CS第1　課題2　演習ガイド

コンピュータ・サイエンス第1
クラス：xx　担当：XX
2016.11.xx

本日の予定

1. 準備
2. ango.rb, hukugo.rb の作成
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. 必要なファイルを kadai2 の部屋へコピーする
3. Terminal を動かす（TSUBAME と直接対話する窓口）
   3.1. cd kadai2　　kadai2 と言う部屋に入る

共通ファイルの置き場所：Desktop/shared/cs/cs00/kadai2b

2. 暗号化，復号プログラムの作成

コンピュータ・サイエンス第1
クラス： xx　担当： XX
2016.11.xx

ango.rb, hukugo.rb の作成

1. まずは，コピーしてきた復習用の code.rb を実行してみよう.

```
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

bun = gets.chomp         # 入力文字列から改行を除去
cc = bun.unpack("C*")    # 文字列懼・文字コードの配列
leng = bun.length        # 文字列の長さ

for i in 0..leng-1
  moji = bun[i]          # bun の i 文字目を得る (i は 0から始まる)
  code = cc[i]           # その文字のコードを得る
  sa = code - code_a  # 文字 a との差分
  if  0 <= sa && sa < kosu                    # 小文字アルファベットなら
     print(moji, ": ", code, ", ", sa, "¥n")  #     差分まで表示する
  else                                        # そうでないときは
     print(moji, ":", code,  "¥n")            #     差分は表示しない
  end
end
```

2. これを参考に，ango.rb, hukugo.rb を完成させよう.

2. 暗号化，復号プログラムの作成

コンピュータ・サイエンス第1
クラス： xx　担当： XX
2016.11.xx

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.

```
# enc(秘密鍵 k, 平文 m) = 暗号文 c
def enc(k, m)
  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 = m.length        # 文字列の長さ
  a = m.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
  c = b.pack("C*")         # コードの配列を文字列に直す
  return(c)
end
# サブルーチン enc（終）
```

ここを作る

使い方

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

2. 暗号化，復号プログラムの作成

コンピュータ・サイエンス第1
クラス： xx　担当： XX
2016.11.xx

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

Terminal 上での使い方

- 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

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

3. 解読プログラムのアイデア

コンピュータ・サイエンス第1
クラス： xx　担当： XX
2016.11.xx

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

宿題：考えてきて下さい

英語の場合

一番多く現れる文字が e のはず！

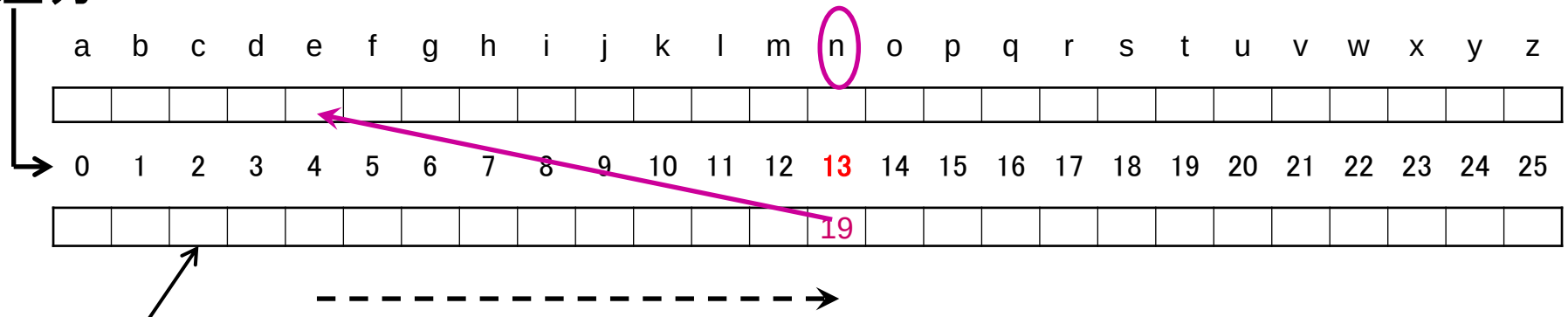
qxuv**n**b qjm k**nn**w b**n**jc**n**m oxa bxv**n** qxdab rw bru**n**wl**n** frcq qrb
uxwp, cqrw kjlt ldae**n**m xe**n**a j lqnvrlju e**n**bb**n**u rw fqrlq q**n**
fjb ka**n**frwp j yjacrldujauh vjuxmxaxdb yaxmdlc. qrb q**n**jm
fjb bdwt dyxw qrb ka**n**jbc, jwm q**n** uxxt**n**m oaxv vh yxrwc xo ...

n が19回出現で最多

qxuvnb qjm knnnw bnjcnm oxa bxvn qxdab rw brunwln frcq qrb
uxwp, cqrw kjlt ldaenm xena j lqnvrlju enbbnu rw fqrlq qn
fjb kanfrwp j yjacrldujauh vjuxmxaxdb yaxmdlc. qrb qnjm
fjb bdwt dyxw qrb kanjbc, jwm qn uxxtnm oaxv vh yxrwc xo ...

差分

n が19回出現で最多



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.

まとめ Terminal 上のコマンド

コンピュータ・サイエンス第1
クラス： xx　担当： XX
2016.11.xx

命令	使用例	意味
mkdir	mkdir kadai2	kadai2 というフォルダ（部屋）を作る
cd	cd kadai2	kadai2 というお部屋に入る
	cd ..	上の（大きな）部屋に戻る
	cd ../..	上の上の部屋に戻る
ls	ls	その部屋にあるファイルを表示する
rm	rm foo.rb	foo.rb を消す（戻らないので注意）
リダイレクト <	ruby xx.rb < aa	xx.rb を実行．入力は aa から取り込む
リダイレクト >	ruby xx.rb > bb	xx.rb を実行．結果は bb へ出す