

Turnin作業7

- chapter 7
- Turnin Code: **c.hw7**
- Due Date: 2025/11/25 23:59 **Hard Deadline**

繳交方式說明

本次Turnin作業包含9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hw7的資料夾，並且切換到該資料夾後，針對第1題所要求的LCM.c進程式編撰：

```
[user@ws ~]$ mkdir hw7
[user@ws ~]$ cd hw7
[user@ws hw7]$ joe LCM.c
```

等到我們完成LCM.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.hw7 LCM.c` 指令將加以上傳：

```
[user@ws c.hw7]$ turnin c.hw7 LCM.c
Turning in:
LCM.c -- ok
All done.
[user@ws c.hw7]$
```

當然，你也可以等到本次作業要求的所有題目都在hw7資料夾裡完成後，再使用 `turnin c.hw7 .` 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hw7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.hw7]$ ls
Factorial.c          LCM.c                PrimeNumSum.c
GCD.c               NumberPyramidWithAlignment.c  PrintDiamond.c
HollowInvertedTriangle.c  NumPalindromePyramid.c
RangeMultiplicationTable.c
a.out
[user@ws c.hw7]$ rm a.out
[user@ws c.hw7]$ ls
Factorial.c          LCM.c                PrimeNumSum.c
```

```
GCD.c          NumberPyramidWithAlignment.c  PrintDiamond.c
HollowInvertedTriangle.c  NumPalindromePyramid.c
RangeMultiplicationTable.c
[user@ws c.hw7]$ turnin c.hw7 .
Turning in:
./Factorial.c -- ok
./LCM.c -- ok
./PrimeNumSum.c -- ok
./HollowInvertedTriangle.c -- ok
./NumberPyramidWithAlignment.c -- ok
./PrintDiamond.c -- ok
./HollowInvertedTriangle.c -- ok
./NumPalindromePyramid.c -- ok
./RangeMultiplicationTable.c -- ok
All done.
[user@ws c.hw7]$
```

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.hw7↵` 指令，例如：

```
[user@ws c.hw7] c.hw7 $ turnin -ls c.hw7
.:
total 40
-rw-rw----. 1 turninman turnin  922 Nov  18 3:25 Factorial.c
-rw-rw----. 1 turninman turnin  506 Nov  18 3:25 GCD.c
-rw-rw----. 1 turninman turnin  468 Nov  18 3:25 HollowInvertedTriangle.c
-rw-rw----. 1 turninman turnin 1418 Nov  18 3:25 LCM.c
-rw-rw----. 1 turninman turnin  549 Nov  18 3:25
NumberPyramidWithAlignment.c
-rw-rw----. 1 turninman turnin  592 Nov  18 3:25 NumPalindromePyramid.c
-rw-rw----. 1 turninman turnin  624 Nov  18 3:25 PrimeNumSum.c
-rw-rw----. 1 turninman turnin 2876 Nov  18 3:25 PrintDiamond.c
-rw-rw----. 1 turninman turnin  508 Nov  18 3:25 RangeMultiplicationTable.c
```

註：本文使用 及 `\n` 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 最小公倍數 LCM.c

題目敘述：

請設計一個 C 語言程式 LCM.c 讓使用者輸入兩個正整數，計算並輸出這兩個數字的最小公倍數 LCM, Least Common Multiple 後加以輸出。假設使用者所輸入的兩個整數分別為 n_1 與 n_2 (如果輸入的任一整數小於等於 0，或者非數字則輸出 **Error!**)，其最小公倍數是同時為 n_1 與 n_2 的倍數中最小者 (意即可以同時被 n_1 與 n_2 整除的數字中最小者)。

此題執行結果可參考如下：

