Cifra de Vigenère

Mário Leite

...

O método de proteção de mensagens conhecido como **"Cifra de Vigenére"** é baseado em uma chave única que serve tanto para cifrar como para decifrar a mensagem; é como uma chave física para "*fechar a porta ao girar na fechadura para a direita*" e "*abrir a porta ao girar na fechadura para a esquerda*". O esquema do **quadro 1** mostra a mensagem "**DESTRUIREMAILSDAPONTE**", cuja chave de cifragem/decifragem é "**PIXULECO**", para encobrir/descobrir algum ilícito administrativo.

Mensagem	D	**E**	S	T	R	U	I	R	**E**	M	A	I	L	S	D	A	P	O	N	T	**E**
Chave	P	I	X	U	L	E	C	O	P	I	X	U	L	E	C	O	P	I	X	U	L
Cifragem	S	**M**	P	N	C	Y	K	F	**T**	U	X	C	W	W	F	O	E	W	K	N	**P**

Quadro 1 - Esquema de cifragem/decifragem de Vigenère

Observe no **quadro 1** que uma mesma letra da mensagem original foi substituída por várias, e não por apenas uma única. Por exemplo, a letra "**E**" foi substituída por "**M**", por "**T**" e por "**P**" nas suas três ocorrências. Se fosse utilizada a "Cifragem de Cesar" simples com deslocamento **3** ela seria substituída somente pela letra "**H**" nas três ocorrências. Outro detalhe a ser observado no método de cifrar/decifrar com a "Cifra de Vigenère" é que no dispositivo montado para o algoritmo a chave abrange toda a mensagem original; a chave "**PIXULECO**" tem oito caracteres, enquanto a mensagem original tem vinte e um; então, é preciso repetir os caracteres da chave até alcançar o mesmo tamanho da mensagem; portanto, é um método de cifragem com substituição *polialfabética*, ao contrário da "Cifragem de Cesar". Na **tabela 1** o esquema de cifrar, por exemplo, a primeira letra "**D**" da mensagem para se transformar em "**S**" pode ser entendido como a intersecção da letra "**D**" da primeira linha com a letra "**P**" da primeira coluna, assim como as demais, dentro de uma matriz quadrada **26x26**.

A	B	C	**D**	**E**	F	G	H	I	J	K	L	M	**N**	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	**M**	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	**P**	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	**S**	**T**	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	Y	Z	A	B	C	D	E	F	G	H	I	J	**K**	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X

Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Tabela 1 - Esquema tabela de cifragem/decifragem de Vigenère

O programa **"ProgCifraVigenere"**, implementado em **VisuAlg** mais adiante, é uma aplicação simples e didática da **"Cifra de Vigenère"** baseado na álgebra do método, que pode ser resumida nas duas equações abaixo, considerando que as letras da grelha são numeradas de **0** (para **A**) até **25** (para **Z**), como mostrado na tabela 9.7.

$C_i = (P_i + K_i)$ **Mod** 26 ... *para cifrar*

$P_i = (C_i - K_i)$ **Mod** 26 ... *para decifrar*

i ==> índice da letra (variando de 0 a 25)

C_i ==> valor numérico do caractere de índice i da mensagem cifrada.

P_i ==> valor numérico do caractere de índice i da mensagem original.

K_i ==> valor numérico do caractere de índice i da chave.

A **tabela 2a** mostra um esquema de como seria aplicada a equação para cifrar a mensagem **DESTRUIREMAILSDAPONTE**, baseada nos índices de **0** a **25** das letras,

Letra Orig.	Valor Orig.	Letra Ch.	Valor Ch.	Soma	Soma **mod** 26	Letra cifrada
D	3	**P**	15	18	18	**S**
E	4	**I**	8	12	12	**M**
S	18	**X**	23	41	15	**P**
T	19	**U**	20	39	13	**N**
R	17	**L**	11	28	2	**C**
U	20	**E**	4	24	24	**Y**
I	8	**C**	2	10	10	**K**
R	17	**O**	14	31	5	**F**
E	4	**P**	15	19	19	**T**
M	12	**I**	8	20	20	**U**
A	0	**X**	23	23	23	**X**
I	8	**U**	20	28	2	**C**
L	11	**L**	11	22	22	**W**
S	18	**E**	4	22	22	**W**
D	3	**C**	2	5	5	**F**
A	0	**O**	14	14	14	**O**
P	15	**P**	15	30	4	**E**
O	14	**I**	8	22	22	**W**
N	13	**X**	23	36	10	**K**
T	19	**U**	20	39	13	**N**
E	4	**L**	11	15	15	**P**

Tabela 2a - Esquema de cifragem pela "álgebra de Vigenère"

