## 情報処理演習 (7)関数その2 配列を引数に取る関数

知能システム学 准教授 万 偉偉(ワン ウェイウェイ)

### 注意事項

- プログラム(行数を含む)と出力分だけスクリーンショットするのは十分
  - 大きい全画面をカットすると見にくくなる.
- Tabを入れなかった方
- プログラムはスクリーンショットではなく、コピペの方→再提出

### 注意事項

- ・ 変数の定義
  - 大文字と小文字の混雑の使用
    - m, M, m\_Mの混雑の使用の汎用性が悪い、特に Windowsシステム用のコンパイラーは認識できない
    - 習慣的に、#defineされたマクロは大文字、それ以外 大文字
    - ・小文字と を活用すること
  - forループの完備化と簡明化
    - for(i=1, j=2; i<5, j<10; i++, j++)などの使い方は理解 し難い、習慣的にfor(i=1;i<5;i++) と if(j>=10) break; の形で実装されること
    - int q=0; for(;q<10;q++)よりfor(q=0;q<10;q++)

# 良いプログラムの第一要素

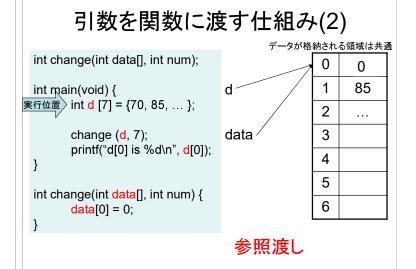
習慣に従って、他人に理解しやすい

```
配列を受け取る関
             配列を引数に渡す
数であることを宣言する
  int minimum(int data[], int num);
  int main(void) {
        int seiseki[10] = {70, 85, ...};
                                    呼び出し側では、
                                   配列の名前のみ指
        int min:
                                     定すればよい
        min = minimum(seiseki, 10);
                                          配列の大きさは別
        printf("minimum is %d\n", min);
                                          の変数で関数に伝
 }
  int minimum(int data[], int num) {
                             受け取り側では、
  }
                             配列の要素数を指 定する必要は無い
```

### 配列の中身の変更

```
int change(int data);
                                int change(int data[], int num);
int main(void) {
                                int main(void) {
       int d = 70;
                                       int d [10] = \{70, 85, \dots \};
       change (d);
                                       change (d, 10);
       printf("d is %d\n", d);
                                       printf("d[0] is %d\n", d[0]);
}
int change(int data) {
                                int change(int data[], int num) {
       data = 0;
                                       data[0] = 0;
}
    dの値は変わらない
                                       d[0]値が変わる
```

# 引数を関数に渡す仕組み(1)



### const 修飾子เตริเสริเ

- const は値の変更を禁じる
- 変数を const にした場合:値の変更が不可能 const int a = 10;

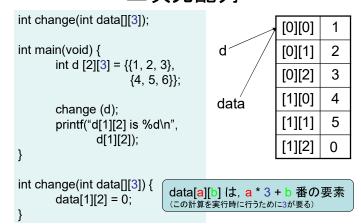
```
a = 20; ← エラー
```

引数を const にした場合:値が変更されない int func (const int data[], int n);
とすると, func 内で data に値を代入できない int func (const int data[], int n) {
 data[0] = 10; ← エラー
}

### 二次元配列の引数渡し

- 二次元配列の場合は、引数の個数を指定する必要がある
  - int func(int data[][]); ← エラー
  - int func(int data[2][3]); ← OK
  - int func(int data[][3]); ← OK
- 二次元配列の場合は、 少なくとも二番目の 引数の個数を指定すべき
- 指定しないと、一つの単元のサイズを決められない

### 二次元配列



### まとめ

- 値渡し
- ・参照渡し
- const修飾子
- 二次元配列の場合