

# 計算機プログラミング

## 第10回目

June 17, 2021

(10:30～12:00)



担当教員: Thi Thi Zin (ティティズイン)  
<thithi@cc.miyazaki-u.ac.jp>



# 本日の内容

---

- 配列

- 一次元配列(前回の内容について復習)
- 例題
- ベクトルの計算

- 列の計算(2次元配列)

# - 配列の復習 - 同じ型の変数をまとめて宣言

型 配列名 [要素数];

例: int tana [5];

tana [1]

tana [0] から tana [4] までしか存在しない  
tana [-1] や tana [5] は×

tana[0]

tana[1]

tana[2]

tana[3]

tana[4]

int 型

int 型

int 型

int 型

int 型

初期化:

int tana [5] = {10,11,12,13,14};

sample1 を変更しましょう。

```
#include<stdio.h>
int main(void)
{
    int i;
    int tana[5] = [10,11,12,13,14];

    for ( i=0 ; i<5 ; i++ )
    {
        printf("%d ¥n", tana[i]);
    }

    return 0;
}
```

forを使う  
場合

考え方  
(説明)

i=0

printf("%d ¥n", tana[0]); ⇒ 10

i=1

printf("%d ¥n", tana[1]); ⇒ 11

:

## 配列の復習

### ＜変数を使用＞

```
int a,b,c,d;
```

```
a = 10;
```

```
b = 11;
```

```
c = a + b;
```

```
d = a - b;
```

### ＜配列を使用＞

```
int tana[4];
```

```
tana[0] = 10;
```

```
tana[1] = 11;
```

```
tana[2] = tana[0] + tana[1];
```

```
tana[3] = tana[0] - tana[1];
```

※整数型の変数が使用可能

# 例題1

配列に予め10個の初期値を設定しておき、キーボードから入力された数値より小さい初期値は全て出力するプログラムを作成しましょう。

## 参考

```

:
{
    int tana[10] = {8, 22, 80, 5, 13, 42, 35, 3, 124, 2};
    int i, data;
    printf("input data:");
    scanf("%d:", &data);
    for(_____) {
        if(_____) {
            printf("%4d", _____);
        }
    }
    return 0;
}

```

## 解答例

```
# include <stdio.h>
int main (void)
{
    int tana[10] = {8, 22, 80, 5, 13, 42, 35, 3, 124, 2};
    int i, data;
    printf("input data: ");
    scanf("%d: ", &data);

    for(i=0; i<10; i++){
        if(data > tana[i]) {
            printf("%4d", tana[i]);
        }
    }
    return 0;
}
```

## 例題 2

int 型 配列 tana の7個の要素(1, 2, 3, 4, 5, 6, 7)の合計を求め、合計値を表示するプログラムを作成しましょう。

### 参考

```

:
{
    int tana[7] = [1, 2, 3, 4, 5, 6, 7];
    int i, s = 0;

    for ( i=0 ; i<7 ; i++ )
    {
        s = s + tana[i] ;
    }
    printf("合計: %d ¥n", s);

    return 0;
}
    
```



# ベクトルの計算を使う

ベクトルの計算は、配列を行うことでできます

$$a \times b = \begin{bmatrix} 10 & 11 & 12 \end{bmatrix} \begin{bmatrix} 13 \\ 14 \\ 15 \end{bmatrix} = c$$

```
int i;
int a[] = {10, 11, 12};
int b[] = {13, 14, 15};
int c = 0;

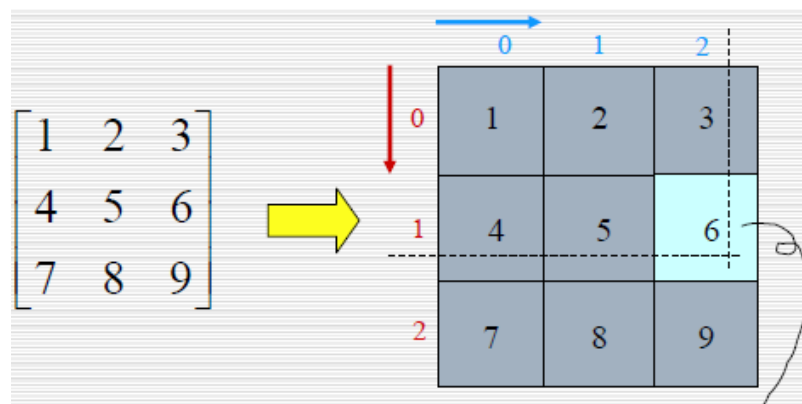
for(i=0; i<3; i++)
{
    c=c+a[i]*b[i];
}
```

# 行列の計算

行列の計算は、どのように実現できますか？

$$B \times b = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \begin{bmatrix} 10 \\ 11 \\ 12 \end{bmatrix}$$

