Turnin作業9

- Turnin Code: c.hw9
- Due Date: 2026/12/01 23:59 **Hard Deadline**

繳交方式說明

同學在經歷前幾次作業的練習後，對於建立資料夾、撰寫程式、編譯，以及使用 turnin 指令上傳的流程應已相當熟悉。

因此，從作業6開始，將省略繳交步驟示範。請同學在自行確認**9個程式題**皆編譯無誤且執行結果正確後，再進行上傳。



若有其他的繳交規定或方式，將會另行公告

如需回顧之前的終端機操作與繳交指令，請參考：c.hw5繳交說明



- 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(get_positive_numbers.c)

請設計一個 C語言程式 get_positive_numbers.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：

```
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Please▲input▲10▲positive▲numbers▲(0▲to▲terminate).↵
#1==>?10↵
#2==>?20↵
#3==>?30↵
#4==>?-40↵
Error!▲Please▲input▲again!↵
```

```
#4==>?p↵
Error!▲Please▲input▲again!↵
#4==>?40↵
#5==>?0↵
Input▲terminated!↵
Numbers[10,20,30,40]↵
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Please▲input▲10▲positive▲numbers▲(0▲to▲terminate).↵
#1==>?12↵
#2==>?35↵
#3==>?74↵
#4==>?65↵
#5==>?23↵
#6==>?57↵
#7==>?89↵
#8==>?es↵
Error!▲Please▲input▲again!↵
#8==>?34↵
#9==>?32↵
#10==>?-100↵
Error!▲Please▲input▲again!↵
#10==>?2↵
Numbers[12,35,74,65,23,57,89,34,32,2]↵
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Please▲input▲10▲positive▲numbers▲(0▲to▲terminate).↵
#1==>?0↵
Input▲terminated!↵
Numbers[empty]↵
[7:17▲user@ws▲in▲hw9]$
```

---

p2 成績統計程式（score_stats.c）

請撰寫一個 C 程式 score_stats.c，讓使用者輸入7筆學生成績，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!。



注意：

- 本題批改輸入皆介於0到100之間，勿需進行檢查。
- 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：

```
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Please▲input▲7▲scores:↵
39▲50▲59▲89▲60▲99▲100↵
Please▲input▲command:▲m↵
Max▲score▲=▲100↵
Please▲input▲command:▲n↵
Min▲score▲=▲39↵
Please▲input▲command:▲s↵
Sum▲score▲=▲496↵
Please▲input▲command:▲a↵
Average▲score▲=▲70.9↵
Please▲input▲command:▲A↵
Error!↵
Please▲input▲command:▲p↵
Pass▲count▲=▲4↵
Please▲input▲command:▲q↵
[7:17▲user@ws▲in▲hw9]$▲
```

---

p3 行列式（determinant.c）

假設有一個3*3的矩陣，如下所示：
$$ A = \left[\begin{array}{ccc} a & b & c \\ d & e & f \\ g & h & i \end{array} \right] $$

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 3×3 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：

```
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Enter▲data:↵
```

```
1▲2▲3↵
4▲5▲6↵
7▲8▲9↵
det(A)▲result:▲0↵
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Enter▲data:↵
20▲67▲78↵
50▲47▲38↵
-30▲-5▲7↵
det(A)▲result:▲1030↵
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Enter▲data:↵
1▲8▲d↵
Error!↵
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Enter▲data:↵
2▲2▲2↵
2▲2▲2↵
e▲2▲2↵
Error!↵
[7:17▲user@ws▲in▲hw9]$
```

---

p4 階乘（factorial.c）

一個正整數的階乘，表示從 1 到該數之間所有正整數的乘積。例如，5 的階乘記作$5!$，計算方法如下：$$5!=1\times2\times3\times4\times5$$ 此外，根據數學定義，$0!$的值被定義為1。

