

独自問題4.2の結果

```
k= 0  x=1.0000000000000000  L=3.0000000000000000
k= 1  x=0.517638090205041  L=3.105828541230249
k= 2  x=0.261052384440103  L=3.132628613281238
k= 3  x=0.130806258460286  L=3.139350203046867
k= 4  x=0.065438165643552  L=3.141031950890509
k= 5  x=0.032723463252974  L=3.141452472285462
k= 6  x=0.016362279207874  L=3.141557607911858
k= 7  x=0.008181208052470  L=3.141583892148318
k= 8  x=0.004090612582328  L=3.141590463228050
k= 9  x=0.002045307360677  L=3.141592105999271
k=10  x=0.001022653814027  L=3.141592516692156
.....
k=25  x=0.000000031208919  L=3.141592653589793
k=26  x=0.000000015604460  L=3.141592653589793
k=27  x=0.000000007802230  L=3.141592653589793
k=28  x=0.000000003901115  L=3.141592653589793
```

独自問題4.2の結果(参考の式)

```
k= 0  x=1.0000000000000000  L=3.0000000000000000
k= 1  x=0.517638090205042  L=3.105828541230250
k= 2  x=0.261052384440103  L=3.132628613281237
k= 3  x=0.130806258460286  L=3.139350203046872
k= 4  x=0.065438165643553  L=3.141031950890530
k= 5  x=0.032723463252972  L=3.141452472285344
k= 6  x=0.016362279207873  L=3.141557607911622
k= 7  x=0.008181208052471  L=3.141583892148936
k= 8  x=0.004090612582340  L=3.141590463236762
k= 9  x=0.002045307360705  L=3.141592106043048
k=10  x=0.001022653813994  L=3.141592516588155
.....
k=25  x=0.000000033320009  L=3.354101966249685
k=26  x=0.000000021073424  L=4.242640687119286
k=27  x=0.000000014901161  L=6.000000000000000
k=28  x=0.000000000000000  L=0.000000000000000
```

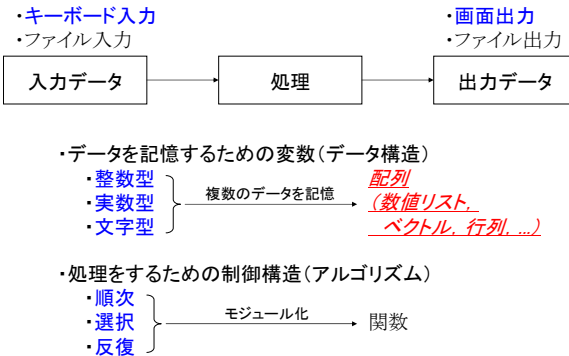
C言語を用いたプログラミング5 補足資料

- ・配列
 - ・1次元配列
 - ・2次元配列

関連項目

- ・内積, 行列計算(線形代数)
- ・並べ替え(統計処理)
- ・時系列データ(常微分方程式)

プログラム



添字(番号)付き変数

- ・統計処理などで多数のデータを扱う.

$$\text{平均 } \bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n}$$

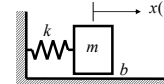
$$\text{最大値 } x_{\max} = \max \{x_1, x_2, \dots, x_n\} \quad \text{最小, 標準偏差, ...}$$

- ・数学や物理学などで多次元空間(二次元以上)を扱う.

$$\text{ベクトル } x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n \quad \text{行列 } A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \in \mathbb{R}^{m \times n}$$

- ・時間経過をシミュレートする.

時間	t_0	t_1	t_2	...	t_k
位置	x_0	x_1	x_2	...	x_k
速度	v_0	v_1	v_2	...	v_k



配列(添字付き変数)

	C言語	数式	
変数(基本データ型)	a	a	スカラー
1次元配列	a[i]	a_i	ベクトル
2次元配列	a[i][j]	a_{ij}	行列
3次元配列	a[i][j][k]	a_{ijk}	...
...	...	...	...

