

SIST. Y APLICACIONES INFORMÁTICAS**CURSO: 2009 - 2010****SEMANA: 11****PROFESOR:** Javier Sánchez Fernández**TEMAS:** Complemento Temas 29 Ejercicios Resueltos*Ejercicio 1:*

Crear un array unidimensional de enteros el cual deberemos rellenar mediante teclado. Se deberá pedir el número de elementos del array antes de empezar a rellenarlo. Una vez rellenado este habrá que mostrar el contenido de este en columnas.

Module Module1

Sub Main()

Dim vector() As Integer

Dim longitud As Integer

Dim i As Integer = 0, j As Integer

Dim salir As Boolean

Try

Console.WriteLine("Introduzca el número de elementos")

longitud = Console.ReadLine()

Catch e As Exception

Console.WriteLine("El error ocurrido es : {0}", e.ToString)

End Try

ReDim vector(longitud)

For i = 0 To longitud

Do

Try

Console.WriteLine("Introduzca el elemento {0}", i)

vector(i) = CType(Console.ReadLine(), Integer)

salir = True

Catch e As Exception

salir = False

Console.WriteLine(e.ToString)

Catch e As InvalidCastException

salir = False

Console.WriteLine("El valor introducido no es un entero")

End Try

Loop Until salir

Next

```

j = 1
For i = 0 To vector.Length - 1
    Console.Write("{0}{1}", vector(i).ToString, vbTab)
    If j = 3 Then
        Console.WriteLine()
        j = 0
    End If
    j += 1
Next
Console.ReadLine()

```

End Sub

End Module

Ejercicio 2:

Introducir números en un array de enteros hasta introducir el número 0. En ese momento, deberá mostrar el contenido del array y dar la media y la suma de los valores de este array.

Module Module1

Introducir números en un array de enteros hasta introducir el número 0. En ese momento, deberá mostrar el contenido del array y dar la media y la suma de los valores de este array

```

Sub Main()
    Dim matriz() As Integer, numero As Integer = 1
    Dim acumulado As Integer, elementos As Integer = -1

    Do
        elementos += 1
        Console.WriteLine("Introduzca un número")
        numero = CType(Console.ReadLine(), Integer)
        acumulado += numero
        ReDim Preserve matriz(elementos)
        matriz(elementos) = numero
    Loop Until numero = 0
    Console.WriteLine("La media es {0}", acumulado / elementos)
    Console.WriteLine("La suma de los elementos es {0}", acumulado)
    Console.ReadLine()

```

End Sub

Ejercicio 3:

Deberemos rellenar dos vectores de números reales. Para ello, iremos introduciendo números hasta introducir un cero en ese momento, comenzaremos a rellenar la segunda matriz hasta que nuevamente introduzcamos un cero. Una vez finalizado este proceso, deberemos rellenar y mostrar el contenido de un tercer vector con todos los valores del primero seguidos de los valores del segundo encolumnados a tres columnas.

Module Module1

Sub Main()

Dim mtr1(), mtr2(), mtr_res(), numero As Single

Dim i As Integer = -1

Console.WriteLine("Introduciendo los valores del primer vector")

Do

Console.Write("Introduzca valor {0}:", i + 1)

numero = CType(Console.ReadLine(), Single)

If numero <> 0 Then

i += 1

ReDim Preserve mtr1(i)

mtr1(i) = numero

End If

Loop Until numero = 0

i = -1

Console.WriteLine("Introduciendo los valores del segundo vector")

Do

Console.Write("Introduzca valor {0}:", i + 1)

numero = CType(Console.ReadLine(), Single)

If numero <> 0 Then

i += 1

ReDim Preserve mtr2(i)

mtr2(i) = numero

End If

Loop Until numero = 0

ReDim mtr_res(mtr1.Length + mtr2.Length - 1)

For i = 0 To mtr1.Length - 1

mtr_res(i) = mtr1(i)

Next

For i = 0 To mtr2.Length - 1

mtr_res(i + mtr1.Length) = mtr2(i)

Next

Console.WriteLine("{0}Listado de la matriz resultante", vbCrLf)

For i = 0 To mtr_res.Length - 1

Console.Write("{0} {1}", mtr_res(i), vbTab)

Next

Console.ReadLine()

