

# 精密工学科プログラミング基礎Ⅱ

## 第3回資料

今回の授業で習得してほしいこと:

- 2次元配列の使い方  
(前回の1次元配列の復習もします.)
  - 宣言の仕方
  - アクセス (値の代入・参照) の方法
- 繰り返し文と配列の組み合わせの方法
- 配列を引数とする関数の作り方

資料のURL : <http://www.den.t.u-tokyo.ac.jp/prog>

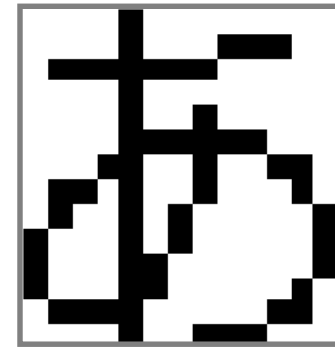
# 「2次元配列」とは？

- 配列を配列として並べたデータ
  - 行列や画像などを扱う時に便利

行列:  $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$

```
int A[2][3];  
A[0][0] = 1;  
A[0][1] = 2;  
A[0][2] = 3;  
A[1][0] = 4;  
A[1][1] = 5;  
A[1][2] = 6;
```

画像:



```
int g[14][13];  
g[0][0] = 1;  
g[0][1] = 1;  
...  
g[0][4] = 0;  
...
```

# 2次元配列の宣言と参照

- 宣言:

型 配列名 [縦の長さ] [横の長さ];

- 参照: (※インデックスは 0～長さ-1 まで)

配列名 [縦のインデックス] [横のインデックス]

例:  $3 \times 4$  の実行列を扱いたい:

$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 3 & 3 & 1 \end{pmatrix}$     宣言: `double A[3][4];`

$A[1][3]$

$A[2][2]$

# 標準入力からの読み込み

例: 正方行列を読み込んで, 表示する.

```
int i, j;  
double a[100][100]; ← とりあえず 100 × 100 個で十分  
int n; ← 実際に使う長さ  
  
scanf("%d", &n);  
for (i=0; i<n; i++) {  
    for (j=0; j<n; j++) {  
        scanf("%lf", &(a[i][j]));  
    }  
}  
  
for (i=0; i<n; i++) {  
    for (j=0; j<n; j++) {  
        printf("%lf ", a[i][j]);  
    }  
    printf("\n");  
}
```

読み込み

プリント

※ include と main は省略されています。

# 配列を引数とする関数

前ページと同じことをするプログラム

```
#include <stdio.h>
#define MAX 100
void scan(double A[MAX][MAX],
          int *pN);
void print(double A[MAX][MAX],
           int N);

int main() {
    int i, j;
    double a[MAX][MAX];
    int n;

    scan(a, &n);

    print(a, n);

    return 0;
}
```

変更があった時に便利

読み込み

長さは  
呼び出し先で  
代入するので  
アドレスを渡す

プリント

```
void scan(double A[MAX][MAX],
          int *pN) {
    int i, j;
    scanf("%d", pN);
    for(i=0; i<*pN; i++) {
        for(j=0; j<*pN; j++) {
            scanf("%lf", &(A[i][j]));
        }
    }
}
```

```
void print(double A[MAX][MAX],
           int N) {
    int i, j;
    for(i=0; i<N; i++) {
        for(j=0; j<N; j++) {
            printf("%lf ", A[i][j]);
        }
        printf("\n");
    }
}
```

# 課題 (前回の復習)

WEBページに置いてある雛形を基に  
以下のプログラムを完成させよ.

1. 2つの3次元ベクトルを入力して、  
それらの内積を計算せよ.

- Main 関数のみで済ませる版 : `product_main.c`
- サブ関数も使う版 : `product_func.c`

2. 整数の列を配列へ入力して、  
配列を昇順に並び替えよ.

- Main 関数のみで済ませる版 : `sort_main.c`
- サブ関数も使う版 : `sort_func.c`

# 課題 (1)

$N \times N$  の実行列を2次元配列へ入力した後、  
以下を行い結果を出力するプログラムを作成せよ。  
(必要ならばWEBの雛形とデータを利用して下さい)

1. 指定した2つの行を交換する.
2. 指定した2つの列を交換する.
3. 行と列を入れ替える(転置).
4. 指定した要素より右下の部分行列において、  
絶対値が最大となる要素を見つける.
5. 行の交換を繰り返し行い、  
**指定した列**の絶対値が降順に並ぶようにする.

## 課題 (2)

以下のプログラムを作成せよ.

1. パスカルの三角形を2次配列を用いて作成し、  
上下反転して表示せよ.
2.  $M \times N$  の実行列  $A$  と  $N$ 次元の縦ベクトル  $v$   
を入力し、掛け算  $Av$  の結果を表示.
3. 2つの  $N \times N$  の正方行列  $A, B$  の積  $AB$  を表示.
4.  $N \times N$  行列のグラムシュミットの直交化を行う.
  - 検算には転置行列との掛け算を行えばよい