

Procédures. Passage d'arguments par valeur ou par adresse.

C40

Méthode

- On compare ici deux modes de passages des arguments. En Visual Basic, le mode par défaut est un passage par adresse. Dans le passage par adresse toute modification apportée à un argument se traduit par une modification de la variable passée en argument au moment de l'appel.

Source

```
Sub Increment1(x As Integer) '--- Passage par adresse
    x = x + 1
End Sub

Sub Increment2(ParVal As Integer) '--- Passage par valeur
    x = x + 1
End Sub

Sub test()
    Dim k As Integer
    k = 15: MsgBox "Valeur initiale de k = " & k
    Call Increment1(k): MsgBox "Après appel Increment1, k = " & k
    k = 48: MsgBox "Nouvelle valeur de k = " & k
    Call Increment2(k): MsgBox "Après appel Increment2, k = " & k
End Sub
```

Méthode

- Le tableau nommé "Table" va comporter la liste des nombres premiers déjà trouvés. Cette table est initialisée aux valeurs des trois premiers nombres premiers: 2, 3 et 5.
- On boucle tant que l'on n'a pas trouvé les 50 premiers nombres premiers.
- On teste successivement si les nombres impairs, à partir de 7, sont premiers.
- Pour savoir si un nombre est premier, on le divise successivement par les nombres premiers déjà trouvés • 3.
- Dès que le carré du diviseur essayé est supérieur au nombre testé, on peut affirmer que le nombre testé est premier, et l'ajouter à la liste.

Sourcec

```
Sub NbPremiers()  
  Const nb = 100: Dim Table(nb) As Integer: NbPrem = 3  
  NbTeste = 5  
  Table(1) = 2: Table(2) = 3: Table(3) = 5: Visu = "2 3 5"  
  While NbPrem < 50  
    NbTeste = NbTeste + 2  
    For k = 2 To NbPrem  
      Diviseur = Table(k)  
      If Diviseur ^ 2 > NbTeste Then  
        NbPrem = NbPrem + 1: Table(NbPrem) = NbTeste  
        Visu = Visu & " " & NbTeste: Exit For  
      End If  
      If NbTeste Mod Diviseur = 0 Then Exit For  
    Next k  
  Wend  
  MsgBox Visu  
End Sub
```

Objectif

- Constituer deux tables de vérité, des opérateurs ET et OU
- On utilise une fonction nommée logique, destinée à transformer une valeur logique en chaîne de caractères.
- On met ensuite en œuvre trois boucles imbriquées: Opérateur, Variables 1 et 2

Source

```
Sub test()  
    Dim p(2) As Integer: Dim q(2) As Integer: Visu = ""  
    p(1) = True: p(2) = False: q(1) = True: q(2) = False  
    For op = 1 To 2  
        For k = 1 To 2  
            For m = 1 To 2  
                wA = Logique(p(k)): wB = Logique(q(m))  
                C1 = Logique(p(k) And q(m)): C2 = Logique(p(k) Or  
q(m))  
                If op = 1 Then w = wA & "AND " & wB & "= " & C1  
                If op = 2 Then w = wA & "OR " & wB & "= " & C2  
                Visu = Visu & w & Chr(13)  
            Next m  
        Next k  
        If op = 1 Then Visu = Visu & Chr(13)  
    Next op  
    MsgBox Visu  
End Sub  
  
Function Logique(ByVal xVal As Integer) As String  
    Logique = Format(xVal, "True/False") & " "  
End Function
```

Objectif

- On utilise l'éditeur de menus pour créer une nouvelle barre de menu nommée "Salut" et deux éléments de menu nommés "Bonjour" et "Bonsoir", qui s'activent et se désactivent réciproquement.

Source

```
Sub Bonjour()  
    MsgBox "Bonjour"  
    Call SetBonjour(Faux)  
End Sub  
  
Sub Bonsoir()  
    MsgBox "Bonsoir"  
    Call SetBonjour(Vrai)  
End Sub  
  
Sub SetBonjour(ByVal x As Boolean)  
    With MenuBars(xlWorksheet).Menus("Salut")  
        .MenuItems("Bonjour").Enabled = x  
        .MenuItems("Bonsoir").Enabled = Not x  
    End With  
End Sub  
  
Sub Init() '--- Chaque fois que la feuille est activée  
    MsgBox "Debut du test"  
    Call SetBonjour(True)  
End Sub  
  
Sub InitP() '--- a lancer au debut du programme  
    Application.OnSheetActivate = "Init"  
End Sub
```

Objectif

- On souhaite créer un fichier texte, au premier niveau du disque dur.
- Ce fichier comportera 100 lignes, et pendant l'écriture, la barre d'état de l'application indiquera le numéro de la ligne en cours d'écriture.

