104 學年度 第一學期 資訊一甲
計算機概論實習 第四次平時上機考試

第一題請使用檔案名稱 "quiz4_dxxxxxxx_1.c", 第二題請使用檔案名稱 "quiz4_dxxxxxxx_2.c". "dxxxxxxx" 為你的學號。同時，請在每個程式原始碼的第一行，以註解方式寫上你的學號、及姓名。未使用規定之檔名或未寫上學號、及姓名者，本次成績將被扣 10% 的分數。當你作答完畢後，請將程式的原始碼 (source code, .c 檔) 上傳到教師機台。

1. (50 points) 你可使用 **quiz4_skeleton_1.c** 的程式架構，並先將其改名為 **quiz4_dxxxxxxx_1.c**. 這個題目是寫一個魔術方陣 (magic square) 的變化題。一個 n 階 (n 為奇數) 的魔術方陣即是一個 $n \times n$ 的二維方陣，它的元素從 1 到 n^2 (不重複)，且方陣中的各個橫列 (row)、直行 (column)、對角斜線 (diagonal)、和反對角線 (anti-diagonal) 上數字之加總起來的和都相等，即是 $(1+n^2)n/2$。一個 n 階魔術方陣產生的演算法如下：
   a) 將 1 放在最左邊行的中間元素。
   b) 放了 k 之後，下次則放 $k+1$，直到 n^2 為止。
   c) 每次要擺放下一個數時，往左下角的方格放。想像中，方陣的上下環狀相連，左右亦環狀相連。
   d) 若下一個欲擺放的值為 $k+1$，而左下角方格已經有放置數字，則將 $k+1$ 擺放在 k 右邊之方格。

   以下是 9 階魔術方陣的執行範例：

```
D:\課程\A1-計算機概論\Exam\Quiz\Au104\Lab\quiz4_1.exe
Enter the size of the magic square: 9

  47  57  67  77   6  16  26  36  37
  58  68  78   7  17  27  28  38  48
  69  79   8  18  19  29  39  49  59
  80   9  10  20  30  40  50  60  70
   1  11  21  31  41  51  61  71  81
  12  22  32  42  52  62  72  73   2
  23  33  43  53  63  64  74   3  13
  34  44  54  55  65  75   4  14  24
  45  46  56  66  76   5  15  25  35

Horizontal row 0 check pass.
Horizontal row 1 check pass.
Horizontal row 2 check pass.
Horizontal row 3 check pass.
Horizontal row 4 check pass.
Horizontal row 5 check pass.
Horizontal row 6 check pass.
Horizontal row 7 check pass.
Horizontal row 8 check pass.
Vertical column 0 check pass.
Vertical column 1 check pass.
Vertical column 2 check pass.
Vertical column 3 check pass.
Vertical column 4 check pass.
Vertical column 5 check pass.
Vertical column 6 check pass.
Vertical column 7 check pass.
Vertical column 8 check pass.
Diagonal check pass.
Anti-diagonal check pass.

The magic square is correct. The total sum is 369.
```

(續下一頁)

2. (50 points) 你可使用 **quiz4_skeleton_2.c** 的程式架構，並先將其改名為 **quiz4_dxxxxxxx_2.c**. 這個題目是寫一個 C 語言的 LU-分割 (LU-decomposition) 和三角矩陣乘法的程式。LU-分割問題就是將一個 n´n 的矩陣 A 分割成兩個 n´n 的三角矩陣 L 和 U。其中 L 是一個下三角矩陣 (lower triangular matrix) 且其對角線 (diagonal) 的元素為 1，U 是一個上三角矩陣 (upper triangular matrix) 且其對角線的元素都不為 0。以下為 LU-分割的矩陣描述：

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & \cdots & a_{0,n-1} \\ a_{1,0} & a_{1,1} & a_{1,2} & \cdots & a_{1,n-1} \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ a_{n-2,0} & a_{n-2,1} & \cdots & a_{n-2,n-2} & a_{n-2,n-1} \\ a_{n-1,0} & a_{n-1,1} & \cdots & a_{n-1,n-2} & a_{n-1,n-1} \end{bmatrix},$$

$$L = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ l_{1,0} & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ l_{n-2,0} & l_{n-2,1} & \cdots & 1 & 0 \\ l_{n-1,0} & l_{n-1,1} & \cdots & l_{n-1,n-2} & 1 \end{bmatrix},$$

$$U = \begin{bmatrix} u_{0,0} & u_{0,1} & \cdots & u_{0,n-2} & u_{0,n-1} \\ 0 & u_{1,1} & u_{1,2} & \cdots & u_{1,n-1} \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ 0 & \cdots & 0 & u_{n-2,n-2} & u_{n-2,n-1} \\ 0 & 0 & \cdots & 0 & u_{n-1,n-1} \end{bmatrix},$$

