

Turnin 作業 2

- Turnin Code: **cpp.hw2**
- Due Date: **3/9 Monday 23:59:00** (midnight) **Hard Deadline**

繳交方式說明

本次 Turnin 作業包含多個程式題，建議先為本次 turnin 要繳交的內層資料夾（例如 hw2）切換到該資料夾後再為每一題建立一個內層資料夾（每一題的資料夾名稱已寫於題目前方，例如第一題的資料夾名為 p1 第二題的為 p2 餘以此類推），進入到內層資料夾才依照題目要求進行編撰。

同學們可參考如下命令列操作：

```
<ssh 登入系計中後>
[user@ws ~]$ mkdir hw2           # 在家目錄建立了一個名為 hw2 的資料夾
[user@ws ~]$ cd hw2             # 進入 hw2 資料夾
[user@ws hw2]$ mkdir p1         # 建立一個名為 p1 資料夾
[user@ws hw2]$ cd p1           # 切換到 p1 資料夾
[user@ws p1]$ joe align.cpp     # 使用 JOE 編輯器對檔名 align.cpp 的檔案進行編輯
```

等到我們完成 p1 的撰寫後，請自行加以編譯與執行程式，確認正確無誤後回到外層資料夾使用 `turnin ^cpp.hw2 ^p1` 指令完成繳交第一題的整個資料夾

```
[user@ws p1]$ cd ..             # 回到上一層資料夾
[user@ws hw2]$ turnin cpp.hw2 p1 # 使用 turnin 指令提交 p1 的程式碼
Turning in:
p1/align.cpp -- ok
All done.
[user@ws hw2]$
```

當然，你也可以等到本次作業要求的所有題目都在 hw2 資料夾裡完成後，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 hw2 資料夾裡完成所有題目，同時確認檔案的繳交格式正確，並且每個題目的程式檔案皆成功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 hw2]$ ls              # 檢視當前資料夾下有哪些內容
p1 p2 p3 p4 p5 p6
[user@ws hw2]$ rm -f */a.out   # 移除所有子資料夾中的 a.out 檔案
```

```
[user@ws hw2]$ turnin cpp.hw2 . # 使用 turnin 指令繳交該資料夾下的所有內容
Turning in:
./p6/receipt.cpp -- ok
./p3/scientific_notation.cpp -- ok
./p1/align.cpp -- ok
./p2/align_center.cpp -- ok
./p5/fp_num_sum.cpp -- ok
./p4/numeral_system.cpp -- ok
All done.
[user@ws hw2]$
```

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 cpp.hw2` 指令，例如：

```
.:
total 24
drwxrwx---. 2 turninman turnin 4096 Feb 26 15:08 p1
drwxrwx---. 2 turninman turnin 4096 Feb 26 15:09 p2
drwxrwx---. 2 turninman turnin 4096 Feb 26 15:09 p3
drwxrwx---. 2 turninman turnin 4096 Feb 26 15:09 p4
drwxrwx---. 2 turninman turnin 4096 Feb 26 15:09 p5
drwxrwx---. 2 turninman turnin 4096 Feb 26 15:09 p6

./p1:
total 0
-rw-rw----. 1 turninman turnin 0 Feb 26 15:08 align.cpp

./p2:
total 0
-rw-rw----. 1 turninman turnin 0 Feb 26 15:09 align_center.cpp

./p3:
total 0
-rw-rw----. 1 turninman turnin 0 Feb 26 15:09 scientific_notation.cpp


./p4:
total 0
-rw-rw----. 1 turninman turnin 0 Feb 26 15:09 numeral_system.cpp

./p5:
total 0
-rw-rw----. 1 turninman turnin 0 Feb 26 15:09 fp_num_sum.cpp

./p6:
total 0
-rw-rw----. 1 turninman turnin 0 Feb 26 15:09 receipt.cpp
```



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，题目的執行結果中，如果出現「(」、「)」、

 「:」、「;」、「.」與「,」等符號，皆為英文半形！


本學期作業繳交需要為每一題建立一個資料夾（資料夾名稱為該題目前方之代號，第一題為p1第二題為p2餘以此類推），繳交方式可參考上述內容，任何未依照正確繳交格式的檔案將以 0 分計。

p1 字串對齊輸出

請設計一個 C++ 語言的程式 align.cpp接收使用者所輸入的一個不超過 20 的整數用以設定字串輸出的寬度，並將字串 "xyz" 與 "abcd" 分別使用靠右與靠左對齊的方式分別輸出，且沒有輸出字元的部分請使用「*」星號填充。

此題的執行結果可參考如下：

