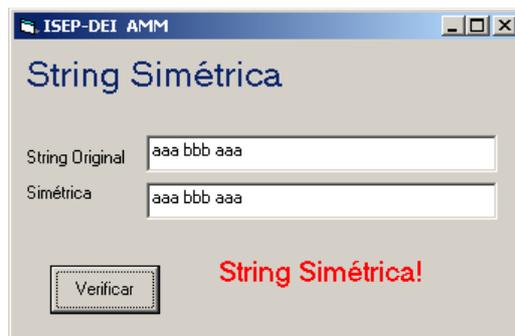


## Ficha nº 5 – Strings

1. Elabore uma função que recebe uma string como parâmetro e retorna uma string simétrica da primeira. Exemplo: "abc cd " -> " dc cba". Utilizando a função, elabore um programa que leia uma string e verifique se esta é simétrica.



Option Explicit

```
Private Sub cmdVer_Click()  
    Dim sto As String, sti As String  
  
    lblRes.Visible = False  
  
    sto = txtSt.Text  
    sti = invString(sto)  
    txtSim.Text = sti  
    If sti = sto Then  
        lblRes.Caption = "String Simétrica!"  
    Else  
        lblRes.Caption = "String Assimétrica!"  
    End If  
    lblRes.Visible = True  
End Sub
```

```
Function invString(ByVal st As String) As String  
    Dim c As String, iv As String  
    Do While st <> ""  
        c = Left(st, 1)  
        st = Right(st, Len(st) - 1)  
        iv = c + iv  
    Loop  
    invString = iv  
End Function
```

Copia para **c** o primeiro caracter da string

`c = Left(st, 1)`

Copia para **st** os restantes caracteres da string

`st = Right(st, Len(st) - 1)`

`iv = c + iv`

Acrescenta **c** a **iv** de forma a a string ficar invertida (o último caracter é adicionado no inicio de **iv**)

2. Elabore uma função que recebe como parâmetros os limites de um intervalo (números inteiros) e retorna uma string com todos os números inteiros desse intervalo (incluindo os extremos). Os números devem estar separados por um espaço.

```
Function GeraNumString(ByVal li As Integer, ByVal ls As Integer) As String
    Dim n As Integer, st As String

    For n = li to ls
        st = st + " " + Str(n)
    Next
    GeraNumString = Ltrim(st)
End Function
```

A função Ltrim() retira o espaço a mais introduzido na primeira iteração do ciclo.

3. Elabore uma função que recebe como parâmetro um vector de palavras (strings) e retorna uma string com todas as palavras do vector, ordenadas por ordem alfabética e separadas por um espaço.

```
Function VecToStr(v() As String, ByVal n As Integer) As String
    Dim p As Integer, q As Integer, st As String, aux As String

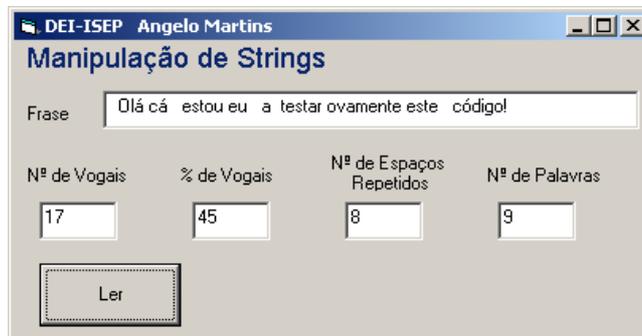
    For p = 1 to n
        For q = p + 1 to n
            If v(p) > v(q) Then
                aux = v(p)
                v(p) = v(q)
                v(q) = aux
            End If
        Next
    Next
    For p = 1 to n
        st = st + " " + v(p)
    Next
    VecToStr = LTrim(st)
End Function
```

Ordenação do vector de strings por ordem alfabética (crescente).

4. Elabore um programa que apresente os seguintes dados estatísticos relativos a uma frase:

- número de vogais;
- percentagem de vogais (os espaços são ignorados);
- número de espaços repetidos na frase;
- número de palavras.

Utiliza funções para obter os dados pedidos.



