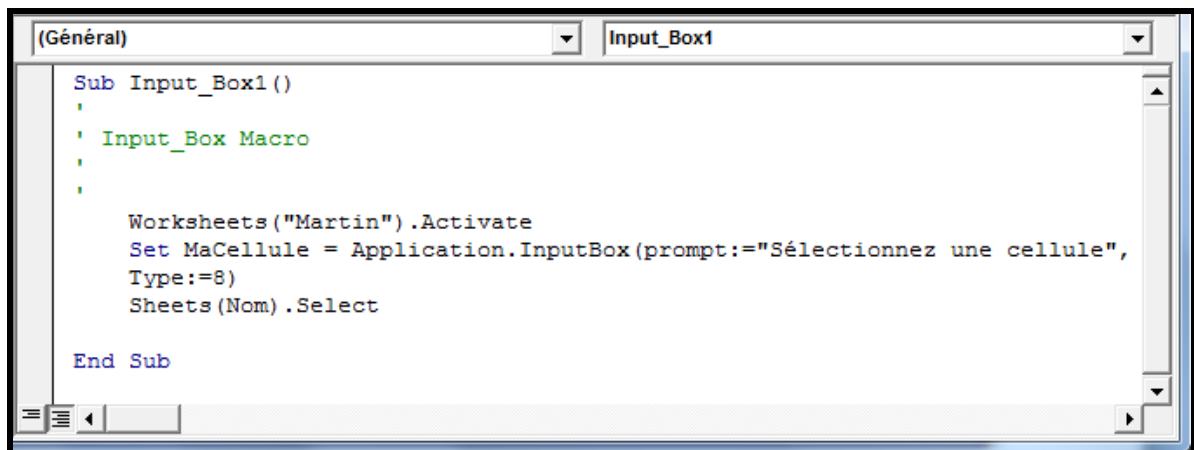
```
Set myRange = Application.InputBox(prompt := "Sample", type := 8)
```

Si vous n'utilisez pas l'instruction **Set**, la valeur dans la plage est affectée à la variable plutôt qu'à l'objet **Range** lui-même.

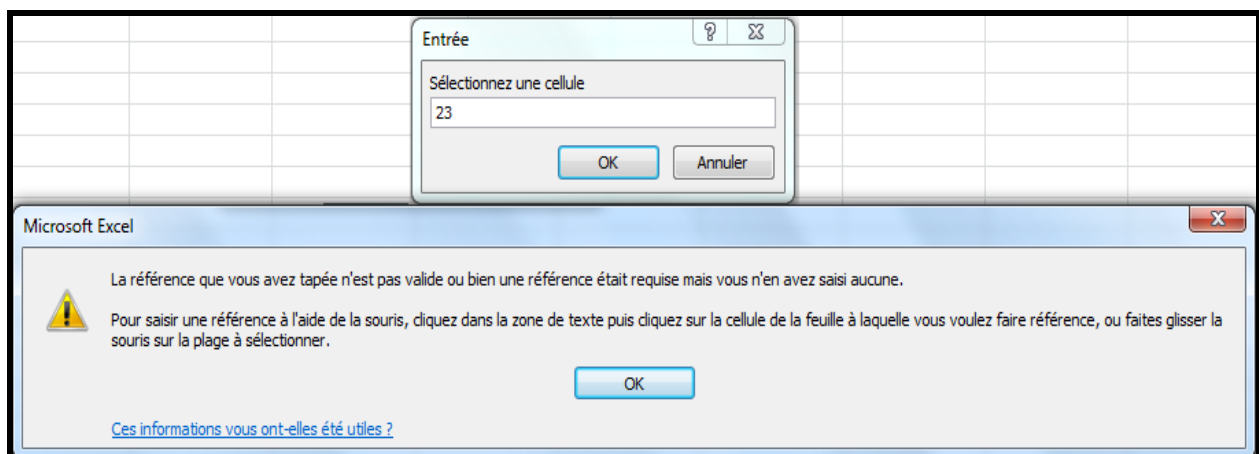
Si vous utilisez la méthode **InputBox** pour demander une formule à l'utilisateur, vous devez utiliser la propriété **FormulaLocal** pour affecter la formule à un objet **Range**. La formule d'entrée est dans la langue de l'utilisateur.

La méthode **InputBox** est différente de la fonction **InputBox**, car elle permet une validation sélective de l'entrée de l'utilisateur et elle peut être utilisée avec des objets Microsoft Excel, des valeurs d'erreur et des formules. Notez que **Application.InputBox** appelle la méthode **InputBox** ; **InputBox** sans qualificatif d'objet appelle la fonction **InputBox**.

Exemple de l'utilisation d'**InputBox** avec l'argument **Type:=8**.

