暗号解読に挑戦！

休校などで時間に余裕のできた皆さん！

暗号解読に挑戦してみませんか？　元の英文を解読できるかな？

チャレンジ暗号文　c.txt

spwwz pgpcjzyp! hpwnzxp ez esp hzcwo zq nzxafepc dntpynp!
nzxafepc dntpynp td l mldtd zq esp xzopcy tyqzcxletzy lyo
nzxxfytnletzy epnsyzwzrj dfns ld mtr olel lylwjdtd, lt, lyo lwdz
nzxafepc rlxp. awpldp pyuzj esp hzcwo zq nzxafepc dntpynp. dpp
jzf lrlty le seea://end.n.etepns.ln.ua/ndmzzv/

これはシーザー暗号というローマ時代にも使われていたと言われている単純な暗号です．暗号の作り方と英文についての基礎知識があれば解読は難しくありません．

でも，面倒だなぁ

そうですよね！
そこでコンピュータでやりましょう！
プログラムで解読してみませんか？

暗号解読に挑戦！ プログラムで！

```
# hukugo.rb
# input: angobun (Caesar ango (k shift))
# output: hirabun (= moto no bun)

#===== koko ha kansuu (subroutine tomo iu) no teigi =====
def dec(k, c)
  code_a = 97
  nagasa = c.length
  a = c.unpack("C*")
  b = Array.new(nagasa)
  for  i in 0..(nagasa-1)
    sa = a[i] - code_a
    if 0 <= sa && sa <= 25
      b[i] = code_a + ((sa - k)%26)   # korede -k shift ga dekiru
    else
      b[i] = a[i]
    end
  end
  m = b.pack("C*")
  return m
end
#===== kokokara program hontai =====
angobun = gets.chomp
hirabun = dec(   0   , angobun)
puts(hirabun)
```

↑ あとで説明します

具体的には，

Ruby お手軽準備 を見ながら準備して

https://tcs.c.titech.ac.jp/cs/jyunbi.pdf

Ruby おためし を見ながらプログラムを実行してみればOK

https://tcs.c.titech.ac.jp/cs/try.pdf

その前にもうちょっと説明を！

補足説明

- シーザー暗号とは，文字をアルファベット上で k 文字シフトさせて作る暗号のこと．たとえば $k = 1$ のときは，

  abc ⇒ bcd　（注：逆に 1 シフトさせれば元に戻る）

  のようになる．このシフト数 k が秘密の鍵．これを知っている者同士が暗号をやり取りできるのだ．

- このプログラムは，暗号文を復号するためのもの．ここに，秘密のシフト数 k を入れれば復号できる．（この例では 0 が入っているのでシフトしない，つまり変化しない）

- あなたに課せられた挑戦は，この秘密のシフト数を当てること．それは英文の性質を使うとできる．しかも，調査プログラムも使えるよ．

```
# hukugo.rb
# input: angobun (Caesar ango (k shift))
# output: hirabun (= moto no bun)

#===== koko ha kansuu (subroutine tomo iu) no teigi =====
def dec(k, c)
  code_a = 97
  nagasa = c.length
  a = c.unpack("C*")
  b = Array.new(nagasa)
  for  i in 0..(nagasa-1)
    sa = a[i] - code_a
    if 0 <= sa && sa <= 25
      b[i] = code_a + ((sa - k)%26) # korede -k shift ga dekiru
    else
      b[i] = a[i]
    end
  end
  m = b.pack("C*")
  return m
end
#===== kokokara program hontai =====
angobun = gets.chomp
hirabun = dec( 0 , angobun)
puts(hirabun)
```

復号用関数 dec の定義

プログラム本体

hukugo.rb

参考リンク

- 英文の法則については，「踊る人形」（シャーロックホームズの帰還，コナン・ドイル）をお勧めします． https://221b.jp/h/danc.html

- ここで使っているプログラムの言葉 Ruby 語を使う準備は

  Ruby お手軽準備 https://tcs.c.titech.ac.jp/cs/jyunbi.pdf

- それを動かしてみるには

  Ruby おためし https://tcs.c.titech.ac.jp/cs/try.pdf

- プログラム（計算）の仕組みは youtube の動画をどうぞ！

  https://www.youtube.com/watch?v=9NUbh6N0ETg

- さらに，勉強したことで「単位」をもらいたければオンラインコースがあります． edX

  https://www.edx.org/course/introduction-to-computer-science-and-programming-3