

BIBLIOTECA TEXTO – PARA TRABALHAR COM MANIPULAÇÃO DE STRINGS(CARACTERES)

Texto

Esta biblioteca contém funções para manipulação de texto (dados do tipo cadeia)

Índice das Funções:

```
funcao cadeia caixa_alta(cadeia cad)
```

```
funcao cadeia caixa_baixa(cadeia cad)
```

```
funcao cadeia preencher_a_esquerda(caracter car, inteiro tamanho,  
cadeia cad)
```

```
funcao caracter obter_caracter(cadeia cad, inteiro indice)
```

```
funcao cadeia substituir(cadeia cad, cadeia texto_pesquisa, cadeia  
texto_substituto)
```

```
funcao inteiro numero_caracteres(cadeia cadeia)
```

```
funcao cadeia extrair_subtexto(cadeia cad, inteiro posicao_inicial,  
inteiro posicao_final)
```

```
funcao inteiro posicao_texto(cadeia texto, cadeia cad, inteiro  
posicao_inicial)
```

DETALHAMENTO DAS FUNÇÕES

caixa_alta

```
funcao cadeia caixa_alta(cadeia cad)
```

Transforma os caracteres de uma cadeia em caracteres maiúsculos

Parâmetros:

cad - um valor qualquer do tipo cadeia

Retorna: a cadeia com os caracteres transformados

```

programa
{
    inclui biblioteca Texto --> txt

    funcao inicio()
    {
        cadeia nome, maiusculo
        nome = "joão da silva"
        escreva(nome)
        escreva("\n\n")
        maiusculo = txt.caixa_alta(nome)
        escreva(maiusculo)
        escreva("\n")
    }
}

```

```

joão da silva
JOÃO DA SILVA

```

caixa_baixa

```
funcao cadeia caixa_baixa(cadeia cad)
```

Transforma os caracteres de uma cadeia em caracteres minúsculos

Parâmetros:

cad - um valor qualquer do tipo cadeia

Retorna: a cadeia com os caracteres transformados

```

programa
{
    inclui biblioteca Texto --> txt

    funcao inicio()
    {
        cadeia nome, minusculo
        nome = "MARINA da SilVeira"
        escreva(nome)
        escreva("\n\n")
        minusculo = txt.caixa_baixa(nome)
        escreva(minusculo)
        escreva("\n")
    }
}

```

```

MARINA da SilVeira
marina da silveira

```

preencher_a_esquerda

```
funcao cadeia preencher_a_esquerda(caracter car, inteiro tamanho, cadeia cad)
```

Concatena o caracter informado, à esquerda da cadeia, até que a cadeia fique do tamanho indicado. Se o tamanho da cadeia for maior ou igual ao tamanho informado, nada é feito

Parâmetros:

car - o caracter que será concatenado á esquerda da cadeia

tamanho - o tamanho final da cadeia

cad - a cadeia que será transformada

Retorna: a cadeia transformada

```

programa
{
    inclui biblioteca Texto --> txt

    funcao inicio()
    {
        cadeia texto, novotexto
        texto = "Texto Qualquer" // Esta variável contém 14 caracteres
        escreva(texto)
        escreva("\n\n")
        //Atenção usar aspa simples pois vamos passar um caracter
        novotexto = txt.preencher_a_esquerda('*',16, texto)
        escreva(novotexto)
        escreva("\n")
    }
}

```

Texto Qualquer

**Texto Qualquer

obter_caracter

```
funcao caracter obter_caracter(cadeia cad, inteiro indice)
```

Obtém um caracter da cadeia a partir de seu índice. O índice deve estar entre 0 e o número de caracteres da cadeia

Parâmetros:

- cad - a cadeia da qual será obtido o caracter
- indice - o índice do caracter que se deseja obter

Retorna: o caracter no índice informado

```

programa
{
    inclui biblioteca Texto --> txt

    funcao inicio()
    {
        cadeia texto
        caracter letra
        texto = "Texto Qualquer" // Esta variável contém 14 caracteres
        escreva(texto)
        escreva("\n\n")
        // Vamos pegar a letra(caracter) Q da palavra qualquer
        // Lembre-se a posição dos caracteres dentro da variável cadeia começa em 0
        letra = txt.obter_caracter(texto,6)
        escreva("letra na posição 16 --> ",letra)
        escreva("\n")
    }
}

```

Texto Qualquer

letra na posição 16 --> Q

substituir

```
funcao cadeia substituir(cadeia cad, cadeia texto_pesquisa, cadeia texto_substituto)
```

Pesquisa por um determinado texto em uma cadeia e substitui todas as ocorrências por um texto alternativo

Parâmetros:

- cad - a cadeia que será pesquisada
- texto_pesquisa - o texto que será pesquisado na cadeia
- texto_substituto - o texto pelo qual as ocorrências serão substituídas

Retorna: a cadeia resultante da substituição

```

programa
{
    inclui biblioteca Texto --> txt

    funcao inicio()
    {
        cadeia texto, novotexto
        texto = "Texto Qualquer"
        escreva(texto)
        escreva("\n\n")
        novotexto = txt.substituir(texto, "Qualquer", "Novo Alterado")
        escreva(novotexto)
        escreva("\n")
    }
}

```

Texto Qualquer

Texto Novo Alterado

numero_caracteres

```
funcao inteiro numero_caracteres(cadeia cadeia)
```

Conta o número de caracteres existentes em uma cadeia

Parâmetros:

cadeia - um valor qualquer do tipo cadeia

Retorna: o número de caracteres na cadeia

```

programa
{
    inclui biblioteca Texto --> txt

    funcao inicio()
    {
        cadeia texto
        inteiro tamanho
        texto = "Texto Qualquer"
        escreva(texto)
        escreva("\n\n")
        tamanho = txt.numero_caracteres(texto)
        escreva("Quantidade de caracteres --> ", tamanho)
        escreva("\n")
    }
}