A **tabela 2b** mostra um esquema de como seria aplicada a equação para decifrar a mensagem gerada como **SMPNCYKFTUXCWWFOEWKNP**

Letra Cifr.	Valor Cifr.	Letra Ch.	Valor Ch.	Subtração	Sub **mod** 26	Letra Orig.
S	18	P	15	3	3	D
M	12	I	8	4	4	E
P	15	X	23	-8		
N	13	U	20	-7		
C	2	L	11	-9		
Y	24	E	4	20	20	U
K	10	C	2	8	8	I
F	5	O	14	-9		
T	19	P	15	4	4	E
U	20	I	8	12	12	M
X	23	X	23	0	03	A
C	2	U	20	-18		
W	22	L	11	11	11	L
W	22	E	4	18	18	S
F	5	C	2	3	3	D
O	14	O	14	0	0	A
E	4	P	15	-11		
W	22	I	8	13	14	O
K	10	X	23	-13		
N	13	U	20	-7		
P	15	L	11	4	4	E

Tabela 2b - Esquema de decifragem pela "álgebra de Vigenère"

Observe na **tabela 2b** que pela álgebra de Vigenère para decifragem (C_i - K_i) **mod** 26 algumas letras da mensagem original não deu certo (faixas sombreadas), pois a subtração nesses casos deu negativa e a expressão para decifrar não pode ser aplicada. Para resolver esses casos, deve-se adicionar **26** ao resultado da subtração; deste modo, a decifragem das letras que não aparecem na **tabela 2b** ficaria assim, na sequência:

```
(-8 + 26) mod 26  =  18  (letra S)
(-7 + 26) mod 26  =  19  (letra T)
(-9 + 26) mod 26  =  17  (letra R)
(-9 + 26) mod 26  =  17  (letra R)
(-18 + 26) mod 26 =   8  (letra I)
(-11 + 26) mod 26 =  15  (letra P)
(-13 + 26) mod 26 =  13  (letra N)
(-7 + 26) mod 26  =  19  (letra T)
```