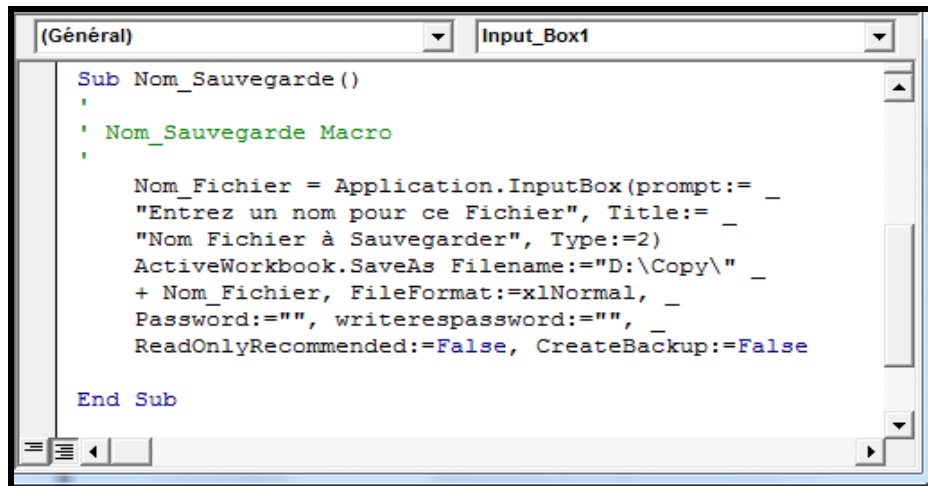


Si nous lançons cette macro, nous obtenons la boîte de dialogue suivante. Si nous entrons une valeur qui ne correspond pas à une référence de cellule valide, nous recevons un message d'erreur, nous obligeant à saisir une nouvelle fois la valeur correcte.

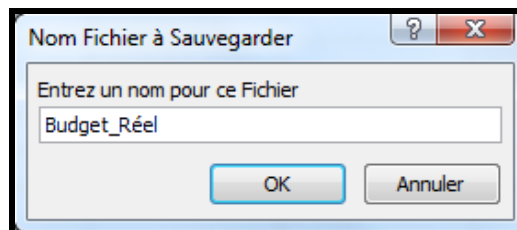


2) - Chargement d'une variable par InputBox et Sauvegarde avec celle-ci

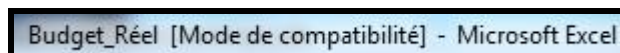
Pour effectuer cela, il faut en premier lieu conserver avec la commande **InputBox** la valeur de la variable que nous allons saisir, puis avec la commande **ActiveWorkbook.SaveAs FileName:=** nous donnons au fichier le nom de la valeur saisie. Dans cet exemple nous chargeons dans la variable **Nom_Fichier** la valeur saisie (Format *Texte*) et nous sauvegardons le fichier avec cette valeur, comme nom du fichier.



Lorsque nous exécutons cette macro, nous ouvrons la boîte de dialogue suivante:

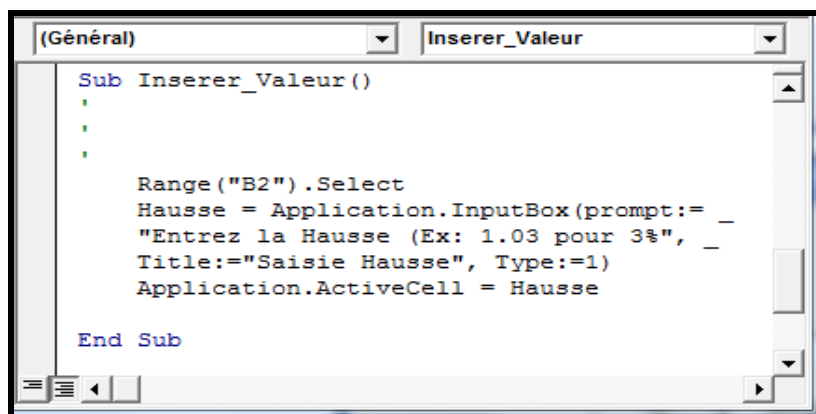


Nous entrons par exemple *Budget_Réel* et nous cliquons sur "OK" pour confirmer. Nous sauvegardons bien notre fichier avec la valeur saisie soit ***Budget Réel***.

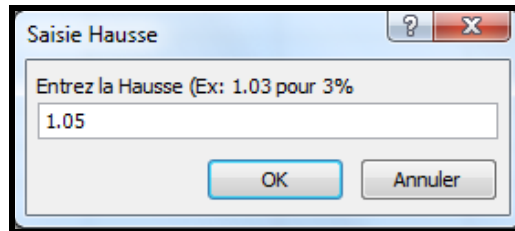


3) - Charger une variable Nombre avec InputBox et utiliser cette variable

Dans cet exemple nous chargeons dans la variable **Hausse**, la valeur saisie (Format *Numérique*) que nous pourrons utiliser dans une formule. Pour effectuer cela, il faut en premier lieu conserver avec la commande **InputBox** la valeur de la variable que nous allons saisir, puis avec la commande **ActiveCell.FormulaR1C1** = nous effectuons le calcul sur cette cellule



Lorsque nous exécutons cette macro, nous ouvrons la boîte de dialogue suivante:



Nous entrons par exemple 1.05 et nous cliquons sur "OK" pour confirmer. Nous obtenons bien dans la cellule active la valeur saisie soit 1.05.

