أكواد منوعة ومبتكرة للمبرمجين بالفيجوال بيسك

الجزء الثاني

تأليف وإعداد / عبدالله خضر عبدالله الهوساوي

E-Mail : prog909@yahoo.com

بسم الله الرحمن الرحيم

هذه مجموعة أكواد مهمة وفنية وتسهل الكثير من الأعمال البرمجية في الفيجوال بيسك 6 ، وهي من تصميمي ، أرجو أن ينتفع بها مبرمجينا الأعزاء ، هي هكذا تباعا :

في هذا الجزء الثاني من هذا الموضوع أود التركيز أكثر على الأداة **ListBox** ليست بوكس ، وأحد أهم الأسباب التي جعلتني أركز عليها هي كثيرة منها :

1 – أنها بمثابة المساحة التي يُحفظ فيها عناصر البيانات الحرفية والعددية إذا كانت مخفية ، بشرط أن تكون محددة في خانة الخصائص هكذا : Visible = False

2 – أنها تقوم بدور أساسي ومحوري في الكثير من البرامج التي تعتمد على نمط (الملف المتسلسل) في وضع بيانات من الملف إلى الليست ، أي من القرص الصلب إلى ذاكرة الرام للكمبيوتر ، أو العكس أي :

3 – حفظ بيانات من الليست إلى الملف ، أي من الذاكرة إلى القرص الصلب (سيأتي الشرح لاحقاً) .

4 – يمكن بواسطة أداة الليست **ListBox** كونها (وعاء جامع للبيانات) البحث عن وجود عنصر ما فيها . كل ماذكرته هو على سبيل المثال .

---

1 - (دالة وظيفية تعطي عدد عناصر الليست List)

حيث الإسم الإفتراضي لهذه الدالة يكون : **LstCnt**

حيث يوضع إسم أداة الليست بين قوسين بعدها أثناء كتابتها في النص البرمجي

شرح مبسط عن عمل الدالة :

مثلاً إسم أداة الليست المطلوب معرفة عدد عناصرها مباشرة هي = **List1**

صيغة الكتابة البرمجية تكون هكذا :

```
Print LstCnt(List1)
```

أو

```
N= LstCnt(List1)
```

فتعطي الدالة هذه النتيجة :

5 أو 10 أو 24 أو 3 أو أي عدد آخر يكون هو عدد عناصر الليست .

ملاحظة مهمة : هذا الكود الآتي يوضع في الفورم في منطقة الإعلانات العامة للفورم الـ **Declarations** ، أو في ملف ميديول **Moduel** تابع لمشروع البرنامج .

نص الكود :

```
Function LstCnt(LstName As ListBox) As Integer

LstName.Refresh

LstCnt = LstName.ListCount

End Function
```

السبب من تصميمي هذه الدالة هو إختصار التكرار ، مع التأكيد على عمل الخاصية Refresh في كل مرة تعمل فيها هذه الدالة ، لأنك مطالب أكثر الأحيان بإثباتها قبل طلب معرفة عدد العناصر ، ولايخفى أن عدم إنعاش Refresh الليست قبل طلب معرفة عدد العناصر يعطي أحيانا نتيجة ناقصة بسبب الإضافات التالية في الأداة التي لم تنتبه الأداة لها مما يتسبب في حصول أخطاء برمجية أنت في غنى عنها ! ، تستطيع شكري ولكن ليس الآن تابع بقية الموضوع ! .

---

2 - (إجراء لحذف عناصر مُختارة من الليست List)

حيث الإسم الإفتراضي لهذا الإجراء يكون : **LstDel**

إنتبه جيداً : يجب أن يُلحق الإسم الإفتراضي في البرنامج لهذا الإجراء بالكلمة **Call** ، أي هكذا :

Call LstDel (إسم أداة الليست)

شرح مبسط عن عمل الإجراء :

إذا كانت العناصر المراد تحديدها أكثر من 1 يجب تحديد الخاصية **MuliSelect** لليست = 1 بدلاً من 0 في خانة خصائص الليست ، وذلك قبل تنفيذ البرنامج ، لكي يتاح لك التحديد المتعدد ! .

فمثلاً حددت بعض العناصر لحذفها عنصراً واحداً أو أكثر في الليست المسمى **List1** مثلاً ، تكتب الأمر في زر كوماند بوتون **CommandButton** مثلاً هكذا :

```
Private Sub Command1_Click()

Call LstDel ( List1 )

End Sub
```

سترى النتيجة أن العناصر التي حددتها قد حُذفت

ملاحظة مهمة : هذا الكود الآتي يوضع في الفورم في منطقة الإعلانات العامة للفورم الـ **Declarations** ، أو في ملف ميديول **Moduel** تابع لمشروع البرنامج .