```
[3:23 user@ws hw] ./a.out↵
Please▲input▲two▲numbers(a,b):▲12▲13↵
LCM(12,13)▲=▲156↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲two▲numbers(a,b):▲12▲5↵
LCM(12,5)▲=▲60↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲two▲numbers(a,b):▲2▲2↵
LCM(2,2)▲=▲2↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲two▲numbers(a,b):▲0▲2↵
Error!↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲two▲numbers(a,b):▲-1▲5↵
Error!↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲two▲numbers(a,b):▲1▲a↵
Error!↵
[3:23 user@ws hw]
```

P2 最大公因數 GCD.c

兩個數字的最大公因數是能夠同時被兩數整除的最大正整數。請設計一個C語言的程式GCD.c讓使用者輸入兩個不超過10000的正整數（如果不符合或是輸入非數字時則請輸出**Error!**），請找出這兩個數字的最大公因數 Greatest Common Divisor GCD後加以輸出。

本題的輸出結果可以參考以下的內容：

```
[3:23 user@ws hw] ./a.out↵
Please▲input▲two▲numbers(a,b):▲18▲4↵
GCD(18,4)▲=▲2↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲two▲numbers(a,b):▲256▲40↵
GCD(256,40)▲=▲8↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲two▲numbers(a,b):▲80▲80↵
GCD(80,80)▲=▲80↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲two▲numbers(a,b):▲-1▲1↵
Error!↵
```

```
[3:23 user@ws hw] ./a.out↵
Please input two numbers(a,b): 1 0↵
Error!↵
[3:23 user@ws hw] ./a.out↵
Please input two numbers(a,b): 10001 1↵
Error!↵
[3:23 user@ws hw] ./a.out↵
Please input two numbers(a,b): a 4↵
Error!↵
[3:23 user@ws hw]
```

P3 輸出菱形 PrintDiamond.c

請設計一個C語言程式PrintDiamond.c讓使用者輸入為一個大於等於3的整數 \$n\$ 使用迴圈以及加號印出邊長等於 \$n\$ 的菱形。

如果輸入不符合 $n \geq 3$ 或是輸入非數字則輸出**Error!**

其執行結果可參考如下：

【注意：為了便利閱讀，在以下的輸出結果裡使用了等寬字型顯示題出所輸出的+與 。儘管使用了灰色底色但它們仍為輸出結果。】

【注意：本題會採用人工批改，請務必使用迴圈配合printf(“+”)與printf(“ ”)來印出加號與空白，不可以直接使用printf()印出整行的結果。】

```
[3:23 user@ws hw] ./a.out↵
Please input an integer greater than or equal to 3: 5↵
```

```
▲▲▲▲+↵
▲▲▲++++↵
▲▲+++++↵
▲++++++↵
+++++++↵
▲++++++↵
▲▲+++++↵
▲▲+++++↵
▲▲▲+++↵
▲▲▲▲+↵
```

```
[3:23 user@ws hw] ./a.out↵
Please input an integer greater than or equal to 3: 3↵
```

```
▲▲+↵
▲+++↵
+++++↵
▲+++↵
▲▲+↵
```

```
[3:23 user@ws hw] ./a.out↵
Please input an integer greater than or equal to 3: 1↵
```

```
Error!↵
[3:23 user@ws hw] ./a.out↵
Please input an integer greater than or equal to 3:↵
Error!↵
[3:23 user@ws hw]
```

P4 計算1到n的質數之和 PrimeNumSum.c

請設計一個C語言程式PrimeNumSum.c讓使用者輸入一個正整數 n 將 1 到 n 之間的所有質數輸出並加總，如果輸入負數或是非數字則輸出**Error!**，其執行結果可參考如下：