End Sub

End Module

Ejercicio 4:

Crear una aplicación con las siguientes restricciones :

- El usuario irá introduciendo valores que corresponderán con radios de circunferencias.
- Cualquier número igual o menor que cero será el que nos indique que no seguirá introduciendo valores.
- Debemos crear una matriz que tendrá tres filas, la primera contendrá el valor del radio de la circunferencia, la segunda el valor del perímetro de esta y la tercera su área. El número de columnas dependerá del momento en el que se introduzca el valor de parada.
- Debemos utilizar un tipo enumerado que contendrá el valor radio, perímetro y área y que utilizaremos para hacer referencia a la fila que estemos leyendo o en la que estemos escribiendo.
- Debemos mostrar el resultado en una matriz en la que la primera columna corresponderá al radio, la segunda al perímetro y la tercera al área. Por ejemplo :

Radio	Perímetro	Área
1	6,28	3,14
2	12,56	12,56

Module Module1

Enum c

radio = 0

perimetro = 1

area = 2

End Enum

Sub Main()

Dim fila As c

Dim radios(), tabla(), numero As Single

Dim i As Integer = -1

```
Console.WriteLine("Introduciendo los valores de los radios")
```

```
Do
```

```
    Console.Write("Introduzca valor {0}:", i + 1)
```

```
    numero = CType(Console.ReadLine(), Single)
```

```
    If numero <> 0 Then
```

```
        i += 1
```

```
        ReDim Preserve radios(i)
```

```
        radios(i) = numero
```

```
    End If
```

```
Loop Until numero = 0
```

```
ReDim tabla(2, radios.Length - 1)
```

```
For i = 0 To radios.Length - 1
```

```
    tabla(c.radio, i) = radios(i)
```

```
    tabla(c.perimetro, i) = 2 * tabla(c.radio, i) * Math.PI
```

```
    tabla(c.area, i) = tabla(c.radio, i) ^ 2 * Math.PI
```

```
Next
```

```
Console.WriteLine("{0}Radio{1}Perimetro{2}Area", vbCrLf, vbTab, vbCrLf)
```

```
For i = 0 To radios.Length - 1
```

```
    Console.WriteLine("{0}{1}{2}{3}{4}", tabla(c.radio, i), vbCrLf, _  
        tabla(c.perimetro, i), vbCrLf, tabla(c.area, i))
```

```
Next
```

```
Console.WriteLine("{0}{1}Pulse INTRO para finalizar", vbCrLf, vbCrLf)
```

```
Console.ReadLine()
```

```
End Sub
```

```
End Module
```

Ejercicio 5:

Vamos a construir una aplicación que nos pida tres números por teclado y de cómo resultado el menor de tres números

Module Module1

Sub Main()

Dim num1, num2, num3, c As Integer

Dim menor As Integer = 100

Console.WriteLine("Introduzca tres numero para saber cual es el menor:

")

Console.Write("Numero 1:")

num1 = Console.ReadLine()

Console.Write("Numero 2:")

num2 = Console.ReadLine()

Console.Write("Numero 3:")

num3 = Console.ReadLine()

If num1 < num2 Then

If num1 < num3 Then

Console.WriteLine("El mas pequeño: {0}", num1)

Else

Console.WriteLine("El mas pequeño: {0}", num3)

End If

Else

If num2 < num3 Then

Console.WriteLine("El mas pequeño: {0}", num2)

Else

Console.WriteLine("El mas pequeño: {0}", num3)

End If

End If

Console.ReadLine()

End Sub

End Module

Ejercicio 6:

Realizar una aplicación que lea una fecha representada por dos enteros , mes y año y dé como resultado los días correspondientes al mes, es decir:

Mes (##): 10

Año (####):2004

El mes de Octubre tiene 31 días

Module Module1

Sub Main()

Dim Mes, Año As Integer

Console.Write("Introduzca el mes: ")

Mes = Console.ReadLine

Console.Write("Introduzca el año: ")

Año = Console.ReadLine

Select Case Mes

Case 1, 3, 5, 7, 8, 10, 12

Console.Write("El mes tiene 31 Dias")

Case 4, 6, 9, 11

Console.Write("El mes tiene 30 dias")

