

# プログラミング概論

後期第15回/演習回

# 本日の内容

- 講義
  - 先週の演習5(解説)
- 演習
  - 試験に向けての演習

# 先週の演習4(余力のある人向け)

- 要素数5の配列にキーボードから整数を読み込み、大きい順(降順)に並べ替えるプログラムを作成しarray\_sort.cというファイル名で保存しなさい



# どのように並べ替えるか？(1/3)

はじめに先頭の2要素を比較する

2	6	4	9	3
---	---	---	---	---

6の方が大きいので2つの要素を入れ替える

6	2	4	9	3
---	---	---	---	---

次の2要素を比較する

6	2	4	9	3
---	---	---	---	---

4の方が大きいので入れ替える

6	4	2	9	3
---	---	---	---	---

次のスライドへ続く

# どのように並べ替えるか？(2/3)

次の2要素を比較する

6	4	2	9	3
---	---	---	---	---

9の方が大きいので入れ替える

6	4	9	2	3
---	---	---	---	---

次の2要素を比較する

6	4	9	2	3
---	---	---	---	---

3の方が大きいので入れ替える(一番小さな数が確定)

6	4	9	3	2
---	---	---	---	---

# どのように並べ替えるか？(3/3)

また先頭の2要素を比較する. 6の方が大きいので入れ替えない

6	4	9	3	2
---	---	---	---	---

次の2要素を比較する. 9の方が大きいので入れ替える

6	4	9	3	2
---	---	---	---	---

次の2要素を比較する. 4の方が大きいので入れ替えない

6	9	4	3	2
---	---	---	---	---

# C言語で実装するには？

1. 配列を先頭から2要素ずつ比較
2. 後ろの要素が大きければ交換
3. すべて並べ替えられるまで繰り返す

# プログラム例(ソート部分のみ抜粋)

```
// 配列をソート
for (i = N-1; i >= 0; i--) {
    // 2要素を繰り返し比較する
    for (j = 0; j < i; j++) {
        /* 後ろの要素の方が大きければ交換 */
        if (numbers[j] < numbers[j+1]) {
            tmp = numbers[j];
            numbers[j] = numbers[j+1];
            numbers[j+1] = tmp;
        }
    }
}
```



# プログラム例(ソート部分のみ抜粋)

```
// 配列をソート
for (i = N-1; i >= 0; i--) {
    // 2要素を繰り返し比較する
    for (j = 0; j < i; j++) {
        /* 後ろの要素の方が大きければ交換 */
        if (numbers[j] < numbers[j+1]) {
            tmp = numbers[j];
            numbers[j] = numbers[j+1];
            numbers[j+1] = tmp;
        }
    }
}
```

前の要素number[j]と1つ後ろの要素number[j+1]を比較  
後ろの要素が大きければ、2つの要素を交換

# プログラム例(ソート部分のみ抜粋)

```
// 配列をソート
```

```
for (i = N-1; i >= 0; i--) {
```

```
    // 2要素を繰り返し比較する
```

```
    for (j = 0; j < i; j++) {
```

```
        /* 後ろの要素の方が大きければ交換 */
```

```
        if (numbers[j] < numbers[j+1]) {
```

```
            tmp = numbers[j];
```

```
            numbers[j] = numbers[j+1];
```

```
            numbers[j+1] = tmp;
```

```
        }
```

```
    }
```

```
}
```

j=0から開始し、確定している要素iまでjの値を増やす

# プログラム例(ソート部分のみ抜粋)

// 配列をソート

```
for (i = N-1; i >= 0; i--) {  
    // 2要素を繰り返し比較する  
    for (j = 0; j < i; j++) {  
        /* 後ろの要素の方が大きければ交換 */  
        if (numbers[j] < numbers[j+1]) {  
            tmp = numbers[j];  
            numbers[j] = numbers[j+1];  
            numbers[j+1] = tmp;  
        }  
    }  
}
```

内側のfor文が回り終わると、一番小さな数が確定していく。  
外側のfor文では末尾の要素( $i=N-1$ )からはじめ、 $i$ の値を  
1つつ小さくしていく( $i--$ )

# 演習

- 先週までの演習で終わっていないものがある  
ばそちらを進めてもok

# 演習1(関数)

- 2つの引数 $a$ ,  $b$ を受け取り、和( $a+b$ ), 差( $a-b$ ), 積( $a*b$ ), 累乗( $a^b$ )を出力する関数を作成しなさい。

なお、関数の形式は以下とする。

```
int calc(int a, int b);
```

## 演習2(再帰)

- トリボナッチ数を入力する以下の形式の関数を記述しなさい。

```
int trib(int a);
```

なお、トリボナッチ数 $F_a$ は以下のように再帰的に定義される数である。

$$F_0 = 0$$

$$F_1 = 1$$

$$F_2 = 1$$

$$F_a = F_{a-1} + F_{a-2} + F_{a-3}$$

たとえば、 $\text{trib}(3) = \text{trib}(2) + \text{trib}(1) + \text{trib}(0) = 0 + 1 + 1 = 2$ となる。

## 演習3(配列)

- 要素数10の配列にキーボードから小数を読み込み、平均値と合計値を表示しなさい。

出力例)

```
$ ./a.out
```

```
numbers[0]: 1
```

```
numbers[1]: 4
```

```
...
```

```
..
```

```
合計値: xx, 平均値: yy
```