

電子計算機入門

池田, 大輔
九州大学附属図書館

<https://hdl.handle.net/2324/2841>

出版情報 : 2004
バージョン :
権利関係 :

電子計算機入門 第5回 (06/02)

池田 大輔

`z4id01in@cse.ec.kyushu-u.ac.jp`

附属図書館

目次

前回課題の回答例

今後の方針

ファイルの利用

ソート (並べかえ)

前回課題の解答例

ある正の整数 n が与えられたとき $\sum_{i=1}^n i$ を計算しなさい

前回課題の解答例

ある正の整数 n が与えられたとき $\sum_{i=1}^n i$ を計算しなさい

```
import sys
n = sys.argv[1] # 入力

i = 0 # カウンタ
sum = 0 # 現在までの合計値
while i <= n:
    sum = sum + i
    i = i + 1 # カウンタの更新
print sum # 結果を出力
```

■ 今後の方針：情報の加工や表現方法

出力の仕方は重要である

例：第1回レポートの場合

出力はある文字列の全ての部分文字列

入力を baa とすると

['b', 'ba', 'a', 'baa', 'aa'] # 重複なし

['a', 'aa', 'b', 'ba', 'baa'] # 辞書式順序

['a', 'b', 'aa', 'ba', 'baa'] # 長さ+辞書式

のいずれも正解だが...

よくある問題

最大共通部分文字列

複数の文字列に共通な部分文字列の中で**最長**のものを
見つける

頻出部分文字列

ゲノム配列の中で繰り返し何度も表われる部分配列を
探す問題

よくある問題

最大共通部分文字列

複数の文字列に共通な部分文字列の中で**最長**のものを
見つける

頻出部分文字列

ゲノム配列の中で繰り返し何度も表われる部分配列を
探す問題

必要な技術：今後の方針

文字列に応じて頻度を数える

単に数えるだけではダメ

それまで数えたすべての文字列と回数の対応表が必要

文字列中から部分文字列を検索する

配列の順序の並べ替え

ファイル中のデータを読み込む

引数 (arguments) からの入力では長い入力文字列や、
すでにあるデータを効率よく利用できない

⇒ プログラムからファイルを読み込み、入力とする

■ プログラムへのファイルの与え方

Pythonwin の引数 (Arguments) にファイル名を入力する
ファイルが Z フォルダにある場合

“Arguments” の欄に “Z:¥FILENAME” と入力する

Z フォルダの folder フォルダの下のファイルの場合は
“Z:¥folder¥FILENAME”

プログラムと読み込みたいファイルがある場所が同じ
場合は “FILENAME” のみでよい

`sys.argv[n]` にファイル名が入る

open () 関数

関数 `open (ファイル名, "r")`

ファイルにアクセスするためのモノ (オブジェクト) を返す

"r" については付録を参照

書式

```
f = open (sys.argv [1] , "r")
```

引数の最初に与えたファイルを開く

`open ()` が返したオブジェクトを変数 `f` に格納

以下では `f` を使ってファイルにアクセスする

ファイル読み込み

```
import sys  
f = open(sys.argv[1], "r")  
all = f.readlines()
```

ファイル読み込み

```
import sys
```

```
f = open(sys.argv[1], "r")
```

```
all = f.readlines()
```

オブジェクト (f) の後にピリオドをつけて、オブジェクトの機能を呼び出す

ファイル読み込み

```
import sys
```

```
f = open(sys.argv[1], "r")
```

```
all = f.readlines()
```

オブジェクト (f) の後にピリオドをつけて、オブジェクトの機能呼び出す

`readlines()` はファイルを全てを読み、1行を1つの要素(文字列)とする配列を返す

`all` にはファイル全ての内容が行ごとに代入される

実習：ファイル読み込み

引数で与えたファイルの各行を表示するプログラムを書き、そのプログラム自身を表示させよ

実習：ファイル読み込み

引数で与えたファイルの各行を表示するプログラムを書き、そのプログラム自身を表示させよ

```
import sys
f = open(sys.argv[1], "r")
all = f.readlines()
for line in all:
    print line # 各行を表示
```

プログラムを“Script.py” とすると、Arguments に Script.py と入力する

■ 実習：ファイル読み込み(2)

ファイルの各行に行番号をつけて表示するプログラムを書け

プログラムを“Script.py” とすると、Arguments に Script.py と入力する

実習：ファイル読み込み(2)

ファイルの各行に行番号をつけて表示するプログラムを書け

プログラムを“Script.py” とすると、Arguments に Script.py と入力する

```
import sys
f = open(sys.argv[1], "r")
i = 1 # 行番号のカウンタ
all = f.readlines()
for line in all:
    print i, ":", line # 各行を表示
```

配列に対する演算

注： s は配列

$\min(s)$, $\max(s) \Leftrightarrow s$ の最小と最大値

$s.sort()$ $\Leftrightarrow s$ の要素を辞書式順序に並べかえ

$s.reverse()$ $\Leftrightarrow s$ の要素を逆順に並べかえ

実習：配列要素の大小関係

以下のプログラムを入力し、実行してみてください

```
array = ["baa", "accc", "abc", "bab"]
print array.max(), array.min()
array.sort()
print array

array = [7, -1, 3, 0]
array.sort()
print array
```

課題(提出不要)：頻度カウント

あなたは、ある自動車メーカーから依頼を受け、街頭を走る車の台数を車種ごとに調べないといけません。

あなたは車に詳しく、実物を見ればその車種名は即座に分かるものとしてします。

問題

どのような紙を用意するか？

新たに一台の車を見たときに、どのような処理をすればよいか？

今日の課題

ファイルの中身を、行ごとに辞書式順序に並べかえて出力せよ

baa

accc

abc

bab

というファイルがあるとき、これを

abc

accc

baa

bab

と出力させること

付録：open () 関数

関数 open (ファイル名、モード)

モードは「読み込み ("r")」、「書き込み ("w")」、「追加 ("a")」などを指定する

書式

```
f = open('sample' , "r")
```

“sample” というファイルを開く

open () が返したオブジェクトを変数 *f* に格納