國立屏東大學 資訊工程系 物件導向程式設計

# Turnin 作業 4

- Turnin Code: cpp.hw4
- Due Date: 3/23 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 要繳交的內容建立一個**外層資料夾**（例如□hw4□□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稱為□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 hw4       # 在家目錄建立了一個名為□hw4□的資料夾
[user@ws ~]$ cd hw4          # 進入□hw4□資料夾
[user@ws hw4]$ mkdir p1      # 建立一個名為□p1□資料夾
[user@ws hw4]$ cd p1         # 切換到□p1□資料夾
[user@ws p1]$ joe sort.cpp   # 使用 JOE 編輯器對檔名 sort.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.hw4▲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 hw4]$ turnin cpp.hw4 p1 # 使用 turnin 指令提交 p1 的程式碼
Turning in:
 p1/sort.cpp -- ok
All done.
[user@ws hw4]$
```

當然，你也可以等到本次作業要求的所有題目都在 hw4 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 hw4 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 hw4]$ ls                  # 檢視當前資料夾下有哪些內容
p1  p10  p2  p3  p4  p5  p6  p7  p8  p9
[user@ws hw4]$ rm -f */a.out       # 移除所有子資料夾中的 a.out 檔案（務必謹慎操作，可
```

```
使用 -i 避免誤刪)
[user@ws hw4]$ rm -f */*.o        #移除所有子資料夾中的 object 檔案（務必謹慎操作,
可使用 -i 避免誤刪)
[user@ws hw4]$ rm -f */*.*~       #移除所有子資料夾中的編輯器暫存檔案（務必謹慎操作,
可使用 -i 避免誤刪)
[user@ws hw4]$ turnin cpp.hw4 .   #使用 turnin 指令繳交該資料夾下的所有內容
Turning in:
 ./p6/decompose.cpp -- ok
 ./p6/main.cpp -- ok
 ./p6/Makefile -- ok
 ./p6/decompose.h -- ok
 ./p9/grade.cpp -- ok
 ./p9/grade.h -- ok
 ./p9/main.cpp -- ok
 ./p9/Makefile -- ok
 ./p3/score.cpp -- ok
 ./p1/sort.cpp -- ok
 ./p10/branch.cpp -- ok
 ./p10/branch.h -- ok
 ./p2/barchart.cpp -- ok
 ./p7/main.cpp -- ok
 ./p7/gcd.h -- ok
 ./p7/Makefile -- ok
 ./p7/gcd.cpp -- ok
 ./p8/main.cpp -- ok
 ./p8/Makefile -- ok
 ./p8/area.cpp -- ok
 ./p8/area.h -- ok
 ./p8/calculator.cpp -- ok
 ./p8/calculator.h -- ok
 ./p5/is_card_vaild.cpp -- ok
 ./p4/1A2B.cpp -- ok
All done.
[user@ws hw4]$
```

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.hw4↵` 指令，例如：

```
.:
total 40
drwxrwx---. 2 turninman turnin 4096 Mar 19 09:32 p1
drwxrwx---. 2 turninman turnin 4096 Mar 19 09:32 p10
drwxrwx---. 2 turninman turnin 4096 Mar 19 09:32 p2
drwxrwx---. 2 turninman turnin 4096 Mar 19 09:32 p3
drwxrwx---. 2 turninman turnin 4096 Mar 19 09:32 p4
drwxrwx---. 2 turninman turnin 4096 Mar 19 09:32 p5
drwxrwx---. 2 turninman turnin 4096 Mar 19 09:32 p6
drwxrwx---. 2 turninman turnin 4096 Mar 19 09:32 p7
drwxrwx---. 2 turninman turnin 4096 Mar 19 09:32 p8
drwxrwx---. 2 turninman turnin 4096 Mar 19 09:32 p9
```

```
./p1:
total 0
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 sort.cpp

./p10:
total 0
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 branch.cpp
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 branch.h

./p2:
total 0
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 barchart.cpp

./p3:
total 0
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 score.cpp

./p4:
total 0
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 1A2B.cpp

./p5:
total 0
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 is_card_vaild.cpp

./p6:
total 0
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 Makefile
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 decompose.cpp
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 decompose.h
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 main.cpp

./p7:
total 0
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 Makefile
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 gcd.cpp
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 gcd.h
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 main.cpp

./p8:
total 0
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 Makefile
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 area.cpp
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 area.h
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 main.cpp

./p9:
total 0
```

```
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 Makefile
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 grade.cpp
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 grade.h
-rw-rw----. 1 turninman turnin 0 Mar 19 09:32 main.cpp
```