1次元配列でやることもできるが、2次元なので、  
2次元配列を使いましょう。



$B[0][0] = 1, B[0][1] = 2, B[0][2] = 3,$

$B[1][0] = 4 \dots$

$B[1][2] = 6$

# 行列の計算

要素3の1次元配列が... 3つある

(これは省略不可)

```

int i, j;
int B[][3] = { {1,2,3}, {4,5,6}, {7,8,9} };
int a[] = {10,11,12};
int c[] = {0,0,0};

for(i=0; i<3; i++)
{
    for(j=0; j<3; j++)
    {
        c[i] += a[j] * B[i][j];
    }
}

printf("[%d, %d, %d]\n", c[0], c[1], c[2]);
    
```

Diagram illustrating the array structure and loop indices:

- The array `B` is a 2D array of type `int` with 3 columns. The first dimension is implicitly 3, indicated by the text "要素3の1次元配列が..." and "3つある".
- The array `a` is a 1D array of type `int` with 3 elements.
- The array `c` is a 1D array of type `int` with 3 elements.
- The loops iterate over `i` (0 to 2) and `j` (0 to 2).
- The expression `B[i][j]` is highlighted in yellow, indicating the element accessed during the calculation.

行列の掛け算

$$b = a x = \begin{bmatrix} 1.2 & 2.3 & 3.4 \\ 5.8 & 2.1 & -4.9 \\ 3.2 & 6.4 & 5.2 \end{bmatrix} \begin{Bmatrix} 3.5 \\ 2.1 \\ 8.8 \end{Bmatrix} \Rightarrow \begin{Bmatrix} b[1] \\ b[2] \\ b[3] \end{Bmatrix}$$

$$\begin{bmatrix} 1.2 & 2.3 & 3.4 \\ 5.8 & 2.1 & -4.9 \\ 3.2 & 6.4 & 5.2 \end{bmatrix} \begin{Bmatrix} 3.5 \\ 2.1 \\ 8.8 \end{Bmatrix}$$



$$b[1] = a[1,1] x[1] + a[1,2] x[2] + a[1,3] x[3]$$

$$\begin{bmatrix} 1.2 & 2.3 & 3.4 \\ 5.8 & 2.1 & -4.9 \\ 3.2 & 6.4 & 5.2 \end{bmatrix} \begin{Bmatrix} 3.5 \\ 2.1 \\ 8.8 \end{Bmatrix}$$



$$b[2] = a[2,1] x[1] + a[2,2] x[2] + a[2,3] x[3]$$

$$\begin{bmatrix} 1.2 & 2.3 & 3.4 \\ 5.8 & 2.1 & -4.9 \\ 3.2 & 6.4 & 5.2 \end{bmatrix} \begin{Bmatrix} 3.5 \\ 2.1 \\ 8.8 \end{Bmatrix}$$



$$b[3] = a[3,1] x[1] + a[3,2] x[2] + a[3,3] x[3]$$

How to multiply 2 matrices?

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 5 \end{bmatrix} \times \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \quad 1 \times 1 + 2 \times 2 = 1 + 4 = 5$$

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 5 \end{bmatrix} \times \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \quad 1 \times 2 + 2 \times 4 = 2 + 8 = 10$$

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 5 \end{bmatrix} \times \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \quad 2 \times 1 + 4 \times 2 = 2 + 8 = 10$$

そこで、青文字は外側の for loop (i を使う)

赤文字は内側の for loop (j を使う)

$a[i][j] * x[j]$  となる

# <課題 1>

右に表す2行3列 a, b の和を求めて表示する  
プログラムを作成せよ。

$$a = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad b = \begin{bmatrix} 6 & 3 & 4 \\ 5 & 1 & 2 \end{bmatrix}$$

<実行結果>

7	5	7
9	6	8

## <課題 2>

東京の平均温度を基に毎月の平均気温及び前の月との気温を表示するプログラムを考えてみましょう。

tana[ 0]	7.7
tana[ 1]	4.9
tana[ 2]	8.4
tana[ 3]	14.3
tana[ 4]	18.2
tana[ 5]	22.3
tana[ 6]	24.8
tana[ 7]	27.7
tana[ 8]	24.2
tana[ 9]	16.6
tana[10]	9.6
tana[11]	5.9

実行結果が以下のようにになるように、プログラムを作成すること。

<実行結果>

```
month  temperature(C)  change
1      7.7            1.8
2      4.9           -2.8
3      8.4            3.5
      ⋮
```

# <課題 3>

4×4の形の配列 a の内容を、4×4の形の配列 b に、  
下の図のように右に90度回転するようにして複写し、  
b の内容を表示するプログラムを作成せよ。

複写する配列 a の内容

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

複写後の配列 b の内容

13	9	5	1
14	10	6	2
15	11	7	3
16	12	8	4



# 本遠隔授業を受けるルール

- 課題の内容、小テスト、レポートの情報は評価に関わるので、他人に提供しないこと
- 遠隔授業の内容を勝手にSNS等で大学外第3者が観覧できるようにアップロードはしないでください。違法行為に当たる場合があります。

提出締切:6月19日(土)20:00 まで

## 出席条件

- 毎回、締め切りまでに問題の解答をレポートとして提出する
- PDFファイルで提出する
- レポートには日付、自分の名前、学籍番号を必ず記載すること