نص الكود :

```
Sub LstDel(LstName As ListBox)

Dim G, I As Integer, U As Boolean

LstName.Refresh
G = LstName.ListCount

If G = 1 Then

U = LstName.Selected(0)

If U = True Then LstName.Clear

Else

For I = G - 1 To 0 Step -1

U = LstName.Selected(I)

If U = True Then LstName.RemoveItem I

Next

End If

LstName.Refresh

End Sub
```

---

3 - (إجراء لنسخ ملف بيانات متسلسل إلى الليست List)

حيث الإسم الإفتراضي لهذا الإجراء يكون : **CopyFL**

حرف **F** إشارة إلى كلمة (ملف **File**) ، وحرف **L** إشارة إلى كلمة (ليست **List**) ، أقصد من الملف إلى الليست **F** to **L**

إنتبه جيداً : يجب أن يُلحق الإسم الإفتراضي في البرنامج لهذا الإجراء بالكلمة **Call** ، أي هكذا **:**

Call CopyFL (إسم أداة الليست , مسار الملف المتسلسل **)**

شرح مبسط عن عمل الإجراء :

أردت مثلاً نسخ جميع محتويات ملف متسلسل إلى ليست في البرنامج أو أكثر من ليست واحد وربما مرات متعددة في أماكن متغيرة في البرنامج ، وأنت تعرف فهي عملية تحتاج لتكرار كود ممل كل مرة ، ولكن إذا أصبح هذا الكود إجراءاً واحداً على شكل أمر واحد فربما تغيرت نظرتك للأمر بشكل أفضل ، إليك الطريقة :

مثلاً : الملف المتسلسل المُراد مساره هو :
P$="C:\Program Data.txt" أو C:\Program Files\Data
وتريد نسخه إلى ليست في البرنامج إسمه **List1** ، سيكون الإجراء هكذا :

Call CopyFL ("C:\Program Data" , List1)

أو

Call CopyFL (P$, List1)

سترى النتيجة أن العناصر قد نُسخت من الملف المتسلسل إلى الليست بكتابة أمر واحد ، وهذا بدون تكرار ممل لكتابة الكود المصدري أو نسخه كل مرة ! ، أليس ذلك أسهل وأيسر ؟

ملاحظة مهمة : هذا الكود الآتي يوضع في الفورم في منطقة الإعلانات العامة للفورم الـ **Declarations** ، أو في ملف ميديول **Moduel** تابع لمشروع البرنامج .

نص الكود :

```
Sub CopyFL(FilePath As String, LstName As ListBox)

Dim U As String

Open FilePath For Append As 1: Close

If FileLen(FilePath) <> 0 Then

LstName.Clear

Open FilePath For Input As 1

Do

Input #1, U

LstName.AddItem U

Loop Until EOF(1)

Close

End If

End Sub
```

4 - (إجراء لحفظ ملف بيانات ليست List إلى ملف متسلسل)

ركز معي ! ، هذا الإجراء هو عكس الإجراء السابق !

حيث الإسم الإفتراضي لهذا الإجراء يكون **CopyLF** (وليس CopyFL) فقط تبديل بين موقع آخر حرفين .

وحرف **L** إشارة إلى كلمة (ليست **List**) ، وحرف **F** إشارة إلى كلمة (ملف **File**) ، ، أقصد من الليست إلى الملف **L to F**

إنتبه جيداً : يجب أن يُلحق الإسم الإفتراضي في البرنامج لهذا الإجراء بالكلمة **Call** ، أي هكذا **:**

Call CopyLF (إسم أداة الليست , مسار الملف المتسلسل **)**

شرح مبسط عن عمل الإجراء :
أردت مثلاً عكس الإجراء السابق ، أي نسخ جميع محتويات الليست إلى ملف متسلسل على القرص الصلب ، الطريقة تكون هكذا :

مثلاً : الملف المتسلسل المُراد مساره وأسمه هو :
P$="C:\Test.txt" أو C:\Program Files\GetData
وتريد نسخ جميع عناصر الليست **List1** مثلاً إلى الملف الذي سميته ، سيكون الإجراء هكذا :

Call CopyLF (P$, List1)

سترى النتيجة أن العناصر قد نُسخت من الليست إلى الملف المتسلسل الذي سميته من عندك إلى بكتابة أمر واحد !

ملاحظة مهمة : هذا الكود الآتي يوضع في الفورم في منطقة الإعلانات العامة للفورم الـ **Declarations** ، أو في ملف ميديول **Moduel** تابع لمشروع البرنامج .

نص الكود :

```
Sub CopyLF(FilePath As String, LstName As ListBox)

Dim U As String: LstName.Refresh

If LstName.ListCount > 0 Then

Open Trim$(FilePath) For Output As 1

For I = 0 To LstName.ListCount - 1

Print #1, Trim$(LstName.List(I))

Next

Close

End If

End Sub
```

5 - (دالة لحساب جميع الأعداد الموجودة في الليست List)

حيث الإسم الإفتراضي لهذه الدالة يكون : **LstCal**

حيث يوضع إسم أداة الليست بين قوسين بعدها أثناء كتابتها في النص البرمجي هكذا :
LstCal (إسم أداة الليست **)**

شرح مبسط عن عمل الدالة : مثلاً إسم أداة الليست المطلوب حساب جميع الأعداد الموجودة فيها أسمها : **List1**

صيغة الكتابة البرمجية تكون هكذا :
Print LstCal(List1)
أو
N= LstCal(List1)
فتعطي الدالة هذه النتيجة وهي المجموع مثلاً : 1200

ملاحظة مهمة : هذا الكود الآتي يوضع في الفورم في منطقة الإعلانات العامة للفورم الـ **Declarations** ، أو في ملف ميديول **Moduel** تابع لمشروع البرنامج .

