

1 はじめに

本レポートではプログラミング序論演習の第 1 回レポートとして、複数の自然数の最大公約数を求めるプログラムについて述べる。プログラミング序論演習の授業で三つの自然数の最大公約数を求めるプログラムを作成した。そこで、複数の自然数の最大公約数を求めるプログラムの作成を試みた。

2 設計

2.1 機能設計

本プログラムでは、複数の自然数の入力を要求し、入力された自然数の最大公約数を求め表示する。自然数の入力は 0 が入力されるまで続くものとし、負の値が入力された場合は、最大公約数を求める対象とはしない。

2.2 アルゴリズムの設計

二つの自然数の最大公約数を求めるアルゴリズムには、プログラミング序論の教科書[1]に掲載されているものを利用する。このアルゴリズムを図示したものを図 1 に示す。

三つの自然数 a_1, a_2, a_3 の最大公約数は、 a_1, a_2 の最大公約数と a_3 の最大公約数と等しい。同様に自然数 $a_1, a_2, a_3, \dots, a_n$ の最大公約数は、 a_1, a_2 の最大公約数 b_1 を求め、それと a_3 の最大公約数 b_2 を求めるという計算を続け最大公約数 b_{n-2} と a_n の最大公約数として求めることができる。

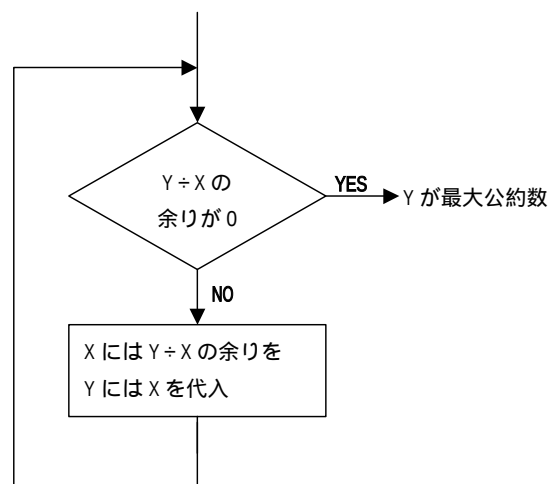


図 1 二つの自然数の最大公約数を求める方法

2.3 プログラムの流れ

プログラムの流れを次に示す．また，図示したものを図 2 に示す．

(1) 一つ目の自然数の入力

一つ目の自然数の入力を要求する．入力された数が 0 以下の場合は，再入力を要求する．0 で終了させないのは，一つも自然数が入力されないのを避けるためである．

(2) 二つ目以降の正の整数の入力

二つ目以降の自然数の入力を要求し，一つの自然数を入力させる．その値が 0 の場合，入力終了とみなす．このとき， a に格納されている値がこれまで入力した自然数の最大公約数となる．入力された数が負の場合は再度入力を要求する．