請設計一個C語言的程式factorial.c，根據使用者輸入的整數，將其階乘的值輸出，若使用者所輸入的整數數值小於0或大於19則輸出Error!。



在儲存階層計算結果時，請將變數型態設為 long long

其輸出結果可以參考以下的內容：

```
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Please▲input▲an▲integer:▲0↵
0!▲=▲1↵
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Please▲input▲an▲integer:▲1↵
1!▲=▲1↵
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Please▲input▲an▲integer:▲-1↵
```

```
Error!↵
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Please▲input▲an▲integer:▲20↵
Error!↵
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Please▲input▲an▲integer:▲7↵
7!▲=▲1▲*▲2▲*▲3▲*▲4▲*▲5▲*▲6▲*▲7▲=▲5040↵
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Please▲input▲an▲integer:▲15↵
15!▲=▲1▲*▲2▲*▲3▲*▲4▲*▲5▲*▲6▲*▲7▲*▲8▲*▲9▲*▲10▲*▲11▲*▲12▲*▲13▲*▲14▲*▲15▲=▲1307674368000↵
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Please▲input▲an▲integer:▲a↵
Error!↵
[7:17▲user@ws▲in▲hw9]$▲
```

---

p5 數字回文金字塔（num_palindrome_pyramid.c）

請設計一個 C 語言程式 num_palindrome_pyramid.c，讓使用者輸入一個正整數以產生一個高度為 n 層的數字回文金字塔。若輸入的$n < 1$或輸入非整數，則輸出 **Error!**。

每層左右皆以中括號 [ ] 包覆並以空白置中對齊，可參考下列的執行結果輸出數字回文：

```
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Please▲input▲the▲number▲of▲the▲layer▲of▲the▲pyramid:▲1↵
[1]↵
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Please▲input▲the▲number▲of▲the▲layer▲of▲the▲pyramid:▲4↵
[▲▲▲1▲▲▲]↵
[▲▲121▲▲]↵
[▲12321▲]↵
[1234321]↵
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Please▲input▲the▲number▲of▲the▲layer▲of▲the▲pyramid:▲7↵
[▲▲▲▲▲▲1▲▲▲▲▲▲]↵
[▲▲▲▲▲121▲▲▲▲▲]↵
[▲▲▲▲12321▲▲▲▲]↵
[▲▲▲1234321▲▲▲]↵
[▲▲123454321▲▲]↵
[▲12345654321▲]↵
[1234567654321]↵
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Please▲input▲the▲number▲of▲the▲layer▲of▲the▲pyramid:▲-1↵
Error!↵
```

```
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Please▲input▲the▲number▲of▲the▲layer▲of▲the▲pyramid:▲a↵
Error!↵
[7:17▲user@ws▲in▲hw9]$▲
```

---

p6 迷宮探險（maze.c）

請設計一個C語言的程式maze.c，使用 5×7 的 `int` 二維陣列建立一個迷宮小遊戲。會先讀取 5×7 的二維陣列地圖，接著重複讀取單一字元指令控制角色移動或退出遊戲。

地圖中的整數定義如下：

- `0`：空地（印出地圖時顯示為 `.`）
- `1`：牆壁或障礙物（印出地圖時顯示為 `#`，無法通行）
- `2`：玩家角色（印出地圖時顯示為 `P`）
- `3`：終點（印出地圖時顯示為 `E`）

遊戲開始時先印出初始地圖，之後每次處理指令時皆先印出地圖，再輸出對應訊息：

- 角色移動：`w` 為往上、`s` 為往下、`a` 為往左、`d` 為往右。
- 目標為空地（`0`）：角色移動（原位置變 `0`）。
- 目標為牆壁（`1`）：角色不移動，並輸出 `Hit wall!`。
- 目標為終點（`3`）：角色移動至終點，輸出 `You win!` 並結束程式。
- 輸入 `q`：輸出 `Quit!` 並結束程式。
- 其他無效指令：輸出 `Invalid command!`。



- 本題地圖大小固定為 5 列 7 行。
- 本題地圖的最外圍一圈皆為牆壁（`1`），且玩家起始位置固定在地圖上的 $(1, 1)$（列與行皆由 0 開始編號）。
- 本題輸出為了方便同學觀看，已將輸出的部分改以等寬字體處理。

```
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
1▲1▲1▲1▲1▲1▲1↵
1▲2▲0▲0▲0▲0▲1↵
1▲0▲1▲1▲0▲0▲1↵
1▲0▲0▲0▲1▲3▲1↵
1▲1▲1▲1▲1▲1▲1↵
```

```
#######
#P....#
#.##..#
#...#E#
```

```
#######
```

Command▲(w/a/s/d/q):▲w↵

```
#######
#P....#
#.##..#
#...#E#
#######
```

Hit▲wall!↵
Command▲(w/a/s/d/q):▲d↵

```
#######
#.P...#
#.##..#
#...#E#
#######
```

