

Cプログラミング演習1(再)

2

講義では、Cプログラミングの基本を学び

演習では、やや実践的なプログラミングを通して学ぶ

今回のプログラミングの課題

次のステップによって、徐々に難易度の高いプログラムを作成する
(参照用の番号は「よくわかるC言語」のページ番号)

1. キーボード入力された整数10個の中から最大のを答える
2. 整数を要素とする配列(p.57-59)に初期値を与えておき、その中から最大の要素を答える
3. 配列を引数としてとり、その配列の中から最大の要素を返す関数を作る
4. 配列を引数としてとり、その配列の中から**最小**の要素を返す関数を作る
5. ファイルから整数を読み込み、その中の最小値、最大値、平均値を答える

数を記憶する

数を記憶するために、「変数」を使える。しかも、いくつでも使って良い。

ただし、どんな「数」かによって宣言の仕方を変えなければならない

```
例 int x; // 整数の記憶用  
    float a; // 浮動小数点数の記憶用
```

同じ種類の数を記憶するには、『配列』を使うことも考えよう

```
例: int x[10]; // 10個の整数を記憶するための配列  
     float a[20]; // 20個の浮動小数点数を記憶するための配列
```

数を入力する

キーボードから数を入力するには scanf (p.18)を使います

ただし、どんな「数」かによって仕方を変えなければならない

例: // 整数の入力の場合

```
int x;  
scanf("%d", &x);
```

// 浮動小数点数の入力の場合

```
float a;  
scanf("%f", &a);
```

初めに数を記憶するための変数を用意

scanfの第1引数に注意

変数の前に&が必要(ポインタ)

注意：実際にはfloat aのような 変数の宣言は実行文(例:scanf文)の「前」に書かないとエラーになります

1) キーボード入力された整数10個の中から最大のを答える

手順: (最初は手順を与えておきます。段々と自分で考えられるようになりましょう)

最初は10個ではなく、3個から始めます

- 1) 整数3個を記憶するための配列を用意
- 2) キーボード入力した整数を配列に入力

繰り返しによる

- 3) これらの要素を1つずつみて、最大のを記憶する

ここも繰り返しによる

条件分岐

- 4) 配列の要素を見終わったら、記憶した「最大要素」を出力

1) 整数3個を記憶するための配列を用意

これはもうできますね。

ここでは配列名を `array` としておきましょう。

2) キーボード入力した整数を配列に入力

for文を使って3個の数をarrayの要素にセットしましょう。

(1) そのために、1) で用意した配列 array と、整数型の変数 (iとしておきます) を用意します

```
int array[3];  
int i;
```

(2) キーボード入力された整数を変数にセットするにはscanfを使います

```
scanf("%d", &x); // 変数xに入力された数をセットする
```

問題 : array[i] にセットするにはどうすればよい?

(3) forを使って、(2)を3回繰り返します

for文による繰り返し

for文は、**特定の回数**、何かを繰り返すために使われます
パターン：

i を整数型の変数とする（前の方で宣言されているとする）

```
for (i=0; i < 3; i++) { // 3回の繰り返し
    何かの仕事;
}
```

質問： $i < 3$ ではなく $i \leq 3$ とするとどうなるでしょう？

「3」ではなく「10」とするとどうでしょうか？

キーボード入力した整数を配列に入力

for文を使って3個の数をarrayの要素にセットしましょう

```
int array[3];  
int i;  
for (i=0; i < 3; i++) {  
    scanf("%d", &array[i]);  
}
```

注目：ここでは、iは繰り返しの回数の記憶だけではなく
配列のどこに数を記憶するかを表すという重要な役割

注意：配列は0番目から始まる！

だから array[3] によって、array[0], array[1], array[2] という3つの記憶場所が作られる

n個の要素から最大値を探す

「配列の要素を1つずつみて、最大のものを記憶する」方法を考えましょう

この作業には以下が必要です(どうでしょう、考えつきましたか?)

- 1) 「今まで見た要素の中で最大のものを記憶する」ための変数を用意する
- 2) その変数に初期値を与える(次で「要素の比較」をするため)
- 3) 配列の要素を1つずつ取り出して、記憶した要素と比較する
必要ならば、変数の値を更新する

n個の要素から最大値を探す(1)

1) 「今まで見た要素の中で最大のものを記憶する」ための変数を用意する

これは簡単ですね。

変数名を（なんでもよいのですが）max としておきましょう

```
int max;
```

n個の要素から最大値を探す(2)

2) その変数に初期値を与える(次で「要素の比較」をするため)
これも簡単ですね...

