

# 第11回

## 配列型の変数

情報処理演習 I , V  
(テキスト: 第4章, 第8章)

# 今日の内容

## 第2回C言語レポート課題の出題

1. 配列の必要性
2. 配列の宣言
3. 配列変数のイメージ
4. 配列変数を使用した例
5. 範囲を超えた添字を使うと？
6. 多次元配列変数
7. 多次元配列変数を使用した例
8. データのソーティング
9. 今日の練習問題

## 第2回C言語レポート課題

**課題:** 繰り返しと配列を利用したプログラムを作成する

**形式:** プログラムだけを提出するのではなく、プログラムの説明を書くこと。実行結果をキャプチャして貼付けること。

**問題:** テキスト pp.117~120 の練習問題から1つ以上を選んで答えなさい

**※切:** クラスによって案内がある

# 1. 配列の必要性

(テキスト31ページ)

## 多数のデータ処理

(要求) 10,000人分の成績データをもらって、合計点の大きい順に並べ替えて出力したい。

(方法) ひとまず、全部のデータを変数に入れておく。

(問題) しかし、`int d1,d2,d3,...,d10000;`のように、10,000個も変数を書けるか？

(解決策) 配列を使おう!!

```
int d[10000];
```

と書くだけで、`d[0]~d[9999]` の 10,000 個の変数を使うことができる。

## 2. 配列の宣言

(テキスト32ページ)

- 一般の変数を宣言する場所で配列も宣言する.

### ■ 宣言の一般形

型 配列名 [配列の要素数];

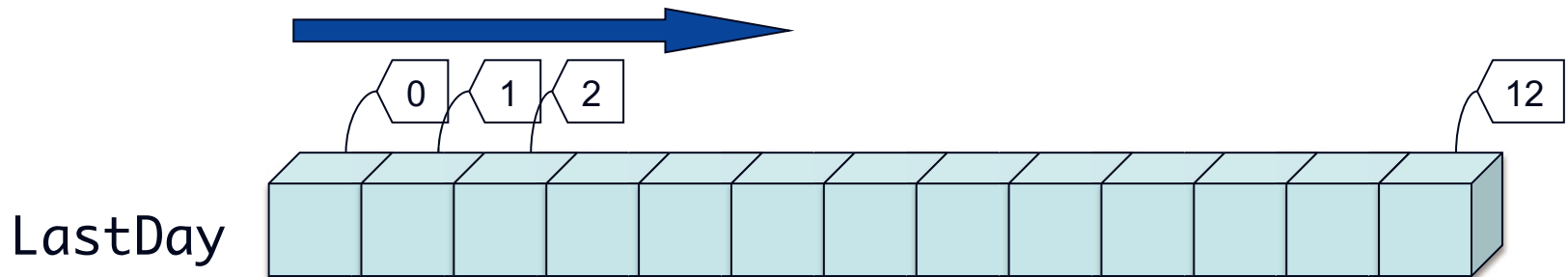
- 配列名は変数名と同様, アルファベット・数字・記号からなる.
  - ◆ 最初の文字はアルファベットであること.
  - ◆ 記号は一部しか使用できない.
- 型は構成要素の型を表す.

