

# 第5回課題

## 課題(10-Equations.pdfの課題19)

10-Equations.pdfの課題19を読んで、連立一次方程式をLU分解を用いて解くプログラムをピボッティングを行うよう拡張してください。なお、LU分解を行うプログラムの解説は、10-Equations-code.pdfにて示しております。

### 入力形式

以下のように表される連立一次方程式の解を求めるとしています。

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 = b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 = b_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 = b_3$$

$$a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 = b_4$$

```
(a11の値) (a12の値) (a13の値) (a14の値)  
(a21の値) (a22の値) (a23の値) (a24の値)  
(a31の値) (a32の値) (a33の値) (a34の値)  
(a41の値) (a42の値) (a43の値) (a44の値)  
(b1の値) (b2の値) (b3の値) (b4の値)
```

入力は**小数**でも受け付けられるようにしてください。**倍精度浮動小数点**の範囲("double"型の範囲)で受け付けられるようにしてください。

### 出力形式

出力は以下のような形式で行ってください。

```
(x1の値)  
(x2の値)  
(x3の値)  
(x4の値)
```

出力行数は**全体最大10桁で小数部5桁**("%10.5lf")で行ってください。

## 具体例

### 入力例1

```
5 4 1 0
0 0 3 1
3 2 1 1
2 1 2 1
16 2 12 9
```

### 出力例1

```
4.00000
-1.00000
-0.00000
2.00000
```

## 注意事項

入力形式が**倍精度浮動小数点**になっていることに注意してください. ("double型"の範囲)

連立一次方程式の未知変数は4つと仮定して、プログラムを作成してください. (10-Equations-code.pdfのように、  
M=4と決めてプログラムを作成して良いということです。)

## コピペ用プログラム

LU分解を用いた連立方程式の解法を入出力を今回の仕様通りに書き換えたものです。中間出力はコメントアウトしています。

```
#include <stdio.h>
#define M 4

int main(){

    //変数宣言
    double a[M][M];
    double b[M];
    double c[M];
    double l[M][M];
    double u[M][M];
    double x[M];
    int i, j, k;

    //入力データの受け取り
    for(i=0;i<M;i++){
        for(j=0;j<M;j++) scanf("%lf",&a[i][j]);
    }
    for(i=0;i<M;i++) scanf("%lf",&b[i]);

    //L行列, U行列の初期化

    //入力行列の出力(ピボッティングでa行列を操作するため事前に出力)
    /*
    printf("入力行列\n");
    for(i=0;i<M;i++){
        for(j=0;j<M;j++) printf("%10.5lf",a[i][j]);
        printf("%10.5lf\n",b[i]);
    }
    */

    //結果の出力
    //L行列
    /*
    printf("\nL行列\n");
    for(i=0;i<M;i++){
        for(j=0;j<M;j++) printf("%10.5lf",l[i][j]);
        printf("\n");
    }
    */

    //U行列
    /*
    printf("\nU行列\n");
    for(i=0;i<M;i++){
        for(j=0;j<M;j++) printf("%10.5lf",u[i][j]);
        printf("\n");
    }
    */
}
```

```
*/  
  
//解の出力  
for(i=0;i<M;i++) printf("%10.5lf\n",x[i]);  
}
```