Case 2

If (Año Mod 4 <> 0) And ((Año Mod 100 <> 0) Or (Año Mod 400 = 0)) Then

Console.Write("El mes tienes 28 dias")

Else

Console.Write("El mes tiene 29 dias")

End If

Case Else

Console.Write("Has introducido un mes incorrecto")

End Select

Console.Read()

End Sub

End Module

Ejercicio 7:

Realizar un programa que saque por pantalla los números comprendidos entre 1 y 50 que cumplan la expresión:

$$z^2 = x^2 + y^2$$

donde z, x e y son números enteros positivos. El resultado se mostrara de la siguiente forma

Z	X	Y
5	3	4
13	5	12

```
Imports System
Module Module1
```

```
Public Sub Main()
```

```
Dim x = 1, y = 1, z As Integer = 0
```

```
Dim Th As Char = Convert.ToChar(&H9)
```

```
Console.WriteLine("Z" & Th & "X" & Th & "Y")
```

```
Console.WriteLine("_____")
```

```
While (x <= 100)
```

```
z = Math.Sqrt(x ^ 2 + y ^ 2)
```

```
While y <= 100 AndAlso z <= 100
```

```
If z ^ 2 = x ^ 2 + y ^ 2 Then
```

```
Console.WriteLine(x & Th & y & Th & z)
```

```
End If
```

```
y += 1
```

```
z = Math.Sqrt(x ^ 2 + y ^ 2)
```

```
End While
```

```
x += 1
```

```
y = x
```

```
End While
```

```
End Sub
```

```
End Module
```


Ejercicio 8:

Realizar una función que reciba dos valores y devuelva el producto. El producto se resuelve mediante sumas sucesivas.

Module Module1

```
Function Multiplicar(ByVal nValor1 As Integer, ByVal nValor2 As Integer)  
As Integer
```

```
    Dim nContador, nTotal As Integer
```

```
    nTotal = 0
```

```
    For nContador = 1 To nValor2
```

```
        nTotal += nValor1
```

```
    Next
```

```
    Return nTotal
```

```
End Function
```

```
Sub Main()
```

```
    Dim nValor1, nValor2 As Integer
```

```
    Console.WriteLine("Introduzca el primer valor: ")
```

```
    nValor1 = Console.ReadLine()
```

```
    Console.WriteLine("Introduzca el segundo valor: ")
```

```
    nValor2 = Console.ReadLine()
```

```
    Console.WriteLine(nValor1 & " x " & nValor2 & " = " &
```

```
Multiplicar(nValor1, nValor2))
```

```
End Sub
```

End Module

Ejercicio 9:

El ejercicio de la calculadora hacerlo con procedimientos.

Realizar una calculadora, el programa nos mostrara un menu:

- 1 Suma
- 2 Resta
- 3 Multiplicar
- 4 Dividir
- 0 Salir

Seleccionaremos una opción y nos solicitara dos números realizaremos la operación y mostrara el resultado por pantalla. El programa se ejecutara hasta que pulsemos la opción 0

Module Module1

```
Function Sumar(ByVal nOp1 As Integer, ByVal nOp2 As Integer)
    Return nOp1 + nOp2
End Function
```

```
Function Restar(ByVal nOp1 As Integer, ByVal nOp2 As Integer)
    Return nOp1 - nOp2
End Function
```

```
Function Multiplicar(ByVal nOp1 As Integer, ByVal nOp2 As Integer)
    Return nOp1 * nOp2
End Function
```

```
Function Dividir(ByVal nOp1 As Integer, ByVal nOp2 As Integer)
    If nOp2 <> 0 Then
        Return nOp1 / nOp2
    End If
End Function
```

```
Sub Main()
    Dim nOpcion, nOp1, nOp2 As Integer
    Do
        Console.WriteLine("1. Suma")
        Console.WriteLine("2. Resta")
        Console.WriteLine("3. Multiplicacion")
        Console.WriteLine("4. Division")
        Console.WriteLine("0. Salir")
        Console.Write("Seleccionar una opcion: ")
        nOpcion = Console.ReadLine()
        If nOpcion <> 0 Then
            Console.Write("Introduzca el primer operando: ")
            nOp1 = Console.ReadLine()
            Console.Write("Introduzca el segundo operando: ")
            nOp2 = Console.ReadLine()
        End If
        Select Case nOpcion
            Case 0
                Console.WriteLine("FIN DE PROGRAMA")
            Case 1
                Console.WriteLine(nOp1 & " + " & nOp2 & " = " & Sumar(nOp1,
nOp2))
            Case 2
```

```
        Console.WriteLine(nOp1 & " - " & nOp2 & " = " & Restar(nOp1,  
nOp2))
```

```
    Case 3
```

```
        Console.WriteLine(nOp1 & " * " & nOp2 & " = " &  
Multiplicar(nOp1, nOp2))
```

```
    Case 4
```

```
        Console.WriteLine(nOp1 & " / " & nOp2 & " = " & Dividir(nOp1,  
nOp2))
```

```
    End Select
```

```
    Loop Until nOpcion = 0
```

```
End Sub
```

```
End Module
```

