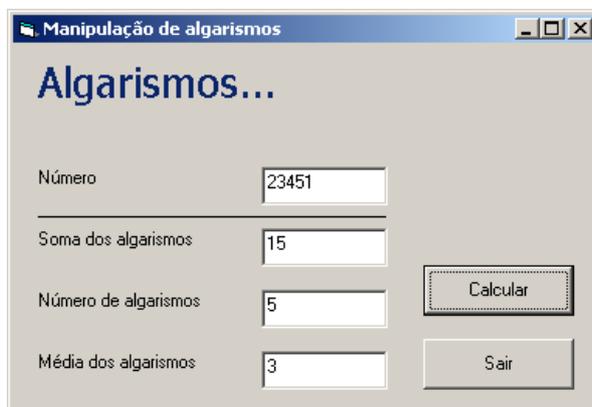


## Ficha nº 2 - Ciclos

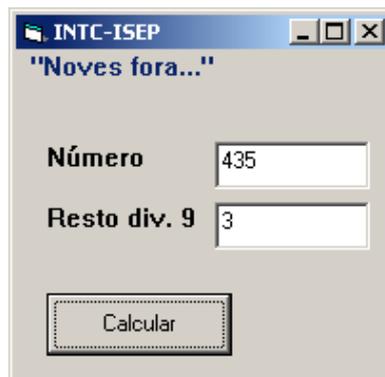
1. Elabore um programa que leia um número e determine a soma e a média dos algarismos que o compõem.



Option Explicit

```
Private Sub cmdCalc_Click()  
    Dim n As Integer, s As Integer, nalg As Integer, alg As Integer  
  
    n = Val(Abs(txtNum.Text))  
    s = 0  
    nalg = 0  
    Do While n <> 0  
        alg = n Mod 10  
        s = s + alg  
        nalg = nalg + 1  
        n = n \ 10  
    Loop  
  
    txtSoma.Text = s  
    If nalg > 0 Then  
        txtNA.Text = nalg  
        txtMed.Text = s / nalg  
    Else  
        txtNA.Text = 1  
        txtMed.Text = 0  
    End If  
End Sub  
  
Private Sub cmdSair_Click()  
    End  
End Sub
```

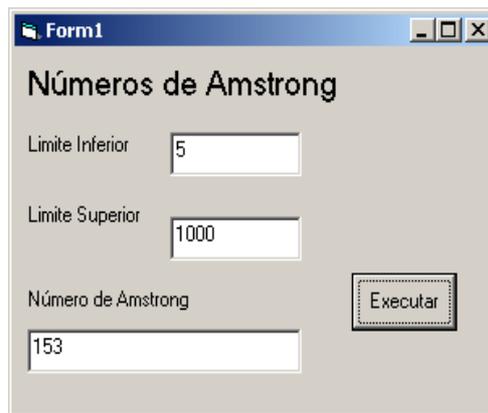
2. Elabore um programa em Visual Basic que calcule o resto da divisão por nove da soma dos algarismos de um número inteiro longo.



```
Option Explicit
```

```
Private Sub cmdCalc_Click()  
    Dim n As Long, alg As Integer, s As Integer  
    n = Abs(Val(txtNum.Text))  
    Do Until n = 0  
        alg = n Mod 10  
        s = s + alg  
        n = n \ 10  
    Loop  
    txtResto.Text = s Mod 9  
End Sub
```

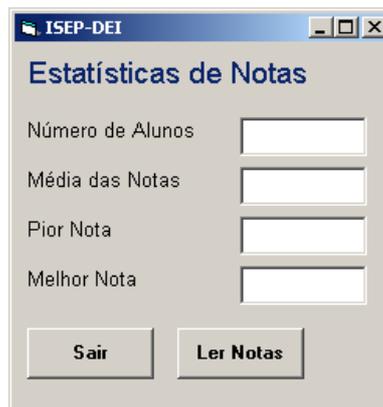
3. Um número diz-se Número de Armstrong se for igual à soma dos cubos dos seus algarismos. Elabore um programa em Visual Basic que apresente o primeiro número de Armstrong que se exista num intervalo dado.



Option Explicit

```
Private Sub Executa_Click()  
Dim intNum As Integer, intSoma As Integer  
Dim intInf As Integer, intSup As Integer, intX As Integer  
Dim fFim As Boolean  
  
fFim = False  
intX = Val(txtInf.Text)  
Do While intX < Val(txtSup.Text) And Not fFim  
    intNum = intX  
    intSoma = 0  
    Do While intNum <> 0  
        intSoma = (intNum Mod 10) ^ 3 + intSoma  
        intNum = intNum \ 10  
    Loop  
    If intSoma = intX Then  
        txtNum.Text = intX  
        fFim = True  
    End If  
    intX = intX + 1  
Loop  
If Not fFim Then  
    txtNum.Text = " não há números de Armstrong"  
End If  
End Sub
```

4. Elabore um programa que leia uma sequência de notas e apresente a média, a maior e a menor das notas introduzidas. O número de notas da sequência é dado pelo utilizador.