(3) 最大公約数の計算

これまでの最大公約数と直前に入力された自然数との最大公約数を計算し，それを a に格納する．

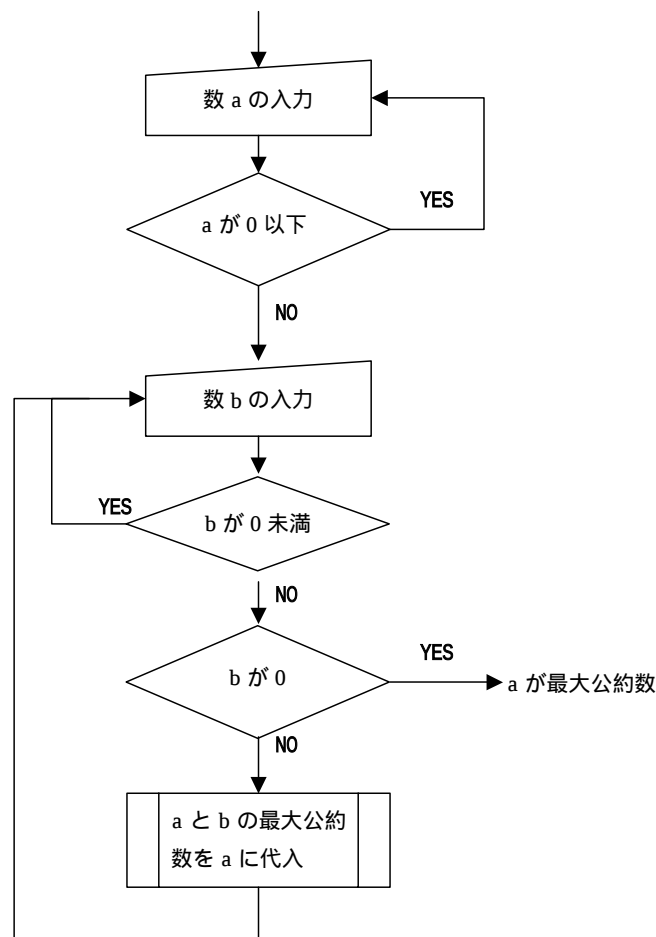


図 2 複数の自然数の最大公約数を求めるプログラムの流れ

3 実現

3.1 開発環境と実行環境

開発環境を表 1 に，実行環境を表 2 に示す．

表 1 開発環境

本体	Frontier 社製 IBM/PC 互換機
OS	Microsoft 社製 Windows2000
開発言語	Microsoft 社製 Visual C++ 6.0

表 2 実行環境

本体	IBM/PC 互換機
OS	Microsoft 社製 Windows98 以上，Windows2000 以上
実行形式	コンソール上

3.2 プログラムリストと説明

作成したプログラムを付録 1 に添付する．使用した変数の一覧を表 3 に示す．次にプログラムの流れに沿って，プログラムの説明を記す．

(1) 一つ目の自然数の入力

Do-while 文を利用し，1 以上の正数が入力されるまでループを繰り返す．Do-while 文を使うことで条件判定が入力後になるので，条件判定に用いる変数の初期化が必要ない．

(2) 二つ目以降の自然数の入力と最大公約数の計算

二つ目以降の自然数の入力部分は，二重ループとなっている．内側のループは負の値を無視するためのループで，do-while 文により実現している．外側のループは 0 が入力されるまでループするためのもので無限ループとし，入力された数が 0 である場合はこの無限ループから抜け，そうでない場合は最大公約数を求める．0 が入力された場合，最大公約数を求める処理を行わないようにするために，このような構造とした．

表 3 使用した変数

変数名	型	用途
a	Int	一つ目の自然数の入力用 求めた最大公約数の保管用
b	Int	二つ目以降の自然数の入力用
c	Int	最大公約数を求めるときの作業用

4 実行結果

作成したプログラムが正しく動作しているかどうかを調べるために、いくつかの数を入力したときの実行結果を次に示す。入力として 252, 660, 90 を入力した場合の実行結果を図 3 に、一つ目の数として 0 を入力した場合と負の数を入力した場合を図 4 に、二つ目以降の数として負の数を入力した場合の実行結果を図 5 に示す。なお、これらの図は実行画面のハードコピーである。

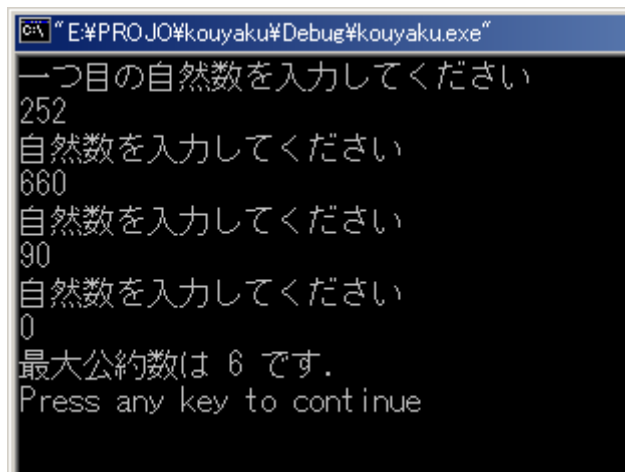


図 3 実行結果 1 (252,660,90 の最大公約数)

