# ...A CIFRA DE VIGENERE...



```
> restart:
  with(StringTools):
```

## Codificacao:

```
> # MENSAGEM :
  Mg:="eles nao sabem nem sonham que o sonho e uma constante da
  vida tao concreta e definida como outra coisa qualquer como esta
  pedra cinzenta em que me sento e descanso como este ribeiro
  manso em serenos sobressaltos como estes pinheiros altos que em
  verde e oiro se agitam como estas aves que gritam em bebedeiras
  de azul eles nao sabem que o sonho e vinho e espuma e fermento
  bichinho alacre e sedento de focinho pontiagudo que fossa
  atraves de tudo num perpetuo movimento eles nao sabem que o
  sonho e tela e cor e pincel base fuste capitel arco em ogiva
  vitral pinaculo de catedral contraponto sinfonia mascara grega
  magia que e retorta de alquimista mapa do mundo distante rosa
  dos ventos infante caravela quinhentista que e cabo da boa
  esperanca ouro canela marfim florete de espadachim bastidor
  passo de danca colombina e arlequim passarola voadora para raios
  locomotiva barco de proa festiva alto forno geradora cisao do
  atomo radar ultra som televisao desembarque em foguetao na
  superficie lunar eles nao sabem nem sonham que o sonho comanda a
  vida que sempre que um homem sonha o mundo pula e avanca como
  bola colorida entre as maos de uma crianca":
> evalb(1+1=2);
> # CHAVE...
  Key:="CONFIDENCIAL":
  lgk:=length(Key):
  printf("Comprimento da chave: %A\n\n",lgk):
  # CRIPTOGRAMA...
  Cr:="": j:=0: printf("Criptograma:\n\n"):
  for i from 1 while evalb(Mg[i]<>"") do
   if Mg[i]<>" " then
    j:=j+1: if j=lgk+1 then j:=1: fi:
    c:=convert(Mg[i],bytes)[1]+convert(Key[j],bytes)[1]-97-65 mod
  26 +65:
    Cr:=cat(Cr,convert([c],bytes)):
   fi:
  od:
  ##########################
```

```
m:=5: lg:=length(Cr): j:=0:
for i from 1 to lg do
 j:=j+1: if j=m+1 then j:=1: fi:
 printf("%A",Cr[i]):
 if j=m then printf(" "): fi:
 if frac(i/(m*12))=0 then printf("\n"): fi:
od:
>
```

# Analise de um Criptograma

```
> Cr:="FPCFN BMHRV PMFCI MWSFF RRVGI VITNA UMZMG YVZSE IYMZE FPCCC
      OILRQ AOIBB RHEOM IKWQS IVQNE JABHB WETPC QAEKU WPLYM SSBMP
      DTRCE EWSPL SVYHI GGKLD RMKTN PULOE WHPMM OZBFI IMLXC QCGWE
      XHSNF SFJEE LMNPI IWSSB WWDPV VFSIS TFMAW TSYEN SOLWR RVSPE
      RFKBQ HRCNI FVNPF EULGF ZNILX CLBII PSLGQ XUDQA UNITT TFXBV
      MITLT ZZNHH YOLCY CGIBV CBUVI GMGKK FORZO MMVQA ALTBT TVCEY
      RXRRT QCTSG ZBZTZ GWARX YMVET LOAWT EWSTG AIQNI GXTPU DMDYT
      XTZJF OBWWI UFSGM AWTSO LSFBR GGMEZ JFARQ HYTTN YWGIF EKYDR
      WHXKE UARVW PYIEE ZEJXN VTEFP SCWPE VEQLB IITEH RFPVR QQIYV