### 3. 配列変数のイメージ

```
int LastDay[13];
```

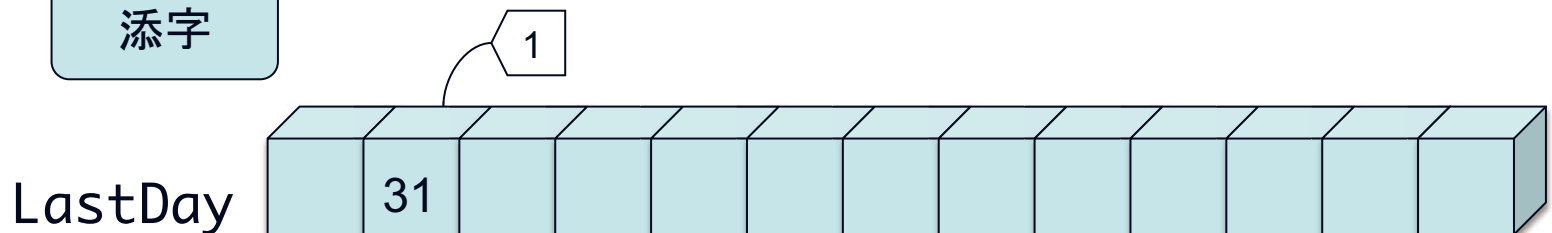
13個の箱が並んでいると考える。

先頭から順番に 0, 1, 2, …12 と添字が付く。



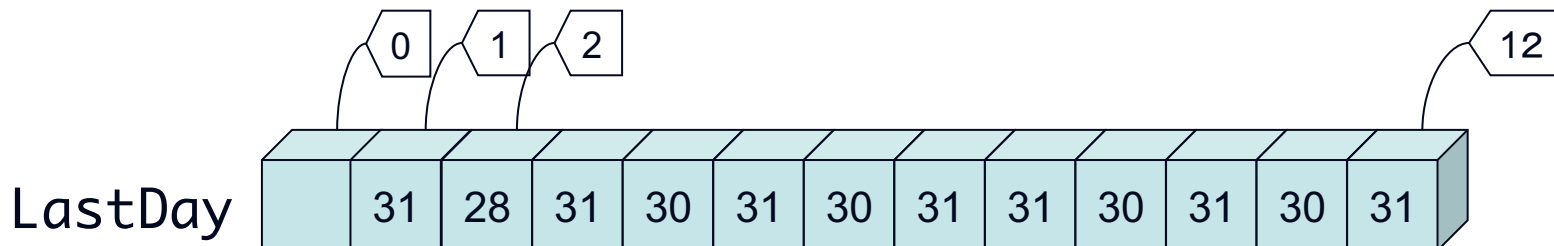
```
LastDay[1] = 31; ← 代入式の文
```

添字



# 配列変数へのデータの代入と参照

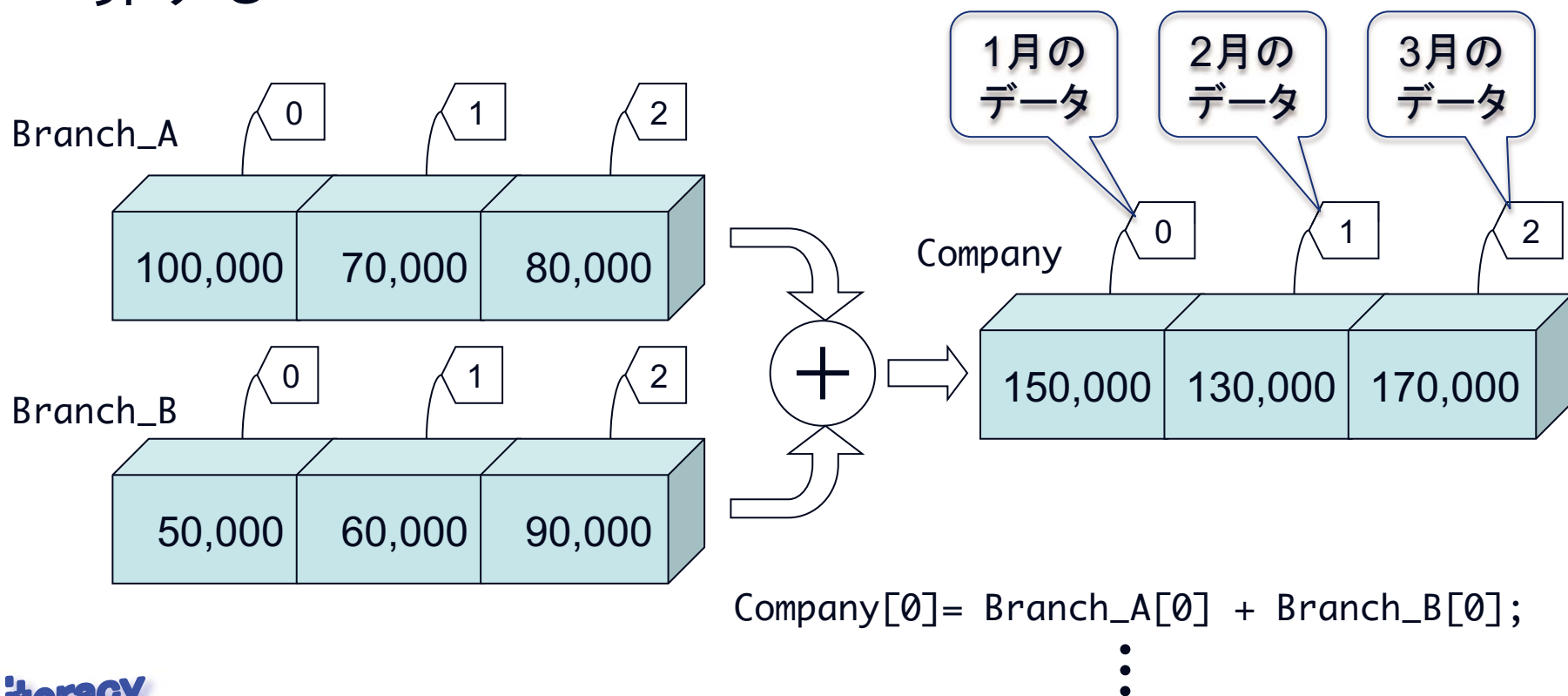
```
int main(int argc, const char * argv[]) {  
    int LastDay[13];  
    int month;  
  
    LastDay[1]=31;    LastDay[2]=28;    LastDay[3]=31;  
    LastDay[4]=30;    LastDay[5]=31;    LastDay[6]=30;  
    LastDay[7]=31;    LastDay[8]=31;    LastDay[9]=30;  
    LastDay[10]=31;   LastDay[11]=30;   LastDay[12]=31;  
  
    printf("月の日数を答えます. 何月を知りたい?");  
    scanf("%d", &month);  
    printf("%d月は%d日です\n", month, LastDay[month]);  
}
```