Option Explicit

```
Private Sub cmdLer_Click()
    Dim frase As String
```

```
    frase = txtFrase.Text
    txtNVogais.Text = NumVogais(frase)
    txtPVogais.Text = PercentVogais(frase)
    txtNPal.Text = NumPal(frase)
    txtEspR = NumEspRep(frase)
```

```
End Sub
```

Calcula o número de vogais na frase.

```
Function NumVogais(ByVal st As String) As Integer
```

```
    Dim c As String
```

```
    st = UCase(Trim(st))
```

A conversão para maiúsculas permite simplificar a comparação abaixo.

```
    Do While st <> ""
```

```
        c = Left(st, 1)
```

```
        If c = "A" Or c = "E" Or c = "I" Or c = "O" Or c = "U" Then
            NumVogais = NumVogais + 1
```

```
        End If
```

```
        st = Right(st, Len(st) - 1)
```

```
    Loop
```

```
End Function
```

Calcula a percentagem de vogais na frase.

```
Function PercentVogais(ByVal st As String) As Integer
```

```
    Dim c As String, n As Integer, nt As Integer
```

```
    st = UCase(Trim(st))
```

```
    Do While st <> ""
```

```
        c = Left(st, 1)
```

```
        If c <> " " Then
```

```
            If c = "A" Or c = "E" Or c = "I" Or c = "O" Or c = "U" Then
```

```
                n = n + 1
```

```
            End If
```

```
            nt = nt + 1
```

```
        End If
```

```
        st = Right(st, Len(st) - 1)
```

```
    Loop
```

```
    If nt <> 0 Then
```

```
        PercentVogais = n / nt * 100
```

```
    Else
```

Verifica se é uma vogal. A conversão prévia de st para maiúsculas permite efectuar só 5 comparações.

```
        PercentVogais = 0
    End If
End Function
```

```
Function NumPal(ByVal st As String) As Integer
    Dim pos As Integer, n As Integer
```

Calcula o número de palavras na frase.

```
    st = Trim(st)
    Do While st <> ""
        pos = InStr(1, st, " ")
        n = n + 1
        If pos > 0 Then
            st = LTrim(Right(st, Len(st) - pos))
        Else
            st = ""
        End If
    Loop
    NumPal = n
End Function
```

```
Function NumEspRep(ByVal st As String) As Integer
    Dim n As Integer, pant As Integer, p As Integer
    Dim c As String
```

Calcula o número de espaços repetidos na frase.

```
    n = 0
    p = 1
    pant = 1
    Do
        p = InStr(p, st, " ")
        If p > 0 Then
            If pant = p - 1 Then
                n = n + 1
            End If
            pant = p
            p = p + 1
        End If
    Loop While p > 0
    NumEspRep = n
End Function
```

5. Elabore uma função que recebe uma frase (string) e um número inteiro *c* como parâmetros e retorne uma string composta pelas palavras da frase original, separadas por um espaço, cujo comprimento seja igual ou superior a *c*.

Elabore um programa que leia uma frase e um número inteiro e que, utilizando a função, apresente numa textbox as palavras de comprimento igual ou superior ao número introduzido.

Exemplo: "Ab desef cBa dk cbAjdkl" e "3" --> "desef cBa cbAjdkl"



Option Explicit

```
Private Sub cmdProc_Click()
    Dim frase As String, n As Integer

    frase = txtF
    n = Val(txtN.Text)
    If n <= 0 Then
        MsgBox "Número inválido! Tente novamente."
        Exit Sub
    End If
    txtPal.Text = truncaPal(frase, n)
End Sub
```

```
Function truncaPal(ByVal st As String, ByVal n As Integer) As String
    Dim p As Integer, pal As String, stout As String
    st = Trim(st)
    Do While st <> ""
        p = InStr(st, " ")
        If p > 0 Then
            pal = Left(st, p - 1)
            st = LTrim(Right(st, Len(st) - p))
        Else
            pal = st
            st = ""
        End If
        If Len(pal) >= n Then
            stout = stout + " " + pal
        End If
    Loop
    truncaPal = Trim(stout)
End Function
```

p vai conter a posição do primeiro espaço de st

Copia p-1 caracteres que formam a palavra para pal

st não tem espaços, logo tem uma só palavra

Verifica se o comprimento da palavra é superior a n

6. Elabore uma função que recebe uma string como parâmetro e retorna uma string em que as letras iniciais da cada palavra, da primeira string, são convertidas para maiúsculas e as restantes letras para minúsculas.

```
Function FormataPal(ByVal st As String) As String
    Dim pal As String, p As Integer, aux As String, res As String

    st = Trim(st)
    If Len(st) > 0 Then
        Do
            p = InStr(1, st, " ")
            If p > 0 Then
                pal = Left(st, p - 1)
                st = Trim(Right(st, Len(st) - p))
            Else
                pal = st
            End If
            aux = UCase(Left(pal, 1)) + LCase(Right(pal, Len(pal) - 1))
            res = res + " " + aux
        Loop While p > 0
        FormataPal = LTrim(res)
    End If
End Function
```

Converte a primeira letra de **pal** para maiúsculas e as restantes para minúsculas ao adicionar à string **aux**.