Ejercicio 10:

Realizar una función que reciba dos enteros y cambie el valor del primero por la suma de ambos y el valor del segundo por la diferencia de ambos.

Module Module1

```
Sub CambiaValores(ByRef nValor1 As Integer, ByRef nValor2 As Integer)
    Dim nAux As Integer
    nAux = nValor1 + nValor2
    nValor2 = nValor1 - nValor2
    nValor1 = nAux
End Sub
```

```
Sub Main()
    Dim nValor1, nValor2 As Integer
    Console.WriteLine("Introduzca un valor: ")
    nValor1 = Console.ReadLine()
    Console.WriteLine("Introduzca un valor: ")
    nValor2 = Console.ReadLine()
    CambiaValores(nValor1, nValor2)
    Console.WriteLine("nValor1 = " & nValor1)
    Console.WriteLine("nValor2 = " & nValor2)
End Sub
```

End Module

Ejercicio 11:

Realizar una función que reciba tres tablas tb1, tb2, tb3 de 10 enteros cada una. La función copiará los valores pares de tb1 en tb2 y los valores impares en tb3. Visualizar el resultado.

Module Module1

```
Sub CopiarTablas(ByVal tb1() As Integer, ByRef tb2() As Integer, ByRef
tb3() As Integer)
    Dim nIndex, nIndex2, nIndex3 As Integer
    nIndex2 = 0 : nIndex3 = 0
    For nIndex = 0 To tb1.Length - 1
        If tb1(nIndex) Mod 2 = 0 Then
            ReDim Preserve tb2(nIndex2)

```

```
tb2(nIndice2) = tb1(nIndice)
nIndice2 += 1
```

```
Else
```

```
ReDim Preserve tb3(nIndice3)
tb3(nIndice3) = tb1(nIndice)
nIndice3 += 1
```

```
End If
```

```
Next
```

```
End Sub
```

```
Sub Main()
```

```
Dim tb1(9), tb2(), tb3(), nIndice As Integer
```

```
For nIndice = 0 To tb1.Length - 1
```

```
Console.WriteLine("tb1(" & nIndice & "): ")
tb1(nIndice) = Console.ReadLine()
```

```
Next
```

```
CopiarTablas(tb1, tb2, tb3)
```

```
Console.WriteLine("PARES")
```

```
For nIndice = 0 To tb2.Length - 1
```

```
Console.WriteLine(tb2(nIndice))
```

```
Next
```

```
Console.WriteLine("IMPARES")
```

```
For nIndice = 0 To tb3.Length - 1
```

```
Console.WriteLine(tb3(nIndice))
```

```
Next
```

```
End Sub
```

```
End Module
```

Ejercicio 12:

Realizar un programa que nos pida una fecha Día, mes, año y de como resultado el nº del tarot. El programa verificara si la fecha es correcta, e. d. Si los valores están en el rango de valores permitidos.

