

Turnin作業8

- chapter 8
- Turnin Code: **c.hw8**
- Due Date: 2025/12/02 23:59 **Hard Deadline**

繳交方式說明

本次Turnin作業包含7個程式題，建議可以為它們建立一個資料夾，並將此次作業所有要上傳的檔案放置於該資料夾後，再使用 `turnin` 指令上傳作業。例如下面的操作在同學的家目錄裡建立了一個名為hw8的資料夾，並且切換到該資料夾後，針對第1題所要求的GetPositiveNumbers.c進行程式編撰：

```
[user@ws ~]$ mkdir hw8
[user@ws ~]$ cd hw8
[user@ws hw8]$ joe GetPositiveNumbers.c
```

等到我們完成GetPositiveNumbers.c的撰寫後，請自行加以編譯與執行程式，確認正確後使用 `turnin c.hw8 GetPositiveNumbers.c` 指令將加以上傳：

```
[user@ws c.hw8]$ turnin c.hw8 GetPositiveNumbers.c
Turning in:
  GetPositiveNumbers.c -- ok
All done.
[user@ws c.hw8]$
```

當然，你也可以等到本次作業要求的所有題目都在hw8資料夾裡完成後，再使用 `turnin c.hw8 .` 指令一次將所有在目前資料夾中的所有檔案都加以上傳。假設你已經在hw8資料夾裡撰寫完所有題目，並且每個題目的程式檔案皆已編譯並確認執行結果正確後，我們可以使用以下指令將多餘的(不需要繳交的)檔案加以刪除後，一次將所有檔案繳交：

```
[user@ws c.hw8]$ ls
Determinant.c          GetPositiveNumbers.c  ScoreStats.c
FindNumber.c          ReverseMatrix.c       MatrixCalcInteractive.c
ScoreBarchart.c       a.out
[user@ws c.hw8]$ rm a.out
[user@ws c.hw8]$ ls
Determinant.c          GetPositiveNumbers.c  ScoreStats.c
FindNumber.c          ReverseMatrix.c       MatrixCalcInteractive.c
ScoreBarchart.c
```

```
[user@ws c.hw8]$ turnin c.hw8 .
Turning in:
./FindNumber.c -- ok
./ScoreStats.c -- ok
./ReverseMatrix.c -- ok
./MatrixCalcInteractive.c -- ok
./GetPositiveNumbers.c -- ok
./ScoreBarchart.c -- ok
./Determinant.c -- ok
All done.
[user@ws c.hw8]$
```

如果繳交後想要查看已繳交的檔案及相關資訊，可以輸入`turnin -ls c.hw8` 指令，例如：

```
[user@ws c.hw8] c.hw8 $ turnin -ls c.hw8
.:
total 28
-rw-rw----. 1 turninman turnin 740 Nov 25 21:41 Determinant.c
-rw-rw----. 1 turninman turnin 1545 Nov 25 21:41 FindNumber.c
-rw-rw----. 1 turninman turnin 2774 Nov 25 21:41 MatrixCalcInteractive.c
-rw-rw----. 1 turninman turnin 1083 Nov 25 21:41 GetPositiveNumbers.c
-rw-rw----. 1 turninman turnin 2524 Nov 25 21:41 ReverseMatrix.c
-rw-rw----. 1 turninman turnin 990 Nov 25 21:41 ScoreBarchart.c
-rw-rw----. 1 turninman turnin 2870 Nov 25 21:41 ScoreStats.c
```

註：本文使用 及 代表空白字元與Enter換行字元，並且將使用者輸入的部份使用灰階方式顯示。另外，出現在程式執行結果中的符號皆為半形。

P1 輸出陣列中的元素 `GetPositiveNumbers.c`

請設計一個 C 語言程式 `GetPositiveNumbers.c` 讓使用者依序輸入不超過 10 個正整數。在使用者輸入的過程中，如果輸入的數值為 0 則提前結束輸入；若輸入的是負整數或是其他字元，則請輸出 `Error! Please input again!` 並要求使用者再次輸入。完成使用者輸入後，請將其所輸入內容依序輸出。此題的執行結果可參考以下的畫面：