      QYTRZ BQHVG TRVVF IGSFZ MZVTS WITWE ZBRVF MPZUF RBKHG QXODI
      EWXPJ LCFUH QTGQY TLANS GSRPD ZZRMM SGZNK MZJXD CAEEI FXKEV
      LMVVG STPGX DFIIE GGCOO KWGEE MUEAV UGSWS UZSAW TSLHG DTRAH
      TXVNT GRMKM LXGXT RUOIF QWTTR AEILI TGAJY HEGXQ LUKQY MSEEL
      OUMAY GSXLL VVGIH PCEEI IYILU WPRUW FSYVG FUDIZ MTPIT AEWTI
      FIQOA GMERT IUBUV ZQEJY GZBIQ TSNQQ FRZVU STWWM SKQGY BPQAO
      IZVGT VFZCF AGEGS FPCFZ EIKHC DEXCA HTTCC TVLNT TVVTD RNEIG
      XGLOD IEWXP JLCVA NVIIK IOKWC SKWGF LRLBG HRVTN LINVX GWAEI
      IEHTM PEEID RRVEQ NIICE KBGCB UVWIE BEHLS KIEHH WTPLM IQSLR
      QDPIW KMFSU XEJMF SIPCY TVTIS EXCLT IMVRT VJZJV LRQTR JL":
```

```
   C0:=SubstituteAll(Cr," ",""):
   Cr:=SubstituteAll(C0,convert([10],bytes),""): # 10 é o código
   ASCII para mudança de linha...
   lg:=length(Cr);
>
```

## Teste de Kasiski:

```
> for i from 3 to lg do
    for j from i+1 to lg do:
     if cat(Cr[i-2],Cr[i-1],Cr[i])= cat(Cr[j-2],Cr[j-1],Cr[j]) then
      bloco:=cat(Cr[i-2],Cr[i-1],Cr[i]):
       for k from 1 while(Cr[i+k]=Cr[j+k]) do
        bloco:=cat(bloco,Cr[i+k]):
       od:
      if length(bloco)>2 then
         printf("(%A,%A) %A %A\n",i-3,j-3,bloco,ifactor(j-i)): i:=
   i+k:
      fi:
     fi:
    od:
   od:
>
```

## Frequencia das letras e indice de coincidencia:

```
> Cr;
> C0:=Cr:
   m:=length(C0):
   ind:=0:
   for j from 0 to 25 do:
     c:=0:
      for i from 1 to m do:
        if C0[i]=convert([65+j],bytes) then c:=c+1: fi: # 97/65 -
   minusculas/maiusculas, respectivamente...
      od:
     printf("(%A) %A\n",convert([65+j],bytes),evalf(c/m)*100):
   ind:=ind+(c/m)^2:
   od:
   printf("indice de coincidencia = %A\n",evalf(ind)):
> interface(displayprecision=3):evalf(ind);
> ICportugues:=0.07813849:
   ICfrances:=0.0778:
   ICespanhol:=0.0775:
   ICalemao:=0.0762:
   ICitaliano:=0.0738:
   ICingles:= 0.0667: # 0.065 e' tambem citado...
   ICrusso:= 0.0529: # o alfabeto russo tem 30 caracteres...
   # Valor aproximado do comprimento da Chave...
   l:=evalf((ICportugues-1/26)*m/(ind*(m-1)+ICportugues-m/26));
> evalf((ICportugues-1/26)/(ind-1/26));
```

```
>
> # Calculo da chave...
  klg:=5: # comprimento da chave...
  iklg:=3: # introduzir valores 1, 2, ..., comprimento da chave
  Cb:="":
  for j from 0 while evalb(Cr[j*klg+iklg]<>"") do
    Cb:=cat(Cb,Cr[j*klg+iklg]):
  od:
  m:=length(Cb):
  for j from 0 to 25 do:
    c:=0:
      for k from 1 to m do:
        if Cb[k]=convert([65+j],bytes) then c:=c+1: fi: # 97/65 -
  minusculas/maiusculas, respectivamente...
