

```

Algoritmo "ProgCifraCesar"
//Encripta/Decripta mensagem com o método "Cifragem de Cesar"
//Autor: Mário Leite - Copyright(2017)
//marleite#gmail.com
//-----
//Variáveis globais
    Var Op, MsgOriginal, MsgCifrada, MsgDecifrada: caractere
        TamMsgOriginal, TamMsgCifrada, Desloc: inteiro
//-----
Funcao CifraCesar(MsgOrig:caractere; TamMsg,Desloc:inteiro): caractere
    var j, NumDecif: inteiro
    VetMsgOrig, VetMsgCif: vetor[1..200] de caractere
    NumDefic, MsgCif, StrDecif : caractere

Inicio
    Para j De 1 Ate TamMsg Faca
        VetMsgOrig[j] <- Copia(MsgOrig,j,1)
        VetMsgOrig[j] <- Maiusc(VetMsgOrig[j]) //converte em maiúsculas
    FimPara
    {Monta a mensagem cifrada}
    MsgCif <- ""
    Para j De 1 Ate TamMsg Faca
        StrDecif <- VetMsgOrig[j]
        NumDecif <- Asc(StrDecif)
        Se((VetMsgOrig[j]=" ") ou (VetMsgOrig[j]=" ")) Entao
            VetMsgCif[j] <- " "
        Senao
            NumDefic <- NumPCarac(NumDecif) //variável estrutura
            Escolha (NumDefic) //trata exceções para cifrar X Y Z
                Caso "88"
                    VetMsgCif[j] <- Carac(65)
                Caso "89"
                    VetMsgCif[j] <- Carac(66)
                Caso "90"
                    VetMsgCif[j] <- Carac(67)
                OutroCaso
                    VetMsgCif[j] <- Carac(NumDecif+Desloc)
            FimEscolha //fim do tratamento das exceções X Y Z
        FimSe
        MsgCif <- MsgCif + VetMsgCif[j]
    FimPara
    Retorne MsgCif
FimFuncao //fim da função "CifraCesar"
//-----
Funcao DecifraCesar(MsgCif:caractere;TamMsg,Desloc:inteiro): caractere
    var j, NumCif: inteiro
    VetMsgCif, VetMsgDecif: vetor[1..200] de caractere
    NumFic, MsgDecif, StrCif : caractere

Inicio
    Para j De 1 Ate TamMsg Faca
        VetMsgCif[j] <- Copia(MsgCif,j,1)
    FimPara
    {Monta a mensagem decifrada}
    MsgDecif <- ""
    Para j De 1 Ate TamMsg Faca
        StrCif <- VetMsgCif[j]
        NumCif <- Asc(StrCif)

```

```

Se((VetMsgCif[j]=" ") ou (VetMsgCif[j]=" ")) Entao
    VetMsgDecif[j] <- " "
Senao
    Se(NumCif<65) Entao
        NumCif <- NumCif+32 //garante caracteres normais
        Se(NumCif=81) Entao //exceção para o ponto (.) na mensagem
            VetMsgDecif[j] <- "."
        Senao
            VetMsgDecif[j] <- Carac(NumCif)
        Fimse
    Senao
        NumFic <- NumPCarac(NumCif)
        Escolha (NumFic) //trata exceções para decifrar A B C
            Caso "65"
                VetMsgDecif[j] <- Carac(88)
            Caso "66"
                VetMsgDeCif[j] <- Carac(89)
            Caso "67"
                VetMsgDeCif[j] <- Carac(90)
            OutroCaso
                VetMsgDecif[j] <- Carac(NumCif-Desloc)
        FimEscolha //fim do tratamento das exceções de A B C
    Fimse
FimSe
MsgDecif <- MsgDecif + VetMsgDecif[j]
FimPara
Retorne MsgDecif
FimFuncao //fim da função "DecifraCesar"
//=====
//Programa principal
Inicio
Op <- "X"
Enquanto ((Op<>"C") e (Op<>"D")) Faca
    Escreva("Para cifrar [C] - Para decifrar [D]: ")
    Leia(op)
    Op <- Maiusc(Op)
FimEnquanto
Escreval("") //salta linha
Desloc <- 0
Enquanto ((Desloc<3) ou (Desloc>25)) Faca
    Escreva("Digite o tamanho do deslocamento [3 a 25]: ")
    Leia(Desloc)
FimEnquanto
Escreval("")
TamMsgOriginal <- 2
Se(Op="C") Entao
    {Lê a mensagem original}
    Enquanto ((TamMsgOriginal<3) ou (TamMsgOriginal>200)) Faca
        Escreval("Digite a mensagem original [de 3 até 200 caracteres]: ")
        Leia(MsgOriginal)
        TamMsgOriginal <- Compr(MsgOriginal)
    FimEnquanto
    MsgCifrada <- CifraCesar(MsgOriginal,TamMsgOriginal,Desloc)
Senao
    TamMsgCifrada <- 2
    Enquanto ((TamMsgCifrada<3) ou (TamMsgCifrada>200)) Faca
        Escreval("Digite a mensagem cifrada [de 3 até 200 caracteres]: ")

```

```

        Leia(MsgCifrada)
        TamMsgCifrada <- Compr(MsgCifrada)
    FimEnquanto
    MsgDecifrada <- DecifraCesar(MsgCifrada,TamMsgCifrada,Desloc)
FimSe
LimpaTela
Escreval("")
Escreval("")
Escreval("Cifragem de Cesar")
Escreval("-----")
Escreval("")
Se(Op="C") Entao
    {Exibe a mensagem cifrada}
    Escreval("Mensagem original: ", MsgOriginal)
    Escreval("Mensagem cifrada:  ", MsgCifrada)
Senao
    {Exibe a decifrada}
    Escreval("Mensagem cifrada:  ",MsgCifrada)
    Escreval("Mensagem decifrada: ",MsgDecifrada)
FimSe
Escreval("")
FimAlgoritmo //fim do programa "ProgCifraCesar"

```