O programa modular **“ProgCifraVigenere”**, a seguir, apresenta uma solução simples para cifragem/decifragem de mensagens com o método “Cifra de Vigenère”.

```
Algoritmo "ProgCifraVigenere"
//Programa de cifragem/decifragem pelo método "Cifra de Vigenère"
//Autor:  Mário Leite
//-------------------------------------------------------------------------
//Variáveis globais
  Var x, TamMsgOrig, TamMsgCif, TamChave: inteiro
      MatVig: vetor[1..26,1..26] de caractere
      MensagemOrig, MensagemCif, Chave: caractere
      VetChaveTot, VetChave: vetor[1..100] de caractere
      VetMsgCif, VetMsgDecif, VetMsgOrig: vetor[1..100] de caractere
      Opcao, Msg, Car: caractere
//-------------------------------------------------------------------------
funcao VerifTexto(Texto,Msg:caractere; TamTexto:inteiro): logico
//Valida a mensagem ou a chave para considerar apenas letras
inicio
   Car <- "X"
   Se(TamTexto<=100) Entao
      Escolha Msg
         Caso "Msg"
           Para x De 1 Ate TamTexto Faca
              VetMsgOrig[x] <- Copia(Texto,x,1)
              Se((Asc(VetMsgOrig[x])<65) ou (Asc(VetMsgOrig[x])>90)) Entao
                 Car <- "?"
                 Interrompa  //encontrou caracter inválido na mensagem
              FimSe
           FimPara
         Caso "Chv"
           Para x De 1 Ate TamTexto Faca
              VetChave[x] <- Copia(Texto,x,1)
              Se((Asc(VetChave[x])<65) ou (Asc(VetChave[x])>90)) Entao
                 Car <- "?"
                 Interrompa  //encontrou caracter inválido na chave
              FimSe
           FimPara
         OutroCaso
           Car <- "?"
      FimEscolha
    FimSe
    Se((Car="?") ou (TamTexto>100)) Entao
       Retorne Falso
    Senao
       Retorne Verdadeiro
    FimSe
fimfuncao  //fim da função "VerifTexto"
//-------------------------------------------------------------------------
procedimento GeraGrelha
//Monta a "Grelha de Vigenère" como uma matriz 26x26 de letras
   var i, j, k, m, n: inteiro
inicio
   Escreval("   A  B  C  E  R  F  G  H  I  J  K  L  M  N  O  P  Q  R
             S  T  U  V  W  X  Y  Z")
   Para i De 1 Ate 26 Faca
       Escreva(Carac(i+64))
       Escreva(" ")
      Para j De 1 Ate 26 Faca
         k <- 63+i+j  //primeira linha da tabela 9.7 (A-Z sem exceção)
         Se(Asc(Carac(k))>90) Entao  //primeira exceção (segunda linha)
            k <- 90 - (27-i)
         FimSe
```

```
            MatVig[i,j] <- Carac(k)
            Escreva(" ", MatVig[i,j], " ")  //imprime letra
            Se((MatVig[i,j]="Z")e(j<26)) Entao //encontrou "Z" antes do fim
               k <- 0
               m <- j+1
               Para n De m Ate 26 Faca
                  k <- k + 1
                  MatVig[i,n]<- Carac(64+k) //imprime a partir da letra "A"
                  Escreva(" ",MatVig[i,n], " ")
               FimPara
               Interrompa  //abandona loop j para pegar nova linha da grelha
            FimSe
         FimPara
         Escreval("")  //salta linha
      FimPara
fimprocedimento   //fim do procedimento "GeraGrelha"
//--------------------------------------------------------------------------------
procedimento MontaEsquema
//Monta um esquema para criar o vetor VetChaveTot[] da chave monogrâmica
   var TamMsg, j, k, n: inteiro
inicio
   LimpaTela
   Escreval("")
   Se(Opcao="2") Entao   //cifrar
      TamMsg <- TamMsgOrig
      Escreval("Processo de cifragem de mensagem pelo método 'Cifra de Vigenère'")
   Senao  //decifrar
      TamMsg <- TamMsgCif
      Escreval("Processo de decifragem de mensagem pelo método 'Cifra de Vigenère'")
   FimSe
   Escreval("-------------------------------------------------------------------")
   Escreval("")
   Se(Opcao="2") Entao
      Escreval("Original: ")
   Senao
      Escreval("Cifrada: ")
   FimSe
   Se(Opcao="2") Entao
      Para j De 1 Ate TamMsg Faca
         Escreva(VetMsgOrig[j], " ")
      FimPara
   Senao
      Para j De 1 Ate TamMsg Faca
         Escreva(VetMsgCif[j], " ")
      FimPara
   FimSe
   Escreval("")
   Escreval("")
   k <- 0
   n <- 0
   Escreval("Chave:     ")
   Para j De 1 Ate TamMsg Faca
      n <- n + 1
      Se(j<=TamChave) Entao  //mensagem ainda do tamanho da chave