```
[3:23 user@ws hw] ./a.out↵
Please input an integer:↵40↵
2+3+5+7+11+13+17+19+23+29+31+37=197↵
[3:23 user@ws hw] ./a.out↵
Please input an integer:↵1↵
There isn't any prime numbers between 1 and 1.↵
[3:23 user@ws hw] ./a.out↵
Please input an integer:↵g↵
Error!↵
[3:23 user@ws hw] ./a.out↵
Please input an integer:↵-1↵
Error!↵
[3:23 user@ws hw]
```

P5 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 $5! = 1 \times 2 \times 3 \times 4 \times 5$ 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，其輸出結果可以參考以下的內容：

【注意：本題的輸入不會是大於19的非負整數，如果不符合定義或是輸入非數字則輸出Error!】

【注意：本題會採用人工批改，請務必使用迴圈計算n的階乘後輸出，而非直接印出1~19的階乘值。】

```
[3:23 user@ws hw] ./a.out↵
Please input an integer:↵0↵
```

```

0!▲=▲1↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲an▲integer:▲1↵
1!▲=▲1↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲an▲integer:▲7↵
7!▲=▲1▲*▲2▲*▲3▲*▲4▲*▲5▲*▲6▲*▲7▲=▲5040↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲an▲integer:▲20↵
Error!↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲an▲integer:▲-1↵
Error!↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲an▲integer:▲a↵
Error!↵
[3:23 user@ws hw]

```

P6 中空倒三角形 `HollowInvertedTriangle.c`

題目敘述：

請設計一個 C `HollowInvertedTriangle.c` 語言程式，要求如下：

- 使用者輸入一個整數 n $3 \leq n \leq 99$ 如果輸入不在這個範圍的整數或是輸入非數字，則輸出 **Error!**
- 程式輸出由星號 (*) 組成的倒三角形圖案；
- 圖案共 n 列，其中第一列印出 n 個 *，最後一列印出單一個 *；
- 第二列到第 $n-1$ 列為「中空」倒三角形，每列僅最左與最右印出 *，中間填充空白；

輸出結果請參考如下：

【注意：為了便利閱讀，在以下的輸出結果裡使用了等寬字型顯示題出所輸出的+與 。儘管使用了灰色底色但它們仍為輸出結果。】

【注意：本題會採用人工批改，請務必使用迴圈配合 `printf("*")` 與 `printf(" ")` 來印出加號與空白，不可以直接使用 `printf()` 印出整行的結果。】

```

[3:23 user@ws hw] ./a.out↵
5↵
*****↵
*▲▲*↵
*▲*↵
**↵
*↵
[3:23 user@ws hw] ./a.out↵
9↵

```

```

*****↵
*  ▲▲▲▲▲*↵
*  ▲▲▲▲▲*↵
*  ▲▲▲▲*↵
*  ▲▲▲*↵
*  ▲▲*↵
*  ▲*↵
**↵
*↵

```

```

[3:23 user@ws hw] ./a.out↵
-1↵
Error!↵
[3:23 user@ws hw] ./a.out↵
100↵
Error!↵
[3:23 user@ws hw] ./a.out↵
n↵
Error!↵
[3:23 user@ws hw] ./a.out↵
3↵

```

```

***↵
**↵
*↵

```

```

[3:23 user@ws hw]

```

P7 區間乘法表 RangeMultiplicationTable.c

題目敘述：

請設計一個 C 語言程式 RangeMultiplicationTable.c 讓使用者輸入兩個整數 a 與 b (不分大小，例如 a=9 b=11 或 a=11 b=9 都可)。

規則說明：

假設輸入兩個整數 a 與 b 程式需先判斷哪一個為較小值、哪一個為較大值，然後印出區間內 (含端點) 所有數字的乘法表。

如果輸入內容超過兩個整數或是輸入非數字，則輸出 Error!↵

每一筆輸出格式為 $x \times y = \text{結果}$

每個 x 的乘法表需包含以下內容：

- y 從較小值到較大值依序列印
- 完成一個 x 的乘法表後，要換行再印下一個 x 的乘法表