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！

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 排序

請設計一個 C++ 語言的程式 sort.cpp，讓使用者輸入 10 個整數後，進行遞減（由大到小）排序加以輸出。此題的執行結果可參考如下：

```
[3:23␣user@ws␣p1]␣./a.out↵
Please␣input␣10␣numbers:␣6␣9␣5␣1␣10␣2␣7␣8␣4␣3↵
The␣numbers␣are␣sorted␣as␣follows:␣↵
10␣9␣8␣7␣6␣5␣4␣3␣2␣1↵
[3:23␣user@ws␣p1]␣./a.out↵
Please␣input␣10␣numbers:␣17␣3␣98␣99␣2␣0␣-1␣4␣221␣-10↵
The␣numbers␣are␣sorted␣as␣follows:␣↵
221␣99␣98␣17␣4␣3␣2␣0␣-1␣-10↵
[3:23␣user@ws␣p1]␣./a.out↵
Please␣input␣10␣numbers:␣9␣2␣1␣88␣2␣34␣2␣7␣11␣0↵
The␣numbers␣are␣sorted␣as␣follows:␣↵
88␣34␣11␣9␣7␣2␣2␣2␣1␣0↵
[3:23␣user@ws␣p1]␣
```

# p2 輸出長條圖

請設計一個 C++ 語言的程式 barchart.cpp，此程式讀入使用者輸入的 7 個介於 0（含）至 10（含）之間整數，並依據其值輸出直式長條圖。本題的執行結果可參考如下：

```
[3:23▲user@ws▲p2]▲./a.out↵
Please▲input▲7▲numbers:▲10▲6▲2▲5▲6▲8▲4↵
#▲.▲.▲.▲.▲.▲.↵
#▲.▲.▲.▲.▲.▲.↵
#▲.▲.▲.▲.▲#▲.↵
#▲.▲.▲.▲.▲#▲.↵
#▲#▲.▲.▲#▲#▲.↵
#▲#▲.▲#▲#▲#▲.↵
#▲#▲.▲#▲#▲#▲#↵
#▲#▲.▲#▲#▲#▲#↵
#▲#▲#▲#▲#▲#▲#↵
#▲#▲#▲#▲#▲#▲#↵
1▲2▲3▲4▲5▲6▲7↵
[3:23▲user@ws▲p2]▲./a.out↵
Please▲input▲7▲numbers:▲1▲2▲3▲4▲5▲6▲7↵
.▲.▲.▲.▲.▲.↵
.▲.▲.▲.▲.▲.↵
.▲.▲.▲.▲.▲.↵
.▲.▲.▲.▲.▲#↵
.▲.▲.▲.▲.▲#▲#↵
.▲.▲.▲.▲#▲#▲#↵
.▲.▲.▲#▲#▲#▲#↵
.▲.▲#▲#▲#▲#▲#↵
.▲#▲#▲#▲#▲#▲#↵
#▲#▲#▲#▲#▲#▲#↵
1▲2▲3▲4▲5▲6▲7↵
[3:23▲user@ws▲p2]▲
```



