

# Turnin作業6

- chapter 7
- Turnin Code: **c.hw6**
- Due Date: 2025/11/11 23:59 **Hard Deadline**

## 繳交方式說明

本次Turnin作業包含10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hw6的資料夾，並且切換到該資料夾後，針對第1題所要求的IsPrimeNumber.c進行程式編撰：

```
[user@ws ~]$ mkdir hw6
[user@ws ~]$ cd hw6
[user@ws hw6]$ joe IsPrimeNumber.c
```

等到我們完成DateUK2US.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.hw6 IsPrimeNumber.c` 指令將加以上傳：

```
[user@ws c.hw6]$ turnin c.hw6 IsPrimeNumber.c
Turning in:
IsPrimeNumber.c -- ok
All done.
[user@ws c.hw6]$
```

當然，你也可以等到本次作業要求的所有題目都在hw6資料夾裡完成後，再使用 `turnin c.hw6 .` 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hw6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.hw6]$ ls
CoffeePoints.c    CountDivisible.c  CumulativeMean.c
DateToEnglish.c  FindAllFactors.c  IsPrimeNumber.c
PerfectNumber.c  SmartVending.c    SquarePyramidalNum.c
StoryMaker.c     a.out
[user@ws c.hw6]$ rm a.out
[user@ws c.hw6]$ ls
CoffeePoints.c    CountDivisible.c  CumulativeMean.c
DateToEnglish.c  FindAllFactors.c  IsPrimeNumber.c
```

```
PerfectNumber.c SmartVending.c SquarePyramidalNum.c
StoryMaker.c
[user@ws c.hw6]$ turnin c.hw6 .
Turning in:
./SmartVending.c -- ok
./PerfectNumber.c -- ok
./IsPrimeNumber.c -- ok
./CoffeePoints.c -- ok
./SquarePyramidalNum.c -- ok
./CountDivisible.c -- ok
./DateToEnglish.c -- ok
./CumulativeMean.c -- ok
./StoryMaker.c -- ok
./FindAllFactors.c -- ok
All done.
[user@ws c.hw6]$
```

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.hw6↵` 指令，例如：

```
[user@ws c.hw6] c.hw6 $ turnin -ls c.hw6
.:
total 40
-rw-rw----. 1 turninman turnin 922 Nov 5 04:42 CoffeePoints.c
-rw-rw----. 1 turninman turnin 506 Nov 5 04:42 CountDivisible.c
-rw-rw----. 1 turninman turnin 468 Nov 5 04:42 CumulativeMean.c
-rw-rw----. 1 turninman turnin 1418 Nov 5 04:42 DateToEnglish.c
-rw-rw----. 1 turninman turnin 549 Nov 5 04:42 FindAllFactors.c
-rw-rw----. 1 turninman turnin 592 Nov 5 04:42 IsPrimeNumber.c
-rw-rw----. 1 turninman turnin 624 Nov 5 04:42 PerfectNumber.c
-rw-rw----. 1 turninman turnin 2876 Nov 5 04:42 SmartVending.c
-rw-rw----. 1 turninman turnin 508 Nov 5 04:42 SquarePyramidalNum.c
-rw-rw----. 1 turninman turnin 1151 Nov 5 04:42 StoryMaker.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 日期英文轉換 `DateToEnglish.c`

### 【題目敘述】

設計一個 C 語言程式 `DateToEnglish.c` 讓使用者輸入一個日期（格式為 `MM/DD/YYYY`）並輸出該日期的英文格式。

例如 `October 21st, 2025`

請注意日期中的日 `Day` 需根據英文序數詞規則輸出：

- 若日為 1[ ]21[ ]31 ⇒ st
- 若日為 2[ ]22 ⇒ nd
- 若日為 3[ ]23 ⇒ rd
- 其他日期 ⇒ th

若輸入的月份、日期或年份不合理，請輸出 “Input error!” 並終止程式。

#### 【輸入說明】

- 輸入格式為 [MM/DD/YYYY]
- 年份輸入將不會超過4位數。
- 月份須介於 1~12。
- 日期須介於 1~31（不需考慮閏年）。
- 若輸入不符合格式或範圍，輸出 “Input error!” 並終止程式。

#### 【附加說明：每個月日期與大小月的天數】

請根據各月天數判斷日期是否合理：

- 大月（31天）：1月、3月、5月、7月、8月、10月、12月
- 小月（30天）：4月、6月、9月、11月
- 2月（28天），不考慮閏年。
- 若使用者輸入的日期超出該月份的天數，請輸出 “Input error!” 並結束程式。

