Turnin作業3

- Turnin Code: cpp.hw3
- Due Date: 3/9 Sunday 23:59 (midnight) **Hard Deadline**
- 本次作業繳交期限為3月9日週日晚上11點59分0秒！

繳交方式說明

本次Turnin作業包含多個程式題，建議同學可以為這次作業先建立一個資料夾hw3，然後在該資料夾內再為每一題建立一個子資料夾，用以進行每一題的作答以及上傳。每一題的子資料夾名稱已寫於題目前方，請務必依照題目的規定建立子資料夾，例如第1題為p1，第2題為p2，餘依此類推。當我們完成某一個題目的作答後，就可以使用turnin指令將該題的答案上傳。以第1題為例，當我們在p1子資料夾裡完成作答後，就可以回到hw3資料夾，使用以下指令將其上傳：

[3:23 user@ws hw1] `turnin▲cpp.hw3▲p1↵`

當然，你也可以等到所有題目都完成後，再回到hw1資料夾，使用以下指令將所有題目都加以上傳：

[3:23 user@ws hw1] `turnin▲cpp.hw3▲.↵`



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，題目的執行結果中，如果出現(、)、:、;、.與,等符號，皆為英文半形！

---

p1 判斷成績是否及格

請設計一個C++語言的程式passfail.cpp，讓使用者輸入成績。並依據所輸入的成績輸出不同的訊息如下：

- 若成績大於等於60分，則輸出“ 及格 ”
- 若剛好60分，則輸出“ 剛好及格 ”
- 若成績大於等於90分，則輸出“ 高分通過 ”
- 若成績小於60分，則輸出不及格“ (還差X分) ”
- 另外，如果使用者所輸入的成績超過100分或低於0分，則請輸出“ 錯誤 ”

```
[3:23 user@ws hw] ./a.out↵
90↵
高分通過
[3:23 user@ws hw] ./a.out↵
59↵
不及格(還差1分)
```

```
[3:23 user@ws hw] ./a.out↵
60↵
剛好及格
[3:23 user@ws hw] ./a.out↵
75↵
及格
[3:23 user@ws hw] ./a.out↵
101↵
錯誤
[3:23 user@ws hw]
```

p2 依照成績輸出其等第

請設計一個C++語言的程式grade.cpp讓使用者輸入成績，並依據所輸入的成績換算為對應的成幾等第後輸出(若使用者輸入的分數超出0~100的範圍，則輸出“ 錯誤 ”)。成績與等第之對應可參考如下：

- A:100~90分
- B:89~80分
- C:79~70分
- D:69~60分
- F:59~0分

此程式執行結果請參考如下：

```
[3:23 user@ws hw] ./a.out↵
90↵
A↵
[3:23 user@ws hw] ./a.out↵
100↵
A↵
[3:23 user@ws hw] ./a.out↵
83↵
B↵
[3:23 user@ws hw] ./a.out↵
55↵
F↵
[3:23 user@ws hw] ./a.out↵
101↵
錯誤
[3:23 user@ws hw]
```

p3 在何時、何地、哪裡做什麼？

請設計一個C++語言的程式story.cpp讓使用者輸入“ 何時？ ”、“ 在哪裡？ ”、與“ 做什麼？ ”等三個

選擇。然後由程式輸出一個簡短小故事。具體來說，我們首先必須輸入一個代表時間的字元，分述如下：

- M:代表早上
- A:代表下午
- E:代表晚上

另外，我們還要輸入一個代表地點的字元，其中

- S:代表學校
- R:代表餐廳
- G:代表花園

最後再輸入一個代表地點的字元，其中

- R:代表看書
- S:代表睡覺
- W:代表工作

請取得使用者的輸入後（大小寫都視為正確），產生一段簡單的故事描述。但輸入超出以上範圍的字元時，必須顯示其錯誤！詳細的輸出結果（包含正確與錯誤）請參考以下的執行結果：

```
[3:23 user@ws hw] ./a.out↵
MGS↵
阿財在早上到花園睡覺
[3:23 user@ws hw] ./a.out↵
aRW↵
阿財在下午到餐廳工作
[3:23 user@ws hw] ./a.out↵
XSZ↵
阿財在錯誤的時間到學校做錯誤的事
[3:23 user@ws hw] ./a.out↵
iha↵
阿財在錯誤的時間到錯誤的地方做錯誤的事
[3:23 user@ws hw]
```