El nº del tarot consiste en sumar los números de su fecha de nacimiento y reducirlos a un único dígito

Module Module1**Sub** Main()

Dim Dia, Mes, Año, Tarot, fallo **As Integer**

fallo = 0

Console.Write("Introduce tu día de nacimiento: ")

Dia = Console.ReadLine

Console.Write("Introduce el mes de tu nacimiento: ")

Mes = Console.ReadLine

Console.Write("Introduce el año de tu nacimiento: ")

Año = Console.ReadLine

Select Case Mes

Case 1, 3, 5, 7, 8, 10, 12

If Dia > 31 **Then**

fallo = 1

End If

Case 4, 6, 9, 11

If Dia > 30 **Then**

fallo = 1

End If

Case 2

If ((Año **Mod** 4 <> 0) **Or** ((Año **Mod** 100 = 0) **And** (Año **Mod** 400 <> 0))) **Then**

If Dia > 28 **Then**

fallo = 1

End If

Else

If Dia > 29 **Then**

fallo = 1

End If

End If

Case Else

fallo = 2

End Select

```
If Dia < 1 Then
    fallo = 1
End If
If fallo = 0 Then
    Tarot = Dia + Año + Mes
    Año = Tarot / 1000
    Mes = Tarot / 100 - (Año * 10)
    Dia = Tarot / 10 - (Año * 100 + Mes * 10)
    Tarot = Tarot - (Año * 1000 + Mes * 100 + Dia * 10)
    Tarot = Tarot + Dia + Mes + Año
    While Tarot > 9
        Año = Tarot / 10
        Tarot = Tarot - (Año * 10)
    End While
    Console.WriteLine("Tu numero del tarot es el " & Tarot)
Elseif fallo = 1 Then
    Console.WriteLine("Ha introducido un dia erroneo")
Elseif fallo = 2 Then
    Console.WriteLine("Ha introducido un mes erroneo")
End If
Console.Read()
End Sub

End Module
```

Ejercicio 13

Vamos a simular el proceso de poner notas a un alumno, el curso se compone de tres evaluaciones, cuya nota va desde el 0 hasta el 10, Como entrada pediremos los datos personales del alumno, y las notas de las tres evaluaciones, debe calcular la media de las tres y poner las notas según el siguiente criterio

9-10	Sobresaliente
7-8.9	Notable
5-6.9	Aprobado
4-4.9	Insuficiente
1-3.9	Muy deficiente

Si alguna de las evaluaciones no se ha presentado el alumno se le calificara con un cero y al final debe dar un mensaje de No Presentado

Module Module1

Sub Main()

```
Dim Alumno As String, not1, not2, not3, resultado As Integer
Console.WriteLine("Introduzca el nombre del alumno:")
Alumno = Console.ReadLine()
Console.WriteLine("Nota de la primera evaluación")
not1 = Console.ReadLine()
Console.WriteLine("Nota de la segunda evaluación")
not2 = Console.ReadLine()
Console.WriteLine("Nota de la tercera evaluación")
not3 = Console.ReadLine()
Console.WriteLine("Alumno:{0},Notas:{1},{2},{3}", Alumno, not1, not2,
not3)
If not1 = 0 Or not2 = 0 Or not3 = 0 Then
    Console.WriteLine("El alumno no se ha presentado a alguna de las
evaluaciones")
Else
    resultado = (not1 + not2 + not3) / 3
    Select Case resultado
        Case Is > 9
            Console.WriteLine("{0} ha obtenido Sobresaliente como nota
final", Alumno)
        Case Is > 7
            Console.WriteLine("{0} ha obtenido Notable como nota final",
Alumno)
```



```
        Case Is > 5
            Console.WriteLine("{0} ha obtenido Aprobado como nota final",
Alumno)
        Case Is > 4
            Console.WriteLine("{0} ha obtenido Insuficiente como nota
final", Alumno)
        Case Else
            Console.WriteLine("{0} ha obtenido Muy deficiente como nota
final", Alumno)
        End Select

    End If

End Sub

End Module
```

Ejercicio 14

Escribir un programa que ordene un array unidimensional con el algoritmo de inserción directa, el algoritmo de selección y el algoritmo de la burbuja

Ordenación por Selección

```
Module Module1
    Const max As Integer = 20000
    Dim vector(max) As Integer

    Private Sub rellenar(ByRef vec() As Integer)

        Dim i As Integer
        vec(0) = 0
        For i = 1 To max
            vec(i) = max + 1 - i
        Next
    End Sub

    Private Sub seleccion()
        Dim i, j, k As Integer
        Dim inicio, fin, tiempo As Long
        Dim diff As TimeSpan
```

```

inicio = Now.Ticks
For i = 1 To vector.Length - 2
    k = i
    vector(0) = vector(i)
    For j = i + 1 To vector.Length - 1
        If vector(j) < vector(0) Then
            k = j
            vector(0) = vector(k)
        End If
    Next
    vector(k) = vector(i)
    vector(i) = vector(0)
Next
fin = Now.Ticks
tiempo = fin - inicio
Console.WriteLine("El tiempo con seleccion es transcurrido es {0}",
tiempo / diff.TicksPerSecond)
End Sub

Sub Main()
    rellenar(vector)
    seleccion()
    Console.Read()
End Sub

End Module

