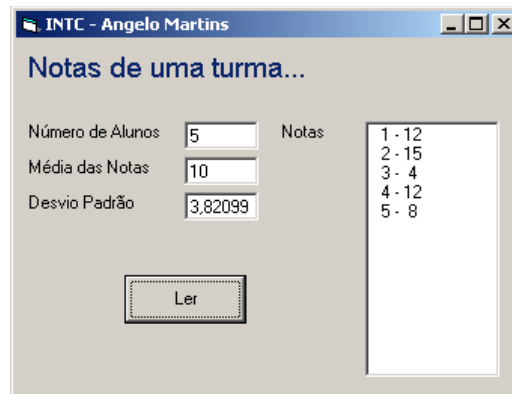


Ficha nº 4 – Vectores

1. Elabore um programa que leia um conjunto de notas (entre 0 e 20) e apresente a média e o desvio padrão deste conjunto. O número de notas a ler é dado pelo utilizador e limitado a 100 unidades.

O desvio padrão é dado pela fórmula
$$S_N = \sqrt{\frac{\sum (x_i - \bar{x})^2}{N}}$$



Option Explicit

```
Const NMax = 100
```

Utilizou-se uma constante para definir o tamanho do vector. O valor de uma constante não pode ser alterado ao longo do programa e é apenas uma forma conveniente de referir o valor 100.

```
Private Sub cmdLer_Click()
```

```
    Dim vec(NMax) As Integer, n As Integer  
    Dim t As Integer, soma As Integer, med As Integer  
    Dim aux As Single, dp As Single
```

```
    lstNotas.Clear          ' limpa os valores anteriores  
    txtDP.Text = ""  
    txtMed.Text = ""
```

```
    n = Val(txtN.Text)  
    If n < 1 Or n > NMax Then  
        MsgBox "Erro: Número de alunos inválido!"  
        Exit Sub  
    End If
```

Validação da entrada das notas, obrigando a que o valor esteja entre 0 e 20.

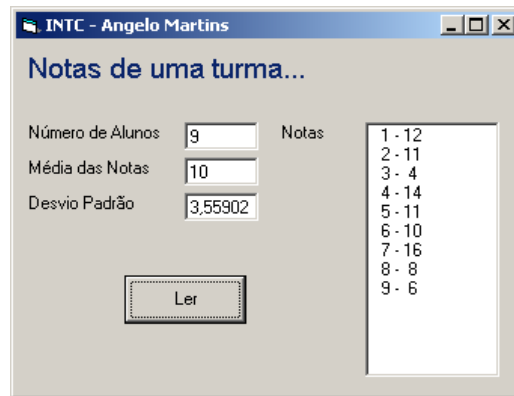
```
    For t = 1 To n
```

```
        Do  
            vec(t) = Val(InputBox("Nota do aluno nº " & t))  
        Loop Until vec(t) >= 0 And vec(t) <= 20
```

```
            soma = soma + vec(t)  
            lstNotas.AddItem Format(t, "@@@") & " - " & Format(vec(t), "@@")  
        Next  
        med = soma / n  
        txtMed.Text = med
```

```
For t = 1 To n
    aux = aux + (vec(t) - med) ^ 2
Next
dp = Sqr(aux / n)
txtDP.Text = dp
End Sub
```

2. Altere o exercício anterior de forma a que a leitura das notas termine quando for introduzido o valor -1. O número de alunos não é fornecido pelo utilizador, mas está ainda limitado às 100 unidades.



```
Option Explicit
Const NMax = 100
```

```
Private Sub cmdLer_Click()
    Dim vec(NMax) As Integer, n As Integer, nota As Integer
    Dim t As Integer, soma As Integer, med As Integer
    Dim aux As Single, dp As Single

    lstNotas.Clear      ' limpa os valores anteriores
    txtDP.Text = ""
    txtMed.Text = ""

    n = 0
    Do
        Do
            nota = Val(InputBox("Nota do aluno nº " & (n + 1)))
            Loop Until nota >= 0 And nota <= 20 Or nota = -1
            If nota <> -1 Then
                n = n + 1
                vec(n) = nota
                soma = soma + nota
                lstNotas.AddItem Format(n, "@@@") & " - " & Format(vec(n), "@@")
            End If
        Loop While n < NMax And nota <> -1

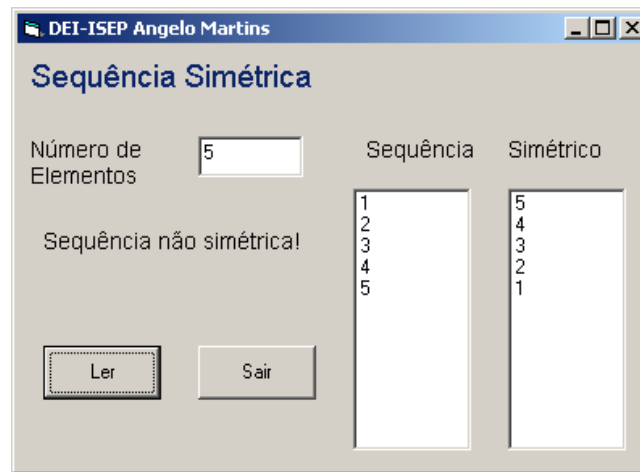
        If n > 0 Then
            med = soma / n
            txtMed.Text = med
            txtN.Text = n

            For t = 1 To n
                aux = aux + (vec(t) - med) ^ 2
            Next
            dp = Sqr(aux / n)
            txtDP.Text = dp
        Else
            MsgBox "Não foram introduzidos valores!"
        End If
    End Sub
```

