

プログラミング演習 第2回

問 A [02A]: 2 番目に大きな整数

図 1 は 標準入力 (端末等) から $N(N \geq 2)$ 個の整数の列を配列 a に読み込み、一行に 1 つずつ出力するプログラムである。このプログラムを変更して、読み込んだ整数の列の中から 2 番目に大きな整数を出力するプログラムを作れ。同じ整数が複数回現れた場合は、これらを区別せよ。たとえば、数列の中に最も大きな整数が二つ以上ある時は、最も大きな整数を出力することを意味する。

N はマクロで 10 と定義せよ。ただし、 N の値を変更しても正しく動作するようにしておくこと。

ヒント

- 全体をソートする必要はないことに注意。
- まず、最大値を求める手順を考え、その手順内に 2 番目に大きな値を求める手順を組み込む (もちろん、別の解法もある)。

```
#include <stdio.h>
#define N 10 /* 整数の個数の定義 */
int a[N];    /* 整数を読み込む配列 */
int main(void) {
    int i;
    for (i = 0; i < N; i++) scanf("%d", &a[i]); /* 読み込み */
    for (i = 0; i < N; i++) printf("%d\n", a[i]); /* 出力 */
    return 0;
}
```

図 1: 整数列を読み込み、そのまま出力するプログラム

入出力例

入力 1

5 -10 3 7 1 4 5 0 3 -2

入力 1 に対応する出力

5

入力 2

5 -10 3 7 1 4 7 0 3 -2

入力 2 に対応する出力

7

問 B [02B]: ユークリッドの互除法

ユークリッドの互除法は世界最古のアルゴリズムとして知られている。そのアルゴリズムとは、二つの整数 n と $m(m \geq n)$ が与えられたとき、次の手順で n と m の最大公約数を求めるものである。

1. $n = 0$ なら計算を終える。このとき、 m が最大公約数である。
2. m を n で割った余りを新たに n と、元の n を新たに m として、1. に戻る。

ユークリッドの互除法を使って、標準入力 (端末等) から与えられる二つの正整数の最大公約数を求めるプログラムを作れ。整数は小さい順に与えられるとは限らないことに注意せよ。

入出力例

```
入力
8 6
出力
2
```

問 C [02C]: 最も長い行

Unix のコマンドには、入力の各行の '#' から行末までをコメントとして無視するものが少なくない。標準入力 (端末等) から空行が現れるまで一行ずつ読み込み、最も長い行 (複数ある時は、それらを辞書順に並べた時に先に最初に現れる行) を出力するプログラムを作れ。ただし、行の途中に '#' が現れた時は、(複数ある時は最初の) '#' を含めて行末まで行の長さに数えず、出力にも含めないものとする。また、空白文字や制御文字も 1 文字と数える。入力される各行の長さは (もしあれば) '#' 以降も含めて 78 文字以下で ASCII 文字のみで構成されると仮定してよい。

入出力例

```
入力
abcd
x y z#XYZ
10.3#!"#$%
出力
x y z
```

ヒント

- 空行が現れるまで 標準入力 (端末等) から文字列を一行ずつ読み込んで出力するプログラムは図 2 のようになる。
- '#' を文字列の終わりを示す文字に書き換えればよい。
- 最も長い行を保持するための配列を追加する必要がある。
- `fgets` は改行文字も読み込むことに注意せよ。
- 文字列の辞書順の比較には `strcmp` を用いる。

```

#include <stdio.h>
#include <string.h>
#define MAXLEN 78 /* 文字列の最大長 */
char buf[MAXLEN+2]; /* 文字列を読み込む配列 */
int main(void) {
    while (1) {
        int len;
        fgets(buf, sizeof(buf), stdin); /* 一行読み込む */
        if (buf[0] == '\n' || buf[0] == '\0') break; /* 空行であれば終了 */
        /* 行末に改行 (\n) があれば取り除く */
        len = strlen(buf);
        if (buf[len - 1] == '\n') buf[len - 1] = '\0';
        printf("%s\n", buf);
    }
    return 0;
}

```

図 2: 文字列を読み込み、そのまま出力するプログラム

問 D [02D]: 行列乗算

図 3 は 標準入力 (端末等) から実数を要素とする $N(N \geq 2)$ 行 N 列の行列を二次元配列 `m` に読み込んで、そのまま出力するプログラムである。これを変更して、読み込んだ行列を $M(M \geq 1)$ 乗した行列を出力するプログラムを作れ。

N 、 M はマクロでそれぞれ 3、4 と定義せよ。ただし、 N と M の値を変更しても正しく動作するようにしておくこと。

入出力例

入力に行列 $\begin{pmatrix} 0.0 & 1.4142 & 0.0 \\ -1.4142 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{pmatrix}$ を与える例を示す。

入力

```

0.0      1.4142 0.0
-1.4142 0.0    0.0
0.0      0.0   1.0

```

出力

```

3.999846 0.000000 0.000000
0.000000 3.999846 0.000000
0.000000 0.000000 1.000000

```

ヒント

- 行列の乗算 $A = AB$ の計算の最中に A を書き換えてはいけない。したがって、一度 $C = AB$ を計算した後、 A に C の内容をコピーする必要がある。

```

#include <stdio.h>
#include <stdlib.h>
#define N 3          /* 行列のサイズ */
#define BUFLLEN 100
float m[N][N];      /* 読み込んだ行列 */
char buf[BUFLLEN]; /* 1行を読み込む配列 */
int main(void) {
    int r, c;
    /* 読み込み */
    for (r = 0; r < N; r++) {
        char* p = fgets(buf, sizeof(buf), stdin);
        for (c = 0; c < N; c++)
            m[r][c] = strtod(p, &p);
    }
    /* 出力 */
    for (r = 0; r < N; r++) {
        for (c = 0; c < N; c++) {
            if (c > 0) printf(" ");
            printf("%f", m[r][c]);
        }
        printf("\n");
    }
    return 0;
}

```

図 3: 行列を読み込み、そのまま出力するプログラム

- n が二進 m 桁の整数で、0 か 1 の値をとる $n_i (0 \leq i \leq m-1)$ を使って $2^{m-1}n_{m-1} + 2^{m-2}n_{m-2} + \dots + 2^0n_0$ と表せるとき、

$$\begin{aligned}
 A^n &= A^{2^{m-1}n_{m-1} + 2^{m-2}n_{m-2} + \dots + 2^0n_0} \\
 &= A^{2^{m-1}n_{m-1}} A^{2^{m-2}n_{m-2}} \dots A^{2^0n_0}
 \end{aligned}$$

となることを利用すると高速に計算できる。余裕があれば挑戦せよ。