



1. (1 punto) Completa los huecos de la siguiente tabla con las equivalencias en las respectivas bases, utilizando los dígitos necesarios en cada caso. Describe escuetamente el método utilizado.

Base 2	Base 4	Base 7	Base 10	Base 16
11000	120	33	24	18
10011001	2121	306	153	99
101101	231	63	45	2d

2. (2 puntos) Escribe un programa VB que para un número natural positivo n ($n > 0$) y un intervalo determinado por dos números enteros positivos p y q , calcule todos los "números amigos" de n en ese intervalo $[p, q]$. Supón que los datos son correctos.

Se dice que dos números naturales positivos x y z son amigos si: "la suma de todos los divisores de x , excepto x , es z y la suma de todos los divisores de z , excepto z , es x ".

Ejemplo: Los números **220** y **284** son *números amigos*.

- La suma de los divisores de **220** (excepto **220**):

$$1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284$$

- La suma de los divisores de **284** (excepto **284**):

$$1 + 2 + 4 + 71 + 142 = 220$$

```
Private Sub cmdCalcular_Click()
    Dim n As Integer
    Dim p As Integer
    Dim q As Integer
    Dim z As Integer
    Dim sumN As Integer

    n = InputBox ("Introduce el número")
    p = InputBox ("Introduce el extremo inferior del intervalo")
    q = InputBox ("Introduce el extremo superior del intervalo")
    sumN = SumaDivisores(n)
    For z = p To q Step 1
        If sumN = z And SumaDivisores(z) = n Then pctRes.Print z
    Next z
End Sub

Function SumaDivisores(ByVal n As Integer) As Integer
    Dim i As Integer
    Dim sum As Integer

    For i = 1 To n - 1 Step 1
        If n Mod i = 0 Then sum = sum + i
    Next i
    SumaDivisores = sum
End Function

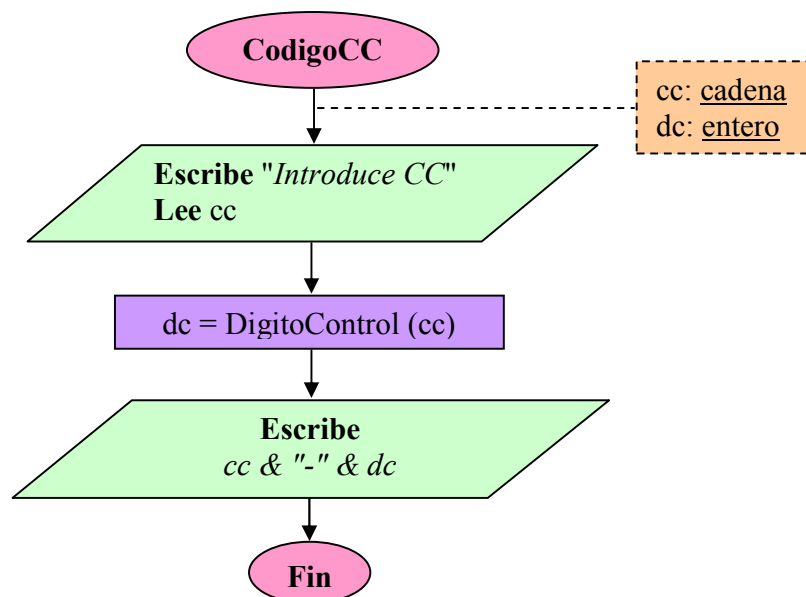
Private Sub cmdCalcular1_Click() ' Otra manera para conseguir lo mismo
    ...
    sumN = SumaDivisores(n)
    If sumN >= p And sumN <= q And SumaDivisores(sumN) = n Then pctRes.Print sumN
End Sub
```