此題的執行結果可參考如下：

```
[3:23 user@ws hw] ./a.out↵
Please▲enter▲a▲date▲(MM/DD/YYYY):▲12/21/2025↵
December▲21st,▲2025↵
[3:23 user@ws hw] ./a.out↵
Please▲enter▲a▲date▲(MM/DD/YYYY):▲8/3/1923↵
August▲3rd,▲1923↵
[3:23 user@ws hw] ./a.out↵
Please▲enter▲a▲date▲(MM/DD/YYYY):▲8/22/123↵
August▲22nd,▲123↵
[3:23 user@ws hw] ./a.out↵
Please▲enter▲a▲date▲(MM/DD/YYYY):▲01/01/0001↵
January▲1st,▲1↵
[3:23 user@ws hw] ./a.out↵
Please▲enter▲a▲date▲(MM/DD/YYYY):▲00/11/1952↵
Input▲error!↵
[3:23 user@ws hw] ./a.out↵
Please▲enter▲a▲date▲(MM/DD/YYYY):▲2/0/2033↵
Input▲error!↵
[3:23 user@ws hw] ./a.out↵
Please▲enter▲a▲date▲(MM/DD/YYYY):▲02/29/2025↵
Input▲error!↵
[3:23 user@ws hw]
```

## P2 判斷是否為質數 `IsPrimeNumber.c` (此處首字母為大寫的I(丕))

質數的定義是只能被1以及該數本身整除的自然數，且必須大於1。

請設計一個C語言程式 `IsPrimeNumber.c` 讓使用者輸入一個正整數，檢查該整數是否為質數。如果使用者所輸入的不是正整數，則請輸出 "Error!" 本題的執行結果可以參考如下：

```
[3:23 user@ws hw] ./a.out↵
Please input a positive integer: 2↵
2 is a prime number.↵
[3:23 user@ws hw] ./a.out↵
Please input a positive integer: 30↵
30 is not a prime number.↵
[3:23 user@ws hw] ./a.out↵
Please input a positive integer: -100↵
Error!↵
[3:23 user@ws hw] ./a.out↵
Please input a positive integer: 0↵
Error!↵
[3:23 user@ws hw] ./a.out↵
Please input a positive integer: 1↵
1 is not a prime number.↵
[3:23 user@ws hw]
```

## P3 計算1至1000間能被a與b整除的數字 `CountDivisible.c`

計算1至1000間能被a與b整除的數字，請設計一個C語言程式 `CountDivisible.c` 讓使用者輸入兩個整數a與b，找出所有介於1(含)至1000(含)間所有可以被a與b整除的數字。若沒有任何符合的數字，則請輸出 `None`；若是使用者所輸入的是小於等於0的整數，則請輸出 `Error` 此題的執行結果可參考如下：

```
[3:23 user@ws hw] ./a.out↵
Please input two positive integers: 115 20↵
460↵
920↵
[3:23 user@ws hw] ./a.out↵
Please input two positive integers: 120 40↵
120↵
240↵
360↵
480↵
600↵
720↵
840↵
960↵
[3:23 user@ws hw] ./a.out↵
```

```
Please input two positive integers: 999 666
None
[3:23 user@ws hw] ./a.out
Please input two positive integers: 0 30
Error
[3:23 user@ws hw]
```

## P4 因數 FindAllFactors.c

### 【題目敘述】

設計一個 C 語言程式 FindAllFactors.c 讓使用者輸入一個正整數  $a$  並輸出該數所有的正因數（由小到大排列）。

### 【輸入說明】

輸入一個正整數  $a$  若輸入的數字小於或等於 0，請輸出 "Input error!" 並終止程式。

此題的執行結果可參考如下：

```
[3:23 user@ws hw] ./a.out
Please enter a positive integer: 100
1,2,4,5,10,20,25,50,100
[3:23 user@ws hw] ./a.out
Please enter a positive integer: 55
1,5,11,55
[3:23 user@ws hw] ./a.out
Please enter a positive integer: 1
1
[3:23 user@ws hw] ./a.out
Please enter a positive integer: 0
Input error!
[3:23 user@ws hw]
```