         Escreva(VetChave[j], " ")  //escreve letra da chave
         VetChaveTot[n] <- VetChaveTot[n] + VetChave[j]
      Senao //posição da mensagem é superior ao tamanho da chave
         k <- k + 1  //recomeça com novo elemento da chave
         Se(k<=TamChave) Entao //chave completa, mas ainda TamMsg>TamChave
            Escreva(VetChave[k], " ")
            VetChaveTot[n] <- VetChaveTot[n] + VetChave[k]
         Senao  //esgotou as letras da chave e ainda TamMsg>TamChave
```

```
            k <- 1
            Escreva(VetChave[k]," ") //recomeça escrever caracteres da chave
            VetChaveTot[n] <- VetChaveTot[n] + VetChave[k]
         Fimse
      FimSe
   FimPara  //fim do loop de varredura da mensagem
   Escreval("")
   Escreval("")
fimprocedimento  //fim do procedimento "MontaEsquema"
//--------------------------------------------------------------------------
funcao PegaNumLetra(Letra:caractere): inteiro
//Obtém o número da letra da mensagem em função da tabela 9.7
   var NumLetra: inteiro
inicio
   Escolha Letra
      Caso "A"
        NumLetra <- 0
      Caso "B"
        NumLetra <- 1
      Caso "C"
        NumLetra <- 2
      Caso "D"
        NumLetra <- 3
      Caso "E"
        NumLetra <- 4
      Caso "F"
        NumLetra <- 5
      Caso "G"
        NumLetra <- 6
      Caso "H"
        NumLetra <- 7
      Caso "I"
        NumLetra <- 8
      Caso "J"
        NumLetra <- 9
      Caso "K"
        NumLetra <- 10
      Caso "L"
        NumLetra <- 11
      Caso "M"
        NumLetra <- 12
      Caso "N"
        NumLetra <- 13
      Caso "O"
        NumLetra <- 14
      Caso "P"
        NumLetra <- 15
      Caso "Q"
        NumLetra <- 16
      Caso "R"
        NumLetra <- 17
      Caso "S"
        NumLetra <- 18
      Caso "T"
        NumLetra <- 19
      Caso "U"
        NumLetra <- 20
      Caso "V"
        NumLetra <- 21
      Caso "W"
        NumLetra <- 22
```

```
      Caso "X"
        NumLetra <- 23
      Caso "Y"
        NumLetra <- 24
      Caso "Z"
        NumLetra <- 25
   FimEscolha
   Retorne NumLetra
fimfuncao  //fim da função "PegaLetra"
//--------------------------------------------------------------------------
procedimento CifraMsgOrig
//Cifra a mensagem original baseando na fórmula: C=(P+K) mod 26
  var j, NumLetraOrig, NumLetraChv, NumLetraCif: inteiro
      LetraCif: caractere
inicio
   Escreval("Cifragem: ")
   Para j De 1 Ate TamMsgOrig Faca
      NumLetraOrig <- PegaNumLetra(VetMsgOrig[j])
      NumLetraChv  <- PegaNumLetra(VetChaveTot[j])
      NumLetraCif  <- NumLetraOrig + NumLetraChv
      NumLetraCif  <- NumLetraCif Mod 26 //obtém número-letra da tab. 9.7
      NumLetraCif  <- NumLetraCif + 65 //obtém código ASCII da letra
      LetraCif <- Carac(NumLetraCif)   //obtém a letra cifrada
      Escreva(LetraCif, " ")
   FimPara
   Escreval("")
fimprocedimento   //fim do procedimento "CifraMsgOrig"
//--------------------------------------------------------------------------
procedimento DeCifraMsgCif
//Decifra a mensagem original baseando na fórmula: P=(C-K) mod 26
   var j, NumLetraCif, NumLetraChv, NumLetraDecif: inteiro
       LetraDecif: caractere
inicio
   Escreval("Decifragem: ")
   Para j De 1 Ate TamMsgCif Faca
      NumLetraChv <- PegaNumLetra(VetChaveTot[j])
      NumLetraCif <- PegaNumLetra(VetMsgCif[j])
      //Verifica exceção na subtração
      Se(NumLetraCif<NumLetraChv) Entao
         NumLetraDecif <- (NumLetraCif + 26) - NumLetraChv
      Senao
         NumLetraDecif <- (NumLetraCif-NumLetraChv)
      FimSe //fim da verificação de exceção na subtração
      NumLetraDecif<-NumLetraDecif Mod 26 //obtém número-letra da tab. 9.7
      NumLetraDecif <- NumLetraDecif + 65 //obtém código ASCII da letra
      LetraDecif <- Carac(NumLetraDecif)  //obtém a letra decifrada
      Escreva(LetraDecif, " ")
   FimPara
   Escreval("")
fimprocedimento  //fim do procedimento "DeCifraMsgCif"
//--------------------------------------------------------------------------
procedimento MontaMenu
inicio
   LimpaTela
   Escreval("")
   Escreval("=============== Menu Principal ================")
   Escreval("Apenas exibir a Grelha de Vigenère.........[1]")