```
[3:23 user@ws hw] ./a.out↵  
Please input two integers: 1 5↵
```

```
1x1=1↵  
1x2=2↵  
1x3=3↵  
1x4=4↵  
1x5=5↵  
↵
```

```
2x2=4↵  
2x3=6↵  
2x4=8↵  
2x5=10↵  
↵
```

```
3x3=9↵  
3x4=12↵  
3x5=15↵  
↵
```

```
4x4=16↵  
4x5=20↵  
↵
```

```
5x5=25↵
```

```
[3:23 user@ws hw] ./a.out↵  
Please input two integers: 1 3↵
```

```
1x1=1↵  
1x2=2↵  
1x3=3↵  
↵
```

```
2x2=4↵  
2x3=6↵  
↵
```

```
3x3=9↵
```

```
[3:23 user@ws hw] ./a.out↵  
Please input two integers: 9 15↵
```

```
9x9=81↵  
9x10=90↵  
9x11=99↵  
9x12=108↵  
9x13=117↵  
9x14=126↵  
9x15=135↵  
↵
```

```
10x10=100↵  
10x11=110↵  
10x12=120↵  
10x13=130↵  
10x14=140↵  
10x15=150↵  
↵
```

```
11x11=121↵
```

```

11x12=132↵
11x13=143↵
11x14=154↵
11x15=165↵
↵
12x12=144↵
12x13=156↵
12x14=168↵
12x15=180↵
↵
13x13=169↵
13x14=182↵
13x15=195↵
↵
14x14=196↵
14x15=210↵
↵
15x15=225↵
[3:23 user@ws hw] ./a.out↵
Please input two integers: 10 8↵
8x8=64↵
8x9=72↵
8x10=80↵
↵
9x9=81↵
9x10=90↵
↵
10x10=100↵
[3:23 user@ws hw] ./a.out↵
Please input two integers: 3 -1↵
-1x-1=1↵
-1x0=0↵
-1x1=-1↵
-1x2=-2↵
-1x3=-3↵
↵
0x0=0↵
0x1=0↵
0x2=0↵
0x3=0↵
↵
1x1=1↵
1x2=2↵
1x3=3↵
↵
2x2=4↵
2x3=6↵
↵
3x3=9↵

```

```

[3:23 user@ws hw] ./a.out↵
Please input two integers: -2 3↵
-2x-2=4↵
-2x-1=2↵
-2x0=0↵
-2x1=-2↵
-2x2=-4↵
-2x3=-6↵
↵
-1x-1=1↵
-1x0=0↵
-1x1=-1↵
-1x2=-2↵
-1x3=-3↵
↵
0x0=0↵
0x1=0↵
0x2=0↵
0x3=0↵
↵
1x1=1↵
1x2=2↵
1x3=3↵
↵
2x2=4↵
2x3=6↵
↵
3x3=9↵
[3:23 user@ws hw] ./a.out↵
Please input two integers: a 1↵
Error!↵
[3:23 user@ws hw]

```

P8 數字回文金字塔 NumPalindromePyramid.c

請設計一個C語言的程式NumPalindromePyramid.c讓使用者輸入一個正奇數 \$n\$ 以產生一個數字回文金字塔，假設輸入的數字為\$n\$如果 \$n \le 0\$ 或是輸入非數字則輸出**Error!**輸出的數字回文金字塔的高度（層數）為 $\lceil \frac{n}{2} \rceil$ 且每層可參考下列的執行結果輸出數字回文：

【注意：為了便利閱讀，在以下的輸出結果裡使用了等寬字型顯示題出所輸出的+與。儘管使用了灰色底色但它們仍為輸出結果。】
 【注意：本題會採用人工批改，請務必使用迴圈配合printf()印出|、數字與空白，不可以直接使用printf()印出整行的結果。】

```

[3:23 user@ws hw] ./a.out↵

```

Please input the number of the layer of the pyramid: 9

```
|▲▲▲▲1▲▲▲▲|
|▲▲▲121▲▲▲|
|▲▲12321▲▲|
|▲1234321▲|
|123454321|
```

[3:23 user@ws hw] ./a.out

Please input the number of the layer of the pyramid: 5