配列の最初の要素を代入すればよいのです
ここで配列の名前を array とすると
`max = array[0];`

n個の要素から最大値を探す(3)

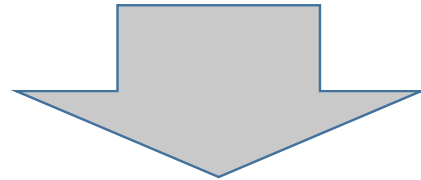
- 3) 配列の要素を1つずつ取り出して、記憶した要素と比較する
必要ならば、変数の値を更新する
これもできますね

配列の要素を1つずつ取り出すには、for文を使います

```
int i; // カウンタ変数
for(i=1; i < n; i++) { // 配列の要素を一個ずつ取り出す
    if (array[i] > max)
        {   なにをしたらよいでしょう？書いてください   }
}
```

必要ならば、変数の値を更新する

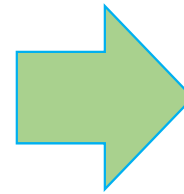
```
int i; // カウンタ変数
for(i=1; i < n; i++) { // 配列の要素を一個ずつ取り出す
    if (array[i] > max)
        {   なにをしたらよいでしょう？書いてください   }
}
```



ここでそれぞれの変数の「意味」を考える

max : 最大値の候補を記憶する変数

array[i] : 読み込んだn個の整数



max よりも array[i] の方が大きいことから...
max の値を書き換える!

```
max = array[i];
```

キーボード入力された整数3個の中から最大のものを答える (完成させてください)

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

```
Int main(void) {
```

```
    //(1) 整数3個を記憶するための配列を用意
```

```
    // (2)キーボード入力した整数を配列に入力
```

```
    // (3) 1)「今まで見た要素の中で最大のものを記憶する」変数
```

```
    // 2) その変数に初期値を与える
```

```
    // 3) 配列の要素を1つずつ取り出して、記憶した要素を比較
```

```
    // (4)配列の要素を見終わったら、記憶した「最大要素」を出力
```

```
    return(0);
```

```
}
```

走らせてみましょう

チェック項目

- 1) コンパイル(ビルド)はできたか？
- 2) 3個のいろいろな整数（マイナスの数も含む）を入力させて、最大の要素を出力できたか？
- 3) 数の入力や、結果の表示の仕方はわかりやすいものだったか？

発展：3個ではなく、10個の整数をあつかえるようにしましょう

2. 整数を要素とする配列(p.57-59)に初期値を与えておき、その中から最大の要素を答える

「よくわかるC言語」 p.59 にあるように、整数を要素とする配列の初期値は次のようにして与えます:

```
int array[10] = {2, 3, -5, 0, 9, 12, -10, 8, -3, 9};
```

2. 整数を要素とする配列に初期値を与えておき、その中から最大の要素を答える(完成させよ)

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

```
Int main(void) {
```

```
    //(1) 整数10個を記憶するための配列を用意、初期値を与える
```

```
    //(2)は不要
```

```
    //(3) 1)「今まで見た要素の中で最大のものを記憶する」変数
```

```
    // 2) その変数に初期値を与える
```

```
    // 3) 配列の要素を1つずつ取り出して、記憶した要素を比較
```

```
    //(4)配列の要素を見終わったら、記憶した「最大要素」を出力
```

```
    return(0);
```

```
}
```

3. 配列と要素数とを引数としてとり、その配列の中から最大の要素を返す関数を作る

そのような関数を `maxInArray` と名付ける。すると2の答は次のようになるだろう:

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

```
// ここに maxInArray の関数定義が入る
```

```
Int main(void) {
```

```
    //(1) 整数10個を記憶するための配列を用意、初期値を与える
```

```
    int array[10] = {2, 3, -5, 0, 9, 12, -10, 8, -3, 9};
```

```
    int max=maxInArray(array,10); // (2-3) 関数の呼び出し
```

```
    // (4) 記憶した「最大要素」を出力
```

```
    printf("Max number in the Array = %d¥n", max);
```

```
    return(0);
```

```
}
```

3. 配列と要素数とを引数としてとり、その配列の中から最大の要素を返す関数を作る

main関数の中で最大の要素を返すことはできても、「関数」を作る、というところで戸惑ってしまった人に

```
#include <stdio.h>
Int main(void) {
    //(1) 整数10個を記憶するための配列を用意、初期値を与える
    //(2)は不要
    //(3) 1)「今まで見た要素の中で最大のものを記憶する」変数
    // 2) その変数に初期値を与える
    // 3) 配列の要素を1つずつ取り出して、記憶した要素を比較
    //(4)配列の要素を見終わったら、記憶した「最大要素」を出力
    return(0);
}
```

この部分を
「関数」 `maxInArray` の呼出とする
そのために考える事はこの関数の
(1) 入力 (引数)
(2) 処理 --- ほぼこのまま
(3) 出力 (帰り値)

関数を作る

```
#include <stdio.h>
Int main(void) {
    //(1) 整数10個を記憶するための配列を用意、初期値を与える
    //(2)は不要
    //(3) 1)「今まで見た要素の中で最大のものを記憶する」変数
    // 2) その変数に初期値を与える
    // 3) 配列の要素を1つずつ取り出して、記憶した要素を比較
    //(4)配列の要素を見終わったら、記憶した「最大要素」を出力
    return(0);
}
```