```
[3:23 user@ws p1] ./a.out↵
width=?▲20↵
*****xyz↵
abcd*****↵
[3:23 user@ws p1] ./a.out↵
width=?▲4↵
*xyz↵
abcd↵
[3:23 user@ws p1]
```

 1. 本題的輸入介於0~20之間。
2. 若指定的寬度少於欲輸出的字串長度時，則必須將字串內容完整呈現。

p2 字串置中輸出

請設計一個 C++ 語言的程式 align_center.cpp接收使用者所輸入的一個不超過 20 的整數用以設定字串輸出的寬度，並將字串 "xyz" 以置中對齊的方式加以輸出，其中沒有輸出字元的部分請用 * 星號填充。

本題的執行結果可參考如下：



```
[3:23 user@ws p2] ./a.out↵
width=?▲20↵
*****xyz*****↵
[3:23 user@ws p2] ./a.out↵
width=?▲5↵
*xyz*↵
[3:23 user@ws p2] ./a.out↵
width=?▲2↵
xyz↵
[3:23 user@ws p2]
```



1. 本題所謂的置中對齊，是指必須在 "xyz" 的左右兩側保留相同的位置空間。但若是左右兩側無法均等時，左側可以比右側少一個位置。
2. 例如：當設定寬度為 20 時 $(20-3) \div 2 = 8.5$ 此時設定左側與右側分別保留 8 與 9 個位置空間。

p3 浮點數與科學記號

請設計一個 C++ 語言的程式 scientific_notation.cpp 接收使用者所輸入的一個浮點數後，將其以科學記號表示法的方式加以輸出。請注意，以科學記號表示的結果中，小數部分請顯示到小數點第五位（即四捨五入到小數第五位）。

此題的執行結果可參考如下：

```
[3:23 user@ws p3] ./a.out↵
Please▲input▲a▲floating-point▲number:▲3.1415926↵
The▲number▲can▲be▲represented▲in▲scientific▲notation▲as▲3.14159e+00.↵
[3:23 user@ws p3] ./a.out↵
Please▲input▲a▲floating-point▲number:▲1234.5678864↵
The▲number▲can▲be▲represented▲in▲scientific▲notation▲as▲1.23457e+03.↵
[3:23 user@ws p3]
```

p4 10 進制轉換 8 與 16 進制

請設計一個 C++ 語言的程式 numeral_system.cpp 接收使用者所輸入的一個十進制的整數，並將其輸出為對應的 8 與 16 進制整數。

本題的執行結果可參考如下：

```
[3:23 user@ws p4] ./a.out↵
```

```

Please▲input▲a▲number:▲10↵
=====↵
decimal:▲↵[TAB]→10↵
octal:▲▲↵[TAB]→012↵
hexdecimal:▲↵[TAB]→0xA↵
=====↵
[3:23 user@ws p4] ./a.out↵
Please▲input▲a▲number:▲128↵
=====↵
decimal:▲↵[TAB]→128↵
octal:▲▲↵[TAB]→0200↵
hexdecimal:▲↵[TAB]→0x80↵
=====↵
[3:23 user@ws p4]

```

請注意，本題的執行結果中的 `↵[TAB]→` 是指 `tab` 符號（即跳脫字元 `'\t'`），請勿使用半形空白代替。另外，其所輸出的表格右側數字欄位中左到右的第一位數字應對齊。為便利同學起見，以上的執行結果以固定寬度字型再次顯示如下：



```

[3:23 user@ws p4] ./a.out↵
Please▲input▲a▲number:▲10↵
=====↵
decimal:▲          10↵
octal:▲▲          012↵
hexdecimal:▲      0xA↵
=====↵
[3:23 user@ws p4] ./a.out↵
Please▲input▲a▲number:▲128↵
=====↵
decimal:▲          128↵
octal:▲▲          0200↵
hexdecimal:▲      0x80↵
=====↵
[3:23 user@ws p4]

```

p5 浮點數加總

請撰寫一個 C++ 程式 `fp_num_sum.cpp` 讓使用者輸入兩個正浮點數，並計算它們的總和。輸出每一個浮點數時，請遵循以下規則：

1. 保留 12 格的寬度，不足的部分用 0 填滿。
2. 固定顯示 6 位小數。

本題的執行結果可參考如下：