Command▲(w/a/s/d/q):▲x↵

```
#######
#.P...#
#.##..#
#...#E#
#######
```

Invalid▲command!↵
Command▲(w/a/s/d/q):▲d↵

```
#######
#..P..#
#.##..#
#...#E#
#######
```

Command▲(w/a/s/d/q):▲d↵

```
#######
#...P.#
#.##..#
#...#E#
#######
```

Command▲(w/a/s/d/q):▲d↵

```
#######
#....P#
#.##..#
#...#E#
```

```
#######
```

Command▲(w/a/s/d/q):▲s↵

```
#######
#.....#
#.##.P#
#...#E#
#######
```

Command▲(w/a/s/d/q):▲s↵

```
#######
#.....#
#.##..#
#...#P#
#######
```

You▲win!↵
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
1▲1▲1▲1▲1▲1▲1↵
1▲2▲1▲0▲0▲0▲1↵
1▲0▲0▲1▲1▲1▲1↵
1▲0▲0▲0▲0▲3▲1↵
1▲1▲1▲1▲1▲1▲1↵

```
#######
#P#...#
#..####
#....E#
#######
```

Command▲(w/a/s/d/q):▲q↵

```
#######
#P#...#
#..####
#....E#
#######
```

Quit!↵
[7:17▲user@ws▲in▲hw9]$▲

---

p7 向量外積乘法（outer_mul.c）

在線性代數中，一個 5×1 的直向陣列（5 列 1 行）與一個 1×5 的橫向陣列（1 列 5 行）相乘，會得到一個 5×5 的二維矩陣，其中第 row 列、第 col 行的數值，就是直向陣列的

第 row 個元素乘上橫向陣列的第 col 個元素。

請設計一個C語言的程式outer_mul.c，先讓使用者輸入 5 個整數作為直向陣列，接著輸入 5 個整數作為橫向陣列。程式需計算出 5×5 的二維整數陣列結果，並逐列印出。

```
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
1▲2▲3▲4▲5↵
6▲7▲8▲9▲10↵
6▲7▲8▲9▲10↵
12▲14▲16▲18▲20↵
18▲21▲24▲27▲30↵
24▲28▲32▲36▲40↵
30▲35▲40▲45▲50↵
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
2▲-1▲0▲3▲2↵
1▲2▲3▲4▲5↵
2▲4▲6▲8▲10↵
-1▲-2▲-3▲-4▲-5↵
0▲0▲0▲0▲0↵
3▲6▲9▲12▲15↵
2▲4▲6▲8▲10↵
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
1▲2▲3▲a▲5↵
Error!↵
[7:17▲user@ws▲in▲hw9]$▲
```

