Turnin作業6

- Turnin Code: c.hw6
- Due Date: 2026/10/20 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，從本次作業6開始，將省略繳交步驟示範。請同學在自行確認**8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 輸出對齊(align.c)

請同學設計一個 C 語言程式 align.c，讓使用者輸入一個浮點數（可帶正負號）後，使用格式化輸出的方式含正負號在內以7個位置靠右對齊加以輸出(小數點顯示至第2位)。

本題的執行結果可參考以下的內容：

```
[7:17▲user@ws▲in▲hw6]$▲./a.out↵
Enter▲number:▲123.2↵
Output:▲[+123.20]↵
[7:17▲user@ws▲in▲hw6]$▲./a.out↵
Enter▲number:▲+1.1666↵
Output:▲[▲▲+1.17]↵
[7:17▲user@ws▲in▲hw6]$▲./a.out↵
Enter▲number:▲-123456.41↵
```

```
Output:▲[-123456.41]↵
[7:17▲user@ws▲in▲hw6]$▲./a.out↵
Enter▲number:▲-123.456↵
Output:▲[-123.46]↵
[7:17▲user@ws▲in▲hw6]$▲./a.out↵
Enter▲number:▲-45.771↵
Output:▲[▲-45.77]↵
[7:17▲user@ws▲in▲hw6]$▲./a.out↵
Enter▲number:▲15917.342↵
Output:▲[+15917.34]↵
[7:17▲user@ws▲in▲hw6]$▲
```

---

p2 計算總位數（count_digits.c）

請同學設計一個 C 語言程式count_digits.c，讓使用者輸入的一個正整數（資料型態為unsigned long）後，計算並輸出該數值的位數。

本題的執行結果可參考以下的內容：

```
[7:17▲user@ws▲in▲hw6]$▲./a.out↵
Number:▲10000↵
Number▲10000▲has▲5▲digit(s)↵
[7:17▲user@ws▲in▲hw6]$▲./a.out↵
Number:▲1237682471↵
Number▲1237682471▲has▲10▲digit(s)↵
[7:17▲user@ws▲in▲hw6]$▲./a.out↵
Number:▲326506210482↵
Number▲326506210482▲has▲12▲digit(s)↵
[7:17▲user@ws▲in▲hw6]$▲
```



本題批改測試輸入數值不超過 18446744073709551615

---

p3 日期曆法轉換（date_convert.c）

世界各地的日期表達方式不盡相同，以台灣、英國和美國為例，其慣用的表示法分別為YYY/MM/DD（DD/MM/YYYY與MM/DD/YYYY）也就是台式的年/月/日、英式的日/月/年，以及美式的月/日/年）。

請設計一個C語言程式date_convert.c讓使用者輸入一個台式日期(年/月/日)後，轉換成英式日期及美式日期後輸出。

本題的執行結果可參考以下的內容：

```
[7:17▲user@ws▲in▲hw6]$▲./a.out↵
TW▲Date▲(YYY/MM/DD)?▲115/10/04↵
UK▲Date:▲04/10/2026↵
US▲Date:▲10/04/2026↵
[7:17▲user@ws▲in▲hw6]$▲./a.out↵
TW▲Date▲(YYY/MM/DD)?▲058/07/11↵
UK▲Date:▲11/07/1969↵
US▲Date:▲07/11/1969↵
[7:17▲user@ws▲in▲hw6]$▲./a.out↵
TW▲Date▲(YYY/MM/DD)?▲001/05/05↵
UK▲Date:▲05/05/1912↵
US▲Date:▲05/05/1912↵
[7:17▲user@ws▲in▲hw6]$▲
```



- 本題測試輸入的數值皆為正數。

---

p4 時間與小時轉換（time_in_hours.c）

請同學設計一個C語言程式time_in_hours.c讓使用者輸入一個時間（必須以HH:MM:SS的格式來輸入時、分、秒，其中各項數值皆不可超過兩位數），然後計算並轉換為小時後將其加以輸出。例如輸入「00:30:00」應該轉換為0.5小時。

