期末程式考試(Open Book)：準時上傳 Q1.c, Q1.pdf

矩陣計算機

(I)考試規定說明：

1. 同學間不能以任何形式討論或傳遞資料
2. 不能向他人或在網路上提問
3. 只可參考書籍、網頁查詢資料、及其他參考資料
4. 獨立完成程式

(II) “矩陣計算機”程式碼與報告說明：

1. 程式 Q1.c

   全英文(不能含有任何特殊符號或中文以避免亂碼)

   有參考到其他程式碼來源時必須在”矩陣計算機” 程式碼參考到”參考資料”位置處之前加入明顯註解，例如

   ```
   /*=========
    102~125[x]
   */=========
   ```

   表示程式碼 102~125 行係參考資料來源[x]

2. 報告 Q1.pdf

   可用中文書寫

   須包含”矩陣計算機”流程圖”(附錄 1)

   參考資料來源(標註編號[x]，例如)

   [1] 作者、書名、頁碼…

   [2] https://baike.baidu.hk/item/矩陣乘法/5446029

   不能含有像如下有%不易理解的編碼

   https://baike.baidu.hk/item/%E7%9F%A9%E9%99%A3%E4%B9%98%E6%B3%95/5446029

   ~~填寫功能表格並勾選已完成功能(附錄 2)~~

   填寫矩陣計算機程式功能自評表格(附錄 2)

(III) “矩陣計算機”程式輸入檔格式

測試輸入檔資料格式一定依照下列說明建置，不須解析錯誤

1. 輸入行可能有前置空格與尾隨空格，輸入行有四類：
2. 註解行：此行第一個非空白字母為#，註解行必須跳過
3. 矩陣名稱行：此行第一個非空白字母為矩陣名稱，是一個大寫英文字母 A~Z，其後為矩陣列數(row)與行數(column)。矩陣名稱 X、列數(row)n 與行數(column)m 之間至少有一個以上的空格

   例如 X n m

   表示 X 是一個大小為 n x m 的矩陣，在此之後輸入檔會有 n 個有效資料行(不包含註解行)，而每個資料行會有 m 個雙精度浮點數(double)
4. 資料行：依矩陣名稱行的資料，有 m 個雙精度浮點數(double)，必須以 double 型態讀取
5. 空白行

(IV)程式功能

由鍵盤讀取指令(提示符號"$ ")，一行一個指令。測試指令格式必定依下列說明下達，不須解析錯誤

1. 指令各符號間可有 0, 1, 或多個空格
2. 計算指令：

    矩陣可用名稱有 A~Z

    矩陣運算符號 op1 ： +(加), -(減), *(乘)

    矩陣運算符號 op2 ： +(正), -(負)

    矩陣運算符號 op3 ： =(儲存運算結果)

    每個計算指令行只有一個運算符號(op1 或 op2)及一個運算符號(op3)

    每個運算結果必須儲存在等號=左邊的矩陣名稱

    合法指令：

A =B + C	
A= A * C	等號=左右兩邊都有 A 表示結果會覆蓋 A 原始資料
A = A + B	允許多個空格

    不合法指令：

B + C	沒有等號=儲存結果
A= -B + C	有兩個運算符號 -、+
A = B + C + D	有兩個運算符號 +、+

3. 子矩陣，例如由 B 矩矩陣取出 row 0,1,2(不含 3)，column 12, 13, 14, 15(不含 16)的 3x4 的子矩陣

    A = B[0:3,12:16]

4. 其他指令

    ? ：輸出已有的所有矩陣狀態，包含矩陣名稱、列數(row)與行數(column)(附錄 3)

    ?X ：X 為矩陣名稱，銀幕輸出該矩陣名稱、列數(row)、行數(column)與所有矩陣元素值

    輸出格式%15.5e、輸出要對齊((附錄 4)

    < filename.txt ：讀取檔案內所有矩陣

    > filename.txt ：將已有的所有矩陣輸出至檔案，輸出檔格式與輸入檔格式相同但儲存的數字盡可能精確。此輸出檔可 讚用指令 < 再讀取

    !X ：刪除矩陣名維 X 的矩陣資料

    !!d ：刪除所有矩陣資料

    !!q ：結束程式

(V)顯示錯誤訊息

檔案錯誤(File error.)、矩陣不存在(Matrix not exist.)、矩陣大小不相容(Matrix size imcompatible.)、矩陣索引超出範圍(Matrix out of index.)、記憶體不足(Memory allocation error.)