請注意，長條圖所輸出的**每一個欄位應對齊**。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 p2] ./a.out
Please input 7 numbers: 10 6 2 5 6 8 4
# . . . . . .
# . . . . . .
# . . . . # .
# . . . . # .
# # . . # # .
# # . # # # .
# # . # # # #
# # . # # # #
# # # # # # #
# # # # # # #
1 2 3 4 5 6 7
[3:23 user@ws p2]
```

# p3 成績報表

假設有一個班級只有 10 個學生，請幫助教設計一個 C++ 語言的程式 score.cpp，瞭解該班的成績狀況。

讓使用者輸入 **10 筆學生的成績**（皆為整數）後：

- 當所輸入的成績低於 0 分時，該成績以 0 分計；所輸入的成績超過 100 時，以 100 分計。
- 計算總平均(無條件捨棄至小數點後一位)。
- 找出最高分與最低分。
- 計算分數最高的 6 名學生平均（四捨五入到整數位）。

此題的執行結果可參考如下：

```
[3:23▲user@ws▲p3]▲./a.out↵
Input▲10▲scores:▲189▲175▲66▲48▲90▲100▲-73▲80▲-2▲0↵
Max▲Score:▲100↵
Min▲Score:▲0↵
Average:▲58.4↵
Average▲of▲the▲top▲6▲scores:▲89↵
[3:23▲user@ws▲p3]▲./a.out↵
Input▲10▲scores:▲10▲20▲30▲40▲90▲90▲90▲90▲100▲100↵
Max▲Score:▲100↵
Min▲Score:▲10↵
Average:▲66.0↵
Average▲of▲the▲top▲6▲scores:▲93↵
[3:23▲user@ws▲p3]▲
```

# p4 1A2B

請設計一個 C++ 語言的程式 1A2B.cpp，讓使用者連續輸入 4 個數字，並在所輸入的數字沒有重複的前提下，與數字 1234 進行比較。若有數值與位置皆相同的數字則記為 A，數值正確但位置不正確的數字則記為 B，請將比較後的結果輸出。此題的執行結果可參考如下:

```
[3:23▲user@ws▲p4]▲./a.out↵
Please▲input▲a▲4-digits▲number:▲1234↵
Checking▲a▲Result:▲4A0B↵
[3:23▲user@ws▲p4]▲./a.out↵
Please▲input▲a▲4-digits▲number:▲4321↵
Checking▲a▲Result:▲0A4B↵
[3:23▲user@ws▲p4]▲./a.out↵
Please▲input▲a▲4-digits▲number:▲3800↵
Duplicated▲numbers!↵
[3:23▲user@ws▲p4]▲./a.out↵
Please▲input▲a▲4-digits▲number:▲1945↵
```