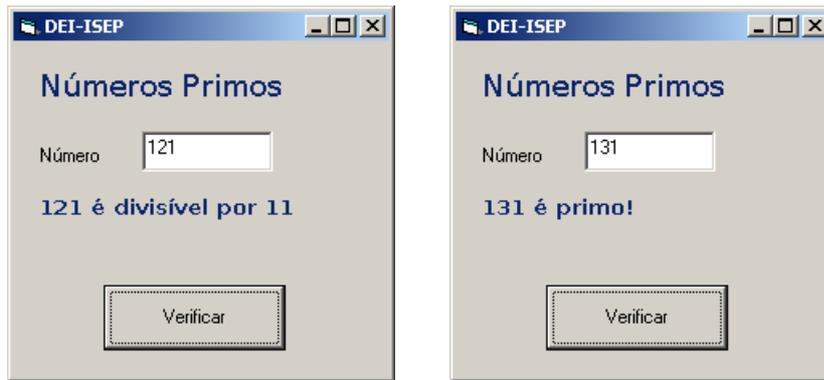


```
Option Explicit
Private Sub cmdLer_Click()
    Dim na As Integer, nota As Integer, soma As Integer
    Dim max As Integer, min As Integer, x As Integer

    na = Val(txtNA.Text)
    If na <= 0 Then
        MsgBox "Erro: Número de alunos inválido!"
        Exit Sub
    End If
    max = -1          ' valores iniciais
    min = 21
    soma = 0
    For x = 1 To na
        nota = Val(InputBox("Nota nº " & x))
        soma = soma + nota
        If nota > max Then      ' compara com melhor anterior
            max = nota
        End If
        If nota < min Then     ' compara com pior anterior
            min = nota
        End If
    Next
    txtMed.Text = soma / na
    txtMelhor.Text = max
    txtPior.Text = min
End Sub

Private Sub cmdSair_Click()
    End
End Sub
```

5. Elabore um programa que verifique se um dado número é primo.



Option Explicit

```
Private Sub cmdVer_Click()  
    Dim num As Integer, k As Integer  
  
    lblRes.Visible = False  
  
    num = Val(txtNum.Text)  
  
    If num > 0 Then  
        k = 2  
        Do While k <= num \ 2 And num Mod k <> 0  
            k = k + 1  
        Loop  
        If k > num / 2 Then  
            lblRes.Caption = num & " é primo!"  
        Else  
            lblRes.Caption = num & " é divisível por " & k  
        End If  
        lblRes.Visible = True  
    Else  
        MsgBox "Só se aceitam números superiores a 0!"  
    End If  
End Sub  
  
Private Sub txtNum_Change()  
    lblRes.Visible = False  
End Sub
```

6. Elabore um programa que verifique se um número é capicua.



Option Explicit

```
Private Sub cmdVer_Click()  
    Dim num As Long, aux As Long, cap As Long, alg As Integer  
  
    num = Abs(Val(txtN.Text))  
    aux = num  
    Do While num > 0  
        alg = num Mod 10  
        num = num \ 10  
        cap = cap * 10 + alg  
    Loop  
    If cap = aux Then  
        MsgBox aux & " é capicua!", , "Teste de Capicua"  
    Else  
        MsgBox aux & " é um número comum!", , "Teste de Capicua"  
    End If  
End Sub
```

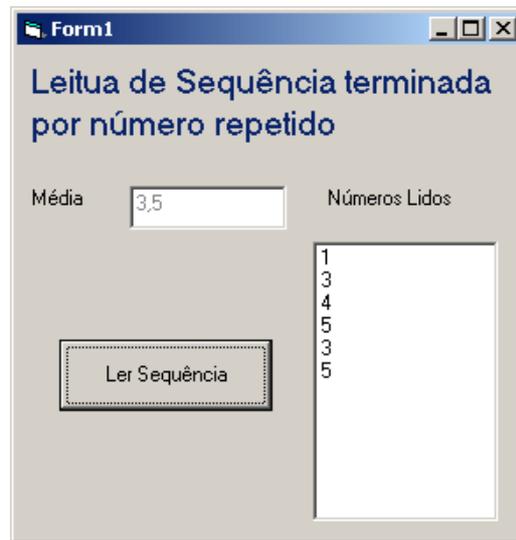
7. Elabore um programa que leia uma sequência de números terminada por zero. O programa deverá dar como saída a média dos números introduzidos, sem considerar o zero. Todos os números introduzidos devem ser visualizados, excepto o zero.



```
Option Explicit
```

```
Private Sub cmdLer_Click()  
    Dim num As Integer, c As Integer, s As Long  
  
    lstNum.Clear  
  
    num = Val(InputBox("Elemento n° 1"))  
    Do While num <> 0  
        lstNum.AddItem num  
        s = s + num  
        c = c + 1  
        num = Val(InputBox("Elemento n° " & (c + 1)))  
    Loop  
    If c <> 0 Then  
        txtMed.Text = s / c  
    Else  
        txtMed.Text = ""  
        MsgBox "Não há elementos a considerar!"  
    End If  
End Sub
```

8. Elabore um programa que leia uma sequência de números, cuja leitura deve terminar quando forem introduzidos dois números consecutivos repetidos. O programa deverá dar como saída a média dos números introduzidos, sem considerar o último. Todos os números introduzidos devem ser visualizados, com a exceção do último.