نص الكود :

```
Function LstCal(LstName As ListBox) As Double

Dim L, I As Integer
Dim U As Double

LstName.Refresh
L = LstName.ListCount

For I = 0 To L - 1

U = U + Val(LstName.List(I))

Next

LstCal = Round(U, 2)

End Function
```

تفيد هذه الدالة في معرفة مجموعة حسابات معينة أدخلت في الليست ، يمكن أن تكون تجارية أو إحصائية أو غير ذلك مما يراه المبرمج .

6 - (دالة تحديد عدد تكرار عنصر محدد في الليست List)

حيث الإسم الإفتراضي لهذه الدالة يكون : **LstItm**

حيث يوضع إسم أداة الليست بين قوسين بعدها أثناء كتابتها في النص البرمجي ثم يكتب العنصر بعد ذلك هكذا :

LstCal (العنصر المراد عدد تكراره **,** إسم أداة الليست **)**

شرح مبسط عن عمل الدالة : مثلاً إسم أداة الليست المطلوب معرفة عدد تكرار عنصر ما فيها أسمها : **List1** ، والكلمة أو العنصر الذي أريد أن أعرف كم تكرر هو (محمد)

صيغة الكتابة البرمجية تكون هكذا :

```
Print LstItm (List1,"محمد")
```

أو

```
N= LstItm (List1,"محمد")
```

فتعطي الدالة هذه النتيجة وهي المجموع مثلاً : 4 ، أي تكرر هذا الإسم أربع مرات

ملاحظة مهمة : هذا الكود الآتي يوضع في الفورم في منطقة الإعلانات العامة للفورم الـ **Declarations** ، أو في ملف ميديول **Moduel** تابع لمشروع البرنامج .

نص الكود :

```
Function LstItm(LstName As ListBox, Word As String) As Integer

Dim L, OK, I As Integer
Dim A, U As String

LstName.Refresh
L = LstName.ListCount

A = UCase$(Trim$(Word))

For I = 0 To L - 1

U = UCase$(LstName.List(I))

If U = A Then OK = OK + 1

Next

LstItm = OK

End Function
```

7 - (إجراء لوضع سلسلة أرقام محددة البداية والنهاية في الليست List)

حيث الإسم الإفتراضي لهذا الإجراء يكون : **LstLoop**

إنتبه جيداً : يجب أن يُلحق الإسم الإفتراضي في البرنامج لهذا الإجراء بالكلمة **Call**، أي هكذا **:**

Call LstLoop (إسم أداة الليست , رقم البداية , رقم النهاية)

شرح مبسط عن عمل الإجراء :
أردت مثلاً وضع سلسلة أرقام متتالية لتواريخ سنوية بين العام **2000** إلى العام **2014** في ليست **List1** ، أو قد تريد وضع أرقام بدأ من **1** حتى **100** ، أو غير ذلك ، إليك الطريقة :

Call LstLoop (List1 , 2000 , 2014)

سترى النتيجة أن الأعداد المطلوبة وضعت في الليست .

ملاحظة مهمة : هذا الكود الآتي يوضع في الفورم في منطقة الإعلانات العامة للفورم الـ **Declarations** ، أو في ملف ميديول **Moduel** تابع لمشروع البرنامج .

نص الكود :

```
Sub LstLoop(LstName As ListBox, StartNumber As Variant, EndNumber As Variant)

Dim A As Integer

LstName.Clear

For A = Val(StartNumber) To Val(EndNumber)

LstName.AddItem A

Next

End Sub
```

8 - (إجراء لنسخ بيانات الليست List إلى ليست آخر مع خيارين لكيفية النسخ)

حيث الإسم الإفتراضي لهذا الإجراء يكون : **CopyLL**

حرف **L** إشارة إلى كلمة (ليست **List**) ، أقصد من الليست إلى الليست **L to L**

إنتبه جيداً : يجب أن يُلحق الإسم الإفتراضي في البرنامج لهذا الإجراء بالكلمة **Call** ، أي هكذا **:**

Call CopyLL (إسم أداة الليست1 , إسم أداة الليست2 , العدد 0 أو 1)

شرح مبسط عن عمل الإجراء :
أردت مثلاً نسخ جميع محتويات ليست في البرنامج إلى ليست آخر ، وأحياناً تريد نسخ عناصر محددة في الليست لتصير لوحدها في ليست آخر ، إليك الطريقتين في أمر واحد :

مثلاً : نسخ جميع محتويات ليست إلى ليست آخر (العدد يكون **0** أي الكل) ، سيكون الإجراء هكذا :

Call CopyLL (List1, List2 , 0)

سترى إن الليست الآخر مطابق للأول تماماً .

أما الطريقة الأخرى ، مثلاً : نسخ بعض محتويات محددة لليست إلى ليست آخر (العدد يكون **1** أي البعض) ، سيكون الإجراء هكذا :

Call CopyLL (List1, List2 , 1)

لاتنسى : إذا كانت العناصر المراد تحديدها أكثر من 1 يجب تحديد الخاصية **MuliSelect** لليست = 1 بدلاً من 0 في خانة خصائص الليست ، وذلك قبل تنفيذ البرنامج ، لكي يتاح لك التحديد المتعدد ! .

سترى وجود العناصر المحددة فقط التي ظللتها قد نُسخت إلى الليست الآخر .

فقط عليك الإختيار للصفر أو الواحد لتحديد نوع النسخ .

ملاحظة مهمة : هذا الكود الآتي يوضع في الفورم في منطقة الإعلانات العامة للفورم الـ **Declarations** ، أو في ملف ميديول **Moduel** تابع لمشروع البرنامج .

