

演習課題 No.11

(注:課題は二つある.)

課題 11-1 【時間内提出】

Step 1. 課題 8-1 で作成した文字列の長さを求めるプログラムを, 関数に書き直して再利用したい. ただし, この関数は, 文字列(文字配列)を引数として, その文字列の文字数が返却値とする. したがって, この関数の宣言は, `int length(char s[])` となる.

Step 2. 上で作成した関数 `length()` を `main()` 関数から呼び出す形に課題 8-1 を書き直せ. 実行例は課題 8-1 と同じ.

!注意! ライブラリ関数 `strlen()` は使用禁止!

課題 11-2

Step 1. 課題 7-3 のデータ入力部分を, データ入力関数 `int input(float data[], int max)` として書き直す. この関数では, 課題 7-2 と同様に, `scanf()` を用いて `float` 型の値を入力し, それを順番に `data[]` 配列に格納する. ここで `data[]` 配列は関数の呼び出し側(この課題では `main()` 関数側)で宣言された配列に実体がある. `main()` 関数側では, 宣言した配列の要素数(入力可能なデータの最大個数)を `max` としてこのデータ入力関数を呼び出すものとする. また, この関数は, データ入力数が最大の `max` 個に達するか, 負値が入力されるまでデータ入力を繰り返すものとする. なお, この関数は実際に入力された有効なデータの個数を返却値として返すものとする.

Step 2. 平均値を求める次の関数を定義する. `float average(float data[], int n)`. この関数は, `data[]` 配列に入っている `n` 個のデータの平均値を求め, 返却値としてその平均値を返す.

Step 3. 不偏分散を求める次の関数を定義する. `float variance(float data[], int n)`. この関数は, `data[]` 配列に入っている `n` 個のデータの不偏分散を求め, 返却値としてその不偏分散値を返す. なお, 不偏分散の計算に必要な平均値は, 関数 `average()` を呼び出して求めること.

Step 4. 上で作成した三つの関数とダウンロードしたソース中の `main()` 関数を用い, 課題 7-3 と同じ動作をするプログラムを作成せよ. なお, 下記の `main()` 関数はそのまま変更せずに用いること.

(`main()` 関数の定義は次ページ)

ダウンロードできるソースプログラムは次のとおり。実行例は課題 7-3 と同じ。

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

int input(float data[], int max)
{
    // この関数は各自で作成
}

float average(float data[], int n)
{
    // この関数は各自で作成
}

float variance(float data[], int n)
{
    // この関数は各自で作成
}

////////// 以下の main() 関数部分を変更してはならない //////////
// この main() 関数のままで正しく動作するように各関数をプログラムせよ

int main(void)
{
    float d[100];    // 処理するデータは最大 100 個まで
    int n;
    double s;

    n = input(d, 100);    // データの入力。実際に入力されたデータの個数を n に入れておく
    printf("データ数は%d です\n", n);

    printf("平均値は%f です\n", average(d, n));    // n 個のデータの平均値の計算と表示

    s = variance(d, n);    // n 個のデータの不偏分散の計算
    printf("不偏分散は%f です\n", s);    // 不偏分散値の表示

    printf("確率誤差は%f です\n", 0.6745*sqrt(s/n)); // n 個のデータの確率誤差の計算と表示
}
```

解説 `input()` 関数は呼び出し側の配列に内容を入力する、すなわち呼び出し側の配列(実引数)の内容を変化させるので、参照渡しの機能を使っていることになる。配列とポインタ以外の引数を用いてこのような関数を作成することはできない。

解説 この様にプログラムされた `input()` 関数、`average()` 関数、`variance()` 関数は、必要に応じて 再利用可能なサブルーチン になっている。