関数名を maxInArray とする

(1) 入力

整数が入った配列 と そのサイズ

(2) 処理

上記の(3)

(3) 出力(帰り値)

配列の中の最大値 = 記憶した最大要素

// (3-4) は「関数の呼び出し」で置き換えます
int max=maxInArray(array, NUM);

関数を定義する

「関数」の定義のパターンは main 関数とほぼ同じ

関数への入力

```
int main(void) {  
    int x,y;    // 変数宣言  
    scanf("%d %d", &x, &y);  
    printf("sum=%d", x+y);  
    return 0;  // 値を返す  
}
```

```
関数の型 関数名(引数リスト) {  
    関数で使う変数の宣言  
    いろいろな処理  
  
    ...  
    値を返す(出力)  
}
```

関数 maxInArray

出力=記憶した最大要素は **int** 型

```
int maxInArray (int ar[] , int n)
```

入力=整数を要素とする配列
int ar[] と要素数 int n

```
{
```

```
int max=ar[0];
```

今まで見た要素の中で最大のものを記憶する」変数を宣言、その変数に初期値を与える

```
int i;
```

```
for (i=1; i< n; i++) {
```

```
    // 一個ずつmax値と比較
```

```
    // maxよりも大きければ、max値を更新
```

```
}
```

要素を比較して最大値を求める

```
return max;
```

出力=最大要素

```
}
```

3. 配列と要素数とを引数としてとり、その配列の中から最大の要素を返す関数 「改訂」

10という数はここでは意味がある数なので「定数」として扱おう

```
#include <stdio.h>
#define NUM 10
// ここに maxInArray の関数定義が入る

Int main(void) {
    //(1) 整数10個を記憶するための配列を用意、初期値を与える
    int array[NUM] = {2, 3, -5, 0, 9, 12, -10, 8, -3, 9};
    int max=maxInArray(array, NUM); // (2-3) 関数の呼び出し
    //(4) 記憶した「最大要素」を出力
    printf("Max number in the Array = %d¥n", max);
    return(0);
}
```


3. 配列と要素数とを引数としてとり、その配列の中から最大の要素を返す関数

これを完成させてください

```
#include <stdio.h>
#define NUM 10
// ここに maxInArray の関数定義が入る

Int main(void) {
    //(1) 整数10個を記憶するための配列を用意、初期値を与える
    int array[NUM] = {2, 3, -5, 0, 9, 12, -10, 8, -3, 9};
    int max=maxInArray(array, NUM); // (2-3) 関数の呼び出し
    //(4) 記憶した「最大要素」を出力
    printf("Max number in the Array = %d¥n", max);
    return(0);
}
```

4. 配列を引数としてとり、その配列の中から最小の要素を返す関数を作る

そのような関数を `minInArray` と名付ける。

その定義はどうなるでしょう？

`maxInArray` とはどこが同じでしょう？

また、どのように違うでしょう？

どのように使えるでしょうか？

考えて書いてみてください。

4. 配列を引数としてとり、その配列の中から最小の要素を返す関数を作る

完成させてください

```
#include <stdio.h>
#define NUM 10
// ここに maxInArray の関数定義が入る
// ここに minInArray の関数定義が入る

Int main(void) {
    //(1) 整数10個を記憶するための配列を用意、初期値を与える
    int array[NUM] = {2, 3, -5, 0, 9, 12, -10, 8, -3, 9};
    int max=maxInArray(array, NUM); // (2-3) 関数の呼び出し
    int min=minInArray(array, NUM);
    //(4) 記憶した「最大要素」を出力
    printf("Max = %d and Min = %d\n", max, min);
    return(0);
}
```

5. ファイルから整数を読み込み、その中の最小値、最大値、平均値を答える

このためには、

- (1) ファイルから整数を読み込んで配列に記憶する(p.103)
- (2) 読み込んだ整数の個数を記憶する
- (3) 最小値、最大値を返す関数に加えて、平均値を返す関数を作る

ことが必要。

さあ、考えてみよう。(次回)

ファイルの操作について...

ファイルの内容を取り込む、ファイルに書出す、どちらにしても基本はほとんど同じ:

(1) **ファイルポインタ**の記憶用の変数を用意

例: `FILE *fp;`

(2) **ファイルを開く**: ファイルへのパスを指定し、「読み込み」または「書き込み」モードで「開く」

例: `fp1 = fopen("H:¥¥data.txt", "r"); // 読み込み(read)`

`fp2 = fopen("H:¥¥rest.txt", "w"); // 書き込み(write)`

(3) **読み込み関数**でファイルからデータを読む

例: `fscanf(fp1, "%d", &x); // 整数を読み込んで変数xに代入`

書き込み関数でファイルにデータを書く

例: `fprintf(fp2, "最大値 = %d¥n", max); // 変数maxの値などを書き込む`

宿題

課題 1 から 4 までについて、プログラム、実行例、プログラムの説明を提出してください

予習：教科書の「ファイルの操作」の箇所を読んでくること

FILE

fopen

fclose

fprintf

fscanf

について調べる