---

p8 打印鑽石（diamond.c）

請設計一個C語言的程式diamond.c，讓使用者輸入一個正整數n以及一個字元產生一個以該字元為底的鑽石。

若$n < 0$則輸出“Height can't be lower than 0!”。



注意：

- 本題輸出為了方便同學觀看，已將鑽石輸出的部分改以等寬字體處理。
- 本題禁止使用 **'\b'** 及 **'\t'** 跳脫序列。

輸出結果可以參考以下的內容：

```
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Enter▲height▲and▲symbol▲(w,s):▲-1,*↵
Height▲can't▲be▲lower▲than▲0!↵
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Enter▲height▲and▲symbol▲(w,s):▲0,s↵
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Enter▲height▲and▲symbol▲(w,s):▲3,*↵
```

```
▲▲*↵
▲***↵
*****↵
▲***↵
▲▲*↵
```

```
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
Enter▲height▲and▲symbol▲(w,s):▲15,@↵
```

```
▲▲▲▲▲▲▲▲▲▲▲▲▲▲@↵
▲▲▲▲▲▲▲▲▲▲▲▲▲@@@↵
▲▲▲▲▲▲▲▲▲▲▲▲@@@@@↵
▲▲▲▲▲▲▲▲▲▲▲@@@@@@@↵
▲▲▲▲▲▲▲▲▲▲@@@@@@@@@↵
▲▲▲▲▲▲▲▲▲@@@@@@@@@@@↵
▲▲▲▲▲▲▲▲@@@@@@@@@@@@@↵
▲▲▲▲▲▲▲@@@@@@@@@@@@@@@↵
▲▲▲▲▲▲@@@@@@@@@@@@@@@@@↵
▲▲▲▲▲@@@@@@@@@@@@@@@@@@@↵
▲▲▲▲@@@@@@@@@@@@@@@@@@@@@↵
▲▲▲@@@@@@@@@@@@@@@@@@@@@@@↵
▲▲@@@@@@@@@@@@@@@@@@@@@@@@@↵
▲@@@@@@@@@@@@@@@@@@@@@@@@@@@↵
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@↵
▲@@@@@@@@@@@@@@@@@@@@@@@@@@@↵
▲▲@@@@@@@@@@@@@@@@@@@@@@@@@↵
▲▲▲@@@@@@@@@@@@@@@@@@@@@@@↵
▲▲▲▲@@@@@@@@@@@@@@@@@@@@@↵
▲▲▲▲▲@@@@@@@@@@@@@@@@@@@↵
▲▲▲▲▲▲@@@@@@@@@@@@@@@@@↵
▲▲▲▲▲▲▲@@@@@@@@@@@@@@@↵
▲▲▲▲▲▲▲▲@@@@@@@@@@@@@↵
▲▲▲▲▲▲▲▲▲@@@@@@@@@@@↵
▲▲▲▲▲▲▲▲▲▲@@@@@@@@@↵
▲▲▲▲▲▲▲▲▲▲▲@@@@@@@↵
▲▲▲▲▲▲▲▲▲▲▲▲@@@@@↵
▲▲▲▲▲▲▲▲▲▲▲▲▲@@@↵
▲▲▲▲▲▲▲▲▲▲▲▲▲▲@↵
```

```
[7:17▲user@ws▲in▲hw9]$▲
```

p9 抽獎箱（lottery_box.c）

請設計一個 C 語言程式 lottery_box.c，宣告一個長度為7的int型態一維陣列。程式執行時，該陣列內初始依序放有 1 到 7 共 7 個號碼球，接著讓使用者輸入指令，若指令錯誤則輸出 “Invalid command!”。

其相關指令如下表：

指令	功能說明
p	從箱子裡隨機抽取一個號碼
q	結束程式



- 本題範例輸出的抽取號碼為隨機產生，因此實際執行結果會跟範例輸出不一樣。



- 因輸出為隨機產生，因此本題將進行**人工批改**

輸出結果可以參考以下的內容：

```
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
The▲box:▲[1,2,3,4,5,6,7]↵
Action▲(p/q):▲x↵
Invalid▲command!↵
↵
The▲box:▲[1,2,3,4,5,6,7]↵
Action▲(p/q):▲p↵
Picked▲number▲7▲out▲of▲the▲box!↵
↵
The▲box:▲[1,2,3,4,5,6,-1]↵
Action▲(p/q):▲p↵
Picked▲number▲2▲out▲of▲the▲box!↵
↵
The▲box:▲[1,6,3,4,5,-1,-1]↵
Action▲(p/q):▲p↵
Picked▲number▲3▲out▲of▲the▲box!↵
↵
The▲box:▲[1,6,5,4,-1,-1,-1]↵
Action▲(p/q):▲p↵
Picked▲number▲1▲out▲of▲the▲box!↵
↵
The▲box:▲[4,6,5,-1,-1,-1,-1]↵
Action▲(p/q):▲p↵
```

```
Picked▲number▲6▲out▲of▲the▲box!↵
↵
The▲box:▲[4,5,-1,-1,-1,-1,-1]↵
Action▲(p/q):▲p↵
Picked▲number▲5▲out▲of▲the▲box!↵
↵
The▲box:▲[4,-1,-1,-1,-1,-1,-1]↵
Action▲(p/q):▲p↵
Picked▲number▲4▲out▲of▲the▲box!↵
↵
The▲box▲is▲now▲empty!↵
Bye!↵
[7:17▲user@ws▲in▲hw9]$▲./a.out↵
The▲box:▲[1,2,3,4,5,6,7]↵
Action▲(p/q):▲q↵
↵
Bye!↵
[7:17▲user@ws▲in▲hw9]$▲
```

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: 343744

Permanent link:
https://junwu.nptu.edu.tw/dokuwiki/doku.php?id=c:2026fall:hw9

Last update: **2026/11/26 13:34**