



## プログラミング演習 ソーティング(2)

---

阿萬裕久  
aman@cs.ehime-u.ac.jp



## ソーティング(前回の話題)

---

- データを何らかの順序で並べ替えること
  - 昇順(小さい順)
  - 降順(大きい順)

基本的なソーティングアルゴリズム  
である『**選択ソート**』を紹介した



## 今回のテーマ

---

- **数値以外**のデータのソーティングを考える

むしろこの方が現実的

- 例題1: 構造体(誕生日)
- 例題2: アルファベット
- 例題3: 文字列



## 例題1

---

- 5人の生年月日(年, 月, 日)を読み込み, 1人ずつ構造体 birthday の配列に格納しなさい。そして, その5人を生まれた順に並び替えて表示しなさい。

構造体  
birthday  
の宣言

```
struct birthday {  
    int year;  
    int month;  
    int day;  
};
```

## 例題1:解説

- 構造体データの大小比較

残念ながら  
< を使った  
比較は  
できない!

```
struct birthday taro;  
struct birthday jiro;  
taro.year = 1986;  
...  
if ( taro < jiro ){  
...  
}
```

taro や jiro の中には year, month, day の **3つの数値が含まれる**ので、**大小比較の基準が分からない!**

## 例題1:解説

- この場合、どちらが「若い」のかを判定する

この順にチェック?

- 生まれた年 (year) が大きい方が若い

```
if ( taro.year < jiro.year ){
```

- 生まれた月 (month) が大きい方が若い

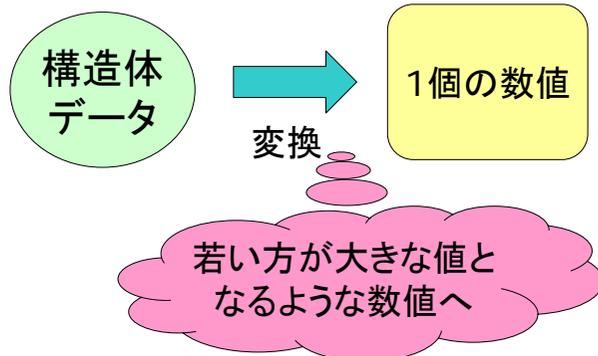
```
if ( taro.month < jiro.month ){
```

- 生まれた日 (day) が大きい方が若い

```
if ( taro.day < jiro.day ){
```

## 例題1:解説

- もっと簡単にならないか?



## 例題1:解説

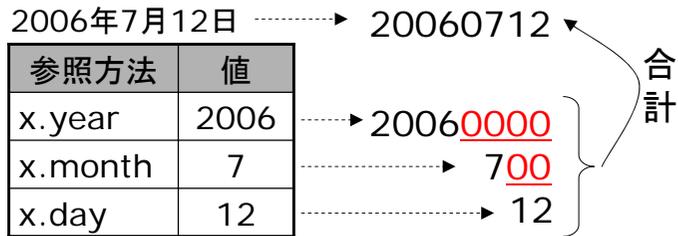
- 変換例

- 2006年7月12日 → 20060712
- 2006年1月1日 → 20060101
- 2005年12月31日 → 20051231

これなら日付が新しいほど大きな値になる

## 例題1: 解説

- 変数 x が誕生日(構造体)データの場合



## 例題1: 解説

Web ページのコーディング例を参照

- 年に 10000, 月に 100 をかけて足す

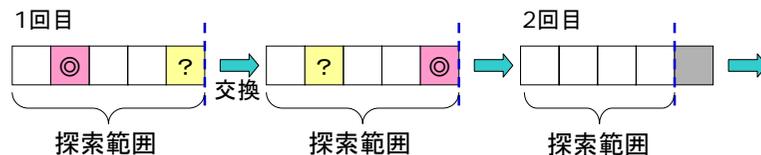
27行目

```
code_m
= a[m].year * 10000
+ a[m].month * 100
+ a[m].day;
```

※ x ではなく a[m] になっている点に注意  
※ code\_m は変換後の数値を意味する

## 例題1: 解説

- 「構造体→数値」の変換により、データの大小関係を評価できる
  - 後は、前回と同様にソーティング
  - 「若い順」=「数値の大きい順(降順)」なので、最小値に対応するデータを見つけ、右端と交換していく



## 例題2

- 5 つの英文字を順に読み込み、配列 a に格納しなさい。  
そして、それらをアルファベット順に並び替えて表示しなさい。ただし、使用する英文字は小文字に限定する。

## 例題2:解説

- やはり「**文字→数値**」の変換をやればよい
- しかし**実際には不要**
  - なぜなら, 文字(char 型データ)の正体は『**文字コード**』つまり『**整数**』

## 例題2:解説

- 文字コードはアルファベット順に割り振られている
- したがって  **$a < b < c < \dots < y < z$**
- 「アルファベット順」=「文字コードの昇順」  
単に数値を昇順(小さい順)にソーティングするのと全く同じ手順でOK

## 例題3

- 5つの文字列を読み込み, 入力順に配列 a へ格納しなさい. そして, それらを辞書式順に並び替えて出力しなさい.  
ただし, 文字列の入力は1行で1つとし, その長さは改行を含めて16文字未満とする.

## 例題3:解説(復習)

- 辞書式順
  - 英和辞書に載せるとした場合の順序のこと
  - 例)  
 **$a < ab < ac < acba < acbx < \dots$**

文字列の場合は **strcmp** 関数を使えばOK

## 例題3: 解説

### ○ strcmp 関数の復習

strcmp( str1, str2 ) の値

- 2 つが同じなら 0
- str1 の方が辞書式順で先なら負の数
- str1 の方が辞書式順で後なら正の数

## 例題3: 解説

Web ページのコーディング例を参照

### ○ 辞書式順にソートするための比較

22行目

```
if ( strcmp(a[m], a[i]) < 0 ){  
    . . . . .
```

これは a[m] の方が a[i] よりも  
辞書式順で先かどうかをチェックしている

※ a[m], a[i] は文字ではなく文字列

## 混乱しないように補足しておきます

※ a[m], a[i] は文字ではなく文字列

10行目 char a[5][16];

a は2次元配列  
要は「配列」の配列

「長さ 16 の配列」を  
要素とした配列(長さ5)

