計算機プログラミングI 講義ノート #7

担当：井上 純一 (情報科学研究科棟 8-13), 赤間 清 (情報基盤センター)
URL：http://chaosweb.complex.eng.hokudai.ac.jp/~j_inoue/PROG2007/PROG2007.html

平成 19 年 5 月 25 日

目 次

17 マクロ定義とヘッダファイル

※ 説明は教科書 pp. 69-82 を用いて行います.

練習問題 17.1 (※ LMS 入力問題)
3 個の引数のうち, 最小のものを返す関数型引数マクロ min3(a,b,c) を作成し, これを確認するメイン関数を書け. メイン関数内では a,b,c の値がキーボード入力されると, その最小値を

```
min=*****
```

の形式で出力するようにせよ.
[**ヒント**] 教科書 p.24 の「3 項演算子」を用いると良い.

練習問題 17.2
練習問題 17.1 で作成した関数マクロを user.h という名前のヘッダファイルに書きこみ, これを読み込んで **練習問題 17.1** と同じ動作をするプログラムを作成せよ.

18 配列

これまでプログラムを書く際には各データを名前をつけた変数に入れて扱ってきました. このような扱い方はデータ数が少ないうちは対応できますが, データ数が多くなると, 変数の定義部だけでも十分に長くなり, 見通しの悪いプログラムになってしまうことがあります. さらに, そのようにして定義した変数の操作も不便です.

この点を実際に見てみるために, 復習になりますが, 次の練習問題をやってもらいましょう.

練習問題 18.1

次のプログラムをファイルに打ち込み, コンパイル・実行せよ.

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

main()
{
        double a1,a2,a3,a4,a5,max;
        scanf("%lf %lf %lf %lf %lf",&a1,&a2,&a3,&a4,&a5);

        /* 最大値を求める*/

        max=a1;
        if(a2 > max){
           max=a2;
           }else{
           max=max;
           }
        if(a3 > max){
           max=a3;
           }else{
           max=max;
           }
        if(a4 > max){
           max=a4;
           }else{
           max=max;
           }
        if(a5 > max){
           max=a5;
           }else{
           max=max;
           }

        printf("max=%lf\n",max);
}
```

練習問題 18.2(※ LMS 入力問題)

変数 a1,a2,a3,a4,a5 に対して, これらの値をキーボードから入力すると, これらの変数の最小値, 平均値, 標準偏差を計算し, 表示させるプログラム作成せよ. ただし, 平均値 m, 標準偏差 d はそれぞれ

$$m = \frac{1}{5}\sum_{i=1}^{5} a_i,\ d = \sqrt{\frac{1}{5}\sum_{i=1}^{5}(a_i - m)^2}$$

で定義されることに注意し, 計算結果を

```
min=***** m=***** d=*****
```

の形式で出力させること.

上記の練習問題のような場合, 次に定義される**配列**を用いると便利です.

18.1 配列の宣言 — 1 次元の場合 —

変数型 配列変数名 [配列要素数];

(例)

```
int a[10]; /* int 型の要素数 10 の配列に a という名前をつける */
double Input[100]; /* double 型の要素数 100 の配列に Input という名前をつける*/
```

これを用いると, 先に見た**練習問題 18.1** は次のように簡潔に書けます.

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

main()
{
        double a[5],max;
        int i;

        for(i=0; i<5; i++)
        {
        scanf("%lf",&a[i]);
        }

        max=a[0];

        for(i=1;i<5;i++){
                if(a[i]>max){
                  max=a[i];
                  }else{
                  max=max;
                  }
                }
        printf("max=%lf\n",max);
}
```

当然のことながら, **練習問題 18.2** で行った最小値, 平均値, 標準偏差を求めるプログラムも配列を用いて

書き直すことができます. これを次の練習問題としてやってみましょう.

練習問題 18.3 (※ LMS 入力問題)
練習問題 18.2 で作成したプログラムを配列を用いて書き直せ.

配列を用いる際には次の事項に注意すべきです.

[**注意事項**]

- 要素数は確定されている必要あり. 要素数は数値定数, あるいは

```
#define N 100
....
....
main()
{
int a[N];
...........
.......
}
```

  のようにして指定する.

- 要素数が N 配列では, 要素番号は 0 から始まり, $N-1$ が最終要素番号である.
- 要素番号の範囲を越えてしまうと, そのデータは無意味であるばかりでなく, プログラムの動作自体も保障されなくなる. これは常に注意を払うべきである.

18.2　配列の初期化 — 1 次元の場合 —

配列変数を初期化するには例えば次のようにします.

```
int a[3]={10,11,12}; /* 要素数は 3 個 */
```

このように初期化すると, a[0]=10,a[1]=11,a[2]=12 のように, 配列の各成分が初期化されます. ここで [] 内に書いた要素数が列挙された要素数よりも多い場合には, 不足している要素は初期化されないことに注意すべきです[1]. 逆に, [] 内に書いた要素数が実際に列挙された要素数よりも少ない場合には文法エラーとなります.

なお, 初期化の方法としては次のようなやり方もあります.

```
int a[]={5,6,7,8};
```

この場合には, **実際に列挙された要素数によって配列の要素数が決定する**ことになります.

練習問題 18.4
次のプログラムをファイルに書き, コンパイル・実行することで動作を確認せよ.

[1] 一見, ゼロで初期化されても良さそうに見えるかもしれないが, 初期化されない.

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

main()
{
        int a[]={1,2,3,4,5,6,7,8,9,10};
        int n=10,sum,i;

        sum=0;
        for(i=0; i<n; i++)
        {
                sum += a[i];
        }

        printf("sum: %d\n",sum);
}
```

今週の LMS 入力問題

サイコロを用いた賭けを行いたいのだが, 当のサイコロが無いことに気がついた. そこで, この賭けには何も利害関係を持たない A さんをつかまえて別室に待機させ, サイコロを振る必要が生じたときに, A さんにその瞬間に頭にひらめいた 1〜6 までの数字の中で好きな数字を言わせ, その数字をもって「サイコロの目の代用」とすることを考えたい. しかし, 例えば A さんはとりわけ「3」という数字が好きで, 3 を他の 5 つの数字よりも頻繁に言う傾向があるかもしれない.

(1) これを確かめるための予備実験として 1〜6 までの数字 120 個を可能な限りでたらめにキーボードから入力すると, 120 回の試行のうち 1 が出た回数, 2 が出た回数, …, 6 が出た回数, 及び打ち込んだ数の総和を表示するプログラムを作成せよ. ただし, そのときの結果を

```
number_1=***
number_2=***
number_3=***
number_4=***
number_5=***
number_6=***
sum_number=*****
```

の形式で表示させること.

(2) キーボードの数字の配置では「1」の隣に「2」が来るが, 120 回の試行のうち 12 あるいは 21 の並びが何回程度現れるかも確かめておきたい. このためのプログラムを作成せよ. 結果は

```
number_12=***
number_21=***
```

の形式で表示させること.