## P5 完美數 PerfectNumber.c

若一個整數所有「真因數」的總和等於其本身的數值，則稱為「完美數 Perfect Number」

例如：6 的真因數有 1、2、3，且  $1 + 2 + 3 = 6$ ，因此 6 是一個完美數。

【注意：真因數是指除自身以外的因數，例如 6 的真因數不包含 6 本身】

### 【題目敘述】

請設計一個 C 語言程式 PerfectNumber.c 讓使用者輸入一個大於 0 的整數  $N$  並找出所有小於等於  $N$  的完美數，依序輸出結果。若找不到完美數則輸出 "None" 如果使用者輸入不正確  $N \leq 0$  請輸出錯誤訊息

“Error!” 並終止程式。

【輸入說明】

- 輸入一個整數  $N$   $1 \leq N \leq 1000$
- 若  $N \leq 0$  輸出 “Error!” 並結束程式。

此題的執行結果可參考如下：

```
[3:23 user@ws hw] ./a.out↵
Please input a positive integer N: 25↵
6 is a perfect number.↵
[3:23 user@ws hw] ./a.out↵
Please input a positive integer N: 50↵
6 and 28 are perfect numbers.↵
[3:23 user@ws hw] ./a.out↵
Please input a positive integer N: -1↵
Error!↵
[3:23 user@ws hw] ./a.out↵
Please input a positive integer N: 5↵
None.↵
[3:23 user@ws hw] ./a.out↵
Please input a positive integer N: 10000↵
6, 28, 496 and 8128 are perfect numbers.↵
[3:23 user@ws hw]
```

## P6 四角錐數 SquarePyramidalNum.c

四角錐數（或稱為金字塔數）是一個表示底面為正方形的金字塔能堆積的球數總和，其公式如下

$$S_n = 1^2 + 2^2 + 3^2 + \dots + n^2$$

其中  $n$  是金字塔的層數。如果輸入的值為負數，則顯示 “Error!” 其輸出結果可以參考以下的內容：

```
[3:23 user@ws hw] ./a.out↵
Please input an integer: 0↵
0 = 0↵
[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: 4↵
1 + 4 + 9 + 16 = 30↵
[3:23 user@ws hw] ./a.out↵
Please input an integer: 7↵
1 + 4 + 9 + 16 + 25 + 36 + 49 = 140↵
[3:23 user@ws hw] ./a.out↵
Please input an integer: 15↵
```

```

1▲+▲4▲+▲9▲+▲16▲+▲25▲+▲36▲+▲49▲+▲64▲+▲81▲+▲100▲+▲121▲+▲144▲+▲169▲+▲196▲+▲225▲=▲1
240↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲an▲integer:▲- 13↵
Error!↵
[3:23 user@ws hw]

```

## P7 累計平均值 CumulativeMean.c

請設計一個C語言程式 CumulativeMean.c。此程式主要請同學們設計讓使用者可以重複輸入多個數值，直到輸入的為負數為止，並且依據所輸入的數值來計算並取得累計平均值，請將所計算出的平均值四捨五入到小數點後第二位輸出。

【注：本題若有使用小數的相關計算請使用 double 型態進行計算】

本題的編譯及執行結果可參考如下：

```

[3:23 user@ws hw] ./a.out↵
Please▲input▲the▲value▲(#1):▲1.5↵
Latest▲input/Cumulative▲Mean=1.500000/1.50↵
Please▲input▲the▲value▲(#2):▲2.732↵
Latest▲input/Cumulative▲Mean=2.732000/2.12↵
Please▲input▲the▲value▲(#3):▲1.2626↵
Latest▲input/Cumulative▲Mean=1.262600/1.83↵
Please▲input▲the▲value▲(#4):▲- 1↵
↵
Finish!!↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲the▲value▲(#1):▲5.35↵
Latest▲input/Cumulative▲Mean=5.350000/5.35↵
Please▲input▲the▲value▲(#2):▲2.574↵
Latest▲input/Cumulative▲Mean=2.574000/3.96↵
Please▲input▲the▲value▲(#3):▲13.64↵
Latest▲input/Cumulative▲Mean=13.640000/7.19↵
Please▲input▲the▲value▲(#4):▲0↵
Latest▲input/Cumulative▲Mean=0.000000/5.39↵
Please▲input▲the▲value▲(#5):▲0↵
Latest▲input/Cumulative▲Mean=0.000000/4.31↵
Please▲input▲the▲value▲(#6):▲- 1↵
↵
Finish!!↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲the▲value▲(#1):▲2.624↵
Latest▲input/Cumulative▲Mean=2.624000/2.62↵
Please▲input▲the▲value▲(#2):▲63.256↵