	A	B
1		
2		1.05
3		

4) - Fonction If Then..Else..End If

Dans l'exemple ci-dessous, nous allons effectuer la sauvegarde d'un fichier situé dans répertoire précis, vers le même répertoire, mais avec un nom de fichier différent, ou bien dans un répertoire différent avec le même nom de fichier. Pour ce faire nous allons ouvrir une boîte de dialogue nous permettant d'indiquer le chemin avec le nom du fichier source que nous appellerons "**Origine\$**" et une autre boîte de dialogue nommée "**Destination\$**" pour indiquer le chemin et le nom du fichier destination.

En même temps, nous allons gérer les erreurs pouvant survenir à la saisie. Si le nom complet de la variable *Origine\$* est vide ou bien égal au contenu de la variable *Destination\$* ou que cette dernière est vide, nous faisons apparaître un message d'erreur et nous recommençons l'ensemble de la saisie.

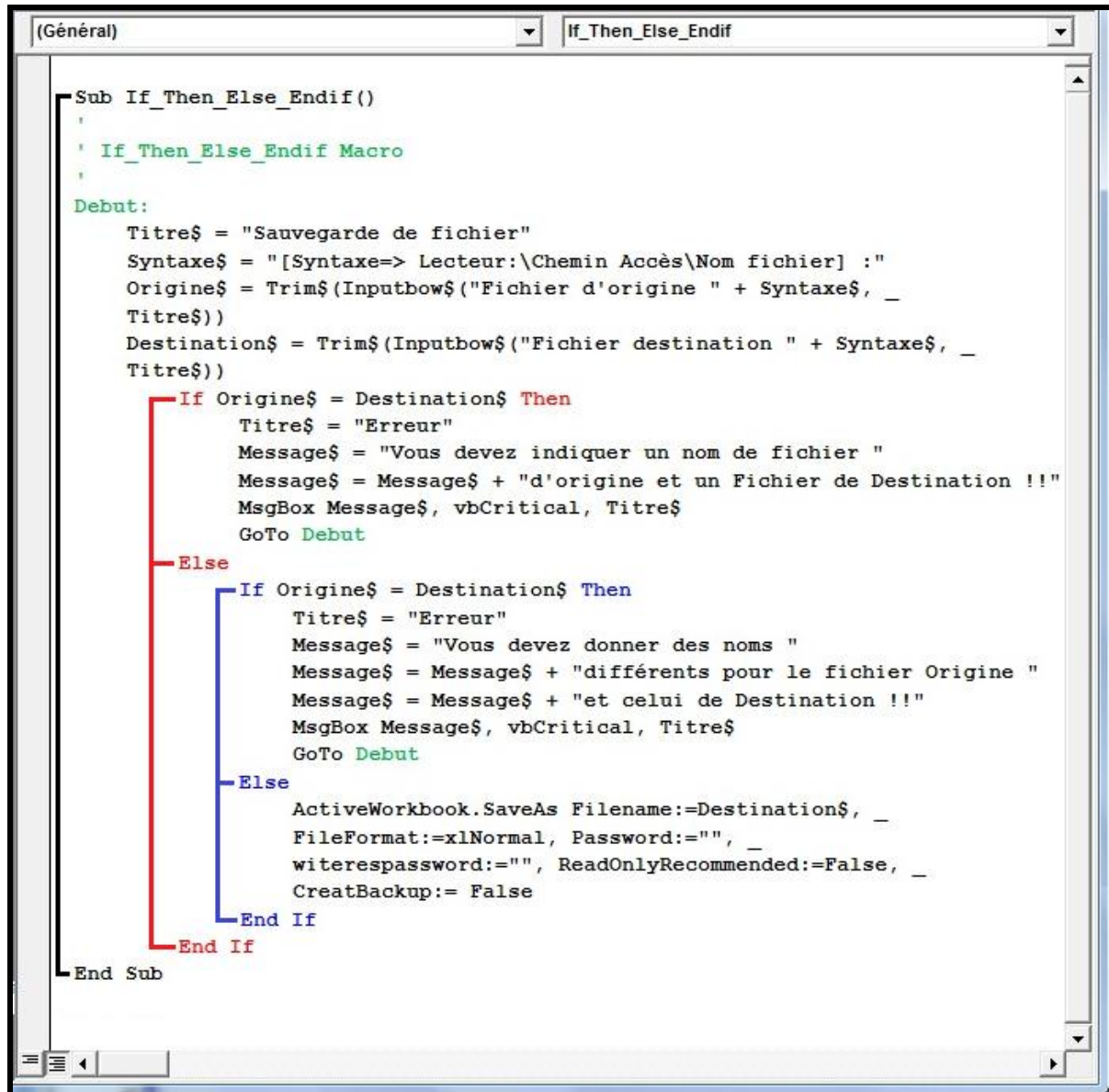
debut: est une étiquette qui fonctionne avec la fonction **GoTo**. C'est à dire que lorsque le programme rencontre cette fonction **GoTo** suivie du nom de l'étiquette, ce dernier repart au niveau du nom de cette étiquette.