نص الكود :

```
Sub CopyLL(LstName1 As ListBox, LstName2 As ListBox, All0Sel1 As Variant)

Dim U As String
Dim I, L, X As Integer

X = Val(All0Sel1)

LstName2.Clear
LstName1.Refresh
L = LstName1.ListCount

' تابع '
```

```
'  تابع  '
For I = 0 To L - 1
If X = 0 Then
LstName2.AddItem Trim$(LstName1.List(I))
End If
'
If X = 1 Then
If LstName1.Selected(I) = True Then
LstName2.AddItem Trim$(LstName1.List(I))
End If
End If
'
Next
End Sub
```

---

9 - (دالة وظيفية تعطي إسم الأرقام بكتابة حرفية عربية)

حيث الإسم الإفتراضي لهذه الدالة يكون : **WrNum**

حيث يوضع رقم من 1 إلى 999,999,999 بين قوسين بعدها أثناء كتابتها في النص البرمجي.

شرح مبسط عن عمل الدالة :

صيغة الكتابة البرمجية تكون هكذا :

Print **WrNum** (3000)

أو

R$ = **WrNum** (66)

فتعطي الدالة هذه النتيجة بالكلمات :
ثلاثة آلاف ، ستة وستون ، وهكذا إلخ .... من العدد **1** حتى **999,999,999**

ملاحظة مهمة : هذا الكود الآتي يوضع في الفورم في منطقة الإعلانات العامة للفورم الـ **Declarations** ، أو في ملف ميديول **Moduel** تابع لمشروع البرنامج .

يجب أن تُترك الفراغات التي بين علامات التنصيص " " كما هي وبدون أي تعديل في آلية الكود ! ، وإلا .....