(VI)提示

1. 銀幕與檔案輸出

```
printf("OK\n");//用於銀幕輸出，
```

fprintf(fptr, “OK\n”);//用於檔案輸出
因為 stdout 內定連結銀幕(若無重新導向)，所以銀幕輸出也可寫成
fprintf(stdout, “OK\n”);

2. 鍵盤與檔案輸入
scanf(“%d”, &var);//用於鍵盤輸入，
fscanf(fptr, “%d”, &var); //用於檔案輸入
因為 stdin 內定與鍵盤連結(若無重新導向)，所以鍵盤輸入也可寫成
fscanf(stdin, “%d”, &var);

3. 程式如何預先偷看檔案資料再決定如何做
int c = getc(fptf); // 讀取一個字元
unget(c, fptr); // 放回一個字元

4. 使用 ch14.pdf 34~37 頁方法動態配置二維陣列

5. 善用 C 結構 struct 儲存”矩陣計算機”的矩陣資料

6. 善用 C 一維陣列儲存所有”矩陣計算機”的矩陣資料

7. 將”矩陣計算機”所需要的每一個小功能寫成一個 C 函式，例如讀取檔案時的
跳過註解行、跳過前置空格、跳過尾隨空格、跳過一行等等

8. V1.2 新增由下列網頁查詢 C 函式用法
https://www.cplusplus.com/

9. V1.2 新增 scanf、fscanf、sscanf 會由不同資料源讀取數據，三者用法幾乎沒有差別
其中 sscanf 的資料來源為字串，例如：

```
char str [80]="A=[1:2]";
int x, y, m, cnt;
char c;
m=sscanf(str,"%c=[%d:%d]%n",&c, &x, &y, &cnt);
printf ("%c %d %d %d %d\n", c, x, y, cnt, m);
```

"%c=[%d:%d]%n"這格式依序匹配：
%c 一個任意字元、一個等號字母=、一個字母[、%d 一個任意整數、一個冒號字母:、%d 一個任意整數、一個字母]而%n 讓 sscanf 告訴你"到目前為止從字串 str 解析了多少個字元"而其中%x 是要依格式 x 讀回的格式碼。sscanf 要由 str 讀取%c, %d, %d 三個，所以傳回值 m=3 (%n 不是讀取值，不算在內)

10. V1.2 新增“矩陣計算機”檔案解析
a. 無法預知檔案每行有多少字元，所以只能一個一個讀入字元加以檢查

b. 前置空格處理：讀進每一個字元，若是空格則放棄直到讀進非空格字元止
c. 用 unget()將此非空格字元放回檔案緩衝區
d. 讀進一個字元，依此字元判斷是註解行、矩陣名稱行或資料行
e. 依行的種類做必要處置
f. 尾隨空格處理：不管是哪種類的行都可能有尾隨空格，讀入每一個字元並放棄直到讀入'\n'為止
g. 重複步驟 b~f

11. V1.2 新增"矩陣計算機"命令列解析
a. 使用 char *fgets(char * str, int num, FILE *stream);
由鍵盤讀取(若 stream 為 stdin)整行命令列字串
b. 去除命令列字串中的所有空格
c. 依命令列字串第一個字元是'A'~'Z'、'?'、'!'做初步命令分類
d. 依初步命令分類做必要處置
e. 重複步驟 a~d

12. V1.3 新增矩陣動態記憶體配置程式範例(附錄 5)

13. V1.3 新增開啟檔案
https://www.cplusplus.com/reference/cstdio/fopen/?kw=fopen
FILE *fopen(const char *filename, const char *mode);
fopen 函數原型表明第一個參數是字串，給字串常數或字串變數皆可
a. 字串常數：
FILE *fptr = fopen("testfile.txt",...);
b. 由鍵盤讀入的字串變數：
```
char filename[128];
scanf("%s", filename); // 輸入不能多於 127 個 characters
FILE *fptr = fopen(filename, ...);
```