$$A = LU.$$

如果 $A^{(k)}$ 代表將 A 的前面 k 行和 k 列移除後的 $(n-k)\times(n-k)$ 矩陣，即

$$A^{(0)} = A,$$

$$A^{(k)} = \begin{bmatrix} a_{k,k} & a_{k,k+1} & \cdots & \cdots & a_{i,n-1} \\ a_{k+1,k} & a_{k+1,k+1} & a_{k+1,i+2} & \cdots & a_{i+1,n-1} \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ a_{n-2,k} & a_{n-2,k+1} & \cdots & a_{n-2,n-2} & a_{n-2,n-1} \\ a_{n-1,k} & a_{n-1,k+1} & \cdots & a_{n-1,n-2} & a_{n-1,n-1} \end{bmatrix}, k < n.$$

自 $A^{(0)}$ 開始到 $A^{(n-1)}$， LU-分割的演算法，即是每個 $A^{(k)}$ 執行下列三個步驟：

1. 計算矩陣 U 的第 k 列元素：$u_{k,j} = a_{k,j}$, for $k \le j \le n-1$,
2. 計算矩陣 L 的第 k 行元素：$l_{i,k} = a_{i,k} / a_{k,k}$, for $k \le i \le n-1$,
3. 計算矩陣 $A^{(k+1)}$ 的元素：$a_{i,j} = a_{i,j} - l_{i,k} \times u_{k,j}$, for $k<i, j \le n-1$.

寫一個 C 語言的程式以執行以下的步驟：

1. 宣告 A、$A1$、L、和 U 為 100×100 的二維陣列；
2. 輸入正整數 n 的值 (小於或等於 100)；
3. 使用副程式 rand() 隨機產生矩陣 A 元素的浮點數值 (假設 A 元素的值在 0 和 1 之間，並取小數點以下六位)；同時，複製矩陣 A 的值到矩陣 $A1$；
4. 輸出矩陣 A (至小數點以下四位)；
5. 計算 LU-分割的矩陣 L 和 矩陣 U 的值，以使得 $A=LU$；
6. 使用三角矩陣乘法，將 L 和 U 相乘以還原 A；
7. 輸出矩陣 L 和 U (至小數點以下四位)；

8. 驗算矩陣 *A* 的是否等於 *A*1 (假設誤差值小於 0.0001)。

將 LU 分割和三角矩陣乘法各寫成一個副程式

void LU_decompose(**int** n, **float** A[100][100], **float** L[100][100], **float** U[100][100]);
void matrix_product_triangular(**int** n, **float** A[100][100], **float** B[100][100],
float C[100][100]);

以下為程式執行範例：

```
D:\課程\A1-計算機概論\Exam\Quiz\Au104\Lab\quiz4_2.exe
Enter matrix size n: 8

Matrix A:
  0.0064    0.0146    0.0092    0.0154    0.0315    0.0119    0.0300    0.0108
  0.0110    0.0221    0.0232    0.0193    0.0050    0.0303    0.0149    0.0034
  0.0208    0.0174    0.0034    0.0291    0.0210    0.0230    0.0052    0.0185
  0.0145    0.0245    0.0282    0.0135    0.0221    0.0267    0.0028    0.0104
  0.0211    0.0198    0.0115    0.0120    0.0212    0.0022    0.0278    0.0134
  0.0137    0.0205    0.0285    0.0110    0.0140    0.0315    0.0200    0.0119
  0.0041    0.0289    0.0219    0.0043    0.0270    0.0102    0.0066    0.0276
  0.0115    0.0307    0.0047    0.0091    0.0052    0.0169    0.0114    0.0307

Matrix L:
  1.0000
  1.7200    1.0000
  3.2390   10.2033    1.0000
  2.2644    2.9202    0.1389    1.0000
  3.2929    9.6366    0.8837    2.0855    1.0000
  2.1260    3.5846    0.1717    0.6575   -0.4940    1.0000
  0.6405   -6.6859   -0.6414    2.6107    2.6308  -17.6913    1.0000
  1.7848   -1.6104    0.0005    3.9409    4.9061  -38.5865    2.3068    1.0000

Matrix U:
  0.0064    0.0146    0.0092    0.0154    0.0315    0.0119    0.0300    0.0108
           -0.0029    0.0073   -0.0072   -0.0491    0.0098   -0.0366   -0.0152
                     -0.1011    0.0528    0.4201   -0.1157    0.2819    0.1388
                               -0.0076    0.0359   -0.0129    0.0027    0.0111
                                         -0.0554   -0.0025    0.0272   -0.0213
                                                   -0.0019    0.0508    0.0019
                                                              0.7433    0.0678
                                                                       -0.0374

The LU-decomposition program is correct.
```