

## Lecture 9

プログラミング演習Ⅰ その9

# 本日の演習の流れ

配列について（続編）

course page

<http://amth.mind.meiji.ac.jp/courses/PE1/>

# 演習問題の目安時間について

1. 10分
2. 10分
3. 20分
4. 20分
5. 10分
6. 10分
7. 10分
8. 20分
9. 20分

# 定数式 (定数定義, #define)の復習

- 1 円の面積などを計算したいとき,  $\pi = 3.1415926536$  をいちいちプログラムの式の中で書くと長いので間違いのもとになる.  
このようなとき,

```
#define PI 3.1415926536
```

と定義(定数式, #define)しておき, 式の中では, PI を使用すれば, 間違いも少なくなり, 見た目にもすっきりする.  
以下のプログラムを作成せよ.

mainの前!



```
1 #include <stdio.h>
2 #define PI 3.1415926536
3 int main(void){
4     double x[10];
5     int i;
6     for(i=0; i<10; i++){
7         x[i]=PI;
8     }
9     for(i=0; i<10; i++){
10         printf("%lf\n", x[i]);
11     }
12     return 0;
13 }
```

最後にセミコロン';'はつけない!

# 配列のコピー

2 配列 a を配列 b にコピーするプログラムを作成せよ.

ヒント 配列は変数のような代入によるコピーはできない.

```
1 #include "everything.h"
2 #define N 4
3
4 int main(void){
5     int k;
6     double a[N] = {1.0,1.0,2.0,3.0};
7     double b[N];
8     b = a;           // no!
9     for(k=0;k<N;k++){
10         printf("%lf %lf\n",a[k],b[k]);
11     }
12     return 0;
13 }
```

# 配列の並べ替え

3 配列aの要素を並べ替え処理をするプログラムを作成せよ.

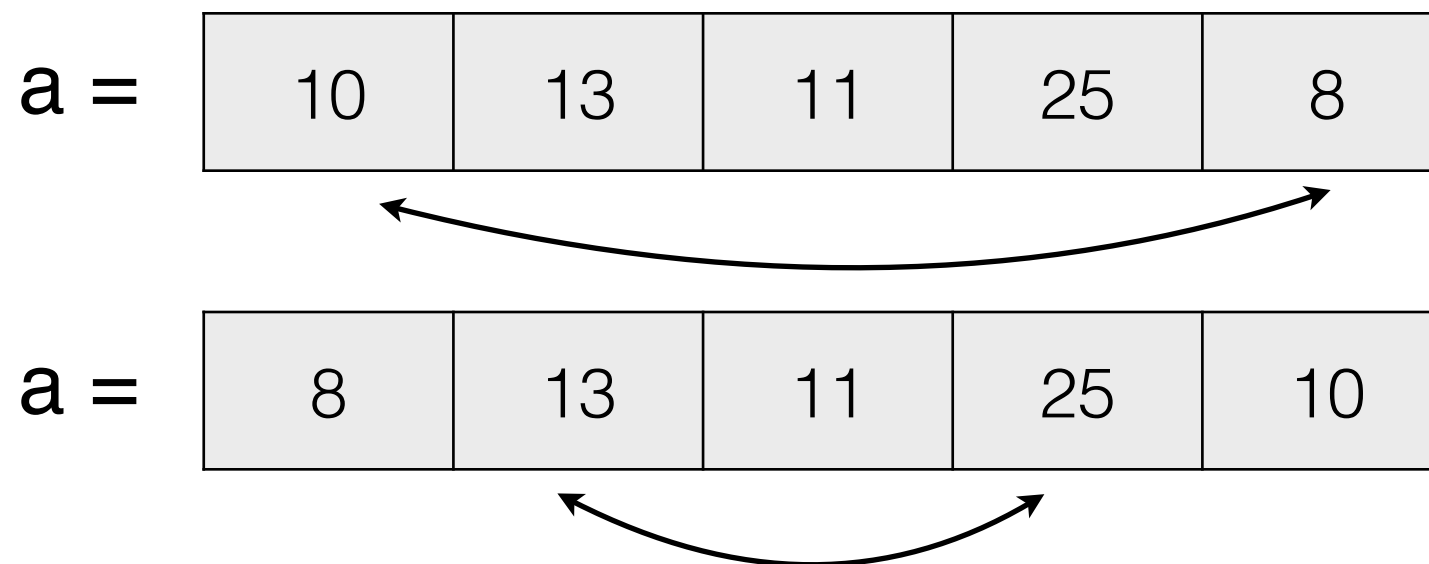
処理前 a = 

10	13	11	25	8
----	----	----	----	---

処理後 a = 

8	25	11	13	10
---	----	----	----	----

ヒント



ヒント while 文を使用すると考えやすい.