【註：此題有使用程式碼區塊段落進行排版，因此有用到 `$#1==>?10 $` 的部分因請 `dokowiki` 形式格式會跑掉成 `#1= ?10`，故使用程式碼區塊，輸入數值為 `?` 後的數字，其餘部分為程式輸出，請同學們注意。】

```
[3:23 user@ws hw] ./a.out
Please input 10 positive numbers (0 to terminate).
#1==>?10
```

```
#2==>?20↵
#3==>?30↵
#4==>? -40↵
```

Error!▲Please▲input▲again!↵

```
#4==>?q↵
```

Error!▲Please▲input▲again!↵

```
#4==>?4↵
#5==>?0↵
```

```
Input▲terminated!↵
Numbers[10,20,30,4]↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲10▲positive▲numbers▲(0▲to▲terminate).↵
```

```
#1==>?2↵
#2==>?4↵
#3==>?6↵
#4==>?8↵
#5==>?10↵
#6==>?12↵
#7==>?14↵
#8==>?16↵
#9==>?18↵
#10==>?20↵
```

```
Numbers[2,4,6,8,10,12,14,16,18,20]↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲10▲positive▲numbers▲(0▲to▲terminate).↵
```

```
#1==>?x↵
```

Error!▲Please▲input▲again!↵

```
#1==>? -99↵
```

Error!▲Please▲input▲again!↵

```
#1==>?q↵
```

Error!▲Please▲input▲again!↵

```
#1==>?0↵
```

```
Input▲terminated!↵
Numbers[empty]↵
[3:23 user@ws hw]
```

P2 反轉矩陣 ReverseMatrix.c

請設計一個 C 語言程式 ReverseMatrix.c 讓使用者先輸入一個矩陣的大小 (列 x 行)，接著輸入該矩陣的每一列內容。程式需先輸出原始矩陣，並請使用者選擇要以 水平反轉 [h] 或 垂直反轉 [v] 的方式進行陣列反轉。最後依照選擇輸出反轉後的矩陣。

【題目輸入要求】

1. 一組矩陣大小 (格式：列 x 行，兩者皆為正整數)，如果輸入非數字字元則輸出Error!並結束程式。為便利起見，本題所輸入的列與行皆不會超過10，因此若使用者輸入的列或行超過10，也要輸出Error!並結束程式。
2. 各列的矩陣內容，如果輸入非數字字元則輸出Error!並結束程式。
3. 反轉方式選擇：如下表

輸入字元	功能說明
h	水平反轉 (每一列左右交換)
v	垂直反轉 (整個矩陣上下交換)