   Escreval("Cifrar uma mensagem........................[2]")
   Escreval("Decifrar uma mensagem......................[3]")
   Escreval("Fechar o programa..........................[4]")
   Escreval("-----------------------------------------------")
   Escreval("")
```

```
fimprocedimento   //fim do procedimento "MontaMenu"
//================================================================================
//Programa principal
Inicio
  Opcao <- ""
  Enquanto (Opcao<>"4") Faca  //mantém o menu de opções na tela
       MontaMenu   //chama procedimento para montar o menu de opções
      //Limpa todos os vetores
      Para x De 1 Ate 100 Faca
         VetMsgOrig[x] <- ""
         VetMsgCif[x] <- ""
         VetChave[x]  <- ""
         VetChaveTot[x] <- ""
      FimPara
        Opcao <- ""
        Escreva("Digite sua opção [1 a 4]: ")
        Leia(Opcao)
        Se(Opcao="4") Entao
           Interrompa  //sai do loop incondicionalmente
      FimSe
      Escreval("")  //salta linha
      Escolha Opcao
         Caso "1"
            GeraGrelha  //chama procedimento para gerar a grelha
         Caso "2"
            TamMsgOrig <- 101
            //Cria/valida a mensagem original
            Msg <- "Msg"
            Enquanto (Nao(VerifTexto(MensagemOrig,Msg,TamMsgOrig))) Faca
               Escreva("Mensagem original [apenas letras e máximo 100]: ")
               Leia(MensagemOrig)
               TamMsgOrig <- Compr(MensagemOrig)
               MensagemOrig <- Maiusc(MensagemOrig)
            FimEnquanto //fim de validação da mensagem original
            Para x De 1 Ate TamMsgOrig Faca
               VetMsgOrig[x] <- Copia(MensagemOrig,x,1)
            FimPara
            //Cria/valida a chave de cifragem
            TamChave <- 101
            Msg <- "Chv"
            Enquanto (Nao(VerifTexto(Chave,Msg,TamChave))) Faca
               Escreva("Chave de cifragem [tamanho máximo da mensagem]: ")
               Leia(Chave)
               Chave <- Maiusc(Chave)
               TamChave <- Compr(Chave)
            FimEnquanto  //fim de validação da chave de cifragem
            Escreval("")
            //Loop para montar o vetor inicial da chave de cifragem
            Para x De 1 Ate TamChave Faca
               VetChave[x] <- Copia(Chave,x,1)
            FimPara
            LimpaTela
            MontaEsquema
            CifraMsgOrig   //chama procedimento para cifrar mensagem
         Caso "3"
            TamMsgCif <- 101
            Msg <- "Msg"
            //Cria/valida a mensagem cifrada
            Enquanto (Nao(VerifTexto(MensagemCif,Msg,TamMsgCif))) Faca
              E screva("Mensagem cifrada[apenas letras e máximo 100]: ")
               Leia(MensagemCif)
```

```
                TamMsgCif <- Compr(MensagemCif)
                MensagemCif <- Maiusc(MensagemCif)
            FimEnquanto  //fim de validação da mensagem cifrada
            Para x De 1 Ate TamMsgCif Faca
                VetMsgCif[x] <- Copia(MensagemCif,x,1)
            FimPara
            //Cria/valida a chave de decifragem
            TamChave <- 101
            Msg <- "Chv"
            Enquanto (Nao(VerifTexto(Chave,Msg,TamChave))) Faca
                Escreva("Chave de decifragem [tamanho máximo da mensagem]: ")
                Leia(Chave)
                Chave <- Maiusc(Chave)
                TamChave <- Compr(Chave)
            FimEnquanto  //fim de validação da chave de decifragem
            Escreval("")
            //Loop para montar o vetor inicial da chave de decifragem
            Para x De 1 Ate TamChave Faca
                VetChave[x] <- Copia(Chave,x,1)
            FimPara
            LimpaTela
            MontaEsquema
            DeCifraMsgCif  //chama procedimento para decifrar a mensagem
        OutroCaso
            //Nada
    FimEscolha
    Escreval("------------------------------------------------------------")
    Escreval("")
    Escreval("")
    Escreval("")
    Escreva("Tecle <Enter> para continuar ")
    Leia(Car) //apenas para interromper temporariamente o processamento
  FimEnquanto
  Escreval("")
  Escreval("")
FimAlgoritmo   //fim do programa "ProgCifraVigenere"
```