      od:
    printf("(%A) %A\n",convert([65+j],bytes),evalf(c/m)*100):
  od:
>
>
```

# Descodificacao da cifra de Vigenere:

```
> Key:="CLARINETE":
  m:=length(Key):
  Dc:="":
  for i from 1 while evalb(Cr[i]<>"") do
   j:=i mod m:
    if j=0 then j:=m:fi:
  c:=convert(Cr[i],bytes)[1]-convert(Key[j],bytes)[1] mod 26 +65:
  #...97->minusculas;65->maiusculas
   Dc:=cat(Dc,convert([c],bytes)):
  od:
  printf("%A",Dc);
>
```

# Um exercicio

```
> # decifrar a seguinte mensagem (cifrada com a cifra de Vigenere)
  Cr:=
  "uszhmgvgldwnbygwezhdgrbkiswrpsvinnokxbkouseyefxrhngeyptwvnaujeo
  yakmylijespcgybyfsebirshblsuvvaojfnpafsrvrwxehtghbirswvsdgtbcozy
  zplviqvnwkevdsqhshwvflnkynsexiypzfefjivsnkerhrhggwgvdwqvsngzrjef
  xbzevsmlnsjngefhnhujmppdaeavajvnpadhrmejvnkakhvztjmgvdgqhuiumcpo
  vivaatyahjgvtljeeqvejebwraqbnefmgvdwnbhosqnkovishraerludeypadins
  aeeqvdwjnyisnbygwezhdgifaenirtfgvghlwdnlmemyuoniplnlsflngzrutsip
  pnuschrssyhnuezlnlsqlavifjotieaaventejmphpwpbztmvpvsfsplnlvbkeus
  acefgblsxmyhswrbymwwqllwmgvrwwflfgvzhrsqchrsgbuswkhprshrkiuegvra
```

eqvamxbyohyosiusflnlmnxuwefhuviqllwrnvejentekqnhahifhrvetyafhrsu
umqlzwgbusumrucaeqvswychpwpqlekgeptgvrkaxmetereqvswyfvcaeypsesha
ohmpvauvrzcahbkemqbaiemftouszvfmxhyovsoyakmyxuwhrpxgygyafwchrwgr
ynswfbakiaarwzvztswwvryiflmhvrhcjiqptgyavbjefplwrblsuvvaojfehsap
rprgefbagfehegvrarsxbkemqchikuhlswhvgmghryngqnzdwyzhdazrysahnkeu
yyaujeylngvzlfsgbjojsnzpspncrswqlfsfvvlmgnznszvzaggeptagnkotvnzi
dnbygwezhdgxehbspuhcgqnjofxehpgwvjaglvztgvvjavsoyafgbwejeaaegrrn
rgibmejiplcgqbyektbztstbzilmihoaqrusgifausvvvdsqvzcakruauebuekwr
hshipaogifjraxbybsmnuokizwrwespreshjoelhtidhnkeiyrhpwrnzotwryvsz
nhshifzoswruagxvuhsteltwrfhovirepdmphrswbjiwhnkeiynudgwrmadeahvs
wghotvnkosygvrdstvswgepamqnpmskrtdwqnnnaxhkevefwejwbuayiazfwqvui
fefhpjsoseeegpcshntudlrybgrvaakiazuspgyafwtyekwbyawmeyeiyvltsuhl
swifaatiylcwrbpmskvuajmbkoyvnudwthilagbseaxbypjmajiheytefxrhpsvg
prvithbjmrsauvncowgnuedezplfsilcwrgvswgvuqmiaaawsvaohevzdggnynsz
nsmapavvwgrutgwrararghemqrhekxelispvaejeepavsrzcjmgvriyrjaemaoom
tnyagvbtafgrkeahrvlgkvhpgpvaiuepvmsthilagnjaghrjauehtidrbceuiaao
kigyifxnltjifhgjeakeyyvuaveqhvahnldssoyasqnkisrnzevepvmgeohnvsav
dgtnytahbjoeyapslerkojszhnuiqlefknqaeiaaohsyptagbhotvnkebsenesqn
koheeaeheehuegnzaeiaaouszhajxrwrgtepaeiaaevmghekwrlnusaargislils