```

Algoritmo de Inserción

```

Module Module1
    Const max As Integer = 20000
    Dim vector(max) As Integer

    Private Sub rellenar(ByRef vec() As Integer)

        Dim i As Integer
        vec(0) = 0
        For i = 1 To max
            vec(i) = max + 1 - i
        Next
    End Sub

```

```
Private Sub Insercion()
```

```
    Dim i, j As Integer
```

```
    Dim inicio, fin, tiempo As Long
```

```
    Dim diff As TimeSpan
```

```
    inicio = Now.Ticks
```

```
    For i = 2 To vector.Length - 1
```

```
        vector(0) = vector(i)
```

```
        j = i
```

```
        Do While (vector(0) < vector(j - 1)) 'AndAlso j > 0
```

```
            vector(j) = vector(j - 1)
```

```
            j = j - 1
```

```
        Loop
```

```
        vector(j) = vector(0)
```

```
    Next
```

```
    fin = Now.Ticks
```

```
    tiempo = fin - inicio
```

```
    Console.WriteLine("El tiempo con insercion es transcurrido es {0}",  
tiempo / diff.TicksPerSecond)
```

```
End Sub
```

```
Sub Main()
```

```
    rellenar(vector)
```

```
    Insercion()
```

```
    Console.Read()
```

```
End Sub
```

```
End Module
```

Algoritmo de la burbuja

```
Module Module1
```

```
    Const max As Integer = 20000
```

```
    Dim vector(max) As Integer
```

```
Private Sub rellenar(ByRef vec() As Integer)
```

```
    Dim i As Integer
```

```
    vec(0) = 0
```

```
    For i = 1 To max
```

```
        vec(i) = max + 1 - i
```

```
    Next
```

End Sub

Private Sub burbuja()

Dim i, j As Integer

Dim inicio, fin, tiempo As Long

Dim diff As TimeSpan

inicio = Now.Ticks

For i = 1 To vector.Length - 2

For j = 1 To vector.Length - (i + 1)

If vector(j) > vector(j + 1) Then

vector(0) = vector(j + 1)

vector(j + 1) = vector(j)

vector(j) = vector(0)

End If

Next

Next

fin = Now.Ticks

tiempo = fin - inicio

Console.WriteLine("El tiempo con burbuja es transcurrido es {0}", tiempo / diff.TicksPerSecond)

End Sub

Sub Main()

rellenar(vector)

burbuja()

Console.Read()

End Sub

End Module

Ejercicio 15

El isótopo radiactivo plutonio-235 tiene una vida media de 26 minutos. La vida media es el tiempo que se necesita para que se degenere la mitad del isótopo. Por tanto, después de 26 minutos sólo quedará la mitad del isótopo. Después de otros 26 minutos solo quedará la mitad de esta mitad, o un cuarto de la cantidad original. El trabajo consiste en escribir un programa que determine cuánto tiempo (el tiempo de vida más cercano) necesita el plutonio-235 para degenerarse hasta que quede un porcentaje determinado. El porcentaje es un valor real introducido por el usuario

Module Module1

Sub Main()

Const VidaMedia = 26

Dim Porcentaje = 0, Desintegrado As Double = 50

Dim Continuar As Boolean = True

Dim Tiempo = 0, Horas = 0, Minutos = 0, cont As Integer = 0

Console.WriteLine("¿Indique el porcentaje de resto deseado?")