```
[3:23 user@ws p5] ./a.out↵
Please input the first floating-point number: 3.1415↵
Please input the second floating-point number: 5674.423↵
  00003.141500↵
+ )05674.423000↵
-----↵
  05677.564500↵
[3:23 user@ws p5] ./a.out↵
Please input the first floating-point number: 8935.4513↵
Please input the second floating-point number: 193.1243↵
  08935.451300↵
+ )00193.124300↵
-----↵
  09128.575600↵
[3:23 user@ws p5]
```

1. 測試檔所輸入的每一個浮點數限制如下：
 1. \$ 0 \le 整數部分 \le 9999\$
 2. \$ 0.9999 \ge 小數部分 \ge 0.0001\$
2. 本題所輸出直式計算式中每一個數字欄位應對齊，可參考如下輸出：



```
Please input the first floating-point number: 3.1415↵
Please input the second floating-point number: 5674.423↵
  00003.141500↵
+ )05674.423000↵
-----↵
  05677.564500↵
```

p6 哈拉少銀行提款機取款收據

請為哈拉少銀行的提款機設計一個能夠列印提款收據的程式 receipt.cpp 此程式在執行時，會將接收一行以半形逗點隔開的「提款交易明細資料」，其格式如下：

交易日期,交易時間,ATM編號,銀行代號,銀行帳號,取款金額,手續費

其各欄位說明如下：

1. 交易日期：7 位數，以民國紀年（為 3 位正整數）、月（為 2 位正整數）、日（為 2 位正整數）組成
2. 交易時間：6 位數，以時、分、秒組成（皆為 2 位正整數）

- 3. ATM編號：9 位數
- 4. 銀行代號：3 位數
- 5. 銀行帳號：12 位數
- 6. 取款金額：5 位數以內
- 7. 手續費：2 位數

請注意，以上所有欄位中除「取款金額」外，若位數不足時，所保留的空間左側皆會補 '0'。以下是一組符合格式要求的提款交易明細資料範例：

```
0971202,070526,003456789,007,123456789012,2000,15↵
```

此題假設 receipt.cpp(經編譯後) 執行時，必須使用 I/O Redirect 的方式將檔案餵給執行檔。

同學們可以將代表提款交易明細資料的字串內容儲存為檔案，再透過 I/O Redirect 的方式將檔案內容輸入給程式使用。為便利同學測試，以上的提款交易明細資料範例已製作為 1.in 檔案，並放置於ws上的 /home/stu/public/cpp2026s/cpp.hw2/p6 裡，有需要的同學可以自行取用。以下是本題的參考執行結果(含取得1.in測試檔案)：

```
[3:23 user@ws p6]▲cp /home/stu/public/cpp2026s/cpp.hw2/p6/1.in .↵
[3:23 user@ws p6]▲cat 1.in↵
0971202,070526,003456789,007,123456789012,2000,15↵
[3:23 user@ws hw]▲./a.out < 1.in
-----↵
▲▲▲Bank▲of▲"Fantastic"▲ATM▲Receipt↵
-----↵
Transaction▲Date:▲▲▲▲▲▲▲▲▲▲097/12/02↵
Transaction▲Time:▲▲▲▲▲▲▲▲▲▲07:05:26↵
ATM▲Number:▲▲▲▲▲▲▲▲▲▲003456789↵
Bank▲Code:▲▲▲▲▲▲▲▲▲▲007↵
Account:▲▲▲▲▲▲▲▲▲▲123456789012↵
Withdrawal▲Amount:▲▲▲▲▲▲▲▲▲▲2000↵
Handling▲Charge:▲▲▲▲▲▲▲▲▲▲15↵
-----↵
[3:23 user@ws hw]
```

由於本題的輸出結果中的每一欄的個位數應與表格右側邊界對齊，為便利同學閱讀起見，以下使用固定寬度字元再次將上述的執行結果顯示如下：



```
-----↵
Bank of "Fantastic" ATM Receipt↵
-----↵
Transaction Date:▲▲▲▲▲▲▲▲▲▲097/12/02↵
```