## 4. 配列を使用した例

(テキスト101ページ)

- ある会社に2つの支店(A支店とB支店)があるとする。
- 各支店で1月～3月の各月の売上額が下の表のようなデータとしてある場合, この会社の各月の売上額を計算する。





# プログラム例

```
int main(int argc, const char * argv[]) {
    int Branch_A[3], Branch_B[3], Company[3];
    int month;

    Branch_A[0] = 100000;  Branch_A[1] = 70000;
    Branch_A[2] = 80000;  Branch_B[0] = 50000;
    Branch_B[1] = 60000;  Branch_B[2] = 90000;

    for(month = 0; month < 3; month++){
        // 各月で計算
        Company[month] = Branch_A[month] + Branch_B[month];
    }
    printf("この会社の売上額");
    for(month = 0; month < 3; month++){
        // 各月で売上額を表示
        printf("%d月は%d円", month + 1, Company[month]);
    }
}
```

## 配列を使用した例(2)

(テキスト103ページ)

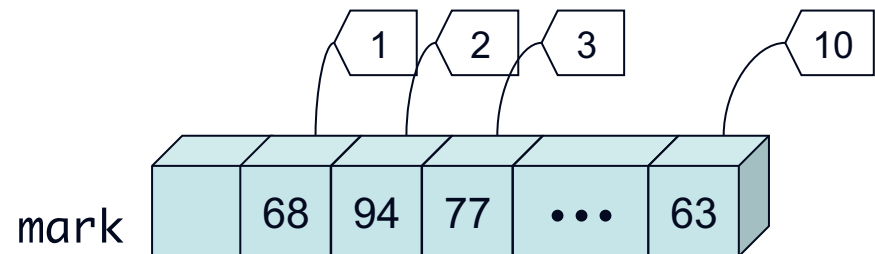
- テストの成績データがあったとして各学生の偏差値を計算するプログラムを作成してみよう。

学生番号	1	2	3	4	5	6	7	8	9	10
成績	68	94	77	52	73	85	79	71	86	63

出力例

学生番号	成績	偏差値
1	68	??
2	94	??
3	77	??
	⋮	
10	63	??