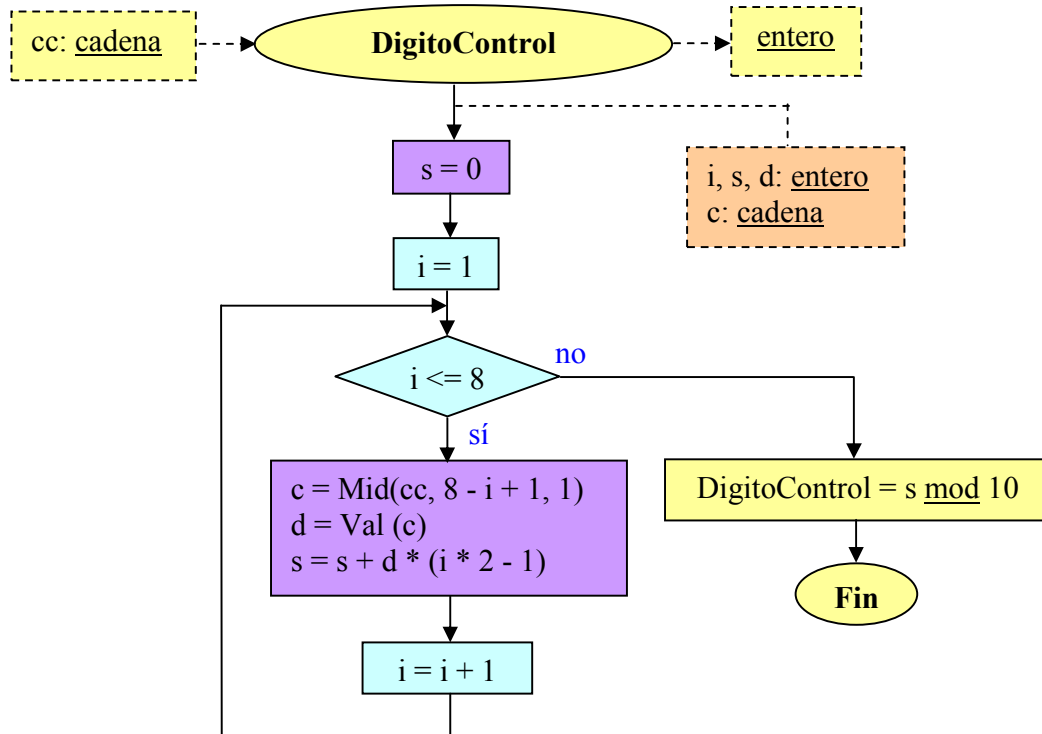
3. (3 puntos) **Diseña** (el diagrama de flujo) y **codifica** un programa VB que **lea** (mediante la función InputBox) un número de cuenta bancaria de 8 dígitos como una cadena de caracteres y **calcule** su dígito de control **utilizando estructuras repetitivas**, mostrando (mediante MsgBox) como resultado la cuenta y dicho dígito.

Se supone que el número de cuenta bancaria que se va a leer es correcto, es decir, contiene 8 dígitos decimales. Ejemplos:

Número de Cuenta	Algoritmo de Dígito de Control	Resultado
11334478	$8 \times 1 + 7 \times 3 + 4 \times 5 + 4 \times 7 + 3 \times 9 + 3 \times 11 + 1 \times 13 + 1 \times 15 = 165$ <i>El resto de dividir 165 entre 10 es 5</i>	11334478-5
10000003	$3 \times 1 + 0 \times 3 + 0 \times 5 + 0 \times 7 + 0 \times 9 + 0 \times 11 + 0 \times 13 + 1 \times 15 = 18$ <i>El resto de dividir 18 entre 10 es 8</i>	10000003-8



```
Private Sub cmdControl_Click()  
    Dim cc As String  
    Dim dc As Integer  
  
    cc = InputBox("Introduce un número de cuenta corriente")  
    dc = DigitoControl(cc)  
    MsgBox (cc & "-" & dc)  
End Sub
```



```
Function DigitoControl(ByVal cc As String) As Integer
    Dim i As Integer
    Dim s As Integer
    Dim d As Integer
    Dim c As String

    sum = 0
    For i = 1 To 8 Step 1
        c = Mid(cc, 8 - i, 1)
        d = Val(c)
        s = s + d * (i * 2 - 1)
    Next i
    DigitoControl = s Mod 10
End Function
```

```
Function DigitoControl(ByVal cc As String) As Integer
    Dim i As Integer
    Dim j As Integer
    Dim s As Integer

    sum = 0
    For i = 8 To 1 Step -1
        s = s + Val(Mid(cc, i, 1)) * j
    Next i
    DigitoControl = s Mod 10
End Function
```

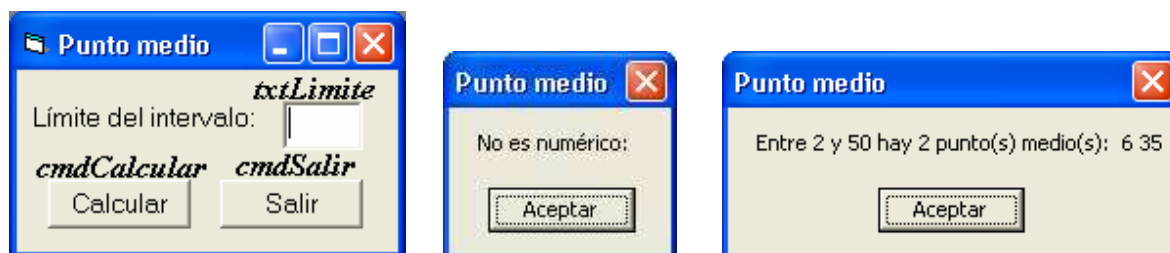


4. (4 puntos) Se dice que un número x ($x \geq 2$) es **punto medio** si el valor que se obtiene al calcular $1 + 2 + 3 + \dots + (x-1)$ se puede obtener también sumando unos cuantos números consecutivos que siguen a x . Ejemplos:

- El número **6** es punto medio porque $1 + 2 + 3 + 4 + 5 = 15$ y por otro lado $7 + 8 = 15$
- El número **35** es también punto medio porque $1 + 2 + \dots + 34 = 595$ y por otro lado $36 + 37 + \dots + 49 = 595$.