如果輸入非上述字元，則顯示Error!並結束程式。

【註：本題的列x行的x為英文字母的小寫x並非數學乘號，請同學注意。】

本題執行結果可參考以下輸出內容：

```
[3:23 user@ws hw] ./a.out
Please input the size of the matrix?x?(positive integer):3x3
Please input matrix row 1:1 2 3
Please input matrix row 2:2 3 4
Please input matrix row 3:4 5 6
Original array:
1 2 3
2 3 4
4 5 6
Please input which way would you like to reverse:h
Reversed array horizontal reversed:
3 2 1
4 3 2
6 5 4
[3:23 user@ws hw] ./a.out
Please input the size of the matrix?x?(positive integer):3x3
Please input matrix row 1:1 2 3
Please input matrix row 2:2 3 4
Please input matrix row 3:4 5 6
Original array:
1 2 3
2 3 4
4 5 6
Please input which way would you like to reverse:v
Reversed array vertical reversed:
4 5 6
```

```

2 3 4
1 2 3
[3:23 user@ws hw] ./a.out
Please input the size of the matrix?x?(positive integer): 1x2
Please input matrix row 1: 1 2
Original array:
1 2
Please input which way would you like to reverse: h
Reversed array horizontal reversed:
2 1
[3:23 user@ws hw] ./a.out
Please input the size of the matrix?x?(positive integer): 1x2
Please input matrix row 1: 1 2
Original array:
1 2
Please input which way would you like to reverse: v
Reversed array vertical reversed:
1 2
[3:23 user@ws hw] ./a.out
Please input the size of the matrix?x?(positive integer): 2x4
Please input matrix row 1: 1 2 3 4
Please input matrix row 2: 3 4 5 6
Original array:
1 2 3 4
3 4 5 6
Please input which way would you like to reverse: v
Reversed array vertical reversed:
3 4 5 6
1 2 3 4
[3:23 user@ws hw] ./a.out
Please input the size of the matrix?x?(positive integer): 2x4
Please input matrix row 1: 1 2 3 4
Please input matrix row 2: 3 4 5 6
Original array:
1 2 3 4
3 4 5 6
Please input which way would you like to reverse: h
Reversed array horizontal reversed:
4 3 2 1
6 5 4 3
[3:23 user@ws hw] ./a.out
Please input the size of the matrix?x?(positive integer): 11x1
Error!
[3:23 user@ws hw] ./a.out
Please input the size of the matrix?x?(positive integer): 1x11
Error!
[3:23 user@ws hw] ./a.out
Please input the size of the matrix?x?(positive integer): 0x1
Error!

```

```
[3:23 user@ws hw] ./a.out↵
Please input the size of the matrix?x?(positive integer):1x0↵
Error!↵
[3:23 user@ws hw] ./a.out↵
Please input the size of the matrix?x?(positive integer):- 1x - 1↵
Error!↵
[3:23 user@ws hw] ./a.out↵
Please input the size of the matrix?x?(positive integer):qx1↵
Error!↵
[3:23 user@ws hw] ./a.out↵
Please input the size of the matrix?x?(positive integer):2x3↵
Please input matrix row1:1 2 q↵
Error!↵
[3:23 user@ws hw]
```

P3 成績統計程式 ScoreStats.c

『題目敘述』

請撰寫一個 C 程式 ScoreStats.c 讓使用者輸入 10 筆學生成績（每筆皆應為 0 – 100 的整數，若有任一筆輸入不是整數或不在範圍內，則輸出 Error!）成績輸入完成後，使用者需輸入一個指令以查詢統計資訊，相關指令如下表：

指令	功能說明
m	顯示最高分 [max]
n	顯示最低分 [min]
s	顯示總分 [sum]
a	顯示平均分 [avg] (四捨五入至小數點後一位)
p	顯示及格 (60 分) 的人數 [pass]
q	結束程式

請依照使用者所輸入的指令輸出對應的統計值；若指令錯誤則輸出 Error!

【註：此程式若有使用小數計算請使用 double 型態作為變數宣告】

本題的執行結果可參考如下：

```
[3:23 user@ws hw] ./a.out↵
Please input 10 scores:↵
10 20 30 40 50 60 70 80 90 100↵
Please input command:m↵
Max score = 100↵
Please input command:n↵
Min score = 10↵
Please input command:s↵
Sum score = 550↵
Please input command:a↵
```

```

Average▲score▲=▲55.0↵
Please▲input▲command:▲p↵
Pass▲count▲=▲5↵
Please▲input▲command:▲q↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲10▲scores:↵
90▲80▲30▲55▲43▲11▲67▲22▲99▲22↵
Please▲input▲command:▲a↵
Average▲score▲=▲51.9↵
Please▲input▲command:▲q↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲10▲scores:↵
10▲20▲30▲-40▲50▲160▲70▲80▲90▲100↵
Error!↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲10▲scores:↵
10▲20▲30▲40▲50▲60▲70▲80▲90▲100↵
Please▲input▲command:▲x↵
Error!↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲10▲scores:↵
10▲20▲30▲40▲50▲60▲70▲80▲90▲100↵
Please▲input▲command:▲9↵
Error!↵
[3:23 user@ws hw]