p4 屏東大哥大電信公司

方案1:月租費499元，每個月不限流量（吃到飽方案）。 方案2:0月租費，每月費用以實際流量計算，每MB 0.1 元。 方案3:月租費299元，包含每個月免費4GB流量，超過的部分，每MB另外加收0.15元。

請設計一個C++語言的程式phonebill.cpp用以計算上網費用。使用者必須輸入其所使用的方案（整數值1-3），並輸入該月份的流量（以MB為單位，且每1024MB等於1GB計算該月的費用後加以輸出。



【注意】：

1. 費用可能會有小數的部分，請四捨五入到整位數。



2. 若有處理浮點數的需求，請使用double型態。

本題的執行結果可參考如下：

```
[3:23 user@ws hw] ./a.out↵
1↵
40000↵
499↵
[3:23 user@ws hw] ./a.out↵
3↵
1050↵
299↵
[3:23 user@ws hw] ./a.out↵
2↵
1000↵
100↵
[3:23 user@ws hw] ./a.out↵
5↵
99↵
Error!↵
[3:23 user@ws hw]
```

p5 計算從1到N的相加數值

請設計一個C++語言的程式sum.cpp，讓使用者輸入一個整數n後，計算並輸出1+2+...+n的數值。此題的執行結果可參考如下:

```
[3:23 user@ws hw] ./a.out↵
n=?▲10↵
Sum▲of▲1▲to▲10▲is▲55↵
[3:23 user@ws hw] ./a.out↵
n=?▲30↵
Sum▲of▲1▲to▲30▲is▲465↵
[3:23 user@ws hw]
```

p6 在1~100中找出可以被a與b整除的數字

請設計一個C++語言的程式divisible.cpp，讓使用者輸入兩個整數a與b，找出所有介於1至100(含)間同時可以被a與b整除的數字。此題的執行結果可參考如下：

```
[3:23 user@ws hw] ./a.out↵
Please▲input▲two▲numbers:▲15▲20↵
60↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲two▲numbers:▲3▲8↵
24↵
48↵
72↵
96↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲two▲numbers:▲76▲3↵
None↵
[3:23 user@ws hw]
```

p7 計算並輸出平均成績

請撰寫一個C++程式average.cpp讓使用者輸入多筆成績，直到輸入的成績為-1時結束，並輸出平均成績。要注意的是，使用者所輸入的成績必須介於0到100的範圍，超出範圍則顯示錯誤。程式執行結果請參考如下:

```
[3:23 user@ws hw] ./a.out↵
Number▲#1:▲100↵
Number▲#2:▲150↵
Wrong▲Input!▲Number▲#2:▲-10↵
Wrong▲Input!▲Number▲#2:▲0↵
Number▲#3:▲0↵
Number▲#4:▲-1↵
Average▲Score▲is▲33.3333↵
[3:23 user@ws hw] ./a.out↵
Number▲#1:▲60↵
Number▲#2:▲70↵
Number▲#3:▲100↵
Number▲#4:▲80↵
Number▲#5:▲-1↵
Average▲Score▲is▲77.5↵
[3:23 user@ws hw] ./a.out↵
Number▲#1:▲100↵
Number▲#2:▲100↵
Number▲#3:▲100↵
Number▲#4:▲0↵
Number▲#5:▲61↵
Number▲#6:▲57↵
Number▲#7:▲-1↵
Average▲Score▲is▲69.6667↵
```

```
[3:23 user@ws hw]
```