配列(添字付き変数)の特長

- ・多数の値を記憶できる
- ・添字(番号)で記憶場所を指定できる
- ・繰り返し処理が容易になる

1次元配列

$$x = (x_0, x_1, x_2, x_3, x_4) \in \mathbb{R}^5$$

意味	書式	例	備考
配列の宣言	型 配列名 [要素数]	int x[5];	要素番号の範囲に注意 [0] ~ [要素数-1]
配列の参照	配列名 [要素番号]	x[2] x[i] x[i*2+1]	要素番号は整数演算で指定できる

(用語) 宣言: メモリの確保, アクセス名の付加
参照: アクセス(値の記憶, 読み出し)

配列の反復処理

i=0, 1, 2, ..., 要素数-1

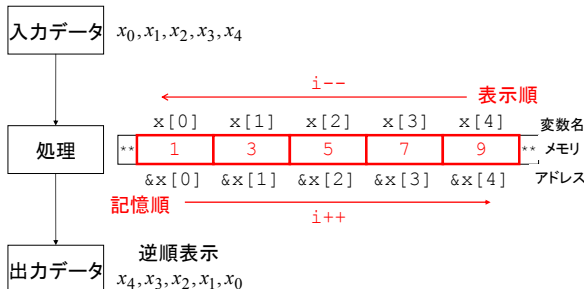
```
for (i=0; i<要素数; i++) {
    x[i]...
}
```

← 「要素数」と「不等号<」で範囲を指定できる

プログラム5.1

5個の整数を入力させ, 逆順に表示するプログラムを作成しなさい

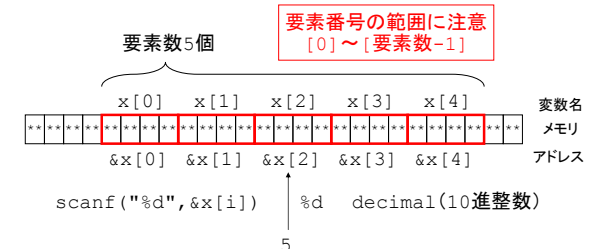
(特徴) 一旦, メモリへの記憶が必要



1次元配列(メモリ上の配置)

```
int x[5];
```

- ・int型 (4byte) × 5 = 20byte
- ・20byteの領域を確保し, 4byte毎に整数を記憶する.
- ・その領域を, 配列名「x」で参照(アクセス)する.