輸出時，請使用8個位置來輸出轉換後的結果（其中包含整數佔2位，小數點佔1位，小數點後的數值佔5位），若有不足位數的情況（不論是整數或是小數部分）則請補0。

本題的執行結果可參考以下的內容：

```
[7:17▲user@ws▲in▲hw6]$▲./a.out↵
Time▲(HH:MM:SS)?▲00:30:00↵
00:30:00▲is▲equal▲to▲00.50000▲hour(s)↵
[7:17▲user@ws▲in▲hw6]$▲./a.out↵
Time▲(HH:MM:SS)?▲00:17:31↵
00:17:31▲is▲equal▲to▲00.29194▲hour(s)↵
[7:17▲user@ws▲in▲hw6]$▲./a.out↵
Time▲(HH:MM:SS)?▲01:45:00↵
01:45:00▲is▲equal▲to▲01.75000▲hour(s)↵
```

```
[7:17▲user@ws▲in▲hw6]$▲
```

---

p5 統一發票（last_4.c）

請同學設計一個C語言程式last_4.c，讓使用者輸入一組統一發票的號碼,並且將其最後4碼印出。

請注意統一發票號碼的格式，其前2碼為英文字母,後面則接8碼的數字。

本題的執行結果可參考以下的內容：

```
[7:17▲user@ws▲in▲hw6]$▲./a.out↵
Please▲input▲a▲uniform-invoice▲number:▲AC38548029↵
The▲last▲4▲digits▲are▲8029.↵
[7:17▲user@ws▲in▲hw6]$▲./a.out↵
Please▲input▲a▲uniform-invoice▲number:▲DE38548324↵
The▲last▲4▲digits▲are▲8324.↵
[7:17▲user@ws▲in▲hw6]$▲./a.out↵
Please▲input▲a▲uniform-invoice▲number:▲NC24480293↵
The▲last▲4▲digits▲are▲0293.↵
[7:17▲user@ws▲in▲hw6]$▲
```

---

p6 發票號碼加總（sum_of_last_4.c）

承上題，請設計一個C語言程式sum_of_last_4.c，將取得的發票最後4碼的值加總後輸出。

本題的執行結果可參考以下的內容：

```
[7:17▲user@ws▲in▲hw6]$▲./a.out↵
Enter▲invoice:▲FB53528029↵
The▲sum▲of▲last▲4▲digits▲is▲19↵
[7:17▲user@ws▲in▲hw6]$▲./a.out↵
Enter▲invoice:▲CK90287910↵
The▲sum▲of▲last▲4▲digits▲is▲17↵
[7:17▲user@ws▲in▲hw6]$▲./a.out↵
Enter▲invoice:▲YT57087402↵
The▲sum▲of▲last▲4▲digits▲is▲13↵
[7:17▲user@ws▲in▲hw6]$▲
```

p7 平均運動速度（average_pace.c）

最近小萱開始進行跑步耐力訓練。為了方便計算每次跑步的配速，她設計了一款手機 APP 來紀錄每次戶外跑步的總時間以及距離，並且透過網路將這些資訊以純文字檔的方式儲存在電腦系統中。

她所設計用以儲存運動資訊的純文字檔，其格式如下：

- 第一行格式為「HH:MM:SS」，代表此次跑步所花費的時間（HH 為小時、MM 為分鐘、SS 為秒），稱為耗時。
- 第二行格式為「XXXX.XX」，代表此次跑步的總距離（公里），整數部分最多 4 位，小數部分最多 2 位，稱為里程。

請同學幫忙設計一個 C 語言程式 average_pace.c，該程式需依序讀取上述檔案格式中的時間與距離，並計算出本次運動的配速。配速的單位為「分鐘/公里」（意即每跑一公里耗費多少分鐘），其計算公式為「耗時/里程」。注意計算前必須先將耗時轉換為分鐘，例如耗時為01:20:30應先轉換為80.5分鐘，若里程為十公里則配速可計算為 80.5/10=8.05分鐘，因此配速應輸出為8:03/KM。