نص الكود :

```
Function WrNum(Number As Variant) As String

Dim I, II, C, L, LN, N, P, PA, PB, P1, P2, P3, P4, P5, P6, P7, PP As Integer
Dim B, B2, D, D1, D2, X, Y, X1, X2, Z As String

Dim G, G9, G8, G7, G6, G5, G4, G3 As String: LN = 9

N = Int(Val(Number)): L = Len(N): B2 = CStr(N)

If L > LN Then Exit Function

B2 = Left$(CStr(N), LN): B = String$(LN - L, "0") + B2

PB = Val(Mid$(B, 1, 1))
PA = Val(Mid$(B, 2, 1))
P1 = Val(Mid$(B, 3, 1))
P2 = Val(Mid$(B, 4, 1))
P3 = Val(Mid$(B, 5, 1))
P4 = Val(Mid$(B, 6, 1))
P5 = Val(Mid$(B, 7, 1))
P6 = Val(Mid$(B, 8, 1))
P7 = Val(Mid$(B, LN, 1))

Y = "": If Val(Right$(B, 8)) <> 0 Then Y = "و"

X = "": If PA = 0 And P1 = 0 Then X = "مليون "

If PB = 1 Then G8 = "مئة " + X + Y
If PB = 2 Then G8 = "مئتي " + X + Y
If PB = 3 Then G8 = "ثلاثمائة " + X + Y
If PB = 4 Then G8 = "أربعمائة " + X + Y
If PB = 5 Then G8 = "خمسمائة " + X + Y
If PB = 6 Then G8 = "ستمائة " + X + Y
If PB = 7 Then G8 = "سبعمائة " + X + Y
If PB = 8 Then G8 = "ثمانمائة " + X + Y
If PB = 9 Then G8 = "تسعمائة " + X + Y

PP = Val(Mid$(B, 2, 2))
If PP >= 1 And PP <= 19 Then

X = "ملايين "
X1 = "مليون "
```

```
Y = "": If Val(Right$(B, 6)) <> 0 Then Y = "و"

If PP = 1 Then G7 = "واحد مليون " + Y
If PP = 2 Then G7 = "مليونين " + Y
If PP = 3 Then G7 = "ثلاثة " + X + Y
If PP = 4 Then G7 = "أربعة " + X + Y
If PP = 5 Then G7 = "خمسة " + X + Y
If PP = 6 Then G7 = "ستة " + X + Y
If PP = 7 Then G7 = "سبعة " + X + Y
If PP = 8 Then G7 = "ثمانية " + X + Y
If PP = 9 Then G7 = "تسعة " + X + Y
If PP = 10 Then G7 = "عشرة " + X + Y
If PP = 11 Then G7 = "إحدى عشر " + X1 + Y
If PP = 12 Then G7 = "إثنا عشر " + X1 + Y
If PP = 13 Then G7 = "ثلاثة عشر " + X1 + Y
If PP = 14 Then G7 = "أربعة عشر " + X1 + Y
If PP = 15 Then G7 = "خمسة عشر " + X1 + Y
If PP = 16 Then G7 = "ستة عشر " + X1 + Y
If PP = 17 Then G7 = "سبعة عشر " + X1 + Y
If PP = 18 Then G7 = "ثمانية عشر " + X1 + Y
If PP = 19 Then G7 = "تسعة عشر " + X1 + Y

End If

If PP > 19 Then

X1 = "مليون "

If P1 = 1 Then Z = "واحد و"
If P1 = 2 Then Z = "إثنين و"
If P1 = 3 Then Z = "ثلاثة و"
If P1 = 4 Then Z = "أربعة و"
If P1 = 5 Then Z = "خمسة و"
If P1 = 6 Then Z = "ستة و"
If P1 = 7 Then Z = "سبعة و"
If P1 = 8 Then Z = "ثمانية و"
If P1 = 9 Then Z = "تسعة و"

Y = "": If Val(Right$(B, 6)) <> 0 Then Y = "و"


If PA = 2 Then G7 = Z + "عشرون " + X1 + Y
If PA = 3 Then G7 = Z + "ثلاثون " + X1 + Y
If PA = 4 Then G7 = Z + "أربعون " + X1 + Y
If PA = 5 Then G7 = Z + "خمسون " + X1 + Y
If PA = 6 Then G7 = Z + "ستون " + X1 + Y
If PA = 7 Then G7 = Z + "سبعون " + X1 + Y
If PA = 8 Then G7 = Z + "ثمانون " + X1 + Y
If PA = 9 Then G7 = Z + "تسعون " + X1 + Y

End If
```

```
Y = "": If Val(Right$(B, 5)) <> 0 Then Y = "و"

X = "": If P3 = 0 And P4 = 0 Then X = "ألف "

If P2 = 1 Then G6 = "مئة " + X + Y
If P2 = 2 Then G6 = "مئتي " + X + Y
If P2 = 3 Then G6 = "ثلاثمائة " + X + Y
If P2 = 4 Then G6 = "أربعمائة " + X + Y
If P2 = 5 Then G6 = "خمسمائة " + X + Y
If P2 = 6 Then G6 = "ستمائة " + X + Y
If P2 = 7 Then G6 = "سبعمائة " + X + Y
If P2 = 8 Then G6 = "ثمانمائة " + X + Y
If P2 = 9 Then G6 = "تسعمائة " + X + Y

PP = Val(Mid$(B, 5, 2)) ' 000 0nn 000

If PP >= 1 And PP <= 19 Then

X = "آلاف "
X1 = "ألف "

Y = "": If Val(Right$(B, 3)) <> 0 Then Y = "و"

If PP = 1 Then G5 = "ألف " + Y
If PP = 2 Then G5 = "ألفين " + Y
If PP = 3 Then G5 = "ثلاثة " + X + Y
If PP = 4 Then G5 = "أربعة " + X + Y
If PP = 5 Then G5 = "خمسة " + X + Y
If PP = 6 Then G5 = "ستة " + X + Y
If PP = 7 Then G5 = "سبعة " + X + Y
If PP = 8 Then G5 = "ثمانية " + X + Y
If PP = 9 Then G5 = "تسعة " + X + Y
If PP = 10 Then G5 = "عشرة " + X + Y
If PP = 11 Then G5 = "إحدى عشر " + X1 + Y
If PP = 12 Then G5 = "إثنا عشر " + X1 + Y
If PP = 13 Then G5 = "ثلاثة عشر " + X1 + Y
If PP = 14 Then G5 = "أربعة عشر " + X1 + Y
If PP = 15 Then G5 = "خمسة عشر " + X1 + Y
If PP = 16 Then G5 = "ستة عشر " + X1 + Y
If PP = 17 Then G5 = "سبعة عشر " + X1 + Y
If PP = 18 Then G5 = "ثمانية عشر " + X1 + Y
If PP = 19 Then G5 = "تسعة عشر " + X1 + Y

End If

If PP > 19 Then

X1 = "ألف "

If P4 = 1 Then Z = "واحد و"
If P4 = 2 Then Z = "إثنين و"
If P4 = 3 Then Z = "ثلاثة و"
If P4 = 4 Then Z = "أربعة و"
If P4 = 5 Then Z = "خمسة و"
```

```
If P4 = 6 Then Z = "ستة و"
If P4 = 7 Then Z = "سبعة و"
If P4 = 8 Then Z = "ثمانية و"
If P4 = 9 Then Z = "تسعة و"

Y = "": If Val(Right$(B, 3)) <> 0 Then Y = "و"

If P3 = 2 Then G5 = Z + "عشرون " + X1 + Y
If P3 = 3 Then G5 = Z + "ثلاثون " + X1 + Y
If P3 = 4 Then G5 = Z + "أربعون " + X1 + Y
If P3 = 5 Then G5 = Z + "خمسون " + X1 + Y
If P3 = 6 Then G5 = Z + "ستون " + X1 + Y
If P3 = 7 Then G5 = Z + "سبعون " + X1 + Y
If P3 = 8 Then G5 = Z + "ثمانون " + X1 + Y
If P3 = 9 Then G5 = Z + "تسعون " + X1 + Y

End If

Y = "": If Val(Right$(B, 2)) <> 0 Then Y = "و"

If P5 = 1 Then G4 = "مئة " + Y
If P5 = 2 Then G4 = "مئتين " + Y
If P5 = 3 Then G4 = "ثلاثمائة " + Y
If P5 = 4 Then G4 = "أربعمائة " + Y
If P5 = 5 Then G4 = "خمسمائة " + Y
If P5 = 6 Then G4 = "ستمائة " + Y
If P5 = 7 Then G4 = "سبعمائة " + Y
If P5 = 8 Then G4 = "ثمانمائة " + Y
If P5 = 9 Then G4 = "تسعمائة " + Y

PP = Val(Mid$(B, 8, 2))
If PP >= 1 And PP <= 19 Then

If PP = 1 Then G3 = "واحد "
If PP = 2 Then G3 = "إثنين "
If PP = 3 Then G3 = "ثلاثة "
If PP = 4 Then G3 = "أربعة "
If PP = 5 Then G3 = "خمسة "
If PP = 6 Then G3 = "ستة "
If PP = 7 Then G3 = "سبعة "
If PP = 8 Then G3 = "ثمانية "
If PP = 9 Then G3 = "تسعة "
If PP = 10 Then G3 = "عشرة "
If PP = 11 Then G3 = "إحدى عشر "
If PP = 12 Then G3 = "إثنا عشر "
If PP = 13 Then G3 = "ثلاثة عشر "
If PP = 14 Then G3 = "أربعة عشر "
If PP = 15 Then G3 = "خمسة عشر "
If PP = 16 Then G3 = "ستة عشر "
If PP = 17 Then G3 = "سبعة عشر "
If PP = 18 Then G3 = "ثمانية عشر "
If PP = 19 Then G3 = "تسعة عشر "

End If
```

```
If PP >= 20 Then

Z = ""

If P7 = 1 Then Z = "واحد و"
If P7 = 2 Then Z = "إثنين و"
If P7 = 3 Then Z = "ثلاثة و"
If P7 = 4 Then Z = "أربعة و"
If P7 = 5 Then Z = "خمسة و"
If P7 = 6 Then Z = "ستة و"
If P7 = 7 Then Z = "سبعة و"
If P7 = 8 Then Z = "ثمانية و"
If P7 = 9 Then Z = "تسعة و"

If P6 = 2 Then G3 = Z + "عشرون "
If P6 = 3 Then G3 = Z + "ثلاثون "
If P6 = 4 Then G3 = Z + "أربعون "
If P6 = 5 Then G3 = Z + "خمسون "
If P6 = 6 Then G3 = Z + "ستون "
If P6 = 7 Then G3 = Z + "سبعون "
If P6 = 8 Then G3 = Z + "ثمانون "
If P6 = 9 Then G3 = Z + "تسعون "

End If

G = G8 & G7 & G6 & G5 & G4 & G3

WrNum = Trim$(G)

End Function
```