Version américaine pour Macintosh

```
Sub Creation()  
  Nom = "HD500:test1.txt"  
  '-- Windows:   Nom = "C:\test1.txt" --  
  Open Nom For Output As 1 '-- Création du fichier et ouverture  
  For k = 1 To 100  
    w = "Ligne " & k '-- Constitution de la ligne  
    Application.StatusBar = w '-- Visualisation de l'avancement  
    Print #1, w '-- Ecriture de la ligne  
  Next k  
  Close '-- Fermeture du fichier (et des autres éventuels)  
  MsgBox "Fichier Créé !": Application.StatusBar = ""  
End Sub
```

Excel - Parcours des barres d'outils et affichage de leurs noms

F01

Source - Parcours et affichage

```
'Parcours des barres d'outils et affichage de leurs noms

Sub ListeBarres()
    Nb = Toolbars.Count
    MsgBox Nb & " barres "
    For k = 1 To Nb
        MsgBox k & " / " & Nb & " ->" & Toolbars(k).Name
    Next k
End Sub
```

Source - Renvoi du numéro de barre

```
' Renvoi du n° d'ordre d'une barre d'outils

Function FindToolBar(ByVal xToolBarName As String) As Integer
    Nb = Toolbars.Count
    FindToolBar = 0
    For k = 1 To Nb
        If Toolbars(k).Name = xToolBarName Then
            FindToolBar = k
            Exit For
        End If
    Next k
End Function
```

Source- Parcours et affichage

```
' Création d'une barre d'outils

Sub CreationBarre()
  Dim w As String
  w = "Barre_123"
  Toolbars(w).Delete: MyBar = FindToolBar(w)
  If MyBar <> 0 Then MsgBox "cette barre existe déjà !": Exit Sub
  Toolbars.Add (w) 'Création de la nouvelle barre
  MyBar = FindToolBar(w)
  If MyBar = 0 Then MsgBox "cette barre n'a pas été créée !": Exit Sub
  For k = 1 To 5
    Toolbars(MyBar).ToolbarButtons.Add (k + 190)
    Toolbars(MyBar).ToolbarButtons(k).Name = "Action " & k
    Toolbars(MyBar).ToolbarButtons(k).OnAction = "Procédure" & k
  Next k
  Toolbars(MyBar).Visible = True
  MsgBox "OK"
End Sub

' Procédure déclenchée par l'outil
Sub Procedure3()
  MsgBox "appel Procédure du 3ème bouton"
End Sub
```

Sélection d'un document Access mdb et description de la structure d'une des tables

```
Sub test()  
    Dim myDB As Database, Dim myFileName As String  
  
    '-- Effacement des trois premières colonnes de la feuille  
    Sheets("Feuill").Columns("A:C").ClearContents  
  
    '-- Recherche des fichiers access  
    wFilter = ("MDB Files (*.mdb), *.mdb")  
    myFileName = Application.GetOpenFilename(wFilter)  
  
    '-- Ouverture de l'objet access sélectionné  
    Set myDB = DBEngine.OpenDatabase(myFileName)  
  
    '-- Parcours et affichage des noms de tables trouvées  
    n = myDB.TableDefs.Count  
    For k = 0 To n - 1  
        w = myDB.TableDefs(k).Name  
        '-- Non prise en compte des tables système  
        If InStr(LCase(w), "msys") <> 1 Then  
            p = MsgBox("Description de la table " & w & " ?", vbYesNo)  
            If p = 6 Then  
                SelTable = w : GoTo Suite  
            End If  
        End If  
    Next k  
    Exit Sub '-- puisque rien de sélectionné  
  
Suite:  
    '-- Description de la table sélectionnée  
    n = myDB.TableDefs(SelTable).Fields.Count  
    For k = 0 To n - 1  
        FName = myDB.TableDefs(SelTable).Fields(k).Name  
        FSize = myDB.TableDefs(SelTable).Fields(k).Size  
        fType = myDB.TableDefs(SelTable).Fields(k).Type  
        wType = "Type " & fType  
        If fType = 4 Then wType = "Nombre"  
        If fType = 10 Then wType = "Chaine"  
        With Sheets("Feuill")  
            .Cells(k + 1, 1) = FName  
            .Cells(k + 1, 2) = FSize  
            .Cells(k + 1, 3) = wType  
        End With  
    Next k  
  
End Sub
```