14. V1.3 新增矩陣計算機指令有下列 13 種，◯為矩陣名稱 A~Z；◯、=、+、-、*、?、!、d、q 各符號之間可能有零個、一個、或多個的空格隔開。aaa, bbb, ccc, ddd 為矩陣的索引值；
◯ = ◯ + ◯
◯ = ◯ - ◯
◯ = ◯ * ◯
◯ = +◯
◯ = -◯
◯ = ◯[aaa : bbb, ccc : ddd]
?
?◯
< filename
> filename
!◯

!!d

!!q

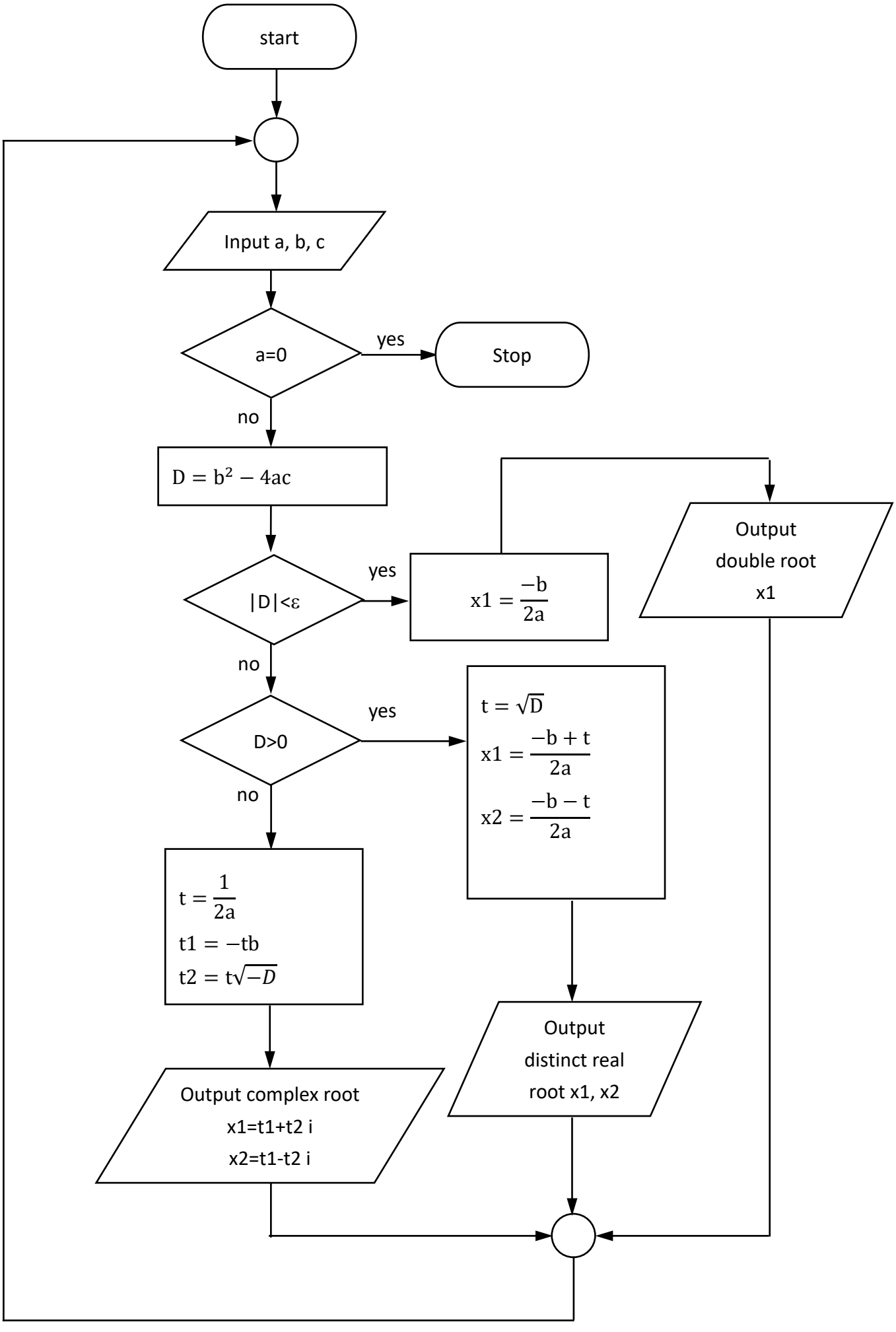
15. V1.3 新增...