- Diseña** (el diagrama de flujo) y **codifica** una **función** VB que calcule si un número x ($x \geq 2$) es **punto medio**.
- Escribir el **programa** VB que obtenga un número n ($n \geq 2$) del cuadro *txtLimite* y calcule (botón **Calcular**) cuántos puntos medios se encuentran en el intervalo $[2, n]$ mostrando los **puntos medios** y el número de ellos según los ejemplos. El programa debe verificar que n es correcto. Al hacer clic sobre el botón **Salir** finalizará la ejecución del programa.

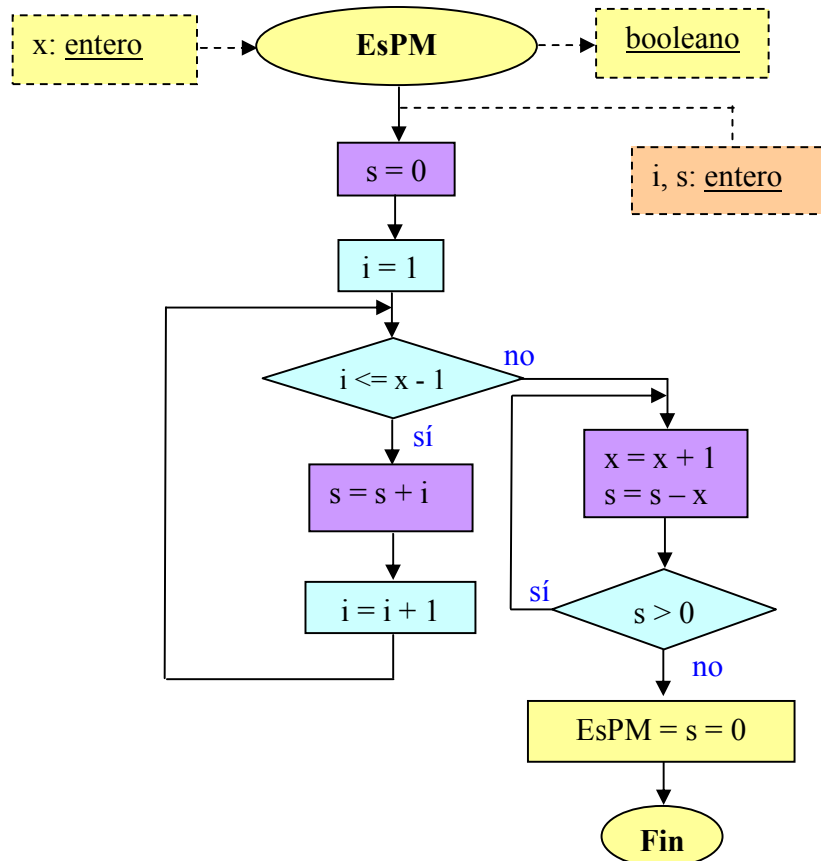
Ejemplos de ejecución (con entradas “” y “50”):



A continuación se propone el resultado a mostrar en pantalla frente a diferentes valores de entrada:

Entrada	Resultado
	No es numérico:
unO	No es numérico: unO
-12	No es mayor o igual que dos: -12
1	No es mayor o igual que dos: 1
3	Entre 2 y 3 no hay ningún punto medio
7	Entre 2 y 7 hay 1 punto(s) medio(s): 6
35	Entre 2 y 35 hay 2 punto(s) medio(s): 6 35

Mid (ByVal cad As String, ByVal ini As Long, [ByVal lon As Long]) As String	Subcadena desde <i>ini</i> hasta la longitud <i>lon</i> indicada, o hasta el final de la cadena si no se indica
IsNumeric (ByVal cad As String) As Boolean	Verifica si una cadena contiene un valor numérico
Val (ByVal cad As String) As Integer	Valor numérico de una cadena



```
Function EsPM (ByVal x As Integer) As Boolean
    Dim i As Integer
    Dim s As Integer

    s = 0
    For i = 1 To x - 1 Step 1
        s = s + i
    Next i

    Do
        x = x + 1
        s = s - x
    Loop While s > 0
    EsPM = s = 0
End Function
```

b)