上の表は下のような配列変数に格納されたデータと考えることができる。



# 偏差値の計算

- 偏差値は下に示す式で求まる。

$$\text{学生}i\text{の偏差値} = \frac{(\text{学生}i\text{の得点} - \text{平均}) \times 10}{\text{標準偏差}} + 50$$

- 平均と標準偏差を求める。

$$\text{平均} = \frac{\text{個々の得点の総和}}{\text{学生数}} = \frac{d_1 + d_2 + d_3 + \dots + d_{10}}{10} = \frac{1}{10} \sum_{i=1}^{10} d_i$$

$$\text{標準偏差} = \sqrt{\frac{(\text{個々の得点} - \text{平均})^2\text{の総和}}{\text{学生数}}} = \sqrt{\frac{1}{10} \sum_{i=1}^{10} (d_i - m)^2}$$

- 総和を求める計算

- 繰り返し制御構造を使う
- 右のプログラム片で計算できる  
ループ終了後, 変数waの値が総和

プログラム片

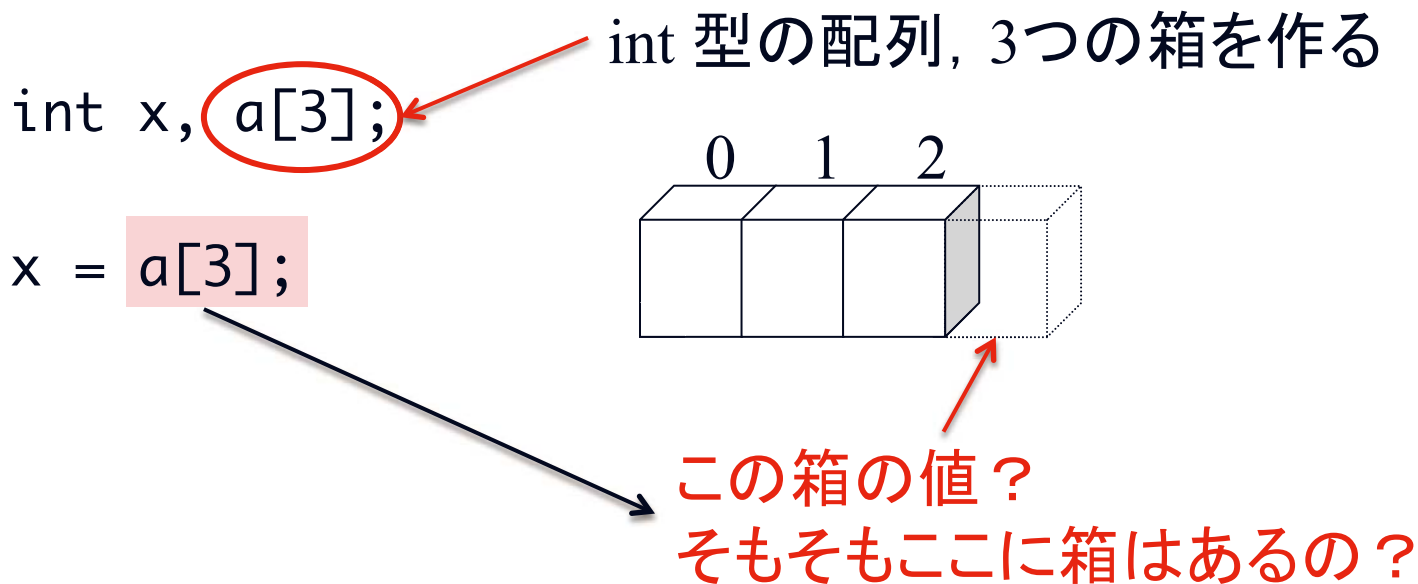
```
wa = 0;
for(i = 1; i <= 10; i++){
    wa = wa + d[i];
}
```

# 偏差値計算のプログラム例

```
#include <math.h>
int main(int argc, const char * argv[]) {
    const int N = 10;
    int mark[] = {0, 68, 94, 77, 52, 73, 85, 79, 71, 86, 63};
    int wa, i;
    double m, sigma, wa2, y;
    wa = 0;
    for(i = 1; i <= N; i++){
        wa = wa + mark[i];
    }
    m = wa / N;
    wa2 = 0;
    for(i = 1; i <= N; i++){
        wa2 = wa2 + pow((mark[i] - m), 2);
    }
    sigma = sqrt(wa2);
    printf("学生番号 成績 偏差値\n");
    for(i = 1; i <= N; i++){
        y = (10 * (mark[i] - m) / sigma + 50);
        printf("%4d      %2d   %2.1f\n", i, mark[i], y);
    }
}
```