p8 找出完美數

若一個整數所有真因數（意即除了本身以外的其它因數）的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與3，又1+2+3=6所以6是一個完美數。
請設計一個C++語言程式perfect.cpp，讓使用者輸入一個大於0的整數N。若使用者輸入不正確的數值，請印出錯誤訊息），找出小於等於N的數字中所有的完美數後加以輸出。此題的執行結果可參考如下：

```
[3:23 user@ws hw] ./a.out↵
Please▲input▲a▲number:▲3↵
Perfect▲number▲was▲not▲found!↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲a▲number:▲10↵
6▲is▲a▲perfect▲number.↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲a▲number:▲1000↵
6▲is▲a▲perfect▲number.↵
28▲is▲a▲perfect▲number.↵
496▲is▲a▲perfect▲number.↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲a▲number:▲0↵
Error↵
[3:23 user@ws hw]
Please▲input▲a▲number:▲-1↵
Error↵
[3:23 user@ws hw]
```

p9 密碼驗證系統

請撰寫一個 C++ 程式password.cpp，模擬一個簡單的密碼驗證系統。程式會要求使用者輸入一個五位數字的密碼(正確的密碼為 “ 12345 ”)。

1. 若輸入的密碼正確，則輸出Access granted! 並結束程式。
2. 若輸入錯誤，則輸出Incorrect password! Try again.並要求使用者重新輸入。
3. 使用者最多可以嘗試 3 次，如果 3 次都輸入錯誤，則輸出Too many failed attempts! Access denied.

```
[3:23 user@ws hw] ./a.out↵
Enter▲password:▲88888↵
Incorrect▲password!▲Try▲again.↵
Enter▲password:▲09876↵
```

```
Incorrect▲password!▲Try▲again.↵
Enter▲password:▲54321↵
Incorrect▲password!▲Try▲again.↵
Too▲many▲failed▲attempts!▲Access▲denied.↵
[3:23 user@ws hw] ./a.out↵
Enter▲password:▲12345↵
Access▲granted!↵
[3:23 user@ws hw]
```

p10 兔子生存

費柏納西曾對提出以下的兔子問題：

- 每一對兔子有“新生”、“成長中”以及“已成長”等三種狀態
- 剛誕生的兔子為“新生”狀態，一個月後成為“成長中”狀態
- “成長中”的兔子還需要再一個月才能成為“已成長”的狀態
- 每對“已成長”的兔子都具有生育能力，每個月固定會生一對兔子
- 假設農場在第一個月有一對“新生”的兔子
- 所有兔子永不死去

請設計一個C++語言的程式 fibonacci.cpp讓使用者輸入月份，計算並輸出該月份兔子的總對數（若使用者輸入小於等於0的月份，則請輸出“Error!”
依據上述的規則，列出前10個月兔子的對數如下：

月份	新生兔子對數	成長中兔子對數	已成長兔子對數	總對數
1	1	0	0	1
2	0	1	0	1
3	1	0	1	2
4	1	1	1	3
5	2	1	2	5
6	3	2	3	8
7	5	3	5	13
8	8	5	8	21
9	13	8	13	34
10	21	13	21	55

```
[3:23 user@ws hw] ./a.out↵
Month:▲0↵
Error!↵
[3:23 user@ws hw] ./a.out↵
Month:▲-1↵
Error!↵
[3:23 user@ws hw] ./a.out↵
Month:▲1↵
There▲is▲1▲pair▲of▲rabbits.↵
```

```
[3:23 user@ws hw] ./a.out↵
Month:▲11↵
There▲are▲89▲pairs▲of▲rabbits.↵
[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: 343751

Permanent link:
https://junwu.nptu.edu.tw/dokuwiki/doku.php?id=cpp:2025spring:hw3

Last update: **2025/03/09 06:47**