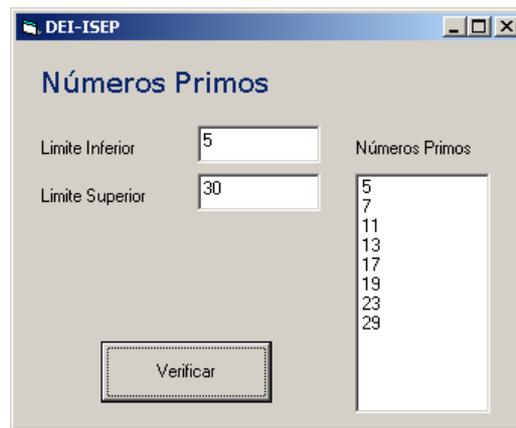
Option Explicit

```
Private Sub cmdLer_Click()  
    Dim num As Integer, c As Integer, s As Long, ant As Integer  
  
    lstNum.Clear  
  
    num = Val(TextBox("Elemento nº 1"))  
    Do  
        lstNum.AddItem num  
        s = s + num  
        c = c + 1  
        ant = num  
        num = Val(TextBox("Elemento nº " & (c + 1)))  
    Loop Until num = ant  
    txtMed.Text = s / c  
  
End Sub
```

guarda o valor anterior que vai ser usado para verificar se o número lido é repetido

ant = num

9. Elabore um programa que apresente numa ListBox todos os números primos que existam num intervalo dado.

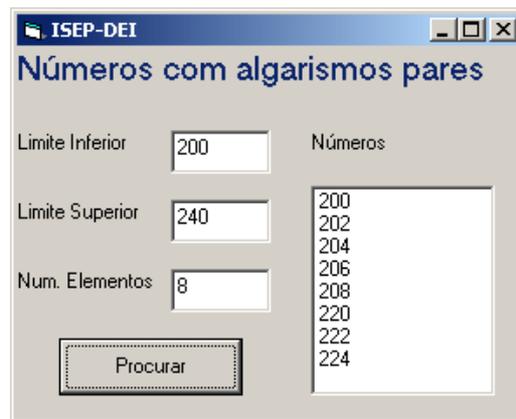


Option Explicit

```
Private Sub cmdVer_Click()  
    Dim inf As Long, sup As Long, k As Long  
    Dim aux As Long, num As Long  
    lstNP.Clear  
  
    inf = Val(txtLI.Text)  
    sup = Val(txtLS.Text)  
    If sup < inf Then  
        aux = inf  
        inf = sup  
        sup = aux  
    End If  
  
    If inf <= 0 Then  
        MsgBox "Os limites estão errados!"  
        Exit Sub  
    End If  
  
    For num = inf To sup  
        k = 2  
        Do While k <= num \ 2 And num Mod k <> 0  
            k = k + 1  
        Loop  
        If k > num / 2 Then  
            lstNP.AddItem num  
        End If  
    Next  
End Sub
```

Não há divisores maiores do que a metade do número e o número só é primo se não for divisível por outro número

10. Elabore um programa que, dado um intervalo e o número máximo de elementos a visualizar, apresente numa ListBox os números do intervalo compostos exclusivamente por algarismos pares.



Option Explicit

```
Private Sub cmdProc_Click()
    Dim inf As Long, sup As Long, ne As Integer, n As Integer
    Dim num As Long, aux As Long, par As Boolean, alg As Integer

    inf = Val(txtInf.Text)
    sup = Val(txtSup.Text)
    If inf > sup Then          ' validar intervalo
        aux = sup
        sup = inf
        inf = aux
    End If

    ne = Val(txtNE.Text)
    If ne <= 0 Then           ' validar nº de elementos
        MsgBox "Nº de elementos errado!", vbCritical
        Exit Sub
    End If

    lstNum.Clear              ' limpar listbox

    num = inf
    Do While num <= sup And n < ne
        aux = num
        par = True
        Do                    ' obter algarismos
            alg = aux Mod 10
            aux = aux \ 10
            If alg Mod 2 <> 0 Then
                par = False
            End If
        Loop While aux <> 0 And par
        If par Then
            lstNum.AddItem num
            n = n + 1
        End If
        num = num + 1         ' próximo número
    Loop
End Sub
```