## 5. 範囲を超えた添字を使うと？

- 配列のサイズは宣言のときに決定する
  - 宣言時にメモリ領域を確保する
- サイズを超えて使用したら？



# 配列のオーバーフロー

- C言語では、コンパイラは配列の添字の範囲のチェックは行わない
  - 配列の添字のチェックはプログラマの責任

```
int x, a[3];
```

```
x = a[3];
```

```
a[10] = 100;
```

この代入式では、何らかの値が  
変数 x に代入される

確保されていないメモリ領域に  
値 100 が書き込まれる

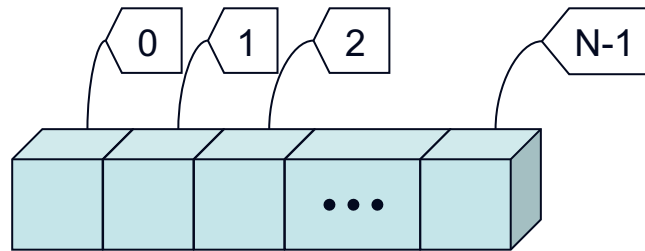
これらはエラーに  
ならない

添字の範囲チェックを怠ると、おかしい動作をする  
プログラムになる

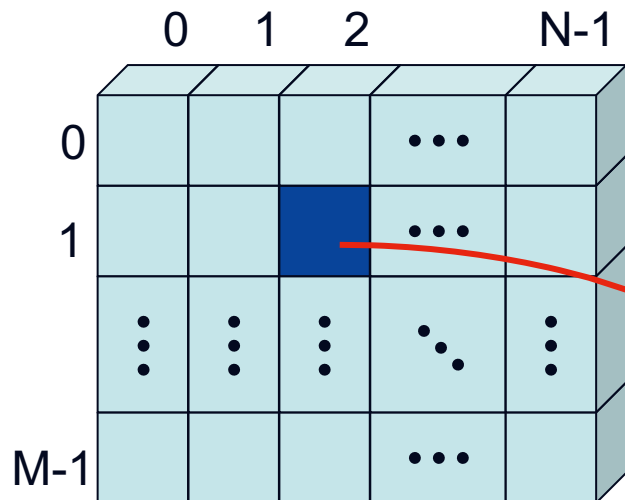
## 6. 多次元配列変数

(テキスト35ページ)

### ■ 1次元配列：一列(直線的)に並べた変数



### ■ 2次元配列：平面的に並べた変数



- 行と列を指定して1つの箱を特定する.
- 添字として2つの値を持つ.

$B[M][N]$

$B[1][2]$

行

列

# 多次元配列変数の数学との対応

意味を与えた添字付きの変数を使う計算

例  $c_{ij} = a_{ij} + b_{ij}$  (行列の和の計算)

C言語		数学	
1次元配列	$a[i]$	数列	$a_i$
2次元配列	$b[i][j]$	行列	$b_{ij}$
3次元配列	$c[i][j][k]$	????	$c_{ijk}$
	⋮		

C言語の配列では、何次元配列でも作ることができる。

例) `int b[n][m];`  
配列  $b$  は,  $n \times m$  行列

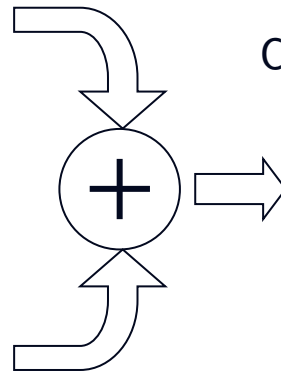


## 7. 多次元配列を使用した例

(テキスト114ページ)

- ある会社に2つの支店(A支店とB支店)があるとする.
- 各支店で1月~3月の各月の商品0, 商品1の売上数が下の表のようなデータとしてある場合, この会社の各月の各商品の売上数を計算する.