A **figura 1** mostra a saída do programa **“ProgCifraVigenere”** para a opção **1** do *menu*: “**Apenas exibir a Grelha de Vigenère**”.

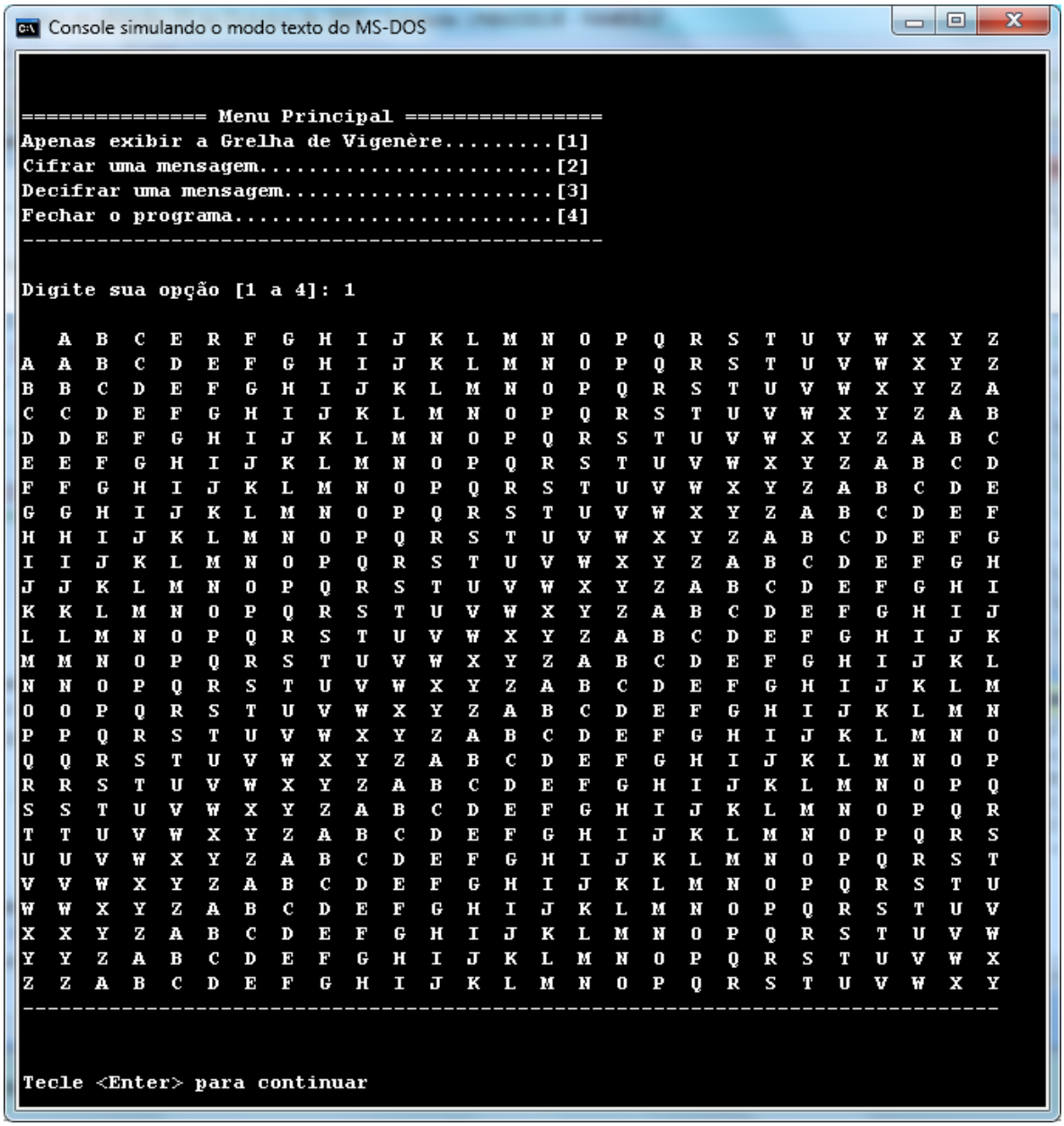

Figura1 - Mostrando a grelha de Vigenére

As **figuras 2a** e **2b** mostram a saída do programa para a opção: "**Cifrar uma mensagem**".

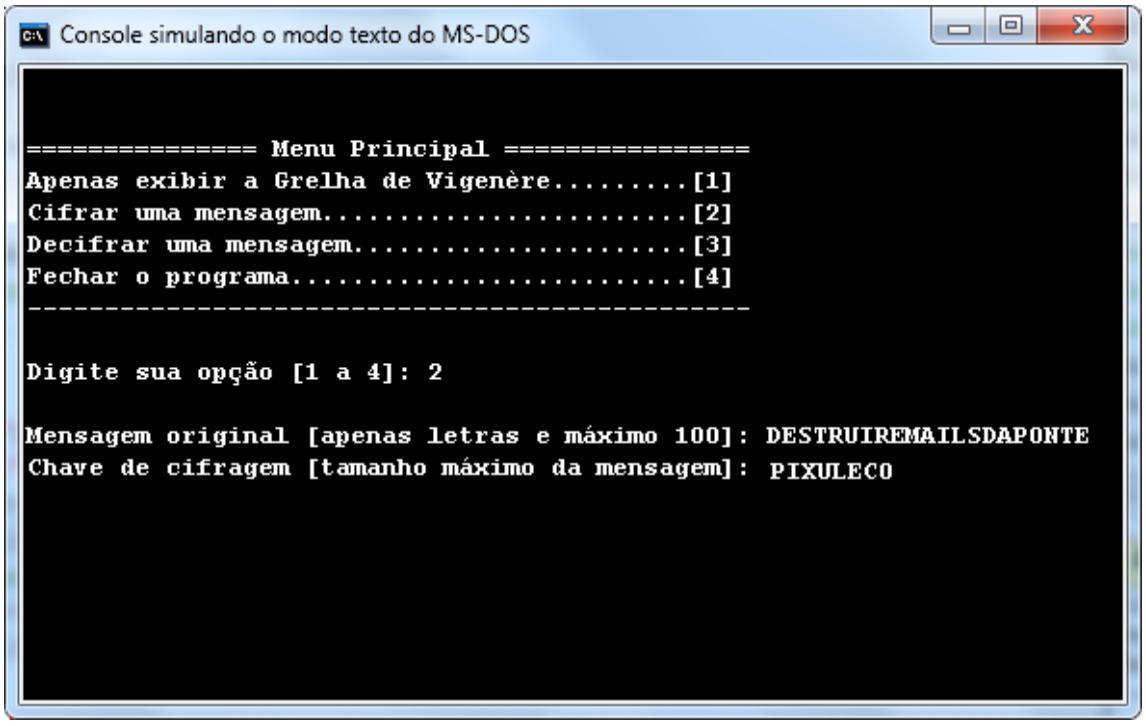

Figura 2a - Optando por cifrar uma mensagem pelo método "Cifra de Vigenère"