pvmteflnswclrksahgwrfmeemapnswntudlrypswfvuswryowtvjefxevdsrnyrs
xvcastnytavqhcjmnjaghrnatvvllsuhlpgwglrasetefxrvlwznyisepyisvqvn
sjyvrlielssfnaikxnltaighdwrgyovecyotprtalmphdgvbtafgrtovieuotvnz
idivyoyeoyiwpnaoeenzswrgvngqhudgjvjcasahlssyhdghrprsgrtakiaoojen
lsuvncaawnbrsgnwilypvnuivjagizbilsfvulvbzmaxbzqminmiugnvbjefplwm
ehcjmbbagpbugghbztwqcvsgwnzpwggvsdmepcgwbhulsetuaxbieeiazaaeehee
qnymgvgvmapavvwgrutgwrararghekivzqmeakouvvvudmipaeeflsgqrutwechr
lmekeyeoyiwpnxuwsqpmwrfpofezlnlsqvfwqvuifsflfabnuojszhnuintavmnu
owegyanifkekwnvbjedbegwcvslyyhdgwcvlaxvjokwnvpgwgvsviyhdgtnyavee
suyeehueerztjygbrsrnyrsxvcaeevzedeovrshnnatvvllsrbkiriekouvvaius
chudschekrnvdwmkhdwwrylavvjahefaojiyhpgvrtdwyzhlavvjaesqlrfeqlsj
szhnlmmhdstepnumchleiaaefsdbevmmyektrptgebhshipaoksppadebouesela
kegprsgbtpsvnyahvbaaysapslenbmstnztgvnlswqqbvahnwedephnvmqlziyrc
ikphtbjeqvaheelcaqrutghnwejwbuayizjoewrbpmieplvifsuefehmwrgvfwie
pcghbjijgbtuaxbjoeyzuoeyakovepbllyehpgthsajzrqaesfjoesnwajiplgsf
epedecllstepmwmehvwdqpafxrkefeppbnifaiveqltjecvsemflrszrpsusolrl
eqltsqnuhswhqeavnxuwiehietbzsazrsvwvyoeswsliusrzeveeshwmqhdwsfja
tiyvsvifnrwruhdgwvtufhbzdwtbvshifkekgnscgwavdwgbyrwvqhnsvehtaznh
pjsghggrvztsipvmheehdsehtaxpbyrgwnyutvnjoegulijsqlcjeivpwvsbmwhr
natvvllsgbtokipvnkxnaaggbuvarplnlieladmftowqppruyaztsrppavmilrke
qvlavvzmgtnztgvrsakirciviajisvnjoesqlswrivlnmzlnlsqhnsvehtaznlsw
viprshrjofxehslirutjibsijmpvegwrusmeyxuwephbjmghdgwrytssrmrmxbko
skelslirwonsahoksbhmgvqlnsgvipgvthbjmrsaliaaalvnusxsetadertueeqh
mshnzoumrkaviflmjifblleqvseefhcaqnkelyqvafeeyalmihcgrghazmfaojmn
kadmolrveqldsgepalyehhmqnuauszvswvhuiusrpnvippfjeillbeqpzaebjrax
vjolvvztssqlallnfdweupslsepavercodyphoksppadhrplzihznssrztshvzsg
gvhdshnoikxbyishnlvgphjagwbjispqlgsfepederuqmeaaoapulukwrnuwrnzu
sfepgstrsohsqlrsxehvwwqvcgrsyofxbkokgbyofivzehsyptagbzqmielpjifl
nlezvpswfhdgibmulyevgsfepedeflgmiypvjizlnlipvmgqbjakizmaemypawjv

```
shshbuavedbevifljseclnswbztsxhzdwezhnlintaverjojxrqaveflmiyrpsks
phslvrhsmefldwhrsitiekavipvmgyapcsjbycswvutwxvgavsehdgwrbivinsdw
zvkagxehtsqrutghnkossntojiznatvvllsgehvgiphnwpnzejzrzeehhcivechr
snhztajvjajsubmgvrvejsgpsescyekiaaekrnvbjebwrgkelsksqlidlrbskmtu
ixmphohvbnrwwfvdgxehtsqrutghnkohiyvhgqrtaeyyoejinvaeseuojszhnuib
zcswbzaesevsgwfholvnaavsfkeesqvektrjisppvmwgnudgtbyoxiapsaenwaab
buavecllgmzwejeqvrhefzafhbwedsevmsrplpswfpofeykekmaoarmaoawsftuf