إغفروا لي هذه الرتابة البرمجية !! ، أعرف الكود طويل ومرعب ! ، ولكني لم أجد أوضح وأدق من هذه الطريقة التي برمجتها ، ولكن تذكر : هذا الكود رغم طوله وغرابته يُنسخ ثم يوضع في ميديول أو أعلى منطقة الإعلانات للفورم ، وما عليك إلا كتابة الأمر **WrNum** فقط عدد ماتريد في أي مكان من البرنامج .

لاتنسى أهمية هذه الدالة **WrNum** , وتوأمتها الأخرى الآتية **WrNumE** ، فأي منهما تفيد مثلاً في برنامج محاسبات تجاري يطلب كتابة المبلغ العددي بصيغة حرفية في فاتورة البيع أو الشراء ، مارأيك ؟؟! .

---

10 - (دالة وظيفية تعطي إسم الأرقام بكتابة حرفية إنجليزية)

حيث الإسم الإفتراضي لهذه الدالة يكون : **WrNumE** ونص كودها يختلف قليلاً عن قرينتها العربية حرف **E** يرمز إلى اللغة الإنجليزية **English** .

حيث يوضع رقم من 1 إلى 999,999,999 بين قوسين بعدها أثناء كتابتها في النص البرمجي.

شرح مبسط عن عمل الدالة :

صيغة الكتابة البرمجية تكون هكذا :
Print **WrNumE** (3000)
أو
R$ = **WrNumE** (66)

فتعطي الدالة هذه النتيجة بالكلمات الإنجليزية :
THREE THOUSAND ، SIXTY-SIX ، وهكذا إلخ **....** من العدد **1** حتى **999,999,999**

ملاحظة مهمة : هذا الكود الآتي يوضع في الفورم في منطقة الإعلانات العامة للفورم الـ **Declarations** ، أو في ملف ميديول **Moduel** تابع لمشروع البرنامج .

يجب أن تُترك الفراغات التي بين علامات التنصيص " " كما هي وبدون أي تعديل في آلية الكود ! ، رجاءً !