```

```

Latest

```

## P8 咖啡打卡集點王☐CoffeePoints.c☐

現代人每天都需要一杯咖啡提神！在「好醒咖啡店」中，顧客每次買一杯咖啡都可以累積點數。集滿 10 點可以兌換一杯免費咖啡，並且點數會歸零重新累積。請設計一個 C 語言程式 CoffeePoints.c☐模擬顧客每天買咖啡累積點數的過程。

### 【題目敘述】

每天購買咖啡後，系統會詢問使用者是否繼續購買（輸入 Y/N☐☐若輸入杯數為負數，則輸出 “Input error!” 並結束程式。當點數達到或超過 10 點時，自動顯示☐Congratulations! You've earned a free coffee!☐☐並將點數歸零繼續累積。每杯咖啡消費 1 杯可得 1 點。最後，程式會輸出總購買杯數與最終剩餘點數。

### 【輸入說明】：

- 每次輸入購買的咖啡杯數（整數）。
- 每次輸入後，系統詢問是否繼續購買☐Y/N☐☐若輸入並非(Y/N)則輸出 “Input error!” 並結束程式。
- 若輸入杯數  $< 0$ ，輸出 “Input error!” 並結束程式。

### 【輸出說明】：

- 每次加點後，若點數達 10，輸出 “Congratulations! You've earned a free coffee!” 並歸零。
- 程式結束時，輸出總購買杯數與最終剩餘點數。

```

[3:23 user@ws hw] ./a.out↵
Please

```

```

Congratulations!▲You've▲earned▲a▲free▲coffee!↵
Do▲you▲want▲to▲continue▲buying▲(Y/N)?▲N↵
Total▲cups▲bought:▲11↵
Free▲coffees▲earned:▲1↵
Remaining▲points:▲1↵
[3:23 user@ws hw] ./a.out↵
Please▲enter▲the▲number▲of▲cups:▲8↵
Do▲you▲want▲to▲continue▲buying▲(Y/N)?▲Y↵
Please▲enter▲the▲number▲of▲cups:▲3↵
Congratulations!▲You've▲earned▲a▲free▲coffee!↵
Do▲you▲want▲to▲continue▲buying▲(Y/N)?▲Y↵
Please▲enter▲the▲number▲of▲cups:▲9↵
Congratulations!▲You've▲earned▲a▲free▲coffee!↵
Do▲you▲want▲to▲continue▲buying▲(Y/N)?▲Y↵
Please▲enter▲the▲number▲of▲cups:▲1↵
Do▲you▲want▲to▲continue▲buying▲(Y/N)?▲N↵
Total▲cups▲bought:▲21↵
Free▲coffees▲earned:▲2↵
Remaining▲points:▲1↵
[3:23 user@ws hw] ./a.out↵
Please▲enter▲the▲number▲of▲cups:▲3↵
Do▲you▲want▲to▲continue▲buying▲(Y/N)?▲7↵
Input▲error!↵
[3:23 user@ws hw] ./a.out↵
Please▲enter▲the▲number▲of▲cups:▲- 1↵
Input▲error!↵
[3:23 user@ws hw]

```

### \*P9 阿文的自動販賣機□SmartVending.c□

阿文來到宿舍自動販賣機前，販賣機提供三種商品：

代號	商品名稱	價格
A	麥神奶茶	15 元
B	伯伯咖啡	30 元
C	超級科學麵	20 元

他可以重複投入 1 元、5 元、10 元、50 元的硬幣。當他投入完後，可以選擇想購買的商品。請設計一個 C 程式 SmartVending.c□模擬投入與購買的過程，並輸出購買結果與剩餘金額。

【輸入說明】：

- 使用者首先輸入多筆硬幣金額（1、5、10、50），以 0 結束投入階段，若輸入為負數，則輸出 “Error!” 並結束程式。
- 若輸入非 1、5、10、50、0，則輸出 “Error!” 並結束程式。
- 接著輸入要購買多少件商品，若輸入並非大於等於 1，則輸出 “Error!”□並結束程式。

- 然後輸入要購買的商品代號 A B C 每輸入一個代號代表購買一件商品。
- 若餘額不足以購買該商品，則輸出 "Insufficient funds!" 並停止購買。
- 若輸入非 A B C 則輸出 "Error!" 並結束程式。
- 所有輸入結束後，輸出購買清單與剩餘金額。