hblvgphpnvspvmsxehiuebkeypbyisibwejhnvswqqbvahnlskidbavvbkeyeylr
aeqlmmpulrwwipsleflmjiyhcssnvaesezeuszwlwxnyausztadzvuaxmyoavipv
rgrrsqmielnwknvdwwgpnghrtouephsshbprshrzaxmnudgsfwrwgbucwmgvswxb
ynsrqvswehuiueclrksahgwqdbewpniojehtanmfhouvvaiueqlskefvcaiqhdwm
yoewrfldgwnuokzvutwzbstsrqvayeoyiwpnhpjsghggrvztstrymsrrjeuszvue
eclrksahgwqgllmvvjavivtefwbciysezefwhhlfeahrjegpvsuhlnssflsmfzlt
weapnyyrtnwqflefuhhdjeahvwpuhomrnuonevshwyfhpwwnydgwrzfgvpvsviah
cafrtqmielrlvnusxsetadenualyelzshrnatvvllsintaawnbtwrgpcshraovef
hshiezofetlnkhbyoeeajeweglnlegpvshrtuveajalseuasyzhmggnarawgvnze
rlnkmzlseeqhokxnaukwbjispnkqmmepdgtrsouefhmwrgvnssglmgqrzmgznsoj
iyhbgvnkohiyhmgvnsbmvtbekebxuwprcagxretgezhdaeavamqspnspslladdba
fhbhpwvfvnskrtdwmkhdwwrydsqnkaksppeveqlensyaaswrysaqcsekwvtbgpbz
epynsevrnjajpbzdweuvlsrqhpjsslsksehdsyapvwvfpdshrlsleqbadhbjesvn":
> lg:=length(Cr);
>
>
```

# Um Maplet para cifrar com a cifra de Vigenere...

```
> restart:
  with(StringTools):
> VigEnc:= proc(Mensagem,Chave)
    local m,j,i,c,k: global Criptograma:
      m:=length(Chave):
      Criptograma:="": j:=0: k:=1:
       for i from 1 while evalb(Mensagem[i]<>"") do
         if Mensagem[i]<>" " then
           j:=j+1: if j=m+1 then j:=1: fi:
           c:=convert(Mensagem[i],bytes)[1]+convert(Chave[j],
  bytes)[1]-97-97 mod 26 +65:
           Criptograma:=cat(Criptograma,convert([c],bytes)):
           k:=k+1: if k=6 then k:=1: Criptograma:=cat(Criptograma,
  " "): fi:
        fi:
      od:
    return Criptograma:
  end proc:
> with(Maplets[Elements]):
> maplet := Maplet(
              Window('title'="Cifra de Vigenere",
               BoxLayout(BoxColumn('halign'='none',
```

```
                ["Mensagem\n (entre aspas)\n (em minúculas):",
    TextBox['TF1'](5..50)],
                BoxRow("Chave (em minúculas): ",TextBox['TF2'](1.
    .5),HorizontalGlue(),HorizontalGlue()),
                ["Criptograma:",TextBox['TF3']('editable'=
    'false',5..50)],
                [Button['B1']("Cifrar", 'onclick'=
                                  Evaluate('TF3'='VigEnc(convert
    (TF1,string),convert(TF2,string))')
                               ),HorizontalGlue(),
                Button['B2']("OK", Shutdown())]
                       ))
                 )):
    Maplets[Display](maplet);
> 
>
```