If Then...End If est une fonction de condition; c'est à dire que si la condition est remplie, nous traitons la tâche qui suit. Sinon nous traitons une autre tâche si nous rencontrons la fonction **Else** ou **ElseIf**. Si nous ne trouvons pas ces fonction; nous nous arrêtons à **End If**.

La façon pour effectuer ce type de contrôle passe par des boucles **If then.. Else... End If**.

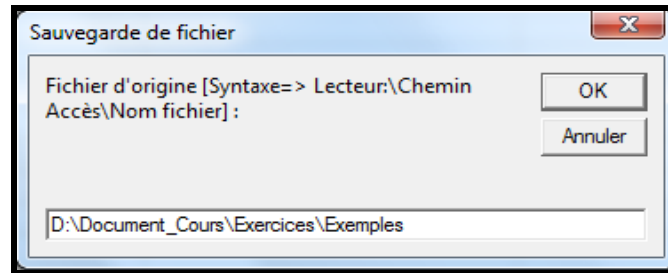
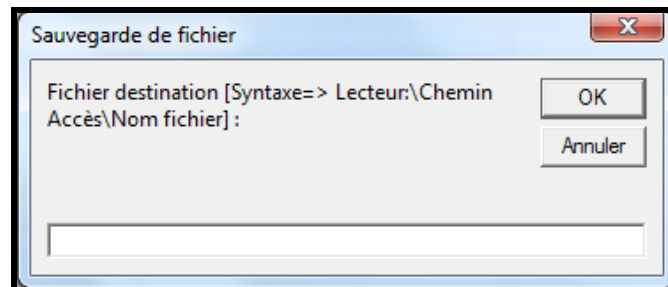
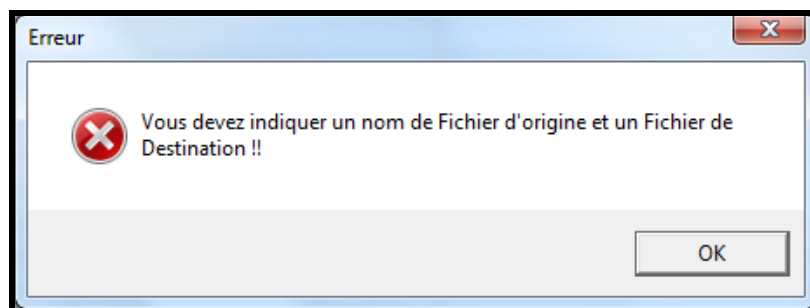
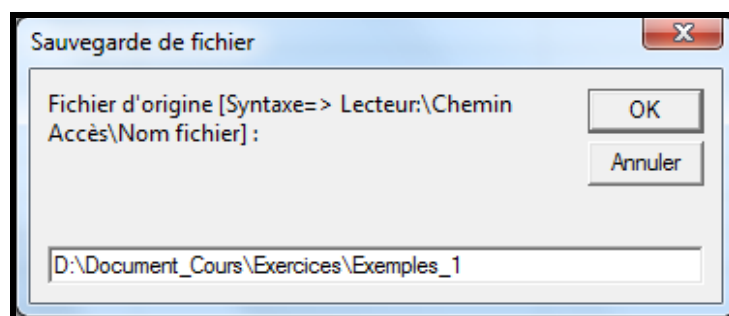
Le fonctionnement d'une telle boucle est le suivant:

- 1) Nous mettons en premier les valeurs des variables que nous voulons contrôler
- 2) Nous entrons dans le début de la boucle par **If** qui est une fonction de condition. C'est-à-dire que si la condition est remplie, nous traitons la tâche qui suit cette condition par la fonction **Then**. Nous traitons donc le message d'erreur et arrivée à la fonction **GoTo**, nous allons ensuite sur le nom suivant cette fonction **GoTo**, c'est à dire que nous remontons sur l'étiquette **Debut**:



```
Sub If_Then_Else_Endif()  
    ' If_Then_Else_Endif Macro  
    '   
    Debut:  
    Titre$ = "Sauvegarde de fichier"  
    Syntaxe$ = "[Syntaxe=> Lecteur:\Chemin Accès\Nom fichier] :"  
    Origine$ = Trim$(Inputbox("Fichier d'origine " + Syntaxe$, _  
    Titre$))  
    Destination$ = Trim$(Inputbox("Fichier destination " + Syntaxe$, _  
    Titre$))  
    If Origine$ = Destination$ Then  
        Titre$ = "Erreur"  
        Message$ = "Vous devez indiquer un nom de fichier "  
        Message$ = Message$ + "d'origine et un Fichier de Destination !!"  
        MsgBox Message$, vbCritical, Titre$  
        GoTo Debut  
    Else  
        If Origine$ = Destination$ Then  
            Titre$ = "Erreur"  
            Message$ = "Vous devez donner des noms "  
            Message$ = Message$ + "différents pour le fichier Origine "  
            Message$ = Message$ + "et celui de Destination !!"  
            MsgBox Message$, vbCritical, Titre$  
            GoTo Debut  
        Else  
            ActiveWorkbook.SaveAs Filename:=Destination$, _  
            FileFormat:=xlNormal, Password:="", _  
            witerespassword:="", ReadOnlyRecommended:=False, _  
            CreatBackup:= False  
        End If  
    End If  
End Sub
```