【輸出說明】：

- 輸出購買的所有商品種類及數量（若有），以及剩餘金額。

```
[3:23 user@ws hw] ./a.out
Please input the amount of money (in dollars): 10
Please input the amount of money (in dollars): 5
Please input the amount of money (in dollars): 1
Please input the amount of money (in dollars): 0
How many products would you like to buy? 1
#1 ==> Please select a product (A/B/C): A
You have purchased 1 product: A
Remaining balance: 1 dollar
[3:23 user@ws hw] ./a.out
Please input the amount of money (in dollars): 50
Please input the amount of money (in dollars): 10
Please input the amount of money (in dollars): 5
Please input the amount of money (in dollars): 1
Please input the amount of money (in dollars): 0
How many products would you like to buy? 3
#1 ==> Please select a product (A/B/C): C
#2 ==> Please select a product (A/B/C): A
#3 ==> Please select a product (A/B/C): B
You have purchased 3 products: ABC
Remaining balance: 1 dollar
[3:23 user@ws hw] ./a.out
Please input the amount of money (in dollars): 50
Please input the amount of money (in dollars): 50
Please input the amount of money (in dollars): 50
Please input the amount of money (in dollars): 50
Please input the amount of money (in dollars): 5
Please input the amount of money (in dollars): 1
Please input the amount of money (in dollars): 0
How many products would you like to buy? 5
#1 ==> Please select a product (A/B/C): A
#2 ==> Please select a product (A/B/C): A
#3 ==> Please select a product (A/B/C): B
#4 ==> Please select a product (A/B/C): C
#5 ==> Please select a product (A/B/C): A
You have purchased 5 products: AAABC
Remaining balance: 111 dollars
[3:23 user@ws hw] ./a.out
Please input the amount of money (in dollars): 5
Please input the amount of money (in dollars): 1
Please input the amount of money (in dollars): 0
```

```

How many products would you like to buy? 1
#1=>Please select a product (A/B/C): A
Insufficient funds!
[3:23 user@ws hw] ./a.out
Please input the amount of money (in dollars): -1
Error!
[3:23 user@ws hw] ./a.out
Please input the amount of money (in dollars): 5
Please input the amount of money (in dollars): 10
Please input the amount of money (in dollars): 50
Please input the amount of money (in dollars): 0
How many products would you like to buy? -1
Error!
[3:23 user@ws hw] ./a.out
Please input the amount of money (in dollars): 10
Please input the amount of money (in dollars): 5
Please input the amount of money (in dollars): 1
Please input the amount of money (in dollars): 0
How many products would you like to buy? 1
#1=>Please select a product (A/B/C): D
Error!
[3:23 user@ws hw] ./a.out
Please input the amount of money (in dollars): 0
How many products would you like to buy? 0
Error!
[3:23 user@ws hw]

```

## P10 阿金的故事產生器 StoryMaker.c

請使用C語言設計一個阿金的故事產生器 StoryMaker.c 讓我們輸入「在哪裡？」、「做什麼？」等兩個選擇，然後由程式輸出產生一個簡短小故事。具體來說，我們首先必須輸入一個代表地點的字元，其中以 S 代表學校 R 代表餐廳 G 代表花園；最後再輸入代表「做什麼」的字元，其中 R 代表看書 S 代表睡覺 W 代表上班。請取得使用者的輸入後，產生一段簡單的故事描述。為便利起見，我們將輸入的字元彙整如下：

代號	地點名稱
S / s	the school
R / r	the restaurant
G / g	the garden
其他英文字元	an unknown place
代號	活動名稱
R / r	reading
S / s	sleeping
W / w	working
其他英文字元	doing an unknown activity

以上的輸入字元，大小寫都視為正確。但輸入超出以上範圍(英文字母A~Z)的字元時，必須顯示[]Error![]  
請參考以下的程式執行結果：

```
[3:23 user@ws hw] ./a.out↵
gg↵
Mr.▲Money▲is▲doing▲an▲unknown▲activity▲in▲the▲garden.↵
[3:23 user@ws hw] ./a.out↵
sR↵
Mr.▲Money▲is▲reading▲in▲the▲school.↵
[3:23 user@ws hw] ./a.out↵
rW↵
Mr.▲Money▲is▲working▲in▲the▲restaurant.↵
[3:23 user@ws hw] ./a.out↵
zR↵
Mr.▲Money▲is▲reading▲in▲an▲unknown▲place.↵
[3:23 user@ws hw] ./a.out↵
!@↵
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: 297445

Permanent link:  
<https://junwu.nptu.edu.tw/dokuwiki/doku.php?id=c:2025fall-hw6>

Last update: **2025/11/30 12:22**