Porcentaje = Console.ReadLine()

Do While Desintegrado < Porcentaje

cont += 1

Desintegrado += Desintegrado / 2

Loop

Tiempo = cont * VidaMedia

Horas = Tiempo \ 60

Minutos = Tiempo Mod 60

Console.WriteLine("El numero de veces que se ha desintegrado es {0}",
cont)

Console.WriteLine("El tiempo que ha pasado para desintegrarse en ese
porcentaje es {0} Horas, {1} minutos", Horas, Minutos)

End Sub

End Module

Ejercicio 16

Acaba de haber unas elecciones al ayuntamiento de la ciudad. Tenemos que hacer un análisis de los votos de los cuatro candidatos presentados en cada uno de los distritos de la ciudad. Queremos saber cuantos votos totales recibio cada candidato y cuantos votos totales hubo en cada distrito. Por ultimo el programa nos debe mostrar el ganador. Los nombres de los candidatos y el nombre de los distritos los introduciremos desde teclado.

Module Module1

Const NUMDISTRITOS = 4, NUMCANDIDATOS As Integer = 3

Sub Main()

Dim contvotos(NUMDISTRITOS, NUMCANDIDATOS) As Integer,
totales(NUMCANDIDATOS)

Dim nombre(NUMCANDIDATOS) As String

Dim i, j As Integer

obtenernombre(nombre)

votoscero(contvotos)

For i = 0 To NUMDISTRITOS

For j = 0 To NUMCANDIDATOS

Console.Write("el numero de votos para el candidato {0} en el
distrito {1} :", nombre(j), i)

contvotos(i, j) = Conversion.Val(Console.ReadLine())

Next

Next

Imptabla(nombre, contvotos)

resultadoporcandidato(nombre, contvotos, totales)

totalvotosdistrito(contvotos)

ganador(nombre, totales)

End Sub

Public Sub obtenernombre(ByRef nombre)

'Este procedimiento introduce en una lista el nombre de los candidatos

Dim i As Integer

Console.WriteLine("Introducir los nombres de los candidatos, un nombre
por linea _

en el orden que aparece en la votación")

```
For i = 0 To NUMCANDIDATOS
    nombre(i) = Console.ReadLine()
Next
```

```
End Sub
```

```
Public Sub votoscero(ByRef contvotos)
    'Este procedimiento inicializa la matriz de votos
```

```
    Dim i, j As Integer
```

```
    For i = 0 To NUMDISTRITOS
        For j = 0 To NUMCANDIDATOS
            contvotos(i, j) = 0
        Next
    Next
```

```
End Sub
```

```
Public Sub Imptabla(ByVal nombre, ByVal contvotos)
    ' Este procedimiento imprime la tabla de candidatos y los votos
    obtenidos por cada uno
```

```
    Dim i, j As Integer
    Dim TH As Char = Convert.ToChar(&H9)
```

```
    Console.WriteLine("INFORME")
    Console.WriteLine("Distrito")
    For i = 0 To NUMCANDIDATOS
        Console.WriteLine(nombre(i) & TH)
    Next
    Console.WriteLine()
```

```
Console.WriteLine("_____")
_____")
```

```
For i = 0 To NUMDISTRITOS
    Console.WriteLine(i & TH)
    For j = 0 To NUMCANDIDATOS
        Console.WriteLine(contvotos(i, j) & TH)
```

```
    Next
    Console.WriteLine()
Next
```

End Sub

Public Sub resultadoporcandidato(ByVal nombre, ByVal contvotos, ByRef
totales)

'Numero total de votos por candidato

Dim i, j, total As Integer

For i = 0 To NUMCANDIDATOS

total = 0

For j = 0 To NUMDISTRITOS

total = total + contvotos(j, i)

Next

Console.WriteLine("El numero total de votos para {0} es : {1}",
nombre(i), total)

totales(i) = total

Next

End Sub

Public Sub ganador(ByVal nombre, ByVal totales)

Dim i, j, valor, cand As Integer

valor = totales(0)

cand = 0

For i = 1 To NUMCANDIDATOS

If totales(i) > valor Then

valor = totales(i)

cand = i

End If

Next

Console.WriteLine("El candidato ganador es: {0} con {1} votos",
nombre(cand), valor)

End Sub

Public Sub totalvotosdistrito(ByVal contvotos)

'Numero total de votos en cada distrito

Dim i, j, total As Integer

For i = 0 To NUMDISTRITOS

total = 0

'sumamos filas

For j = 0 To NUMCANDIDATOS

total += contvotos(i, j)

Next


```
Console.WriteLine("Total votos por distrito {0} es: {1}", i, total)
```

```
Next
```

```
End Sub
```

```
End Module
```