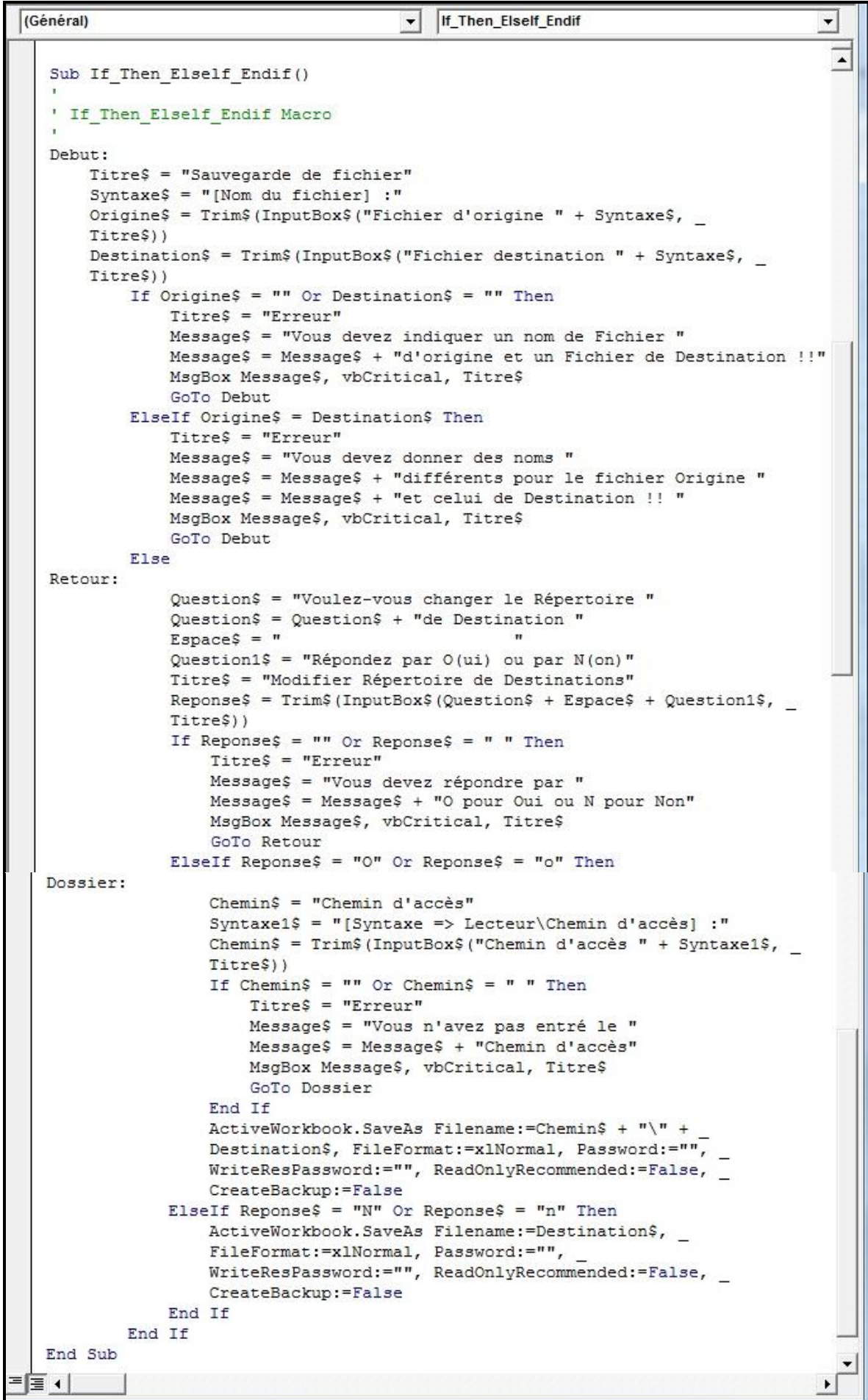
- 3) Si la condition du point précédent n'est pas remplie, nous allons sur la fonction **Else** et nous traitons la seconde boucle **If** qui est une autre fonction de condition contrôlant si le contenu des variables sont différentes ou identiques. Si le contenu est identique, nous traitons la tâche qui suit cette condition par la fonction **Then**. Nous traitons donc le message d'erreur et arrivée à la fonction **GoTo**, nous allons ensuite sur le nom suivant cette fonction **GoTo**, c'est à dire que nous remontons sur l'étiquette **Debut**:
- 4) Si la condition du point 3 n'est pas remplie, nous allons sur la fonction **Else** et nous traitons la commande qui suit, c'est à dire l'enregistrement du fichier selon les critères de la variable **"Destination\$"**. Lorsque nous lançons cette macro, nous obtenons la boîte de dialogue de la Figure 1, dans laquelle nous devons indiquer le chemin et le nom du fichier Source et confirmons par **"OK"**. Nous recevons ensuite la boîte de dialogue de la figure 2, dans laquelle nous devons indiquer le chemin et le nom du fichier de Destination. Si nous confirmons sans indication de valeur, nous recevons le message d'erreur de la figure 3.