# 配列のソート (sorting of arrays)

ソート（整列, sort）は、アルゴリズムの定番課題である。

4 配列  $a$  を次とする。

$a =$	9	1	3	7	0	5	4	2	8	6
-------	---	---	---	---	---	---	---	---	---	---

$a$ を以下のように小さい値から大きな値に並べ替えるプログラムを作成せよ。

$a =$	0	1	2	3	4	5	6	7	8	9
-------	---	---	---	---	---	---	---	---	---	---

※ ソート方法は自由とする。

NOTE: 値を入れ替える操作をスワップ (swap) と呼ぶ。

# 配列のソート (sorting of arrays)

ヒント バブルソート (bubble sort)

バブルソートは隣り合う要素同士  $a[k]$ ,  $a[k+1]$  を比較し, もし  $a[k] > a[k+1]$  なら  $a[k]$  と  $a[k+1]$  の要素を入れ替える. これにより, 配列  $a$  の要素を昇順 (ascending) にソートするアルゴリズムになる. 疑似コードは以下の通りである.

```
for i = 1 to N {  
    for k = 0 to N-i {  
        if (  $a[k] > a[k+1]$  ){  
            k 番目の要素と k+1 番目の要素を入れ替える.  
        }  
    }  
}
```



# 配列のソート (sorting of arrays)

5 配列  $a$  を以下とする.

$a =$	9	1	3	7	0	5	4	2	8	6
-------	---	---	---	---	---	---	---	---	---	---

選択ソート (selection sort) は, 対象のグループから最小値・最大値をみつけ, それを順に並び替える手法である. 選択ソートは次のようなアルゴリズムとなる.

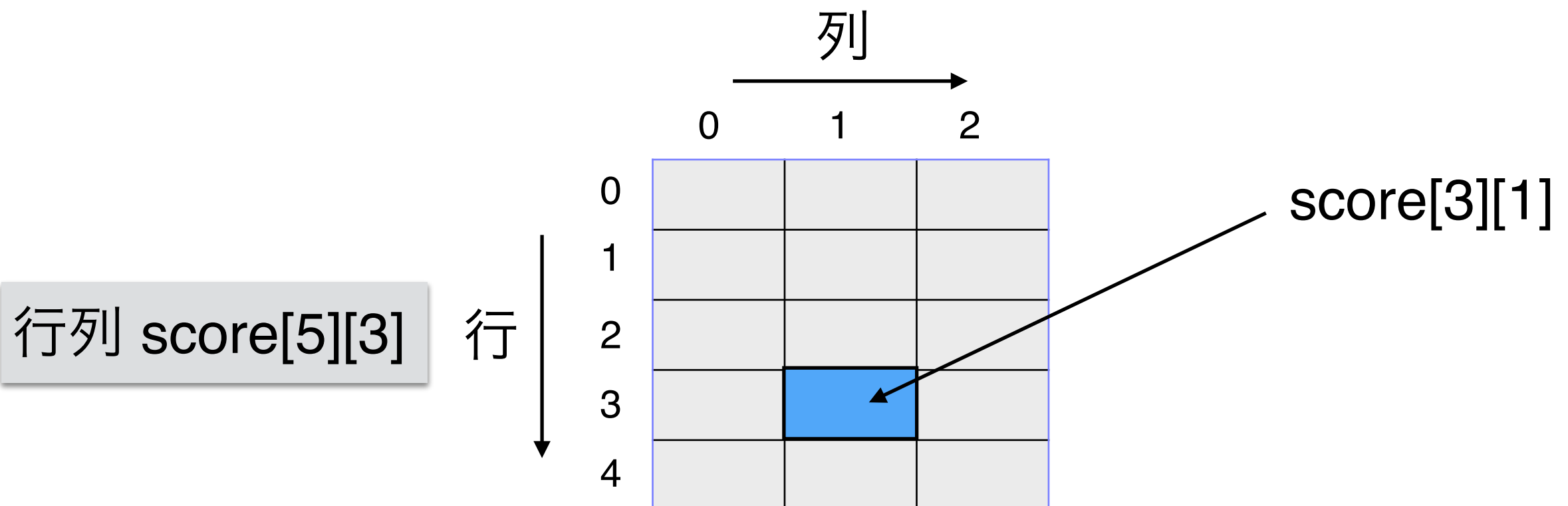
1.  $a[0] \sim a[n-1]$ の範囲で最小値をみつけ, それと $a[0]$ を交換する.
2.  $a[1] \sim a[n-1]$ の範囲で最小値をみつけ, それと $a[1]$ を交換する.
3. 以下同様に,  $a[i] \sim a[n-1]$ の範囲での最小値を,  $a[i]$ と交換する処理を $i=n-2$ まで繰り返す.