-1 também é um valor válido!

A leitura termina quando for atingida a capacidade do vector ou quando for introduzido o valor -1

3. Elabore uma função que verifique se um vector é simétrico. Elabore depois um programa que permita ler uma sequência de números inteiros (número de elementos dado pelo utilizador, limitado a 100 unidades) e indique se a sequência lida é simétrica. O programa deve ainda apresentar a sequência original e o seu simétrico em duas *ListBoxes*.



Option Explicit

```
Private Sub cmdLer_Click()
    Dim seq(100) As Integer, smt(100) As Integer
    Dim x As Integer, ne As Integer

    lstSeq.Clear
    lstSim.Clear
    lblRes.Visible = False

    ne = Val(txtNE.Text)
    If ne < 1 Or ne > 100 Then
        MsgBox "Erro: N° de elementos inválido!", vbCritical + vbOKOnly
        Exit Sub
    End If

    LerSeq ne, seq
    SimVec ne, seq, smt

    For x = 1 To ne
        lstSeq.AddItem seq(x)
        lstSim.AddItem smt(x)
    Next
    If simetrico(ne, seq) Then
        lblRes.Caption = "Sequência simétrica!"
    Else
        lblRes.Caption = "Sequência não simétrica!"
    End If
    lblRes.Visible = True
End Sub

Sub LerSeq(ByVal n As Integer, v() As Integer)
    Dim x As Integer
    For x = 1 To n
        v(x) = Val(InputBox("Insira elemento n° " & x))
    Next
End Sub
```

Definem-se dois vectores de 100 elementos cada.

Leitura dos elementos da sequência para o vector

Subrotina que gera um vector (smt) simétrico do primeiro (seq).

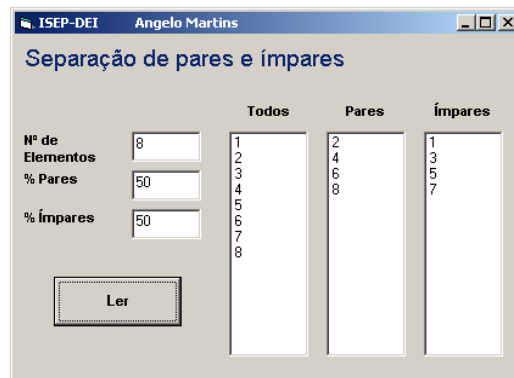
Função booleana que verifica se o vector é simétrico.

```
Sub SimVec(ByVal n As Integer, v() As Integer, vs() As Integer)
  Dim y As Integer, z As Integer
  For y = 1 To n
    vs(n - y + 1) = v(y)
  Next
End Sub

Function simetrico(ByVal n As Integer, v() As Integer) As Boolean
  Dim y As Integer, z As Integer, sim As Boolean
  sim = True
  y = 1
  z = n
  Do While y < z And sim
    If v(y) <> v(z) Then
      sim = False
    End If
    y = y + 1
    z = z - 1
  Loop
  simetrico = sim
End Function
```

Quando sim toma o valor false o ciclo termina..