*Figure 1**Figure 2**Figure 3**Figure 4*

Nous recommandons donc par la saisie demandée dans la figure 1 puis ensuite nous voyons de nouveau la boîte de dialogue de la figure 2, dans laquelle nous devons indiquer le chemin et le nom du fichier de Destination et nous confirmons par "OK". La sauvegarde s'effectue alors.

5) - Fonction If Then..Else..End If et Traitement des Erreurs

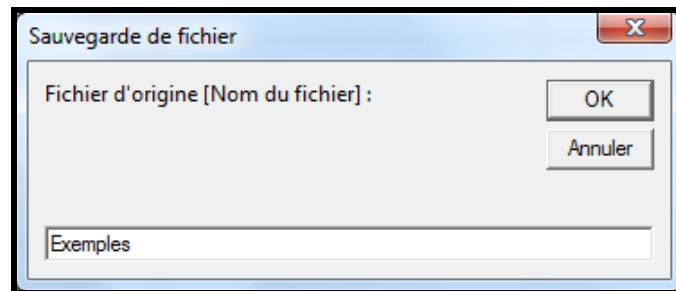
Dans l'exemple ci-dessous, nous allons effectuer le même traitement que dans l'exemple précédent, mais nous allons gérer le changement de répertoire, incorporer deux étiquettes et gérer des erreurs multiples.



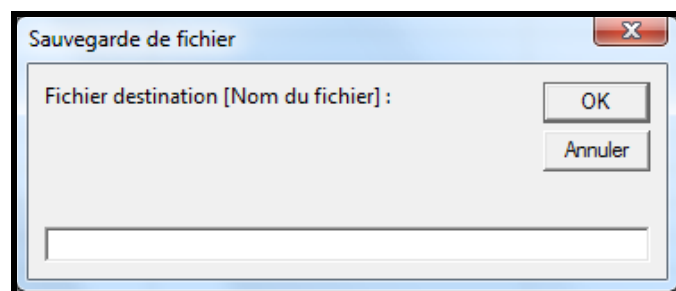
The screenshot shows a VBA editor window with the title bar '(Général) If_Then_Elself_Endif'. The code is as follows:

```
Sub If_Then_Elself_Endif()  
,  
,  
, If_Then_Elself_Endif Macro  
,  
,  
Debut:  
Titre$ = "Sauvegarde de fichier"  
Syntaxe$ = "[Nom du fichier] :"  
Origine$ = Trim$(InputBox("Fichier d'origine " + Syntaxe$, _  
Titre$))  
Destination$ = Trim$(InputBox("Fichier destination " + Syntaxe$, _  
Titre$))  
If Origine$ = "" Or Destination$ = "" Then  
Titre$ = "Erreur"  
Message$ = "Vous devez indiquer un nom de Fichier "  
Message$ = Message$ + "d'origine et un Fichier de Destination !!"  
MsgBox Message$, vbCritical, Titre$  
GoTo Debut  
ElseIf Origine$ = Destination$ Then  
Titre$ = "Erreur"  
Message$ = "Vous devez donner des noms "  
Message$ = Message$ + "différents pour le fichier Origine "  
Message$ = Message$ + "et celui de Destination !! "  
MsgBox Message$, vbCritical, Titre$  
GoTo Debut  
Else  
Retour:  
Question$ = "Voulez-vous changer le Répertoire "  
Question$ = Question$ + "de Destination "  
Espace$ = " "  
Question1$ = "Répondez par O(ui) ou par N(on)"  
Titre$ = "Modifier Répertoire de Destinations"  
Reponse$ = Trim$(InputBox$(Question$ + Espace$ + Question1$, _  
Titre$))  
If Reponse$ = "" Or Reponse$ = " " Then  
Titre$ = "Erreur"  
Message$ = "Vous devez répondre par "  
Message$ = Message$ + "O pour Oui ou N pour Non"  
MsgBox Message$, vbCritical, Titre$  
GoTo Retour  
ElseIf Reponse$ = "O" Or Reponse$ = "o" Then  
Dossier:  
Chemin$ = "Chemin d'accès"  
Syntaxe1$ = "[Syntaxe => Lecteur\Chemin d'accès] :"  
Chemin$ = Trim$(InputBox("Chemin d'accès " + Syntaxe1$, _  
Titre$))  
If Chemin$ = "" Or Chemin$ = " " Then  
Titre$ = "Erreur"  
Message$ = "Vous n'avez pas entré le "  
Message$ = Message$ + "Chemin d'accès"  
MsgBox Message$, vbCritical, Titre$  
GoTo Dossier  
End If  
ActiveWorkbook.SaveAs Filename:=Chemin$ + "\" + _  
Destination$, FileFormat:=xlNormal, Password:="", _  
WriteResPassword="", ReadOnlyRecommended:=False, _  
CreateBackup:=False  
ElseIf Reponse$ = "N" Or Reponse$ = "n" Then  
ActiveWorkbook.SaveAs Filename:=Destination$, _  
FileFormat:=xlNormal, Password:="", _  
WriteResPassword="", ReadOnlyRecommended:=False, _  
CreateBackup:=False  
End If  
End If  
End Sub
```