Affichage de données d'une table Access dans une feuille Excel, par une requête SQL semi dynamique

J04

```
Sub test4()  
    Dim myDB As Database  
    Dim myFileName As String  
    Dim myQuery As QueryDef  
    Dim MyRS As Recordset  
  
    '-- Effacement des trois premières colonnes  
    With Sheets("Feuill1")  
        Columns("A:C").ClearContents  
    End With  
  
    '-- Ouverture de l'objet access désigné  
    myFileName = "c:\AccessVBA\Communes.mdb"  
    Set myDB = DBEngine.OpenDatabase(myFileName)  
  
    '-- Suppression de la requête si elle existe déjà  
    Call DelQuery(myDB, "SQL1")  
  
    w = InputBox("Introduisez quelques lettre de la commune recherchée :")  
    w = "*" & w & "*"  
  
    '-- Application d'une requête SQL sur la table  
    wSQL = "Select * from Communes where nom like " & w & " order by  
    Nom"  
    Set myQuery = myDB.CreateQueryDef("SQL1", wSQL)  
    Set MyRS = myQuery.OpenRecordset  
    k = 0  
    Do While Not MyRS.EOF  
        k = k + 1  
        If k > 100 Then GoTo suite2  
        With Sheets("Feuill1")  
            .Cells(k, 1) = MyRS.Fields("NUMCOMMUNE").Value  
            .Cells(k, 2) = MyRS.Fields("NOM").Value  
            .Cells(k, 3) = MyRS.Fields("DEP").Value  
        End With  
        MyRS.MoveNext '-- Passage à l'enregistrement suivant  
    Loop  
  
suite2:  
  
End Sub
```

Insertions dans une table Access depuis VBA, par des ordres SQL

J06

```
Sub test6()  
'-- Insertion de données dans une table  
  Dim myDB As Database  
  Dim myQuery As QueryDef  
  
  '-- Ouverture de l'objet access désigné  
  myFileName = "c:\AccessVBA\Communes.mdb"  
  Set myDB = DBEngine.OpenDatabase(myFileName)  
  
  '-- Suppression de la requête si elle existe déjà  
  Call DelQuery(myDB, "SQL1")  
  
  '-- Création de la requete avec un ordre de commande  
  wSQL = "Insert into MaTable (n,c) values (1000,""Patricia"")"  
  Set myQuery = myDB.CreateQueryDef("SQL1", wSQL)  
  
  myQuery.Execute  
  
  wSQL = "Insert into MaTable (n,c) values (1500,""Romuald"")"  
  myQuery.Sql = wSQL  
  myQuery.Execute  
  
  MsgBox "2 insertions effectuées"  
  
End Sub
```

Mise à jour d' une table Access depuis VBA, par un ordre SQL

J07

```
Sub test7()  
'-- Mise à jour d'une table  
    Dim myDB As Database  
    Dim myQuery As QueryDef  
  
    '-- Ouverture de l'objet access désigné  
    myFileName = "c:\AccessVBA\Communes.mdb"  
    Set myDB = DBEngine.OpenDatabase(myFileName)  
  
    '-- Suppression de la requête si elle existe déjà  
    Call DelQuery(myDB, "SQL1")  
  
    '-- Création de la requete avec un ordre de commande  
    wSQL = "Update MaTable set n = n + 1 where n > 100"  
    Set myQuery = myDB.CreateQueryDef("SQL1", wSQL)  
  
    myQuery.Execute  
  
    MsgBox "Mise à jour effectuée"  
  
End Sub
```

