



aprenderaprogramar.com

Ejercicios resueltos Visual Basic y .NET con arrays (arreglos), Do While, For, mod, ficheros, Ubound. Cálculo serie numérica (CU00344A)

Sección: Cursos

Categoría: Curso Visual Basic Nivel I

Fecha revisión: 2029

Autor: Mario R. Rancel

29

EJERCICIO RESUELTO

Generar el código correspondiente al siguiente ejercicio, relativo al cálculo de un valor medio a partir de una serie de datos que terminan en un señalero (centinela).

En un fichero tenemos las notas de un grupo de alumnos de una clase en un array Nota(1), Nota(2), ..., Nota(n - 1), establecidas entre 0 y 10. El último dato es un señalero con valor - 50. Se desea crear un programa que determine la nota media usando un array para almacenar las notas extraídas del fichero que inicialmente tenga índice máximo 10 y se amplíe a 20 en caso de que llegue a completarse el número de elementos admitidos en el array (y en caso de llegar al índice 20 se amplíe a 30 y así sucesivamente).

SOLUCIÓN

Consideraremos que el fichero notas.dat contiene los siguientes datos que nos servirán para probar el programa: 5, 7, 2, 3, 9, 5, 6, 8, - 50 (cada dato estará en una línea distinta, aquí los ponemos separados por comas por motivo de espacio).

Usaremos un label denominado Labelfinal para mostrar los resultados.

Código versiones VB menos recientes:

```
'[Curso VB aprenderaprogramar.com]
'[Determinación de media

Option Explicit
Dim Canal As Integer
Dim i%, Suma As Single
Dim Nota() As Single
Dim Media As Single

Private Sub Form_Load()
Form1.Caption = "Nota media"
Labelfinal.Alignment = 2
Labelfinal.FontBold = True
ReDim Nota(10)
Canal = FreeFile
Open "C:\notas.dat" For Input As Canal
i = 1
Input #Canal, Nota(i)
Do While Nota(i) >= 0
Suma = Suma + Nota(i)
If UBound(Nota) = i Then ReDim Preserve
Nota(i + 10)
i = i + 1
Input #Canal, Nota(i)
Loop
Media = Suma / (i - 1)
Close
Labelfinal = "La nota media de la clase es " &
Media
End Sub
```

Código versiones VB más recientes:

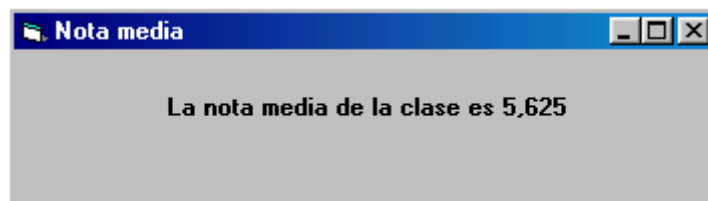
```
REM Curso Visual Basic aprenderaprogramar.com
Option Explicit On
Public Class Form1

Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
 MyBase.Load
Dim i As Integer
Dim Suma As Single
Dim Nota() As Single
Dim Media As Single

Me.Text = "Nota media"
Labelfinal.Font = New Font("Arial", 12,
FontStyle.Bold)
Labelfinal.TextAlign = ContentAlignment.MiddleCenter
ReDim Nota(10)
Dim myFileToRead As New System.IO.StreamReader
("C:\Users\Asus\Desktop\notas2.dat", False)
i = 1
Nota(i) = myFileToRead.ReadLine()

Do While Nota(i) >= 0
Suma = Suma + Nota(i)
If UBound(Nota) = i Then ReDim Preserve
Nota(i + 10)
i = i + 1
Nota(i) = myFileToRead.ReadLine()
Loop
Media = Suma / (i - 1)
myFileToRead.Close()
Labelfinal.Text = "La nota media de la
clase es " & Media
End Sub
End Class
```

El aspecto gráfico resultado al ejecutar el programa será similar a este:



Comentarios: En este caso, se establece una extracción de paquetes de datos de 10 en 10. Fijémonos en una línea: `If UBound(Nota) = i Then ReDim Preserve Nota(i + 10)`

Aquí establecemos que si el número de elementos que se extraen coincide con el tamaño máximo del vector nota que está recibiendo los datos del archivo, dicho vector se amplía en otros diez elementos. Cuando se llega al señalero de terminación es posible que cierto número de elementos del array, entre uno y diez, queden vacíos. Cuanto más grande hagamos el intervalo asociado a cada redimensionamiento, menor número de procesos es necesario. En cambio, mayor cantidad de elementos del array corremos el riesgo de que queden vacíos.

EJERCICIO RESUELTO

Generar el código para un programa que pide un número impar y realiza cálculos numéricos. En concreto el programa presentará las siguientes opciones:

- Introducir un valor entero impar comprendido entre 1 y 19
- Calcular la serie numérica $1 + 3 + 5 + \dots + n$
- Calcular $1 * 3 * 5 * \dots * n$
- Salir del programa.

