

# プログラミング練習 配列

瀧澤 宮腰

## ・オブジェクト形式マクロ

プログラム例 1 の下線部は、オブジェクト形式マクロを定義するための特殊な宣言であり、**#define** 指令と呼ばれます。

「**#define a b** //この指令以降の **a** を **b** に置換せよ」

という形をしており「この指令以降の **a** を **b** に置換せよ」という指示です。そのため、**a** が **b** に置換されたうえでプログラムが翻訳、実行されることになります。プログラム例 1 では、マクロ名（置換の対象になる **a**）は **NUMBER** であり、**NUMBER** が **5** に置換されます。

## ・メリット

値の管理を 1 か所に集約できる

定数に対して名前が与えられるため、プログラムが読みやすくなる。

※プログラムに直接 **5** や **8** などの定数を書くと何を表すための数値なのかわかりません。  
（この数をマジックナンバーと言います。）

## プログラム例 1

テストの合計点と平均点を表示するプログラム(.c ファイル)を以下に示します。

---

// 学生の点数を読み込んで合計点と平均点を表示（人数をマクロで定義）

```
#include <stdio.h>
```

```
#define NUMBER 5 //学生の人数
```

```
int main(void)
```

```
{
```

```
    int tensu[NUMBER]; //学生の点数
```

```
    int sum = 0; //合計点
```

```
    printf("%d 人の点数を入力せよ。\\n", NUMBER);
```

```
    for ( int i = 0; i < NUMBER; i++)
```

```
    {
```

```
        printf("%2d 番 :", i + 1);
```

```
        scanf("%d",&tensu[i]);
```

```
        sum += tensu[i];
```

```
    }
```

```
printf ("合計点 : %5d\n", sum);
printf ("平均点 : %5.1f\n", (double) sum / NUMBER);
return 0;
}
```

---

#### 練習問題①

学生の点数を読み込んで最高点と最低点を表示する。

以下の実行結果を示すプログラムを書いてみよう。(参考書 list5-11) (学生の点数は入力)

5 人の点数を入力せよ。

1 番 : 83

2 番 : 95

3 番 : 85

4 番 : 63

5 番 : 89

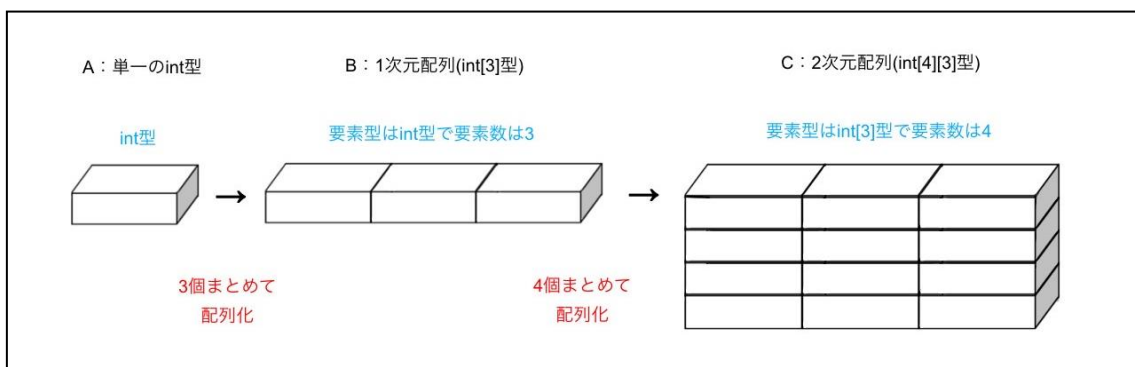
最高点 : 95

最低点 : 63

#### ・多次元配列とは

今まで学習した配列の要素は、`int` や `double` などの単一型でした。実は、配列の要素自体が『配列』である配列も作れます。

配列を要素型とするのが 2 次元配列であり、2 次元配列を要素型とするのが 3 次元配列です。もちろん、4 次元 5 次元、6 次元といった配列も作れます。2 次元以上の配列の総称が、多次元配列 (multidimensional array) です。とはいえ、言葉だけではイメージがわかりと思います。以下の図を見てください。



2 次元配列は、要素が縦横に並んで、行と列で構成される表のイメージです。そのため、図の配列は、「4 行 3 列の 2 次元配列」と呼ばれます。

1 次元配列と同様に、 多次元配列の全要素・全構成要素は記憶域上に直線状に連続して並びます。

構成要素の並びでは、まず末尾側の添字が順に 0、1、...と増えていき、それから先頭側の添字が 0、1、...と増えていく順番です。

↓こんな感じ

`a[0][0] a[0][1] a[0][2]`   `a[1][0] a[1][1] a[1][2]`   `a[3][0] a[3][1] a[3][2]`   ...

## プログラム例 2

テストの点数を表示するプログラム(.c ファイル)を以下に示します。

---

// 4 人の学生の 3 科目のテスト結果を表示

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int tensu1[4][3] = { { 91, 63, 78}, { 67, 72, 46}, { 89, 34, 53}, { 32, 54, 34} };
```

```
    int tensu2[4][3] = { { 97, 67, 82}, { 73, 43, 46}, { 97, 56, 21}, { 85, 46, 35} };
```

//1 回目の点数を表示

```
    puts("1 回目の点数");
```

```
    for ( int i = 0; i < 4; i++)          //4 人分の
```

```
    {
```

```
        for ( int j = 0; j < 3; j++)      //3 科目の
```

```
        printf("%4d", tensu1[i][j]);    //点数を表示
```

```
        putchar ('\n');
```

```
    }
```

//2 回目の点数を表示

```
    puts("2 回目の点数");
```

```
    for ( int i = 0; i < 4; i++)
```

```
    {
```

```
        for ( int j = 0; j < 3; j++)
```

```
        printf ("%4d", tensu2[i][j]);
```

```
        putchar ('\n');
```

```
    }
```

```
return 0;
}
```

---

### 練習問題②

テストの点数を表示するプログラムを参考に、4 人の学生の 3 科目のテスト 2 回分の合計 を求めて表示するプログラムを作成してください。

---

### 実行結果

#### 1 回目の点数

91	63	78
67	72	46
89	34	53
32	54	34

#### 2 回目の点数

97	67	82
73	43	46
97	56	21
85	46	35

#### 合計点

188	130	160
140	115	92
186	90	74
117	100	69

---

☆応用問題(余裕があったらやってみてください！)

3 行 2 列の行列  $X$  を読み込み、 $XT$  ( $X$  の転置行列) と  $XTX$  ( $X$  の転置行列と  $X$  の積) を求め、それぞれの結果を配列に代入すること。  $X$ ,  $XT$ ,  $XTX$  を出力すること。

### ---参考動画---

- ・【大学数学】線形代数入門②(行列)【線形代数】 <https://youtu.be/ltFI0FpLTzQ>
- ・ 転置行列の意味・重要な 7 つの性質と証明 | 高校数学の美しい物語 <https://manabitimes.jp/math/1046>

→後半の方は読むと頭がこんがらがると感じると思うので、最初の方だけ読んで転置行列がどのような行列なのか理解できればいいと思います。

行列は高校で学習していないと思うので、難しかったら 2 年生の先輩に聞いてください。  
きっと答えてくれるはず...！

---

実行例

行列  $X$

1 2

3 4

5 6

転置行列  $X^T$

1 3 5

2 4 6

転置行列  $X^T$  と行列  $X$  の積  $XTX$

35 44

44 56

---