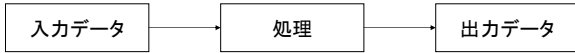
プログラム5.2

10

設計仕様 n 次元ベクトル x, y の内積 S_n を求めなさい。

$$x = (x_0, x_1, \dots, x_{n-1})^T \in \mathbb{R}^n$$

$$y = (y_0, y_1, \dots, y_{n-1})^T \in \mathbb{R}^n$$



$$S_n = (x, y) = \sum_{i=0}^{n-1} x_i y_i$$

アルゴリズム

初期値 $S_0 = 0$

反復処理 $S_{i+1} = S_i + x_i y_i \quad (i = 0, \dots, n-1)$

繰返し範囲

総和と同様のアルゴリズム

プログラム5.3

11

設計仕様 データを昇順に並べ替えなさい。

入力データ 9, 23, 5, 7, 43, 51, 47, 3, 24, 8

$a_0, a_1, \dots, a_{n-2}, a_{n-1}$

・最後尾を最大値に並べ替える。

$$M_n = \max\{a_0, a_1, \dots, a_{n-2}, a_{n-1}\}$$

$$M'_{n-1} = \max\{a'_0, a'_1, \dots, a'_{n-2}\} \leq M_n = a'_{n-1}$$

・範囲を縮小し、反復処理

・先頭から順に隣接2値を昇順に並べ替える

$$a''_0 \leq a''_1 \leq \dots \leq a''_{n-2} \leq a''_{n-1}$$

3, 5, 7, 8, 9, 23, 24, 43, 47, 51

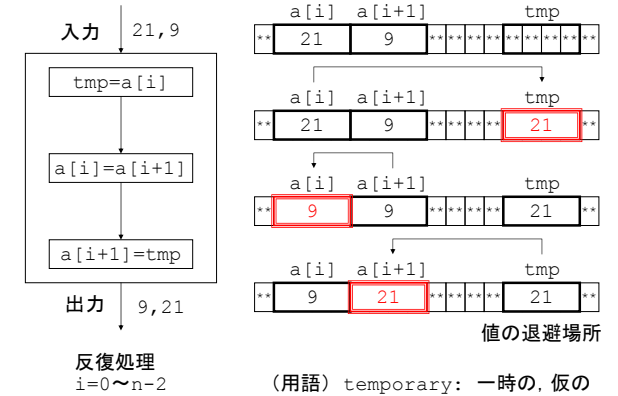
入力データ

処理
(昇順並べ替え)

出力データ

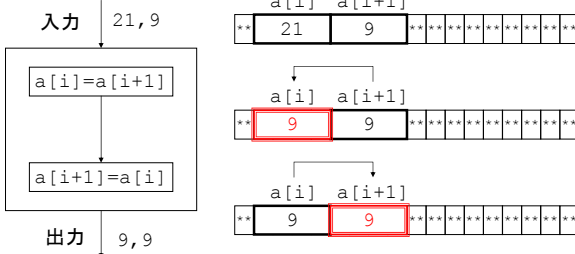
隣接2値の昇順並べ替え

12



値を一時退避しないと失敗

13



2次元配列

14

$$A = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \end{pmatrix} \in \mathbb{R}^{3 \times 4}$$

添字の範囲に注意
行番号: [0] ~ [行数-1]
列番号: [0] ~ [列数-1]

意味	書式	例
宣言	型 配列名 [行数] [列数]	double a[3][4];
参照	配列名 [行番号] [列番号]	a[1][2] a[i][j] a[i*2+1][j*3]

配列の
反復処理

```
for (i=0; i<行数; i++) {
    for (j=0; j<列数; j++) {
        a[i][j]...
    }
}
```

(用語)
行: row
列: column

プログラム5.4

15

設計仕様 行列サイズ3行4列の各要素に数値を入力させ、
行列を表示するプログラムを作成しなさい。

領域の確保

double a[ROW][COLUMN];

入力

要素の記憶

```
for (i=0; i<ROW; i++) {
    for (j=0; j<COLUMN; j++) {
        scanf(..., &a[i][j]);
    }
}
```

処理

```
for (i=0; i<ROW; i++) {
    for (j=0; j<COLUMN; j++) {
        printf(..., a[i][j]);
    }
}
```

出力

行列の表示

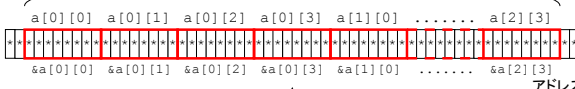
2次元配列(メモリ上の配置)

16

double a[3][4];

- ・double型(8byte) × 3 × 4 = 96byte
- ・96byteの領域を確保し、8byte毎に実数を記憶する。
- ・その領域を配列名「a」で参照(アクセス)する。

要素数3 × 4個



scanf("%lf", &a[i][j]);

%lf

Long Floating Point
(倍長浮動小数点)

4.0

プログラム5.5

17

設計仕様 3行4列の二つの行列A, Bの和をCに記憶し、
結果を表示するプログラムを作成しなさい。

入力 $A \in \mathbb{R}^{3 \times 4}, B \in \mathbb{R}^{3 \times 4}$

処理 $C = A + B$

$$c_{ij} = a_{ij} + b_{ij}$$

出力 $C \in \mathbb{R}^{3 \times 4}$