**SQL Dynamique vers de données Access depuis Excel VBA.
Prise en compte du SQL de sélection ou de commande**

```
Sub test10()  
    Dim myDB As Database  
    Dim myFileName As String  
    Dim myQuery As QueryDef  
    Dim MyRS As Recordset  
  
    On Error GoTo Erreur  
    '-- Effacement des premières colonnes  
    Sheets("Feuill1").Columns("A:Z").ClearContents  
  
    '-- Ouverture de l'objet access désigné  
    myFileName = "c:\AccessVBA\Communes.mdb"  
    Set myDB = DBEngine.OpenDatabase(myFileName)  
  
    '-- Suppression de la requête si elle existe déjà  
    Call DelQuery(myDB, "SQL1")  
  
    w = InputBox("Introduisez votre requête SQL")  
  
    '-- Application d'une requête SQL sur la table  
    wSQL = w  
    Set myQuery = myDB.CreateQueryDef("SQL1", wSQL)  
  
    '-- Raffraichissement du type de requete  
    myDB.QueryDefs.Refresh  
    '-- Test du type d'ordre SQL  
    If Not (myQuery.Type = dbQSelect) Then GoTo NonSelect  
    '-- Cas d'un select ....  
    Set MyRS = myQuery.OpenRecordset  
  
    With Sheets("Feuill1")  
        nbCols = MyRS.Fields.Count  
        For j = 0 To nbCols - 1  
            .Cells(1, j + 1) = MyRS.Fields(j).Name  
        Next j  
        k = 0  
        Do While Not MyRS.EOF  
            k = k + 1  
            If k > 100 Then GoTo suite2  
            For j = 0 To nbCols - 1  
                .Cells(k + 1, j + 1) = MyRS.Fields(j).Value  
            Next j  
            MyRS.MoveNext '-- Passage à l'enregistrement suivant  
        Loop  
    End With  
  
    suite2:  
        Exit Sub  
  
    NonSelect:    '-- cas d'un ordre non Select  
        myQuery.Execute : MsgBox "Ordre SQL effectué !" : Exit Sub  
  
    Erreur:  
        MsgBox "Erreur " & Err() & " - " & Error(Err()) : Exit Sub  
  
End Sub
```

WORD - VBA - Parcours d'un Dynaset Oracle Objects for OLE et écriture dans un tableau Word

K03

```
Public OraSession As Object    '-- Pour travailler avec Oracle
Public OraDatabase As Object   '-- Pour se connecter à Oracle
Public MyDynaset As Object     '-- Pour obtenir les résultats d'une requête

...    (Connexion et requete)

Sub Macro1()
    Dim NbL, NbC As Integer '-- Variables destinées à recevoir les nb de lignes/colonnes

    Call ConnexionOracle '-- Appel Connexion à Oracle
    Call RequeteOracle   '-- Appel Requete vers Oracle

    '-- Obtention des nombres de lignes et de colonnes renvoyées par la requete
    NbL = MyDynaset.RecordCount
    NbC = MyDynaset.Fields.Count
    MsgBox "Le résultat de la requete comporte " & NbL & " lignes et " & NbC & " colonnes
! "

    '-- Creation d'un nouveau document Word vide
    Documents.Add Template:="D:\Office97\ModÈles\Normal.dot", NewTemplate:=False

    '-- Ecriture de texte dans le document Word :
    Selection.TypeText Text:= _
        "Voici la liste des clients sous forme de tableau dans Word :"

    '-- Ecriture de lignes vides
    Selection.TypeParagraph
    Selection.TypeParagraph

    '-- Creation d'un tableau avec un nombre variable de lignes et de colonnes
    ActiveDocument.Tables.Add Range:=Selection.Range, NumRows:=NbL, NumColumns:=NbC

    '-- Boucle de transfert DyNaset -> Tableau Oracle
    For R = 1 To NbL
        For C = 1 To NbC
            '-- AccÈs au contenu de la C iÈme colonne de la ligne (active) du Dynaset
            w = MyDynaset.Fields(C - 1).Value

            '-- Ecriture de cette valeur dans le tableau Word
            ActiveDocument.Tables(1).Cell(R, C).Range.Text = w
        Next C '-- Fin parcours des colonnes
        MyDynaset.DbMoveNext '-- Lecture de la ligne suivante dans le Dynaset
        If MyDynaset.EOF Then Exit For '-- Sortie imperative si fin de Dynaset
    Next R '-- Fin parcours des lignes

End Sub
```