```

P4 成績統計直方圖 [ScoreBarchart.c]

請設計一個 C 程式ScoreBarchart.c讓使用者可以連續輸入不超過100個學生的成績（0–100的整數，若輸入非數字或不在0–100範圍內，則程式須輸出 **Error!** 並結束。），直到輸入 -1 時表示輸入結束。程式需統計每個成績區間的人數（0–9, 10–19, ..., 90–100），並以簡單的橫式文字長條圖呈現（每滿5個學生使用一個*代表，剩餘不足5人時，則不予顯示）。例如，如果0–9區間有3人時，則不顯示任何星號；但如果0–9區間有23人時，則顯示4個星號。

此題的輸出結果可參考以下結果。

【註：此題有使用程式碼區塊段落進行排版，因此在Score▲Barchart:↵下方的區塊並非輸入，而是程式碼輸出，請同學們注意。】

```

[3:23 user@ws hw] ./a.out↵
Please▲input▲scores▲(-1▲to▲end):↵
35▲31▲30▲20▲70▲70▲70▲70▲70▲70▲32▲20▲33▲31▲20▲37▲30▲35▲31▲22▲70▲2
3▲28▲37▲0▲-1↵
Score▲Barchart:↵

```

```

▲0▲-▲▲9▲▲|▲↵
10▲-▲19▲▲|▲↵
20▲-▲29▲▲|▲*↵
30▲-▲39▲▲|▲**↵
40▲-▲49▲▲|▲↵
50▲-▲59▲▲|▲↵
60▲-▲69▲▲|▲↵
70▲-▲79▲▲|▲**↵
80▲-▲89▲▲|▲↵
90▲-100▲▲|▲↵

```

[3:23 user@ws hw] ./a.out↵

Please▲input▲scores▲(-1▲to▲end):↵

```

100▲100▲100▲100▲90▲90▲90▲90▲90▲90▲90▲90▲90▲90▲90▲90▲90▲93▲81▲80▲87▲80▲85▲71▲72▲
70▲63▲68▲67▲60▲60▲61▲60▲60▲61▲67▲67▲67▲66▲66▲66▲66▲60▲62▲66▲66▲-1↵

```

Score▲Barchart:↵

```

▲0▲-▲▲9▲▲|▲↵
10▲-▲19▲▲|▲↵
20▲-▲29▲▲|▲↵
30▲-▲39▲▲|▲↵
40▲-▲49▲▲|▲↵
50▲-▲59▲▲|▲↵
60▲-▲69▲▲|▲****↵
70▲-▲79▲▲|▲↵
80▲-▲89▲▲|▲*↵
90▲-100▲▲|▲***↵

```

[3:23 user@ws hw] ./a.out↵

Please▲input▲scores▲(-1▲to▲end):↵

-1↵

Score▲Barchart:↵

```

▲0▲-▲▲9▲▲|▲↵
10▲-▲19▲▲|▲↵
20▲-▲29▲▲|▲↵
30▲-▲39▲▲|▲↵
40▲-▲49▲▲|▲↵
50▲-▲59▲▲|▲↵
60▲-▲69▲▲|▲↵
70▲-▲79▲▲|▲↵
80▲-▲89▲▲|▲↵
90▲-100▲▲|▲↵

```

[3:23 user@ws hw] ./a.out↵

Please▲input▲scores▲(-1▲to▲end):↵

```

40▲10▲60▲40▲40▲40▲40▲40▲20▲30▲10▲20▲30▲10▲20▲30▲10▲60▲30▲40▲70▲80▲90▲20▲1
0▲14▲23▲25▲24▲28▲75▲84▲-1↵