نص الكود :

```
Function WrNumE(Number As Variant) As String

Dim I, II, C, L, LN, N, P, PA, PB, P1, P2, P3, P4, P5, P6, P7, PP As Integer
Dim B, B2, D, D1, D2, X, Y, X1, X2, Z As String

Dim G, G9, G8, G7, G6, G5, G4, G3 As String: LN = 9

N = Int(Val(Number)): L = Len(N): B2 = CStr(N)

If L > LN Then Exit Function

B2 = Left$(CStr(N), LN): B = String$(LN - L, "0") + B2

PB = Val(Mid$(B, 1, 1))
PA = Val(Mid$(B, 2, 1))
P1 = Val(Mid$(B, 3, 1))

P2 = Val(Mid$(B, 4, 1))
P3 = Val(Mid$(B, 5, 1))
P4 = Val(Mid$(B, 6, 1))

P5 = Val(Mid$(B, 7, 1))
P6 = Val(Mid$(B, 8, 1))
P7 = Val(Mid$(B, LN, 1))
```

```
Y = "": If Val(Right$(B, 8)) <> 0 Then Y = "AND "
X = "": If PA = 0 And P1 = 0 Then X = "MILLION "

If PB = 1 Then G8 = "ONE HUNDRED " + X + Y
If PB = 2 Then G8 = "TWO HUNDRED " + X + Y
If PB = 3 Then G8 = "THREE HUNDRED " + X + Y
If PB = 4 Then G8 = "FOUR HUNDRED " + X + Y
If PB = 5 Then G8 = "FIVE HUNDRED " + X + Y
If PB = 6 Then G8 = "SIX HUNDRED " + X + Y
If PB = 7 Then G8 = "SEVEN HUNDRED " + X + Y
If PB = 8 Then G8 = "EIGHT HUNDRED " + X + Y
If PB = 9 Then G8 = "NINE HUNDRED " + X + Y

PP = Val(Mid$(B, 2, 2)) ' 0nn 000 000

If PP >= 1 And PP <= 19 Then

X = "MILLION ": X1 = X

Y = "": If Val(Right$(B, 6)) <> 0 Then Y = "AND "

If PP = 1 Then G7 = "ONE " + X + Y
If PP = 2 Then G7 = "TWO " + X + Y
If PP = 3 Then G7 = "THREE " + X + Y
If PP = 4 Then G7 = "FOUR " + X + Y
If PP = 5 Then G7 = "FIVE " + X + Y
If PP = 6 Then G7 = "SIX " + X + Y
If PP = 7 Then G7 = "SEVEN " + X + Y
If PP = 8 Then G7 = "EIGHT " + X + Y
If PP = 9 Then G7 = "NINE " + X + Y
If PP = 10 Then G7 = "TEN " + X + Y
If PP = 11 Then G7 = "ELEVEN " + X1 + Y
If PP = 12 Then G7 = "TWELVE " + X1 + Y
If PP = 13 Then G7 = "THIRTEEN " + X1 + Y
If PP = 14 Then G7 = "FOURTEEN " + X1 + Y
If PP = 15 Then G7 = "FIFTEEN " + X1 + Y
If PP = 16 Then G7 = "SIXTEEN " + X1 + Y
If PP = 17 Then G7 = "SEVENTEEN " + X1 + Y
If PP = 18 Then G7 = "EIGHTTEEN " + X1 + Y
If PP = 19 Then G7 = "NINETEEN " + X1 + Y

End If

If PP > 19 Then

X1 = "MILLION "

If P1 = 1 Then Z = "-ONE"
If P1 = 2 Then Z = "-TWO"
If P1 = 3 Then Z = "-THREE"
If P1 = 4 Then Z = "-FOUR"
If P1 = 5 Then Z = "-FIVE"
If P1 = 6 Then Z = "-SIX"
If P1 = 7 Then Z = "-SEVEN"
```

```
If P1 = 8 Then Z = "-EIGHT"
If P1 = 9 Then Z = "-NINE"

Y = "": If Val(Right$(B, 6)) <> 0 Then Y = "AND "

If PA = 2 Then G7 = "TWENTY" + Z + X1 + Y
If PA = 3 Then G7 = "THIRTY" + Z + X1 + Y
If PA = 4 Then G7 = "FORTY" + Z + X1 + Y
If PA = 5 Then G7 = "FIFTY" + Z + X1 + Y
If PA = 6 Then G7 = "SIXTY" + Z + X1 + Y
If PA = 7 Then G7 = "SEVENTY" + Z + X1 + Y
If PA = 8 Then G7 = "EIGHTY" + Z + X1 + Y
If PA = 9 Then G7 = "NINETY" + Z + X1 + Y

End If

Y = "": If Val(Right$(B, 5)) <> 0 Then Y = "AND "

X = "": If P3 = 0 And P4 = 0 Then X = "THOUSAND "

If P2 = 1 Then G6 = "ONE HUNDRED " + X + Y
If P2 = 2 Then G6 = "TWO HUNDRED " + X + Y
If P2 = 3 Then G6 = "THREE HUNDRED " + X + Y
If P2 = 4 Then G6 = "FOUR HUNDRED " + X + Y
If P2 = 5 Then G6 = "FIVE HUNDRED " + X + Y
If P2 = 6 Then G6 = "SIX HUNDRED " + X + Y
If P2 = 7 Then G6 = "SEVEN HUNDRED " + X + Y
If P2 = 8 Then G6 = "EIGHT HUNDRED " + X + Y
If P2 = 9 Then G6 = "NINE HUNDRED " + X + Y

PP = Val(Mid$(B, 5, 2))

If PP >= 1 And PP <= 19 Then

X = "THOUSAND ": X1 = X

Y = "": If Val(Right$(B, 3)) <> 0 Then Y = "AND "

If PP = 1 Then G5 = "ONE " + X + Y
