Cプログラミング演習1(再)
3

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

1. キーボード入力された整数10個の中から最大のものを答える
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
6. ファイルから浮動小数点数を読み込み、その中の最小値、最大値、平均値、標準偏差を(別の)ファイルに書出す

学習済み

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。

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

(4) ファイル操作が終わったら「閉じる」　例: fclose(fp1);

ファイルの操作について…解説

(1) ファイルポインタの記憶用の変数を用意

例: FILE *fp;

この例における変数名は fp

「FILE *fp」　とは

変数 fp に　FILEへの手がかり(ポインタ) を記憶することを意味

ファイルそのものは「変数の値」にはならない！

内部的には、ファイル FILE はいろいろな情報の塊、それは一個の変数では記憶できない、そこでその塊への手がかり(ポインタ)を 変数(この例ではfp) に記憶させる、そのことを表すために * が必要

ファイルの操作について…解説

(2) ファイルを開く: ファイルへのパスを文字列で指定し、「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)

パス(path – 道)

パスをきちんと書かないと、どこにファイルがあるか、コンピュータがわからなくなる=読み込めなかったり、変なところに作られたりする --- ファイルのプロパティ「場所」で確認できる

読み込みには "r"

書き込みには "w" または "a" を用いる　--- この違いを確認しよう

ファイルの操作について…解説

(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

キーボード入力の場合 scanf

変数の値の書き出しには printf を使っていた

--- どこが同じか、どこが違うかを確認しよう

ファイルの操作について…解説

(4) ファイル操作が終わったらファイルを「閉じる(close)」

例: fclose(fp1);

一つのプログラムで同時に開くことができるファイルの個数が決まっています(Windowsでは512個)

- たくさんのファイルを扱う場合には不要になったファイルを閉じるようにする
- 一旦閉じたファイルでも再度「開く」と最初から読んだり書いたりできる

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

(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
次のようになるだろう:

```
FILE *fp;
int n=0, array[NUM];      // nは読み込んだ要素数を記憶
                          // NUMは定数 (100くらい）として宣言しておく
fp=fopen(”H:¥¥data.txt”,”r”);
while (fscanf(fp,”%d”, &array[n]) != EOF) {
      n++;
}
```

質問: EOFとは何か？

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

(3) 平均値を返す関数を作る

平均値は浮動小数点数にしたほうが良い

例えば {1,2}の平均は 1.5 としたい --- 整数なら 1 になってしまう

次のような形の関数になるだろう:

```
float  average(int array[], int n) {
        // 配列arrayのn個の要素の和を求める
        // それを n で割った値（浮動小数点数)をreturn
 }
```

注: 配列の要素の和を求める関数sumを作っておき、それを利用するのは良いアイデア

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

次のようになるだろう:

```
#include <stdio.h>
// 最大100個を要素数とする
#define NUM 100
// 配列と要素数を引数とし最大値を返す関数 maxInArray をここに書く
// 配列と要素数を引数とし最小値を返す関数 minInArray をここに書く
// 配列と要素数を引数とし平均値を返す関数 average をここに書く
int main(void) {
    int n, array[NUM];
    // ファイルを開いてarrayにデータを書き込み、
    // 読み込んだ個数を nの値とする手続きをここに書く
    printf("最大値=%d, 最小値=%d, 平均=%f¥n",
                    maxInArray(array,n), minInArray(array,n), average(array, n));
    // ファイルを閉じる
    return(0);
}
```

完成させ、ビルドし、適当なファイルで試してみよう！

6.ファイルから浮動小数点数を読み込み、その中の最小値、最大値、平均値、標準偏差を(別の)ファイルに書出す

この課題のためには、新たに

(1) プログラムの出力を「ファイルに書出す」方法を学ぶ

(2) 標準偏差を求める関数を作る

ことが必要 --- どちらも簡単。

配列arrayのn個の要素の標準偏差 ： 平均をmeanとすると、

$$\sqrt{\frac{1}{n}\sum_{i=0}^{n-1}(array[i]-mean)^2}$$

6.ファイルから浮動小数点数を読み込み、その中の最小値、最大値、平均値、標準偏差を(別の)ファイルに書出す

完成させよう

data.txt に書かれている整数(100個以下)を読み込み、
配列に記憶し、
最大値、最小値、平均値を求め
平均値を使って標準偏差を求め
result.txt ファイルに次の形式で書出す

ファイルdata.txtの要素数は10
最大値=12, 最小値=-10, 平均= 2.50, 標準偏差 = 4.25