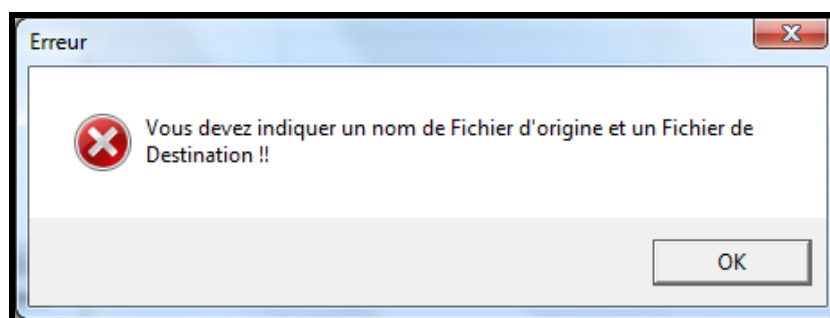
Nous sauvegardons dans la Variable **Titre\$** le titre de la Boîte de dialogue ci-dessous et dans la variable **Syntaxe\$** le texte *[Nom du Fichier]*. Dans la variable **Origine\$**, au moyen de la fonction **InputBox\$**, nous demandons à l'utilisateur de donner un nom au Fichier en cours.



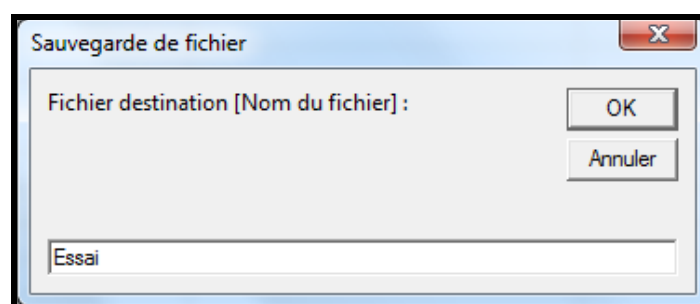
Dans la variable **Destinations\$**, au moyen de la fonction **InputBox\$**, nous demandons à l'utilisateur de donner un nom au Fichier de Destination. Si nous ne donnons pas de nom et confirmons par "OK".



Nous entrons dans la première fonction de comparaison **if..Then**, qui teste si le contenu des variables **Origine\$** ou **Destination\$** sont VIDES. Si c'est **VRAI**, nous chargeons dans la variable **Titre\$** la valeur *Erreur* et dans la variable **Message\$** la valeur *Vous devez indiquer un fichier d'origine et un fichier de Destination*. Avec la fonction **MsgBox**, nous affichons un message d'erreur tel que ci-dessous.



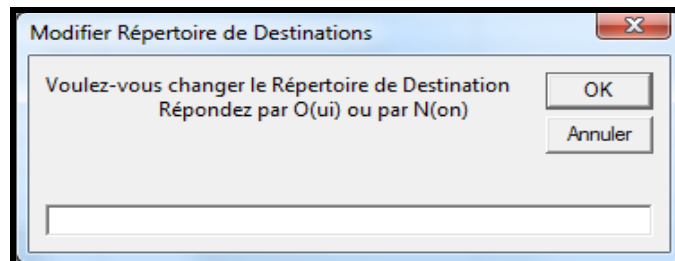
A la ligne du dessous, nous indiquons au programme de reprendre à l'étiquette **Debut**:



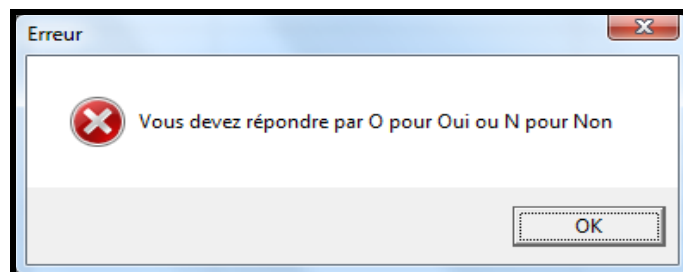
Si la condition précédente était **VRAI**, nous poursuivons à la ligne **ElseIf** où nous testons que les valeurs contenues dans les variables **Origine\$** et **Destination\$** sont différentes. Si c'est **FAUX**, nous

chargeons dans la variable **Titre\$** la valeur *Erreur* et dans la variable **Message\$** la valeur *Vous devez donner des noms différents pour le fichier d'origine et le fichier de Destination*. Avec la fonction **MsgBox**, nous affichons un message d'erreur tel que ci-dessus et à la ligne du dessous, nous indiquons au programme de reprendre également à l'étiquette **Debut**.

