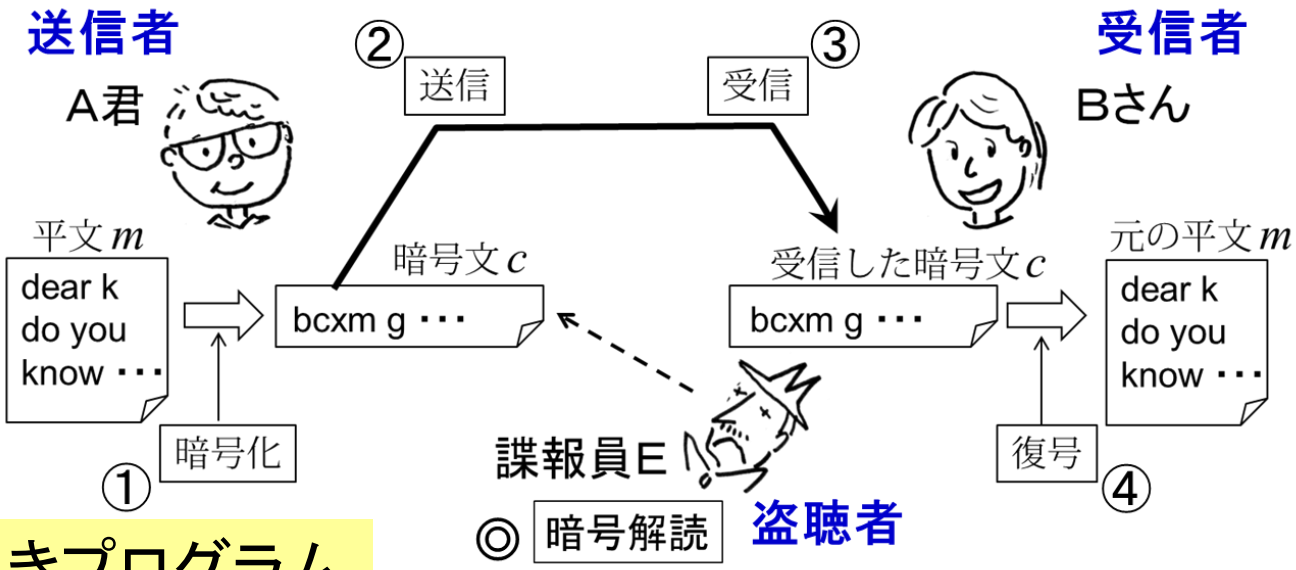
CS第1　レポート課題3

コンピュータ・サイエンス第1
クラスCS1b　担当：渡辺
2017.11.20

課題　暗号解読に挑戦



作成すべきプログラム

1. 暗号化プログラム ango.rb
2. 復号プログラム hukugo.rb
3. 暗号解読プログラム kaidoku.rb

（オプショナル）

4 (a) 自分流の暗号方式の myango.rb, myhukugo.rb もしくは
　(b) チャレンジ暗号を解読するための mykaidoku.rb

CS第1　レポート課題3

提出:11月27日
試験中に集めます

レポートの内容と採点基準(満点 25)

(数字は配点:加点 15 まで)

1. 暗号解読プログラムの使い方の説明

2. 暗号解読プログラムの仕組みの説明(15 ＋加点)
   工夫した点も書くこと(これは別途**加点項目**となるので重要)

3. オプショナル(加点)
   - 自分独自の暗号方式の提案と暗号化，復号プログラムの説明
   - チャレンジ暗号の解読(そのための補助のプログラムの説明)

付録:暗号化，復号，暗号解読のプログラムコード(あれば OK，10)

採点者は，解読の考え方や計算法は
知らないと想定して説明すること

採点者はプログラムは読みません！
レポートに書かれたことのみで採点します！

発展課題（オプショナル）について

発展課題として以下もどうぞ.

(a) 自分流の暗号方式の myango.rb, myhukugo.rb

・自分流の方式の提案
・暗号化や復号の方法の説明（工夫点など）
・付録でプログラムを印刷したものを付ける

(b) チャレンジ暗号を解読するための mykaidoku.rb

・チャレンジ暗号 angobunX.txt の解読文
・どうやって解読したかの説明
注）適切な説明があればプログラムを使わなくてもOK.
　また，プログラムを道具として使って解読したら点が高い.
・そのために使ったプログラム mykaidoku.rb の説明

CS第1　演習ガイド

本日の予定

1. 準備
2. ango.rb, hukugo.rb の復習
3. kaidoku.rb のアイデア
4. kaidoku.rb の作成など

1. 準備

1. ログインする.
2. Terminal を動かす（TSUBAME と直接対話する窓口）.
   2.1. cd kadai3　課題2の部屋（フォルダ）へ.
   2.2. angobunX.txt を共通のお部屋から kadai3 へ
   コピーしておく。