在執行測試時，請使用**輸入重導向 (Input Redirection, <)** 將測試檔的內容傳入程式。

本題的執行結果可參考以下的內容：

```
[7:17▲user@ws▲in▲hw6]$▲cat▲1.in↵
01:20:30↵
10↵
[7:17▲user@ws▲in▲hw6]$▲./a.out < 1.in↵
8:03/KM↵
[7:17▲user@ws▲in▲hw6]$▲cat▲2.in↵
02:35:00↵
33.10↵
[7:17▲user@ws▲in▲hw6]$▲./a.out < 2.in↵
4:40/KM↵
[7:17▲user@ws▲in▲hw6]$▲cat▲3.in↵
15:45:31↵
233.5↵
[7:17▲user@ws▲in▲hw6]$▲./a.out < 3.in↵
4:02/KM↵
[7:17▲user@ws▲in▲hw6]$▲
```



- 本題測試輸入的數值皆為**正數**。



- 所有關於浮點數的計算請使用 **double型態** 處理。



本題有提供測試檔，同學可透過以下路徑將測試檔複製下來進行測試：

- /home/stu/public/c2026f/c.hw6/p7/1.in
- /home/stu/public/c2026f/c.hw6/p7/2.in
- /home/stu/public/c2026f/c.hw6/p7/3.in

---

p8 輸入重導向與管線輸出（redirect_in.c, pipe_out.c）

請設計一個 C 語言程式redirect_in.c，該程式會透過輸入重導向 (redirect)傳入五個整數，並將這五個整數分別乘以2後輸出（數字之間請以空白字元隔開）。

接著，請設計另一個 C 語言程式pipe_out.c，透過管線(pipeline)接收前一個程式的輸出做為其輸入，並將其全部相加計算後輸出。



1. **指定輸出程式檔名**：在編譯時加上 `-o` 檔名 參數，即可自訂編譯出的執行檔名稱，例如`cc redirect_in.c -o redirect_in`會編譯出名為**redirect_in**的執行檔，而不是預設的**a.out**。
2. **輸入重導向 (<)**：在終端機執行時加上 < 檔案名稱，即可將該檔案的內容當作程式的輸入。因此在 redirect_in.c 中，只需像平常一樣使用 **scanf** 讀取即可。
3. **管線 (|)**：在終端機執行時使用 |，作業系統會自動將左邊程式（**redirect_in**）印在螢幕上的輸出，轉交給右邊程式（**pipe_out**）做為輸入。同樣的在 pipe_out.c 中，只需使用 **scanf** 來接收資料即可。

本題的執行結果可參考以下的內容：

```
[7:17 user@ws in hw6]$ cat 1.in
1 2 3 4 5
[7:17 user@ws in hw6]$ ./redirect_in < 1.in | ./pipe_out
Sum: 30
[7:17 user@ws in hw6]$ cat 2.in
-1 2 -3 4 -5
[7:17 user@ws in hw6]$ ./redirect_in < 2.in | ./pipe_out
Sum: -6
[7:17 user@ws in hw6]$ cat 3.in
1 3 5 7 9
[7:17 user@ws in hw6]$ ./redirect_in < 3.in | ./pipe_out
Sum: 50
```

```
[7:17▲user@ws▲in▲hw6]$▲
```



本題需要同時繳交以下2個檔案：

- redirect_in.c
- pipe_out.c

若以上檔案缺失一個將視為**0分**



本題有提供測試檔，同學可透過以下路徑將測試檔複製下來：

- /home/stu/public/c2026f/c.hw6/p8/1.in
- /home/stu/public/c2026f/c.hw6/p8/2.in
- /home/stu/public/c2026f/c.hw6/p8/3.in

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:hw6

Last update: **2026/10/14 06:05**