4. Elabore uma subrotina que receba três vetores: um vector contendo números inteiros positivos e nulos; e dois vectores "vazios". A rotina deve separar os números pares e ímpares do vector original, colocando-os nos outros dois vectores. Elabore um programa para testar a subrotina.



```
Option Explicit
Const NMax = 50
```

```
Private Sub cmdLer_Click()
    Dim vect(NMax) As Integer, vecp(NMax) As Integer, veci(NMax) As Integer
    Dim ne As Integer, nep As Integer, nei As Integer
    Dim y As Integer

    txtPP.Text = ""
    txtPI.Text = ""
    lstTodos.Clear
    lstP.Clear
    lstI.Clear

    ne = Val(txtNE.Text)
    If ne <= 0 Or ne > NMax Then
        MsgBox "Atenção: nº de elementos inválido!"
        Exit Sub
    End If

    Ler ne, vect

    Separa ne, vect, nep, vecp, nei, veci

    For y = 1 To ne
        lstTodos.AddItem vect(y)
    Next

    For y = 1 To nep
        lstP.AddItem vecp(y)
    Next

    For y = 1 To nei
        lstI.AddItem veci(y)
    Next

    txtPP.Text = nep / ne * 100
    txtPI.Text = nei / ne * 100
End Sub
```

```

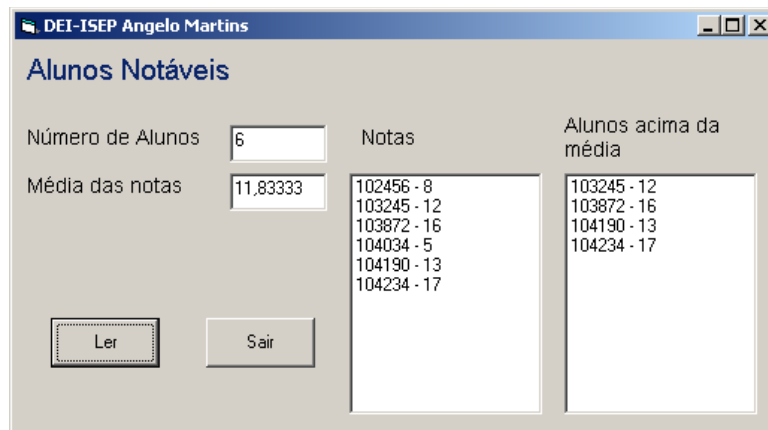
Sub Separa(ByVal n As Long, vt() As Integer, np As Integer, vp() As Integer, _
    ni As Integer, vi() As Integer)
    Dim s As Integer
    np = 0
    ni = 0
    For s = 1 To n
        If vt(s) Mod 2 = 0 Then
            np = np + 1
            vp(np) = vt(s)
        Else
            ni = ni + 1
            vi(ni) = vt(s)
        End If
    Next
End Sub

Sub Ler(ByVal n As Integer, va() As Integer)
    Dim x As Integer

    For x = 1 To n
        Do
            va(x) = Val(InputBox("Elemento n° " & x))
        Loop While va(x) < 0
    Next
End Sub

```

5. Elabore uma subrotina que leia as notas de uma turma (número de alunos dado) para um vector e uma função que calcule a média dessas mesmas notas. Utilizando estas rotinas, elabore um programa que apresente numa ListBox as notas da turma que estejam acima da média.



```
Option Explicit
```

```
Option Explicit
```

```
Private Sub cmdLer_Click()
```

```
    Dim notas(100) As Integer, alunos(100) As Long
    Dim x As Integer, med As Single, na As Integer
```

```
    lstNotas.Clear
    lstQH.Clear
    txtMed.Text = ""
```

```
    na = Val(txtNA.Text)
    If na < 1 Or na > 100 Then
        MsgBox "Erro: N° de alunos inválido!", vbCritical + vbOKOnly
        Exit Sub
    End If
```

```
    LerNotas na, notas, alunos
```

Chamada da subrotina de leitura de notas.
Note-se a ausência de parêntesis nos
vectores notas e alunos.

```
    med = MediaVec(na, notas)
```

```
    lstNotas.Clear
    For x = 1 To na
        lstNotas.AddItem alunos(x) & " - " & notas(x)
        If notas(x) > med Then
            lstQH.AddItem alunos(x) & " - " & notas(x)
        End If
    Next
```

```
    txtMed.Text = med
```

```
End Sub
```

A subrotina tem três argumentos, o primeiro passado por valor e os restantes dois por referência. Só as alterações nestes dois últimos parâmetros são visíveis no exterior da subrotina. Os vectores são sempre passados por referência.

```
Sub LerNotas(ByVal n As Integer, vn() As Integer, va() As Long)
```

```
    Dim x As Integer
    For x = 1 To n
        va(x) = Val(InputBox("Insira Número do " & x & "° aluno"))
        Do
            vn(x) = Val(InputBox("Nota do aluno " & va(x)))
        Loop While vn(x) < 0 Or vn(x) > 20
    Next
```

```
End Sub
```



```
Function MediaVec(ByVal n As Integer, vec() As Integer) As Single
    Dim y As Integer, soma As Long
    For y = 1 To n
        soma = soma + vec(y)
    Next
    MediaVec = soma / n
End Function
```

```
Private Sub cmdSair_Click()
    End
End Sub
```