Nota: Los cálculos se han de realizar a través de un vector que almacene los valores impares comprendidos entre 1 y n.

El pseudocódigo es el siguiente:

PROGRAMA IMPARES01 Curso Visual Basic aprenderaprogramar.com

Variables
Enteras: E, Dato, Cantidad, Valor()

1. Inicio

2. Mientras E <> 4 Hacer

2.1 Si E = 1 Entonces
Llamar Genera
FinSi

2.2 Si E = 2 y Dato <> 0 Entonces
Llamar CalculaSuma(Cantidad) PorValor
FinSi

```

2.3 Si E = 3 y Dato <> 0 Entonces
    Llamar CalculaMult(Cantidad) PorValor
FinSi
2.4 Mostrar "Elija opción: 1. Introducir dato
                2. Calcular 1 + 3 + 5 + ... + n
                3. Calcular 1 * 3 * 5 * ... * n
                4. Salir"

Pedir E

Repetir
3. Fin

Módulo Genera
Variables
    Enteras: Impar, i
    Booleanas: Par
1. Dato = 0
2. Mientras Dato <= 0 ó Dato > 19 ó Par = Verdadero Hacer
    Par = Falso
    Mostrar "Introduzca un número impar entre 1 y 19"
    Pedir Dato
Si Dato mod 2 = 0 Entonces
    Par = Verdadero : Mostrar "El número debe ser impar"
FinSi
Repetir
3. Impar = - 1 : i = 0
4. Hacer
    i = i + 1
    Impar = Impar + 2
    Redimensionar Valor(i)
    Valor(i) = Impar
Repetir Mientras Impar < Dato
5. Cantidad = i
FinMódulo

Módulo CalculaSuma(Numero: Enteros)
Variables
    Enteras: i, Suma
1. Suma = 0
2. Desde i = 1 hasta Numero Hacer
    Suma = Suma + Valor(i)
Siguiente
3. Mostrar "La suma 1 + 3 + 5 + ... + n vale", Suma
FinMódulo

Módulo CalculaMult(Cifra: Enteros)
Variables
    Enteras: i, Acumulado
1. Acumulado = 1
2. Desde i = 1 hasta Cifra Hacer
    Acumulado = Acumulado * Valor(i)
Siguiente
3. Mostrar "El producto 1 * 3 * 5 * ... * n vale", Acumulado
FinMódulo

```

SOLUCIÓN

Vamos a plantear un programa donde inicialmente se nos mostrarán cuatro botones: uno que llevará por título "Introducir valor" y que permitirá establecer el valor de n, otro denominado "Calcular serie 1+3+5..." que hará que se ejecute el cálculo con el valor de n existente, otro denominado "Calcular serie 1*3*5..." con igual función para obtener el valor de las multiplicaciones, y finalmente un botón "Salir" que permitirá salir del programa.

Código versiones VB menos recientes:

```

'[Curso VB aprenderaprogramar.com]
'[PROGRAMA IMPARES01]

Option Explicit
Dim Dato%, Cantidad%, Valor() As Integer

Private Sub Form_Load()
Form1.Caption = "Impares01"
LabelResultado.Alignment = 2
LabelResultado.FontBold = True
Command1.Caption = "Introducir valor"
Command2.Caption = "Calcular serie
1+3+5..."
Command3.Caption = "Calcular serie 1*3*5..."
Command4.Caption = "Salir"
End Sub

Private Sub Command1_Click()
Call Genera
End Sub

Private Sub Command2_Click()
If Dato <> 0 Then Call
CalculaSuma(Cantidad)
End Sub

Private Sub Command3_Click()
If Dato <> 0 Then Call CalculaMult(Cantidad)
End Sub

Private Sub Command4_Click()
Unload Form1
End
End Sub

Private Sub Genera()
Dim Impar%, i As Integer
Dim Par As Boolean
Dato = 0
Do While Dato <= 0 Or Dato > 19 Or Par =
True
Par = False
Dato = Val(InputBox("Introduzca un número
impar entre 1 y 19", "Impar"))

If Dato Mod 2 = 0 Then
Par = True
MsgBox("El número debe ser impar")
End If
Loop

'Continúa...

```

Código versiones VB más recientes:

```

REM Curso Visual Basic aprenderaprogramar.com
'[PROGRAMA IMPARES01]

Option Explicit On
Public Class Form1
Dim Dato, Cantidad, Valor() As Integer

Private Sub Form1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
Me.Text = "Impares01"
LabelResultado.Font = New Font("Arial", 12,
FontStyle.Bold)
LabelResultado.TextAlign =
ContentAlignment.MiddleCenter
Button1.Text = "Introducir valor"
Button2.Text = "Calcular serie 1+3+5..."
Button3.Text = "Calcular serie 1*3*5..."
Button4.Text = "Salir"
End Sub

Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
Call Genera()
End Sub

Private Sub Button2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
If Dato <> 0 Then Call
CalculaSuma(Cantidad)
End Sub

Private Sub Button3_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
If Dato <> 0 Then Call
CalculaMult(Cantidad)
End Sub

Private Sub Button4_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
Me.Close()
End
End Sub

Private Sub Genera()
Dim Impar, i As Integer
Dim Par As Boolean
Dato = 0
Do While Dato <= 0 Or Dato > 19 Or Par = True
Par = False
Dato = Val(InputBox("Introduzca un
número impar entre 1 y 19", "Impar"))

If Dato Mod 2 = 0 Then
Par = True
MsgBox("El número debe ser impar")
End If
Loop

'Continúa...