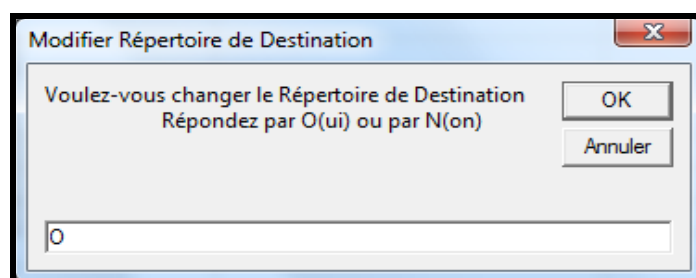
Sinon avec la ligne **Else**, nous poursuivons avec la question relative au changement du répertoire du Fichier de Destination. Nous chargeons au préalable dans la variable **Question\$**, le texte *Voulez-vous changer le Répertoire de Destination? Répondez par O(ui) ou par N(on)*. Dans la variable **Reponse\$**, au moyen de la fonction **InputBox\$**, nous demandons à l'utilisateur de répondre **O** pour oui ou **N** pour non.



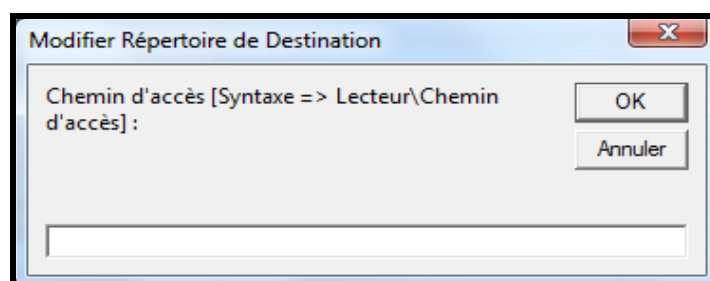
Nous testons ensuite si la valeur contenue dans la variable **Reponse\$** est vide. Si c'est le cas, nous chargeons dans la variable **Titre\$** la valeur *Erreur* et dans la variable **Message\$** la valeur *Vous devez répondre par O pour Oui ou N pour Non*. Avec la fonction **MsgBox**, nous affichons un message d'erreur tel que ci-dessus et à la ligne du dessous, nous indiquons au programme de reprendre à l'étiquette **Retour**.



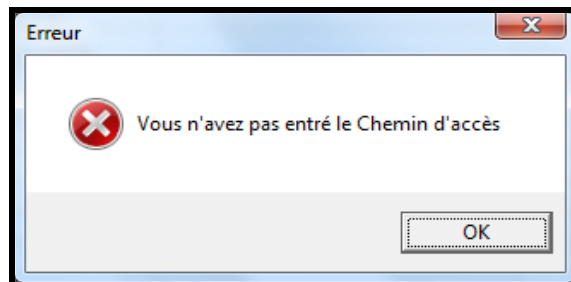
Si le test préalable était FAUX, grâce à la fonction **ElseIf**, nous testons si la valeur est **O** pour Oui.



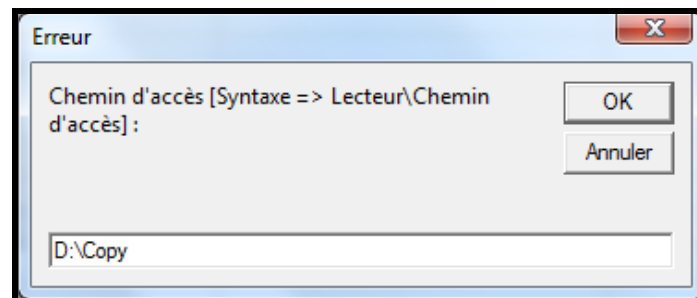
Si c'est le cas, nous chargeons dans la variable **Chemin\$** la valeur *Chemin d'Accès* et dans la variable **Syntaxe1\$** la valeur *[Syntaxe => Lecteur\Chemin Accès]*. Dans la variable **Chemin\$**, au moyen de la fonction **InputBox\$**, nous demandons à l'utilisateur de donner le chemin d'accès.



Nous testons avec la fonction **If..Or..Then..** si la valeur contenue dans la variable est VIDE. Si la réponse est VRAI, nous nous chargeons dans la variable **Titre\$** la valeur *Erreur* et dans la variable **Message\$** la valeur *Vous n'avez pas indiqué le Chemin d'Accès*. Avec la fonction **MsgBox**, nous affichons un message d'erreur tel que ci-dessus et à la ligne du dessous, nous indiquons au programme de reprendre à l'étiquette **Dossier:**.



Si le test ci-dessus était bon (Chemin d'accès correct) alors nous lançons l'enregistrement du fichier



avec la fonction **ActiveWorkbook.SaveAs**en indiquant dans la zone **FileName:=Chemin\$ + "\" + Destination\$**. Si la valeur dans la variable **Reponse\$** était N, alors à la ligne **ElseIf** nous lançons l'enregistrement du fichier avec la fonction **ActiveWorkbook.SaveAs**en indiquant dans la zone **FileName:= Destination\$**. Chaque fonction **IF..** doit être terminée par **End If**.