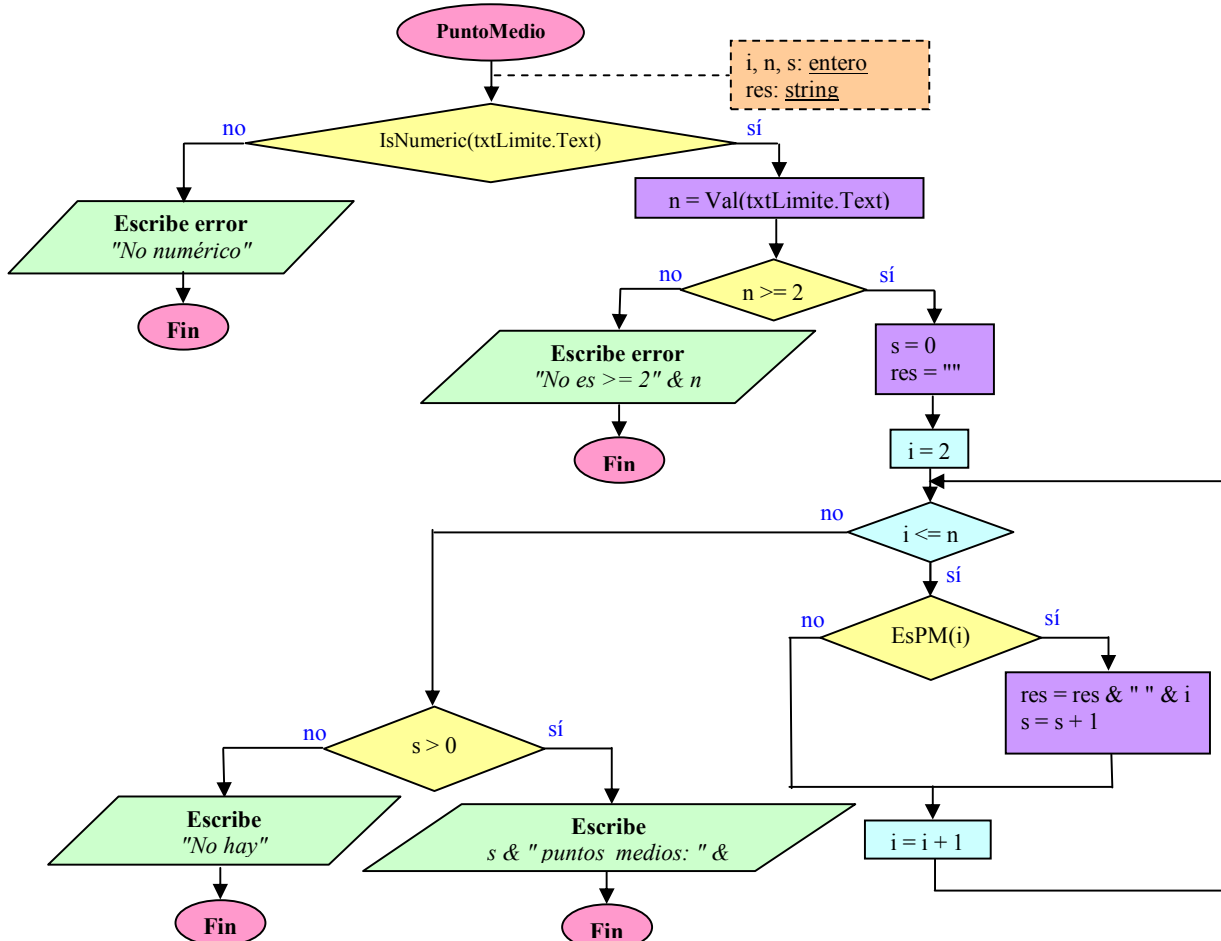
```
Private Sub cmdSalir_Click()
    End
End Sub
```



Fundamentos de Informática

Examen – 18 de Septiembre de 2006

Mecánica y Electricidad



```

Private Sub cmdCalcular_Click()
    Dim i As Integer, n As Integer, s As Integer
    Dim res As String
    If Not IsNumeric(txtLimite.Text) Then
        MsgBox ("No es numérico: " & txtLim.Text)
    Else
        n = Val(txtLimite.Text)
        If n < 2 Then
            MsgBox ("No es mayor o igual que dos: " & n)
        Else
            res = ""
            s = 0
            For i = 2 To n Step 1
                If EsPM(i) Then
                    res = res & " " & i
                    s = s + 1
                End If
            Next i
            If s > 0 Then MsgBox "Entre 2 y " & n & " hay " & s & " punto(s) medio(s): " & res
            Else
                MsgBox "Entre 2 y " & n & " no hay ningún punto medio"
            End If
        End If
    End Sub
  
```