Ejercicio 17

Un número primo es cualquier entero divisible por sí mismos y por la unidad. La criba de Eratóstenes es un método para encontrar números primos cuyo funcionamiento es el siguiente:

Si tenemos un array con los 10 primeros números, comenzamos por el dos y recorremos hasta el final el array dividiendo todos los números por el dos, los que son divisibles por el dos, los “eliminamos” del array, continuamos por el siguiente número de este array, que en este caso sería el tres y volvemos a recorrer todo el array hasta el final dividiendo todos los números y aquellos que son divisibles por el tres los eliminamos del array, y así continuamos con los números que queden en el array hasta el final.

1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Después de la primera pasada tendríamos

1, 2, 3, 5, 7, 9

Después de la segunda pasada tendríamos

1, 2, 3, 5, 7

Cuando el proceso termina solo tenemos en el array los números primos. Escribir un programa que utilice un array de 1000 elementos para imprimir aquellos que sean primos.

Module Module1

'Programa que realiza la criba de Erastotenes

Sub Main()

Const max As Integer = 1000

Dim Criba(max) As Boolean, i, j As Integer

'Inicializamos la matriz

For i = 1 To Criba.Length - 1

Criba(i) = True

Next

For i = 2 To Criba.Length - 1

For j = i + 1 To Criba.Length - 1

If (Criba(j) = True) AndAlso (j Mod i = 0) Then

Criba(j) = False

```
End If
Next
Next

For i = 1 To Criba.Length - 1
    If Criba(i) = True Then
        Console.Write(" {0}", i)
    End If
Next

End Sub
```

```
End Module
```

Ejercicio 18

Un palíndromo es una frase o incluso una palabra que se lee igual en sus dos direcciones (de izquierda a derecha o de derecha a izquierda) p. e. dabale arroz a la zorra el abad. Escriba un programa que compruebe si una cadena de caracteres es un palíndromo. Hacerlo de dos formas, una iterativa y otra recursiva

La forma iterativa

```
Module Module1
```

```
Sub Main()
    Dim cadena As String
    Dim es_palindromo As Boolean = False
    Dim cadchar1(), cadchar2() As Char
    Dim i As Integer

    Console.WriteLine("¿Que frase queremos comprobar?")
    cadena = Console.ReadLine()

    ReDim cadchar1(cadena.Length - 1)
    ReDim cadchar2(cadena.Length - 1)

    cadena.CopyTo(0, cadchar1, 0, cadena.Length)

    For i = cadena.Length - 1 To 0 Step -1
        cadchar2((cadena.Length - 1) - i) = cadchar1(i)
```

Next

```
For i = 0 To cadena.Length - 1
    If cadchar1(i) <> cadchar2(i) Then
        es_palindromo = False
    Exit For
    Else : es_palindromo = True
End If
Next
```

```
If es_palindromo Then
    Console.WriteLine("la frase introducida es un palindromo")
Else
    Console.WriteLine("la frase introducida no es un palindromo")
End If
End Sub
```

De forma recursiva

Module Module1

```
Sub Main()
    Dim cadena As String
    Dim es_palindromo As Boolean = False
    Dim cadchar1() As Char

    Console.WriteLine("¿Que frase queremos comprobar?")
    cadena = Console.ReadLine()

    ReDim cadchar1(cadena.Length - 1)
    cadena.CopyTo(0, cadchar1, 0, cadena.Length)
    es_palindromo = comprueba(cadchar1, cadena.Length - 1)
    If es_palindromo Then
        Console.WriteLine("la frase introducida es un palindromo")
    Else
        Console.WriteLine("la frase introducida no es un palindromo")
    End If
End Sub
```

```
Public Function comprueba(ByVal cad1() As Char, ByVal tam As Integer)
    Dim compruebaux As Boolean = False
    Dim i As Integer
```

Dim cad2() As Char

If tam <= 1 Then

compruebaux = True

Elseif (cad1(0) <> cad1(tam)) Then

compruebaux = False

Else

ReDim cad2(tam - 2)

For i = 1 To cad1.Length - 2

cad2(i - 1) = cad1(i)

Next

compruebaux = comprueba(cad2, tam - 2)

End If

Return compruebaux

End Function

End Module