共通ファイルの置き場所：
Desktop/shared/CS/2017/cs1-1b/kadai3

2. 暗号化，復号プログラムの復習

ango.rb, hukugo.rb の作成

2. これを参考に，ango.rb, hukugo.rb を完成させよう.

```
# dec(秘密鍵 k, 暗号文 c) = 平文 m)
def dec(k, c)
  code_a = 97              # 文字 a の文字コード
  kosu = 26                # 英字アルファベットの数
  leng = c.length          # 文字列の長さ
  a = c.unpack("C*")       # 文字列から文字コードの配列へ変換
  b = Array.new(leng)      # 暗号文（のコード）格納用配列
  for i in 0..leng-1
      code = a[i]                # i 文字目のコードを得る
      sa = code - code_a         # 文字 a からの差分

      b[i] =
  end
  m = b.pack("C*")           # コードの配列を文字列に直す
  return(m)
end
# サブルーチン dec（終）
```

ここを作る

使い方

```
k = 3
angobun = gets.chomp          # 暗号文を入力
hirabun = dec(k, angobun)     # 平文に変換
puts(hirabun)                 # 平文を出力
```

2. 暗号化，復号プログラムの作成

3. 作った ango.rb, hukugo.rb の使い方

```
$ ruby ango.rb
  Hello, love you!
  Hhoor, oryh brx!
$
```

m.txt

Hello, love you!

前もって安全なところで
作っておく

```
$ ruby ango.rb < m.txt
  Hhoor, oryh brx!
$
```

<u>Terminal 上での使い方</u>

- 入力データをファイルから読み込む
  ruby ango.rb < ファイル名
- 出力をファイルに書き出す
  ruby hukugo.rb > ファイル名

※ 読み込んで書き出すことも可能

ruby ango.rb < hirabun.txt > angobun.txt

3. 解読プログラムのアイデア

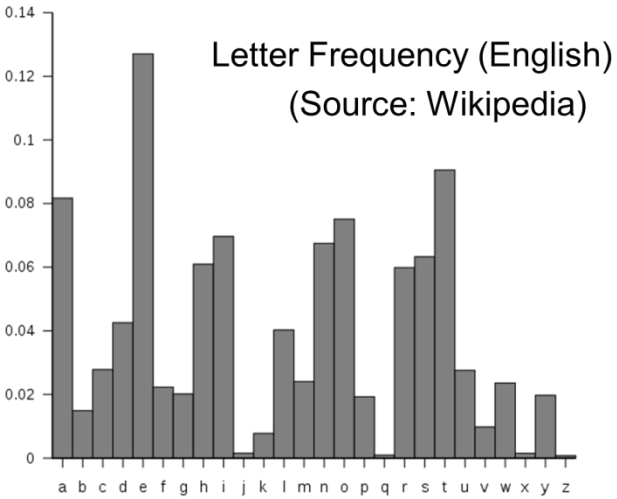
解読 ⟷ 秘密鍵を知らない者が暗号文から平文を得ること

比較的長い英文を暗号化したものを解読したい
どうすればよいか？

明らかだよ
ワトソン君

英語の場合

一番多く現れる文字が e のはず！



qxuv**n**b qjm k**nn**w b**n**jc**n**m oxa bxv**n** qxdab rw bru**n**wl**n** frcq qrb
uxwp, cqrw kjlt ldae**n**m xe**n**a j lqnvrlju e**n**bb**n**u rw fqrlq q**n**
fjb ka**n**frwp j yjacrldujauh vjuxmxaxdb yaxmdlc. qrb q**n**jm
fjb bdwt dyxw qrb ka**n**jbc, jwm q**n** uxxt**n**m oaxv vh yxrwc xo ...

n が19回出現で最多

qxuv**n**b qjm k**nn**w b**n**jc**n**m oxa bxv**n** qxdab rw bru**n**wl**n** frcq qrb
uxwp, cqrw kjlt ldae**n**m xe**n**a j lqnvrlju e**n**bb**n**u rw fqrlq q**n**
fjb ka**n**frwp j yjacrldujauh vjuxmxaxdb yaxmdlc. qrb q**n**jm
fjb bdwt dyxw qrb ka**n**jbc, jwm q**n** uxxt**n**m oaxv vh yxrwc xo ...

差分

n が19回出現で最多

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

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
													19												

13－4 = 9 だけずれた ⇒ k = 9

頻度配列と呼ぼう

注意！ maxj < 4 のときも大丈夫！？

アイデア

1. 頻度配列 hindo を作る.
2. 最大頻度の場所 maxj を見つける.
3. k = maxj － 4 で求め, dec(k, angobun) で平文を求める.