Branch_A	0	1	2
0	190	140	80
1	120	75	155



Branch_B	0	1	2
0	80	95	150
1	70	100	140

Company	0	1	2
0	270	235	230
1	190	175	295

```
Company[0][0]= Branch_A[0][0] + Branch_B[0][0];  
⋮
```

# 月別，商品別の売上げの計算 (main関数の中)

```
int Branch_A[2][3], Branch_B[2][3], Company[2][3];
int good, month;

Branch_A[0][0]=190; Branch_A[0][1]=140; Branch_A[0][2]= 80;
Branch_A[1][0]=120; Branch_A[1][1]= 75; Branch_A[1][2]=155;
Branch_B[0][0]= 80; Branch_B[0][1]= 95; Branch_B[0][2]=150;
Branch_B[1][0]= 70; Branch_B[1][1]=100; Branch_B[1][2]=140;

for(good = 0; good < 2; good++){ //各商品で処理
    for(month = 0; month < 3; month++){ //各月で処理
        Company[good][month] =
            Branch_A[good][month] + Branch_B[good][month];
    }
}
printf("この会社の売上数\n");
for(good = 0; good < 2; good++){ //商品ごとに1行で表示
    printf("商品%d\n", good);
    for(month = 0; month < 3; month++){
        printf("%2d月 : %4d個\n", month + 1, Company[good][month]);
    }
}
```

# 3行3列行列の和の計算 (main関数の中)

```
const int N = 3;
int a[N][N], b[N][N], c[N][N];
int i, j;

// 行列Aの成分を入力
printf("行列A=?\n");
for(i = 0; i < N; i++){
    for(j = 0; j < N; j++){
        scanf("%d", &a[i][j]);
    }
}

// 行列Bの成分を入力
printf("行列B=?\n");
for(i = 0; i < N; i++){
    for(j = 0; j < N; j++){
        scanf("%d", &b[i][j]);
    }
}
```

```
// 和C=A+Bを計算
for(i = 0; i < N; i++){
    for(j = 0; j < N; j++){
        c[i][j] = a[i][j]+b[i][j];
    }
}

// 和C=A+Bを出力
printf("A+B=\n");
for(i = 0; i < N; i++){
    for(j = 0; j < N; j++){
        printf("%3d", c[i][j]);
    }
    printf("\n");
}
```

# 8. データのソート

(テキスト110ページ)