上記に従って, 選択ソートのプログラムを作成せよ.

# 多次元配列 (multidimensional arrays)

2次元配列（行列）をまとめて多次元配列とよぶ。多次元配列を使うことで、より複雑なデータ構造を実現することができる。

**EX** 5名の学生が3回テストを受けたとき、`score[5][3]`という2次元配列を準備すれば、配列名1つだけで、各人の平均点など様々な計算が可能になる。



**NOTE:** このほかにも、 $n$ 行 $\times$  $m$ 列の行列計算などにも多用される。

# 多次元配列 (multidimensional arrays)

二次元配列を使うときには、1次元配列の場合と同じく配列を宣言する。要素数の指定方法は一次元配列の場合と同じである。

## 2次元配列の宣言の構文

型 配列名 [要素数1][要素数2];

例) double A[2][2];

**NOTE:** [要素数 3]を追加することにより 3次元配列も使用できる

例) int A[3][3][3];

# 2次元配列へのデータの入出力

2次元配列にデータを入出力するときには、1次元配列と同じく for文を使うが、添字が2つあるので for文を2重にする。

6 以下を実行してみよう。

```
1 #include <stdio.h>
2
3 int main(void){
4     int i,j;
5     int score[3][2];
6     for(i=0;i<3;i++){
7         for(j=0;j<2;j++){
8             scanf("%d",&score[i][j]);
9         }
10    }
11    for(i=0;i<3;i++){
12        for(j=0;j<2;j++){
13            printf("%d ",score[i][j]);
14        }
15        printf("\n");
16    }
17    return 0;
18 }
```

← 多次元配列の宣言

← キーボードからデータを入力する

← 2次元配列scoreのデータを出力

← 改行

# 2次元配列の初期化

2次元配列のデータがあらかじめ分かっている場合には、宣言するときに、データを配列にセットすることができる。

7 次のプログラムを実行してみよう。

```
1 #include <stdio.h>
2
3 int main(void){
4     int i,j;
5     int a[3][2] = {{1,2},{3,4},{5,6}}; ← 多次元配列の宣言と初期化
6     for(i=0;i<3;i++){
7         for(j=0;j<2;j++){
8             printf("%d ",a[i][j]);
9         }
10        printf("\n");
11    }
12    return 0;
13 }
```

# 行列の計算

8 2つの行列  $A = \begin{bmatrix} 49 & 87 \\ 52 & 28 \end{bmatrix}$  と  $B = \begin{bmatrix} 86 & 98 \\ 75 & 74 \end{bmatrix}$  がある.

(1) 行列の和  $A + B$  を計算して表示するプログラムを作成せよ.

(2) 行列の積  $AB$  を計算して表示するプログラムを作成せよ.

ヒント  $(AB)_{ik} = \sum_j A_{ij} B_{jk}$

# 統計関数

- 9 double 型配列  $x, y$  (等しい大きさ) を引数で取り, 相関係数を計算する関数 `corr` を定義せよ. なお, 相関係数  $r$  とは,  $\bar{x}, \bar{y}$  を相加平均としたとき, 以下の通り定義される.

$$r = \frac{\sum_{i=0}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=0}^n (x_i - \bar{x})^2 \sum_{i=0}^n (y_i - \bar{y})^2}}$$

ただし,  $x, y$  は以下とする.

```
x double x[N] = {0.25,0.81,0.24,0.93,0.35,0.20,0.25,0.62,0.47,0.35};  
y double y[N] = {0.83,0.59,0.55,0.92,0.29,0.76,0.75,0.38,0.57,0.08};
```