```

Texto Qualquer

Quantidade de caracteres --> 14

extrair_subtexto

```
funcao cadeia extrair_subtexto(cadeia cad, inteiro posicao_inicial,
inteiro posicao_final)
```

Extrai uma parte da cadeia delimitada pela posição inicial e final. Exemplos:

```
extrair_subtexto("salgado", 0, 3) // resultado: "sal"
```

```
extrair_subtexto("salgado", 3, 7) // resultado: "gado"
```

```
extrair_subtexto("salgado", 1, 5) // resultado: "alga"
```

Parâmetros:

cad - a cadeia a partir da qual será extraído o subtexto

posicao_inicial - a posição dentro da cadeia onde começará o subtexto

posicao_final - a posição dentro da cadeia onde terminará o subtexto

Retorna: uma cadeia contendo o subtexto

```
programa
{
    inclui biblioteca Texto --> txt

    funcao inicio()
    {
        cadeia texto,pedaco
        texto = "Texto Qualquer Digitado"
        escreva(texto)
        escreva("\n\n")
        pedaco = txt.extrair_subtexto(texto, 6, 14)
        escreva("Pedaco retirado da posição 6 à 14 --> " , pedaco)
        escreva("\n")
    }
}
```

```
Texto Qualquer Digitado
```

```
Pedaco retirado--> Qualquer
```

posicao_texto

```
funcao inteiro posicao_texto(cadeia texto, cadeia cad, inteiro posicao_inicial)
```

Procura por um texto dentro de uma cadeia e, caso encontrado, retorna a posição da primeira ocorrência

Parâmetros:

texto - o texto que será procurado na cadeia

cad - a cadeia dentro da qual o texto será procurado

posicao_inicial - a posição inicial a partir da qual o texto será procurado. Para procurar a partir do início da cadeia deve-se informar a posição 0

Retorna: a posição da primeira ocorrência do texto, caso ele seja encontrado. Caso o texto não seja encontrado na cadeia o valor retornado é -1

```
programa
{
    inclui biblioteca Texto --> txt

    funcao inicio()
    {
        cadeia texto
        inteiro encontrado
        texto = "Texto Qualquer Digitado"
        escreva(texto)
        escreva("\n\n")
        encontrado = txt.posicao_texto("xt", texto, 0)
        escreva("Posição onde foi encontrado 'xt' ", encontrado)
        escreva("\n")
    }
}
```

```
Texto Qualquer Digitado
```

```
Posição onde foi encontrado 'xt' 2
```

Vamos usar o que vimos até aqui

- 1 – Crie um programa onde será digitado uma palavra/frase qualquer, o programa deverá retornar a nova palavra escrita em maiúscula e ao contrário.

Vamos analisar o que precisamos fazer, primeiro precisamos criar as variáveis necessárias e claro uma variável que guarde o tamanho da palavra, e vamos reescrever esta palavra de trás para frente

```

programa
{
    inclui biblioteca Texto --> txt

    funcao inicio()
    {
        cadeia texto,maiuscula,novapalavra=""
        inteiro tamanho,i
        escreva("Digite um palavra/frase qualquer: ")
        leia(texto)
        maiuscula = txt.caixa_alta(texto)
        tamanho = txt.numero_caracteres(maiuscula)
        para(i=tamanho -1; i>=0; i--){
            novapalavra = novapalavra + txt.obter_caracter(maiuscula,i)
        }
        escreva(novapalavra)
    }
}

```

```

Digite um palavra/frase qualquer: ola como vai tudo bem
MEB ODUT IAV OMOC ALO

```

- 2 – Crie um programa onde será digitado uma palavra/frase qualquer, o programa deverá retornar a nova palavra escrita somente com as vogais.

```

inclua biblioteca Texto --> txt

funcao inicio()
{
    cadeia texto,novapalavra=""
    caracter letra
    inteiro tamanho,i
    escreva("Digite um palavra/frase qualquer: ")
    leia(texto)
    tamanho = txt.numero_caracteres(texto)
    para(i=0; i< tamanho; i++){
        letra = txt.obter_caracter(texto,i)
        se(letra == 'a' ou letra=='A' ou letra=='e' ou letra=='E' ou letra=='i' ou letra=='I' ou
        |letra=='o' ou letra=='O' ou letra=='u' ou letra=='U'){
            novapalavra =novapalavra + letra
        }
    }
    escreva(novapalavra)
}

```

```

Digite um palavra/frase qualquer: Ola como vai tudo bem
Oaooiuoe

```

DESAFIOS

1 – Crie um programa onde o usuário irá digitar uma palavra/frase qualquer, no final o programa deverá retornar a palavra/frase somente com as vogais em maiúsculas.

Exemplo

Casa Feliz → deve retornar → cAsA fElIz

2 – Crie um programa onde o usuário irá digitar uma palavra/frase qualquer, no final o programa deverá a palavra/frase sem nenhum espaço em branco.

Exemplo

Ola como vai tudo bem → deve retornar → Olacomovaitudobem

3 – Crie um programa onde o usuário irá digita uma palavra/frase e depois uma letra qualquer, no final o programa deverá retornar a frase reescrita de forma que onde ele encontrar a letra digitada deve ser substituída por @(arroba)

Exemplo

Frase → Ola como vai

Letra → a

Retorno da palavra reescrita → Ol@ como v@i