7. Elabore uma subrotina que recebe como parâmetro uma string contendo uma frase e retorna a mesma frase mas com as palavras invertidas. Eventuais espaços repetidos podem ser eliminados.

```
Public Sub InvPal(st As String)
    Dim palavra As String, p As Integer, palInv As String
    Dim aux As String

    st = Trim(st)
    If Len(st) > 0 Then
        Do
            p = InStr(1, st, " ")
            If p > 0 Then
                palavra = Left(st, p - 1)
                st = Trim(Right(st, Len(st) - p))
            Else
                palavra = st
            End If
            aux = ""
            Do While Len(palavra) > 0
                aux = Left(palavra, 1) + aux
                palavra = Right(palavra, Len(palavra) - 1)
            Loop
            palInv = palInv + " " + aux
        Loop While p > 0
        st = LTrim(palInv)
    End If
End Sub
```

Para inverter a palavra adicionam-se os caracteres ao início da string com a nova palavra.

Para manter a ordem das palavras adicionam-se as palavras ao fim da string com a nova frase.

8. Elabore uma função que receba duas strings como parâmetros, uma contendo uma frase e a outra um caracter, e retorna a palavra da frase com maior número de ocorrências do caracter. Não é feita qualquer distinção entre letras maiúsculas e minúsculas na pesquisa.

```
Function MaisLetra(ByVal st As String, ByVal c As String) As String
    Dim palavra As String, max As Integer, maxP As String, p As Integer
    Dim aux As String, n As Integer, x As Integer

    c = UCase(Trim(c))
    st = UCase(Trim(st))
    If Len(st) > 0 And Len(c) = 1 Then
        Do
            p = InStr(1, st, " ")
            If p > 0 Then
                palavra = Left(st, p - 1)
                st = Trim(Right(st, Len(st) - p))
            Else
                palavra = st
            End If

            aux = palavra
            n = 0
            Do
                x = InStr(1, aux, c)
                If x > 0 Then
                    n = n + 1
                    aux = Right(aux, Len(aux) - x)
                End If
            Loop While x > 0

            If n > max Then
                max = n
                maxP = palavra
            End If

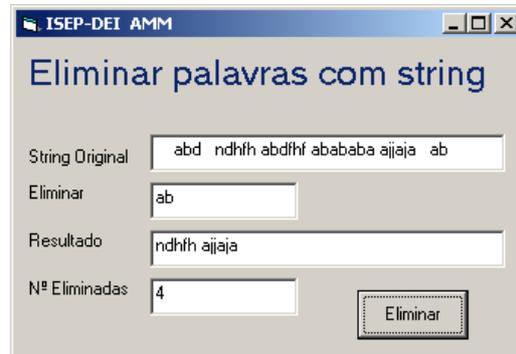
            Loop While p > 0
            MaisLetra = maxP
        End If
    End Function
```

Converte tudo para maiúsculas para facilitar as comparações

Separa a palavra em caracteres e conta as ocorrências do caracter guardado em c

Determina se o número de ocorrências da palavra actual é superior ao máximo corrente. Se o for, passa a ser um novo máximo.

9. Elabore uma subrotina que receba uma frase (string) e uma sequência de caracteres como parâmetros e retorne uma frase semelhante à original em que todas as palavra que contenham a sequência de caracteres (não deve ser feita qualquer distinção entre maiúsculas e minúsculas) são eliminadas, bem como os espaços repetidos. A subrotina deve ainda retornar o número de palavras eliminadas. Elabore um programa que leia a frase e a sequência de caracteres e apresente em TextBoxes a frase corrigida e o número de palavras eliminadas. Apresente a interface do programa, indicando os nomes de todos os objectos.



Option Explicit

```
Sub CortaString (ByVal st As String, ByVal stb As String,
```

```
stres As String, npc As Integer)
```

```
Dim p As Integer, pal As String, aux As String
```

```
stres = ""  
npc = 0
```

Trata-se de uma subrotina e os parâmetros passados por referência trazem valores. Inicializam-se aqui.

```
st = Trim(st)  
stb = UCase(Trim(stb))
```

```
Do While st <> ""  
    p = InStr(st, " ")  
    If p > 0 Then  
        pal = Left(st, p - 1)  
        st = Right(st, Len(st) - p)  
        st = LTrim(st)
```

```
    Else  
        pal = st  
        st = ""
```

```
    End If  
    aux = UCase(pal)
```

```
    If InStr(aux, stb) > 0 Then  
        npc = npc + 1  
    Else  
        stres = stres + " " + pal  
    End If
```

Verifica se a string a procurar existe na palavra. Basta haver uma ocorrência para a palavra ser eliminada.

```
Loop  
stres = Trim(stres)
```

```
End Sub
```

```
Private Sub cmdElim_Click()
```

```
Dim sto As String, stc As String, pb As String, npe As Integer
```

```
sto = txtFrase.Text  
pb = txtElim.Text  
CortaString sto, pb, stc, npe  
txtRes.Text = stc  
txtNPE.Text = npe
```

```
End Sub
```