```

Score▲Barchart:↵

```

▲0▲-▲▲9▲▲|▲↵
10▲-▲19▲▲|▲*↵

```

```

20▲-▲29▲▲|▲*↵
30▲-▲39▲▲|▲↵
40▲-▲49▲▲|▲*↵
50▲-▲59▲▲|▲↵
60▲-▲69▲▲|▲↵
70▲-▲79▲▲|▲↵
80▲-▲89▲▲|▲↵
90▲-100▲▲|▲↵

```

```

[3:23 user@ws hw] ./a.out↵
Please▲input▲scores▲(-1▲to▲end):↵
20▲29▲50▲23▲66▲33▲77▲q▲a↵
Error!↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲scores▲(-1▲to▲end):↵
-100↵
Error!↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲scores▲(-1▲to▲end):↵
150▲-1↵
Error!↵
[3:23 user@ws hw]

```

P5 互動式矩陣運算 `MatrixCalcInteractive.c`

請設計一個C語言程式MatrixCalcInteractive.c可處理兩個列與行皆不超過5的矩陣運算，程式功能如下：

1. 先詢問使用者要進行哪種矩陣運算：

運算子	說明
+	矩陣加法
-	矩陣減法

2. 提示使用者輸入第一個矩陣的維度(其列與行皆不超過5，且不可以小於等於0；若使用者輸入了其他的字元，則視為錯誤，請顯示Error!)至於第二個矩陣的維度則不需輸入，其列與行皆與第一個矩陣相同。

3. 決定好矩陣維度後，依提示要求輸入兩個矩陣的元素(皆為大於-999(含)到小於999(含)間的整數，若輸入不正確的元素時(包含維度不符、元素非整數或是非法運算子，請輸出 Error!並結束程式)後，計算並輸出運算結果。

【註：本題的列x行的x為英文字母的小寫x並非數學乘號，請同學注意。】

本題的輸出結果可參考如下：

```

[3:23 user@ws hw] ./a.out↵

```

```

Input operation (+/-): +
Enter matrix A size (m x n): 2 x 4
Enter matrix A (2 x 4):
1 4 2 6
3 5 2 6
Enter matrix B (2 x 4):
1 6 8 8
12 12 30 30
Result:
2 10 10 14
15 17 32 36
[3:23 user@ws hw] ./a.out
Input operation (+/-): -
Enter matrix A size (m x n): 3 x 2
Enter matrix A (3 x 2):
15 114
66 11
222 931
Enter matrix B (3 x 2):
251 113
67 12
221 912
Result:
-99 1
-1 -1
1 19
[3:23 user@ws hw] ./a.out
Input operation (+/-): 1
Error!
[3:23 user@ws hw] ./a.out
Input operation (+/-): +
Enter matrix A size (m x n): 0 x 1
Error!
[3:23 user@ws hw] ./a.out
Input operation (+/-): +
Enter matrix A size (m x n): -3 x 1
Error!
[3:23 user@ws hw] ./a.out
Input operation (+/-): -
Enter matrix A size (m x n): 2 x q
Error!
[3:23 user@ws hw] ./a.out
Input operation (+/-): +
Enter matrix A size (m x n): 2 x 3
Enter matrix A (2 x 3):
1 2 3
5 6 7
Enter matrix B (2 x 3):
6 a 1
Error!
[3:23 user@ws hw] ./a.out

```

```

Input operation (+/-): +
Enter matrix A size (m x n): 2 x 1
Enter matrix A (2 x 1):
1 q
Error!
[3:23 user@ws hw] ./a.out
Input operation (+/-): q
Error!
[3:23 user@ws hw]

```

P6 行列式 `Determinant.c`

假設有一個3*3的矩陣，如下所示：

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

A的行列式被記作“ $\det(A)$ ”且“ $\det(A)$ ”的值運算方法如下： $\det(A) = aei + bfg + cdh - ceg - afh - bdi$