## 並べ替えイメージ(その1)

小さい順

	0	1	2	3	4	5
並べ替え前	4	89	6	2	23	21

						89	1~6の最大
--	--	--	--	--	--	----	--------

				23	89		1~5の最大
--	--	--	--	----	----	--	--------

			21	23	89		1~4の最大
--	--	--	----	----	----	--	--------

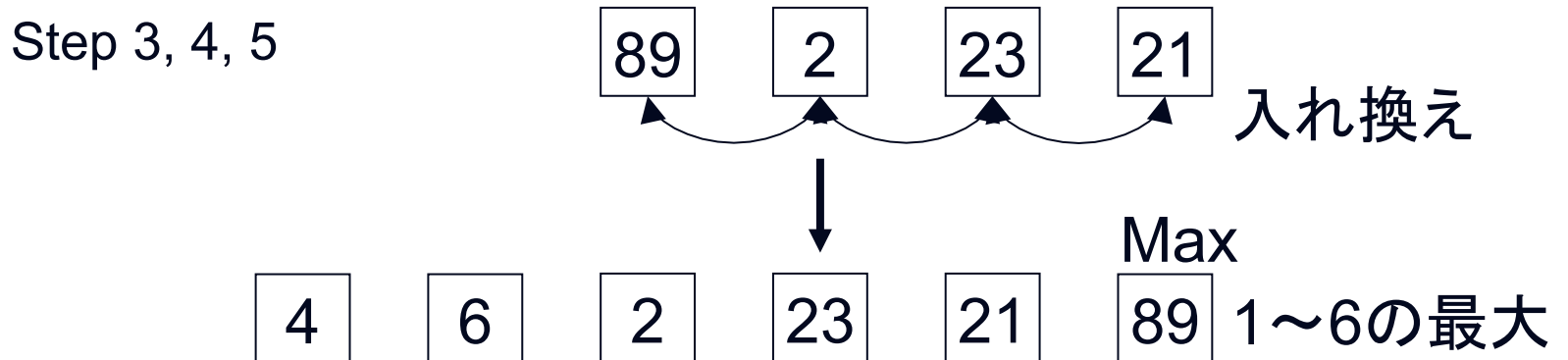
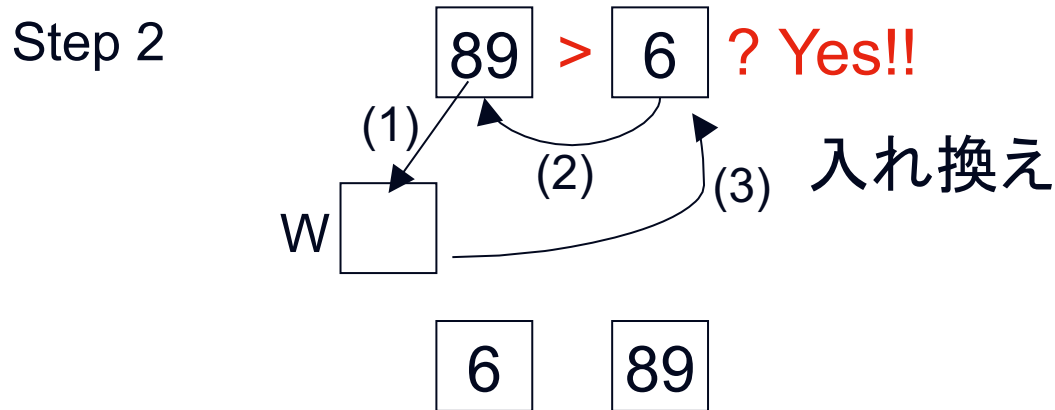
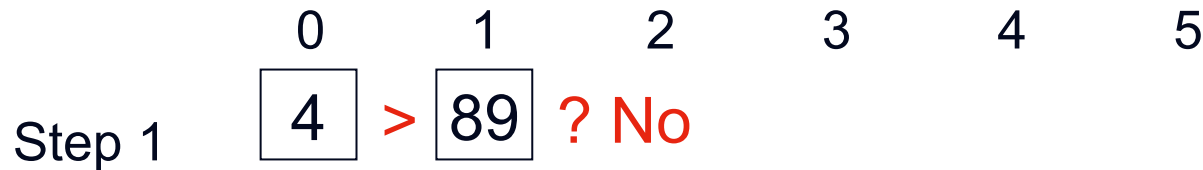
		6	21	23	89		1~3の最大
--	--	---	----	----	----	--	--------



並べ替え後	2	4	6	21	23	89
-------	---	---	---	----	----	----

# 並べ替えイメージ(その2)

隣り合った数を比べ、前の数が大きければ入れ替える。



## プログラム例 (main関数の中)

```
const int N = 6;
int d[]={4,89,6,2,23,25}, sd[N];
int i,j,w;
// 配列dを配列sdにコピー
for(i = 0; i < N; i++){
    sd[i] = d[i];
}
for(i = 0; i < N; i++){
    for(j = 0; j < N - i - 1; j++){
        if(sd[j] > sd[j+1]){
            w=sd[j];
            sd[j] = sd[j+1];
            sd[j+1] = w;
        }
    }
}
printf("Sorted data=\n");
for(i = 0; i < N; i++){
    printf("sd[%d]=%d\n", i, sd[i]);
}
```

## 9. 今日の練習問題

Level	問題
C	多次元配列の例題の会社において, 3ヶ月間の全売上数を求めるプログラムを作成せよ.
C	月と日を入力すると, その年の1月1日からの経過日数を表示するプログラムを作成せよ(閏年は考えない).
B	配列内の変化の様子を観察できるようにソートの例題プログラムを変更せよ.
B+	n個のデータをもらって, 各データの値だけ「*」を横に並べる棒グラフを描くプログラムを作成せよ. (参考: テキスト p.108 例8.7.2)
B+	3×3行列の積を求めるプログラムを作成せよ.
A	上の棒グラフを縦棒のグラフにせよ.
A	棒グラフにおいて, 平均値の「*」を「+」に換えて表示せよ.