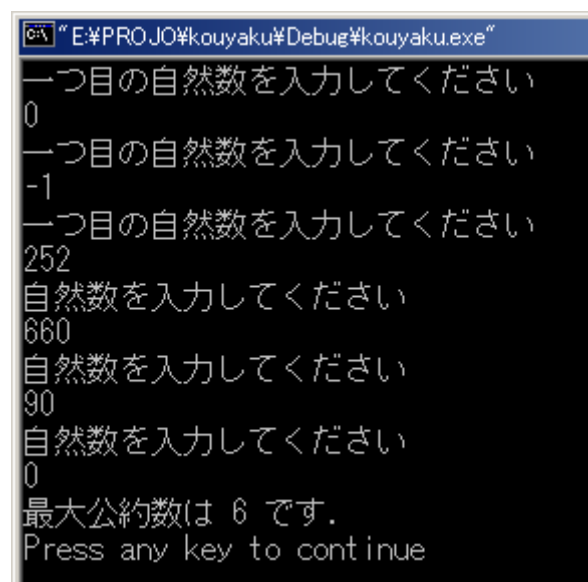
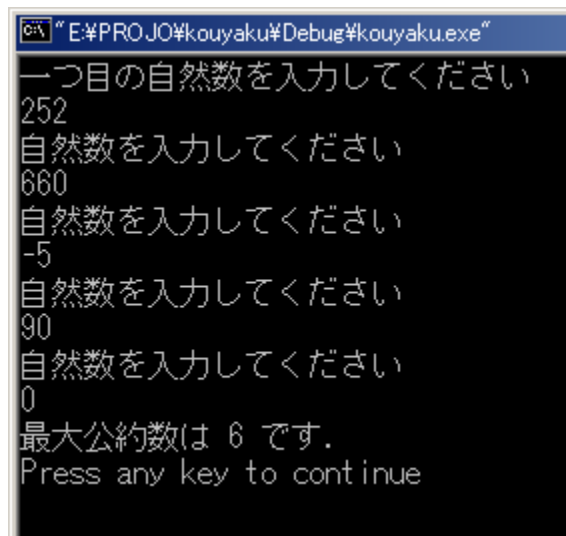


図 4 実行結果 2 (一つ目の数として 0 または負を入力した場合)



```
E:\PROJ0\kouyaku\Debug\kouyaku.exe
一つ目の自然数を入力してください
252
自然数を入力してください
660
自然数を入力してください
-5
自然数を入力してください
90
自然数を入力してください
0
最大公約数は 6 です。
Press any key to continue
```

図 5 実行結果 3 (二つ目以降の数として負の値を入力した場合)

5 考察

図 3 の実行結果で入力した 252, 660, 90 を素因数分解すると, それぞれ, $252=2 \times 2 \times 3 \times 3 \times 7$, $660=2 \times 2 \times 3 \times 5 \times 11$, $90=2 \times 3 \times 3 \times 5$ となるので, 最大公約数は $6=2 \times 3$ である. 図 3 より, 正しく求まっていることがわかる. また, 一つ目の数として 0 または負の数を入力した場合, 二つ目以降の数として負の値を入力した場合, その値は最大公約数の計算に含まれておらず, きちんと再入力が要求されていることが図 4,5 からわかる. これらの結果から, 作成したプログラムは正常に動作していることがわかる.

二つ目の数として 0 を入力すると, 最大公約数として一つ目に入力した数が表示される. 最大公約数は二つ以上の整数を割り切ることができる最大の整数という定義であるから, この動作は正確ではない. 二つ目の数も一つ目の入力と同様に 1 以上の数が入力されるようにすべきであった.

6 おわりに

本レポートでは, 複数の自然数の最大公約数を求めるプログラムについて述べた. 異常値が入力された場合には, その値を無視するようにすることができた. 一回目のレポートということで簡単なプログラムになってしまったが, 二回目のレポートまでにはもっと大きなプログラムが作れるようになりたいと考えている.

参考文献

[1] 金子他, プログラミング序論 2001 年度版, 東京農工大学情報コミュニケーション工学科.

付録 作成したプログラムのリストソース

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

void main()
{
    int a,b,c;

    /*** 0 より大きな数を入力 ***/
    do
    {
        printf( "一つ目の自然数を入力してください\n" );
        scanf( "%d", &a );
    }while( a <= 0 );

    /*** 無限ループ ***/
    while( 1 )
    {
        /*** 0 以上の数を入力 ***/
        do
        {
            printf( "自然数を入力してください\n" );
            scanf( "%d", &b );
        }while( b < 0 );

        /*** 0 だったらループをぬける ***/
        if( b == 0 ) break;

        /*** a と b の最大公約数を a に格納 ***/
        while( b%a != 0 )
        {
            c = b%a;
            b = a;
            a = c;
        }

    }

    /* 表示 */
    printf( "最大公約数は %d です.\n", a );
}
```