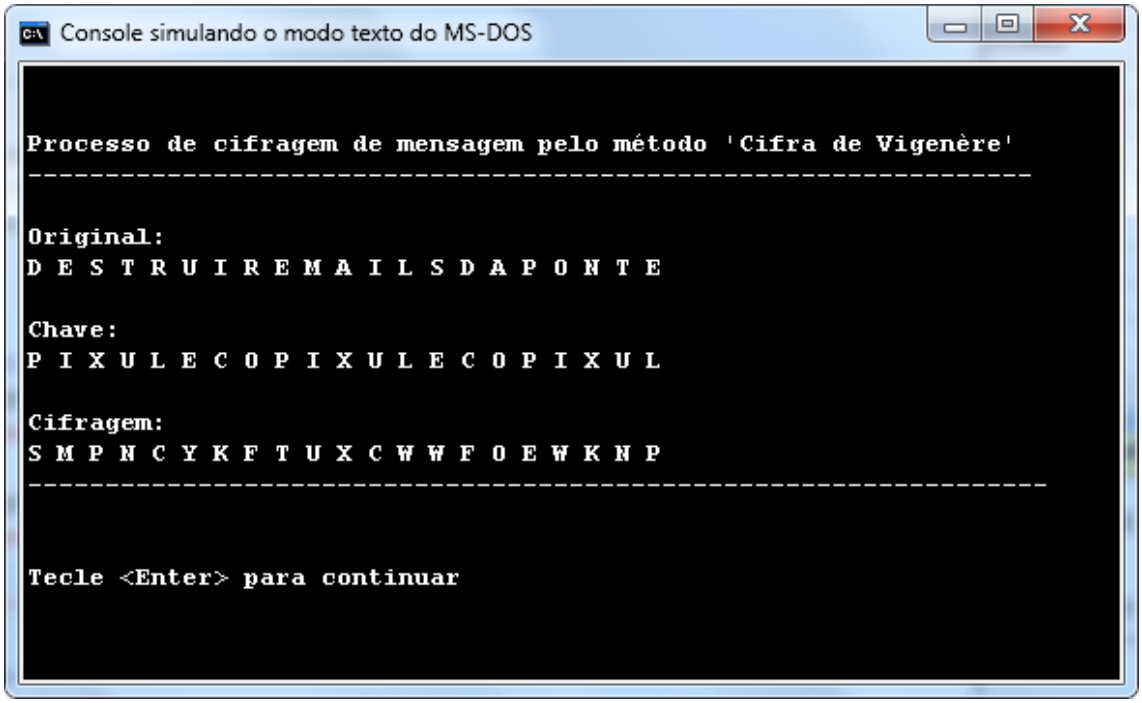

Figura 2b - Cifrando uma mensagem pelo método "Cifra de Vigenère"

As **figuras 3a** e **3b** mostram a saída do programa para a opção: "**Decifrar uma mensagem**".