If PP = 2 Then G5 = "TWO " + X + Y
If PP = 3 Then G5 = "THREE " + X + Y
If PP = 4 Then G5 = "FOUR " + X + Y
If PP = 5 Then G5 = "FIVE " + X + Y
If PP = 6 Then G5 = "SIX " + X + Y
If PP = 7 Then G5 = "SEVEN " + X + Y
If PP = 8 Then G5 = "EIGHT " + X + Y
If PP = 9 Then G5 = "NINE " + X + Y
If PP = 10 Then G5 = "TEN " + X + Y
If PP = 11 Then G5 = "ELEVEN " + X1 + Y
If PP = 12 Then G5 = "TWELVE " + X1 + Y
If PP = 13 Then G5 = "THIRTEEN " + X1 + Y
If PP = 14 Then G5 = "FOURTEEN " + X1 + Y
If PP = 15 Then G5 = "FIFTEEN " + X1 + Y
If PP = 16 Then G5 = "SIXTEEN " + X1 + Y
```

```
If PP = 17 Then G5 = "SEVENTEEN " + X1 + Y
If PP = 18 Then G5 = "EIGHTTEEN " + X1 + Y
If PP = 19 Then G5 = "NINETEEN " + X1 + Y

End If

If PP > 19 Then

X1 = " THOUSAND "

If P4 = 1 Then Z = "-ONE"
If P4 = 2 Then Z = "-TWO"
If P4 = 3 Then Z = "-THREE"
If P4 = 4 Then Z = "-FOUR"
If P4 = 5 Then Z = "-FIVE"
If P4 = 6 Then Z = "-SIX"
If P4 = 7 Then Z = "-SEVEN"
If P4 = 8 Then Z = "-EIGHT"
If P4 = 9 Then Z = "-NINE"

Y = "": If Val(Right$(B, 3)) <> 0 Then Y = "AND "

If P3 = 2 Then G5 = "TWENTY" + Z + X1 + Y
If P3 = 3 Then G5 = "THIRTY" + Z + X1 + Y
If P3 = 4 Then G5 = "FORTY" + Z + X1 + Y
If P3 = 5 Then G5 = "FIFTY" + Z + X1 + Y
If P3 = 6 Then G5 = "SIXTY" + Z + X1 + Y
If P3 = 7 Then G5 = "SEVENTY" + Z + X1 + Y
If P3 = 8 Then G5 = "EIGHTY" + Z + X1 + Y
If P3 = 9 Then G5 = "NINETY" + Z + X1 + Y

End If

Y = "": If Val(Right$(B, 2)) <> 0 Then Y = "AND "

If P5 = 1 Then G4 = "ONE HUNDRED " + Y
If P5 = 2 Then G4 = "TWO HUNDRED " + Y
If P5 = 3 Then G4 = "THREE HUNDRED " + Y
If P5 = 4 Then G4 = "FOUR HUNDRED " + Y
If P5 = 5 Then G4 = "FIVE HUNDRED " + Y
If P5 = 6 Then G4 = "SIX HUNDRED " + Y
If P5 = 7 Then G4 = "SEVEN HUNDRED " + Y
If P5 = 8 Then G4 = "EIGHT HUNDRED " + Y
If P5 = 9 Then G4 = "NINE HUNDRED " + Y

PP = Val(Mid$(B, 8, 2))

If PP >= 1 And PP <= 19 Then

If PP = 1 Then G3 = "ONE "
If PP = 2 Then G3 = "TWO "
If PP = 3 Then G3 = "THREE "
If PP = 4 Then G3 = "FOUR "
If PP = 5 Then G3 = "FIVE "
```

```
If PP = 6 Then G3 = "SIX "
If PP = 7 Then G3 = "SEVEN "
If PP = 8 Then G3 = "EIGHT "
If PP = 9 Then G3 = "NINE "
If PP = 10 Then G3 = "TEN "
If PP = 11 Then G3 = "ELEVEN "
If PP = 12 Then G3 = "TWELVE "
If PP = 13 Then G3 = "THIRTEEN "
If PP = 14 Then G3 = "FOURTEEN "
If PP = 15 Then G3 = "FIFTEEN "
If PP = 16 Then G3 = "SIXTEEN "
If PP = 17 Then G3 = "SEVENTEEN "
If PP = 18 Then G3 = "EIGHTTEEN "
If PP = 19 Then G3 = "NINETEEN "

End If

If PP >= 20 Then

Z = ""

If P7 = 1 Then Z = "-ONE"
If P7 = 2 Then Z = "-TWO"
If P7 = 3 Then Z = "-THREE"
If P7 = 4 Then Z = "-FOUR"
If P7 = 5 Then Z = "-FIVE"
If P7 = 6 Then Z = "-SIX"
If P7 = 7 Then Z = "-SEVEN"
If P7 = 8 Then Z = "-EIGHT"
If P7 = 9 Then Z = "-NINE"

If P6 = 2 Then G3 = "TWENTY" + Z
If P6 = 3 Then G3 = "THIRTY" + Z
If P6 = 4 Then G3 = "FORTY" + Z
If P6 = 5 Then G3 = "FIFTY" + Z
If P6 = 6 Then G3 = "SIXTY" + Z
If P6 = 7 Then G3 = "SEVENTY" + Z
If P6 = 8 Then G3 = "EIGHTY" + Z
If P6 = 9 Then G3 = "NINETY" + Z

End If

G = G8 & G7 & G6 & G5 & G4 & G3

WrNumE = Trim$(G)

End Function
```

إخيراً إنتهت الرحلة الطويلة لهذا الكود التوأم !

وفي الختام أتمنى أن يفيد هذا الموضوع من يهمه الأمر من المبرمجين بالفيجوال بيسك ، وإلى مجموعة أكواد مبتكرة أخرى في الجزء الثالث إن شاء الله ،،

تأليف وإعداد / عبدالله خضر عبدالله الهوساوي

E-Mail : prog909@yahoo.com