```

'Continuación del código

```

Impar = -1: i = 0
Do
    i = i + 1
    Impar = Impar + 2
    ReDim Preserve Valor(i)
    Valor(i) = Impar
Loop While Impar < Dato
Cantidad = i
LabelResultado = " "
End Sub

Private Sub CalculaSuma(ByVal Número As Integer)
Dim i%, Suma As Integer
Suma = 0
For i = 1 To Número
    Suma = Suma + Valor(i)
Next i
LabelResultado = vbCrLf & "La suma
1+3+5+...+n vale " & Suma
End Sub

Private Sub CalculaMult(ByVal Cifra As Integer)
Dim i As Integer
Dim Acumulado As Single
Acumulado = 1
For i = 1 To Cifra
    Acumulado = Acumulado * Valor(i)
Next i
LabelResultado = vbCrLf & "El producto
1*3*5*...*n vale " & Acumulado
End Sub

```

'Continuación del código

```

Impar = -1 : i = 0
Do
    i = i + 1
    Impar = Impar + 2
    ReDim Preserve Valor(i)
    Valor(i) = Impar
Loop While Impar < Dato
Cantidad = i
LabelResultado.Text = " "
End Sub

Private Sub CalculaSuma(ByVal Número As Integer)
Dim i, Suma As Integer
Suma = 0
For i = 1 To Número
    Suma = Suma + Valor(i)
Next i
LabelResultado.Text = vbCrLf & "La suma
1+3+5+...+n vale " & Suma
End Sub

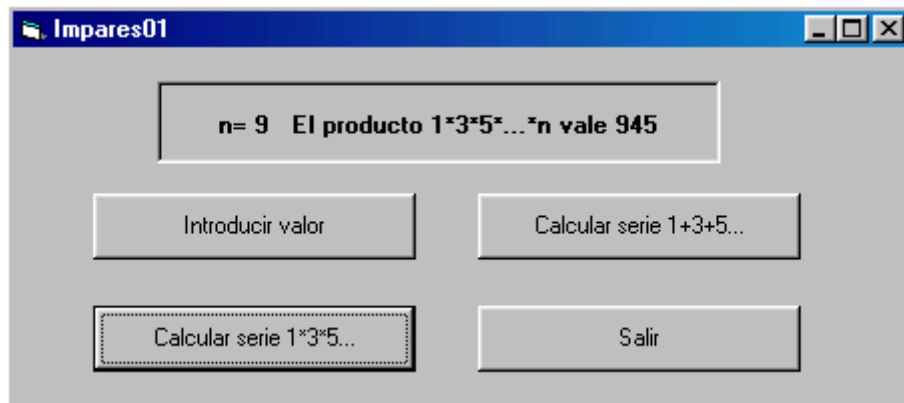
Private Sub CalculaMult(ByVal Cifra As Integer)
Dim i As Integer
Dim Acumulado As Single
Acumulado = 1
For i = 1 To Cifra
    Acumulado = Acumulado * Valor(i)
Next i
LabelResultado.Text = vbCrLf & "El
producto 1*3*5*...*n vale " & Acumulado
End Sub
End Class

```

Comentarios: Distintas variantes pueden ser introducidas respecto a lo planteado en pseudocódigo. Sin embargo, hemos tratado de respetar en la medida de lo posible el esquema allí propuesto. La única variación introducida es la definición de la variable Acumulado como tipo real (Single) en vez de entera (Integer). A ello estamos obligados ya que el rango de valores admisible para un tipo Integer con Visual Basic puede resultar insuficiente para los rangos numéricos con que se hace trabajar a esta variable, que pueden ser muy grandes. Obviar este hecho puede dar lugar a que a partir de un valor introducido igual o superior a cierto número se nos presente un mensaje de "Error 6 en tiempo de ejecución: desbordamiento". Para solventar este problema definimos la variable como tipo Single.

Hemos usado Redim Preserve en lugar de Redim para conservar los valores anteriores del vector Valor. Dicho vector se redimensiona en cada pasada del bucle, aunque podíamos haber optado por redimensionarlo cada cierto número de pasadas o asignarle un valor grande inicialmente y reducirlo al valor definitivo después. Estos y otros aspectos del programa se comentan en el curso "Bases de la programación nivel I" de aprenderaprogramar.com, donde se explica la lógica para la creación del programa y el diagrama de flujo del mismo.

El aspecto gráfico será algo así:



Próxima entrega: CU00345A

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:

http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=37&Itemid=61