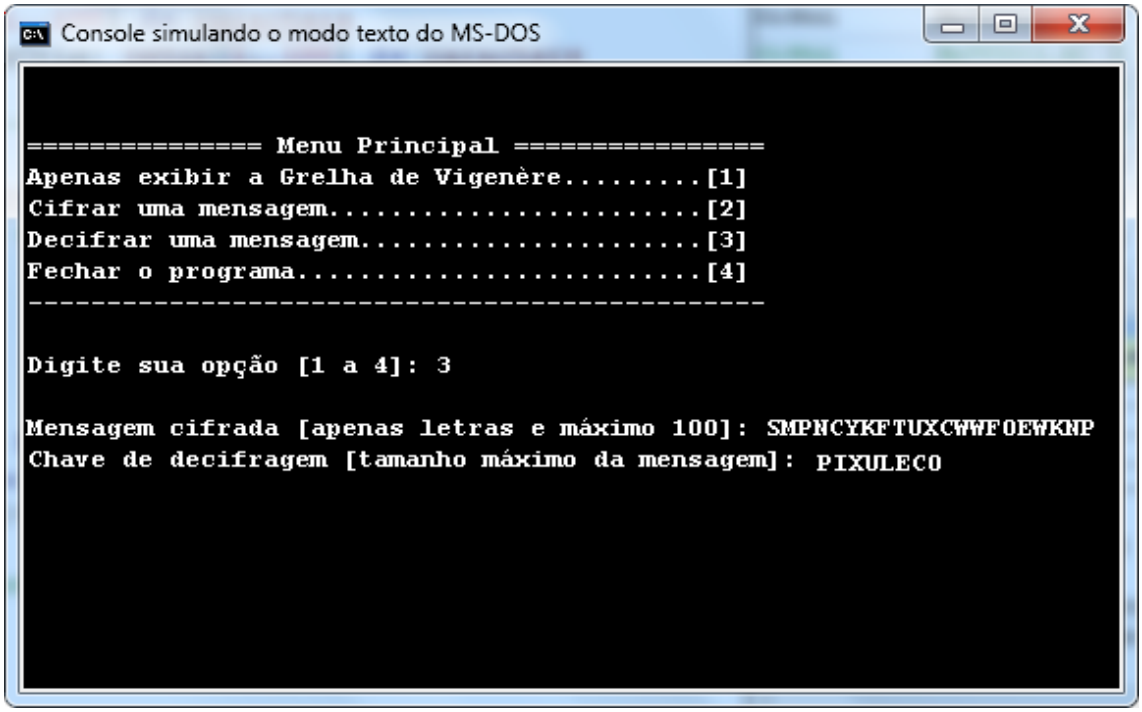

Figura 3a - Optando por decifrar uma mensagem pelo método “Cifra de Vigenère”

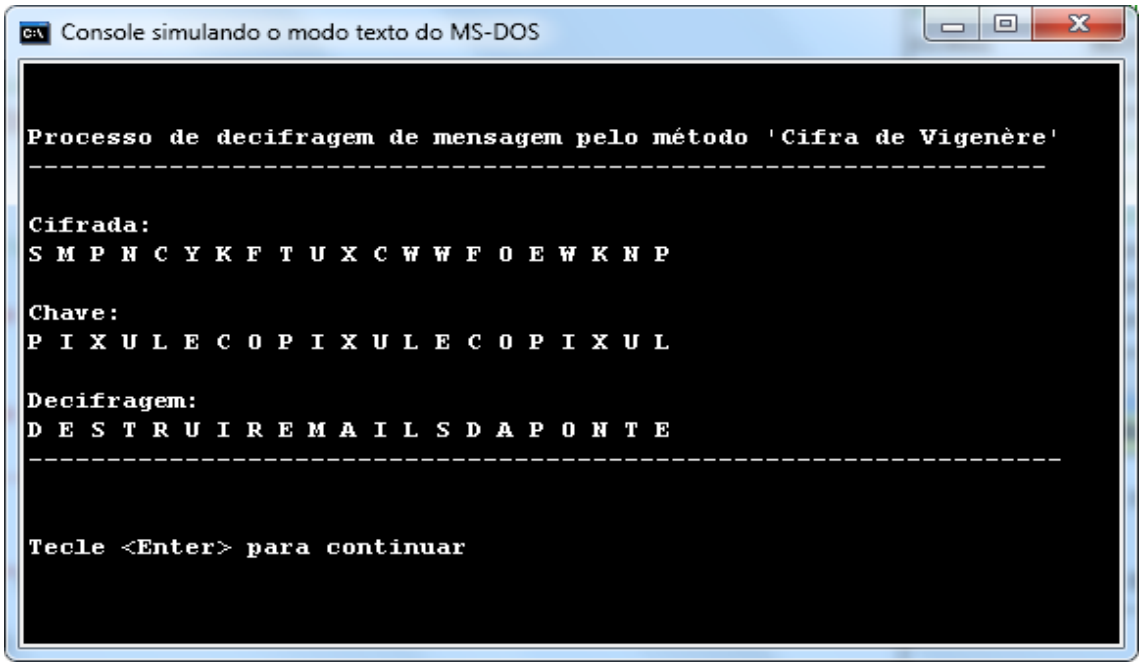

Figura 3b - Decifrando uma mensagem pelo método “Cifra de Vigenère”

Nota1: Esta postagem foi tirada do livro do autor: “Curso Básico de Programação: Teoria e Prática” - Editora Ciência Moderna da Computação” Rio, 2017. Acesse o *link* abaixo da Amazon.

https://www.amazon.com.br/Curso-B%C3%A1sico-Programa%C3%A7%C3%A3o-Teoria-Pr%C3%A1tica/dp/8539908700

Nota2: Acesse o *link* abaixo para ver os meus mais recentes sobre Python publicado pela Amazon em volume único (1001 programas) ou em volumes individuais da coleção “***1001 Programas em Python Para Você Aprender Praticando***”:

Volume único: 1001 programas
Volume1: Nível Básico (500 programas)
Volume2: Nível Intermediário (300 programas)
Volume3: Nível Avançado (201 programas)

https://www.amazon.com.br/s?k=1001%20programas%20em%20Python&rh=n%3A6740748011&crid=1K3A6FH0KZA9P&sprefix=1001%20programas%20em%20python%20%2Cstripbooks%2C418&ref=nb_sb_noss&s=03

Nota3: Acesse o *link* abaixo para ver os meus mais recentes livros da coleção “***1001 Programas em Python Para Você Aprender Praticando***”, publicados pelo “Clube de Autores” em volumes individuais e no formato impresso.

https://clubedeautores.com.br/livros/autores/mario-leite

Para adquirir PDF de todos os meus livros: contato marleite@gamil.com