```
|▲▲1▲▲|
|▲121▲|
|12321|
```

[3:23 user@ws hw] ./a.out

Please input the number of the layer of the pyramid: 1

```
|1|
```

[3:23 user@ws hw] ./a.out

Please input the number of the layer of the pyramid: 4

Error!

[3:23 user@ws hw] ./a.out

Please input the number of the layer of the pyramid: -1

Error!

[3:23 user@ws hw] ./a.out

Please input the number of the layer of the pyramid: 0

Error!

[3:23 user@ws hw] ./a.out

Please input the number of the layer of the pyramid: a

Error!

[3:23 user@ws hw]

P9 數字金字塔 NumberPyramidWithAlignment.c

題目敘述：請設計一個 C 語言程式，要求如下：

使用者輸入一個整數 n ($1 \leq n \leq 9$) 如果輸入不在這個範圍的數值或是輸入非數字則輸出 **Error!**

接著輸入一個字元 c 用來控制輸出對齊方向：

- 如果輸入非 l, c, r 的英文字元，則輸出 **Error!**
- **l** 表示 靠左輸出；
- **c** 表示 靠中輸出，每列數字從左到右依序排列，並在每列前加適當空格。使圖形整體水平居中，如果使用靠中輸出，則必須輸入奇數，否則輸出 **Error!**
- **r** 表示 靠右輸出，每列數字從左到右依序排列，並在每列前加適當空格，使整體右對齊；

請印出數字金字塔圖案：

- 每列從 1 開始累加到該列的列數；
- 根據對齊設定調整每列前的空格數量；
- 每列印完後換行；

執行結果請參考如下：

【注意：為了便利閱讀，在以下的輸出結果裡使用了等寬字型顯示題出所輸出的+與 。儘管使用了灰色底色但它們仍為輸出結果。】

【注意：本題會採用人工批改，請務必使用迴圈配合printf()印出數字與空白，不可以直接使用printf()印出整行的結果。】

```
[3:23 user@ws hw] ./a.out↵
Please▲input▲a▲integer▲and▲a▲char:▲5▲r↵
▲▲▲▲▲1↵
▲▲▲▲1▲2↵
▲▲▲1▲2▲3↵
▲▲1▲2▲3▲4↵
1▲2▲3▲4▲5↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲a▲integer▲and▲a▲char:▲6▲l↵
1↵
1▲2↵
1▲2▲3↵
1▲2▲3▲4↵
1▲2▲3▲4▲5↵
1▲2▲3▲4▲5▲6↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲a▲integer▲and▲a▲char:▲9▲c↵
▲▲▲▲▲1↵
▲▲▲▲1▲2↵
▲▲▲▲1▲2▲3↵
▲▲▲▲1▲2▲3▲4↵
▲▲▲1▲2▲3▲4▲5↵
▲▲1▲2▲3▲4▲5▲6↵
▲▲1▲2▲3▲4▲5▲6▲7↵
▲1▲2▲3▲4▲5▲6▲7▲8↵
1▲2▲3▲4▲5▲6▲7▲8▲9↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲a▲integer▲and▲a▲char:▲2▲l↵
1↵
1▲2↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲a▲integer▲and▲a▲char:▲2▲r↵
▲▲1↵
1▲2↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲a▲integer▲and▲a▲char:▲2▲c↵
Error!↵
[3:23 user@ws hw] ./a.out↵
```

```
Please input an integer and a char: 0 c
Error!
[3:23 user@ws hw] ./a.out
Please input an integer and a char: 3 q
Error!
[3:23 user@ws hw] ./a.out
Please input an integer and a char: 1 2 3
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: 271165

Permanent link:
<https://junwu.nptu.edu.tw/dokuwiki/doku.php?id=c:2025fall-hw7>

Last update: 2025/11/26 14:21