請設計一個C語言的程式 `Determinant.c` 讓使用者輸入一個3x3的整數矩陣，並輸出其行列式的值。要注意的是若使用者輸入非整數字元時，請則輸出Error!並結束程式。

本題的執行結果可參考以下輸出內容：

```

[3:23 user@ws] ./a.out
Enter data:
1 2 3
4 5 7
2 3 4
det(A) result: 1
[3:23 user@ws] ./a.out
Enter data:
3 4 7
12 32 5
11 -11 -11
det(A) result: -3531
[3:23 user@ws] ./a.out
Enter data:
0 1 0
1 0 1
0 1 0
det(A) result: 0
[3:23 user@ws] ./a.out
Enter data:
44 78 59
-23 -100 -13

```

```

-32^0^0^
det(A)^result:^-156352^
[3:23 user@ws] ./a.out^
Enter^data:^
32^11^54^
10^33^41^
9^7^123^
det(A)^result:^98975^
[3:23 user@ws hw] ./a.out^
Enter^data:^
1^2^a^
Error!^
[3:23 user@ws hw] ./a.out^
Enter^data:^
4^1^2^
5^1^2^
a^6^2^
Error!^
[3:23 user@ws hw]

```

P7 找出數字 FindNumber.c

請設計一個C語言的程式 FindNumber.c 讓使用者依序輸入 10 個整數並保留於程式當中；後續再讓使用者輸入一個整數（如果輸入非數字的字元則輸出Error! 輸出該整數為之前所輸入的第幾個整數（若該整數出現兩次以上則請輸出最後一次出現的位置；若找不到則輸出錯誤訊息）。其執行結果可參考以下輸出內容：

```

[3:23 user@ws] ./a.out^
Enter^data:^1^2^3^4^5^6^7^8^9^10^
Enter^a^number^to^find:^5^
5^is^the^5th^number.^
[3:23 user@ws] ./a.out^
Enter^data:^0^1^0^1^1^0^0^1^1^0^
Enter^a^number^to^find:^1^
1^is^the^9th^number.^
[3:23 user@ws] ./a.out^
Enter^data:^-4^-3^-2^-1^0^1^1^2^2^3^
Enter^a^number^to^find:^-3^
-3^is^the^2nd^number.^
[3:23 user@ws] ./a.out^
Enter^data:^-4^-3^-2^-1^0^1^1^2^2^3^
Enter^a^number^to^find:^-2^
-2^is^the^3rd^number.^
[3:23 user@ws] ./a.out^
Enter^data:^123^414^23984^2352^12515^124^29957^-1235215^-12^51^
Enter^a^number^to^find:^123^

```

```

123▲is▲the▲1st▲number.↵
[3:23 user@ws] ./a.out↵
Enter▲data:▲123▲414▲23984▲2352▲12515▲124▲29957▲-1235215▲-12▲51↵
Enter▲a▲number▲to▲find:▲-1↵
-1▲is▲not▲found↵
[3:23 user@ws hw] ./a.out↵
Enter▲data:▲a↵
Error!↵
[3:23 user@ws hw] ./a.out↵
Enter▲data:▲1▲a▲5▲2▲6▲1▲0▲4▲6▲2↵
Error!↵
[3:23 user@ws hw] ./a.out↵
Enter▲data:▲8▲2▲5▲1▲0▲5▲2▲3▲9▲2↵
Enter▲a▲number▲to▲find:▲a↵
Error!↵
[3:23 user@ws hw]

```

From:
<https://junwu.nptu.edu.tw/dokuwiki/> - Jun Wu的教學網頁
 國立屏東大學資訊工程學系
 CSIE, NPTU
 Total: 290665

Permanent link:
<https://junwu.nptu.edu.tw/dokuwiki/doku.php?id=c:2025fall-hw8>

Last update: **2025/12/02 06:47**