(VII)測試資料(更多的測試資料之後再附)

1. TestFile_1.txt
2. TestFile_2.txt
3. TestFile_3.txt
4. V1.2 新增 TestFile_4.txt
5. V1.3 新增 TestFile_5.tx

(VII) 附錄

附錄 1 一元二次方程式求解流程圖範例



附錄 2 V1.3 新增矩陣計算機程式功能自評表格

矩陣計算機程式功能自評表格：通過指令組測試且結果正確才能勾選 V (輸出結果可與範例程式 FinalExam.exe 比較)

自評方式：不能更改測試檔內容，除第 1、12 組外，其餘依自訂順序執行測試直到結束。

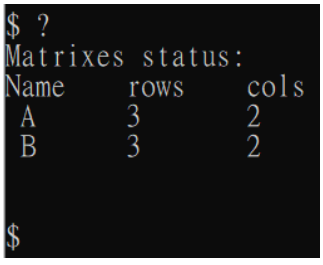
測試時不能中途結束你的程式，然後再重新執行你的程式

矩陣計算機程式功能自評表格(填寫測試順序，通過指令組測試且結果正確才能勾選 V)

組	功能勾選	測試順序	指令組	功　能
1		1	< TestFile_1.txt ? ?A !!d ? ?A	第 1 組指令務必達標才能繼續完成其他測試 正確讀取測試檔 TestFile_1.txt 顯示目前所有矩陣名稱與大小 輸出 A 矩陣所有元素 確認已刪除所有矩陣
2			< TestFile_2.txt ? ?B !!d	正確讀取測試檔 TestFile_2.txt
3			< TestFile_3.txt ? ?B !!d	正確讀取測試檔 TestFile_3.txt
4			< TestFile_1.txt ?Q !!d	判別不存在矩陣
5			< TestFile_1.txt D=-A ?D D=+A ?D !!d	矩陣正(+)負(-)運算
6			< TestFile_1.txt C=A+B ?C A=A+B ?A D=A-B ?D D=A+Q !!d	兩矩陣加(+)減(-)運算 運算結果儲存 運算結果儲存，但會覆蓋原矩陣 不存在矩陣的運算

7			< TestFile_5.txt ? D=B*A ?D D=B*C ?D !!d	矩陣乘法 矩陣大小不合無法相乘
8			< TestFile_4.txt Z=X*Y ?Z W=U*V !!d	太大的矩陣不要在銀幕輸出，以免輸出很久很久 矩陣乘法 視情況而定記憶體可能不足
9			< TestFile_5.txt ?C D=C[0:2,1:3] ?D F=C[1:3,0:2] ?F !!d	取出子矩陣 取出子矩陣，索引超出範圍
10			< TestFile_5.txt ?A !A ?A ? !!d	刪除矩陣 程式結束
11			< TestFile_1.txt !B ? > out.txt	矩陣輸出
12		12	!!q	第 12 組指令務必於最後測試

附錄 3 銀幕輸出已存在的所有矩陣狀態



附錄 4 銀幕輸出 A 矩陣資料

```
$ ?A
A 3 2
    1.00000e+00     2.00000e+00
    3.00000e+00     4.00000e+00
    5.00000e+00     6.00000e+00


$
```

附錄 5 矩陣動態記憶體配置程式範例

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

double **Array2D(size_t rows, size_t cols)
{
    double **p = (double **)malloc(rows*sizeof(double *)
                                   + rows*cols*sizeof(double));
    if(p==NULL)
        return NULL;

    for(int i=0; i<rows; i++)
        p[i] = (double *)(p + rows) + i*cols;

    return p;
}

int main()
{
    int n = 3, m = 4;

    // Allocate n x m 2D array
    double **my2DArray = Array2D(n, m);
    if(my2DArray==NULL)
    {
        puts("Memory allocation error.");
        return 1;
    }

    int cnt = 0;
    for(int i=0; i<n; i++)
        for(int j=0; j<m; j++)
            my2DArray[i][j] = cnt++;

    for(int i=0; i<n; i++)
    {
        for(int j=0; j<m; j++)
            printf(" %10f", my2DArray[i][j]);
        printf("\n");
    }

    free(my2DArray);

    return 0;
}
```