```
Checking▲a▲Result:▲1A1B↵
[3:23▲user@ws▲p4]▲
```

# *p5 有效信用卡

信用卡、簽帳金融卡等「塑膠貨幣」在我們的生活中扮演著重要的角色，當我們要進行線上刷卡的時候，通常需要輸入 16 位數的卡號（**本題以 16 位數卡號為例**，台灣以外之其他地區銀行所發行卡號可能為 15～19 位數）：

- 前 6 位為的發行者代碼（Issuer Identification Number，IIN）
- 中間 9 位為的帳戶號碼
- 最後 1 位為的校驗碼

線上的刷卡系統在使用者按下送出後，都會先驗證其卡號是否有效，我們以卡號「4013-7356-3380-0642」為例，進行如下的驗證步驟（此法稱為 Luhn 演算法）：

我們先把每一個位數從左到右依序稱為「第 1 位（n1）、第 2 位（n2）... 、第 16 位（n16）」如下表：

| **4** | **0** | **1** | **3** | **7** | **3** | **5** | **6** | **3** | **3** | **8** | **0** | **0** | **6** | **4** | **2** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $n_1$ | $n_2$ | $n_3$ | $n_4$ | $n_5$ | $n_6$ | $n_7$ | $n_8$ | $n_9$ | $n_{10}$ | $n_{11}$ | $n_{12}$ | $n_{13}$ | $n_{14}$ | $n_{15}$ | $n_{16}$ |

接下來除最後一位的校驗碼外，將所有的奇數位數（n1、n3、n5...,n15）乘上 2，其計算過程如下表：

| **原值** | **4** | **0** | **1** | **3** | **7** | **3** | **5** | **6** | **3** | **3** | **8** | **0** | **0** | **6** | **4** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 奇數位 * 2 | 8 | | 2 | | 14 | | 10 | | 6 | | 16 | | 0 | | 8 |
| 結果 | 8 | 0 | 2 | 3 | 14 | 3 | 10 | 6 | 6 | 3 | 16 | 0 | 0 | 6 | 8 |
| 變數 | $n_1$ | $n_2$ | $n_3$ | $n_4$ | $n_5$ | $n_6$ | $n_7$ | $n_8$ | $n_9$ | $n_{10}$ | $n_{11}$ | $n_{12}$ | $n_{13}$ | $n_{14}$ | $n_{15}$ |

在上述的計算結果裡，若有任何位數的數值大於等於 10 時，則將該值減去 9，計算過程如下表:

| **原值** | **8** | **0** | **2** | **3** | **14** | **3** | **10** | **6** | **6** | **3** | **16** | **0** | **0** | **6** | **8** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 雙位數值 | | | | | 14 | | 10 | | | | 16 | | | | |
| 雙位數值減去 9 | | | | | 5 | | 1 | | | | 7 | | | | |
| 結果 | 8 | 0 | 2 | 3 | 5 | 3 | 1 | 6 | 6 | 3 | 7 | 0 | 0 | 6 | 8 |
| 變數 | $n_1$ | $n_2$ | $n_3$ | $n_4$ | $n_5$ | $n_6$ | $n_7$ | $n_8$ | $n_9$ | $n_{10}$ | $n_{11}$ | $n_{12}$ | $n_{13}$ | $n_{14}$ | $n_{15}$ |

再接下來，將包含**校驗碼（此例為 2）**在內的所有位數的值加總：

- 所有位數加總：$8+0+2+3+5+3+1+6+6+3+7+0+0+6+8 = 58$
- 再加上校驗碼的值：$58 + 2 = 60$

最後，如果得到的值能夠被 10 整除，則該卡號為有效號碼：

- $60 (mod 10) = 0$（**60 能被 10 整除，該卡號為有效卡號**）

反之該卡號無效。

p.s. 依照 維基百科-支付卡號 的驗證方法，奇數位數（n1、n3、n5...,n15）乘上 2 前**<u>應將所有數字反方向排列</u>**。

| **反轉前** | **4** | **0** | **1** | **3** | **7** | **3** | **5** | **6** | **3** | **3** | **8** | **0** | **0** | **6** | **4** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 反轉後 | 4 | 6 | 0 | 0 | 8 | 3 | 3 | 6 | 5 | 3 | 7 | 3 | 1 | 0 | 4 |
| 變數 | n1 | n2 | n3 | n4 | n5 | n6 | n7 | n8 | n9 | n10 | n11 | n12 | n13 | n14 | n15 |

由於本題限定處理 16 位卡號，扣除最後一位校驗碼後剩餘 15 位。經評估，在此固定長度下，直接由左至右對奇數索引位進行兩倍運算，其結果與維基百科所述「反轉後運算」之數學結果等價，故為簡化程式邏輯，省略反轉步驟。

若該卡號經判定有效後，請接著判斷其發行者代碼：

- Visa：第一位以 4 開頭
- Mastercard：前 2 位以 50～55 開頭
- JCB：前 4 位以 3528～3589 開頭
- none of Visa, Mastercard and JCB：上述範圍以外

請設計一個 C++ 語言的程式 is_card_vaild.cpp（該處檔名拼寫錯誤，應為 is_card_valid.cpp，批改時此兩檔名都被接受），讓使用者輸入 16 位數的卡號（使用者一定會輸入 16 位數），並根據上述方法進行針對卡號的驗證。若卡號經驗證判定為無效卡號，請輸出：Invalid Card!↵；最後，若卡號被判定為有效卡號請依照下方執行結果的格式輸出卡號，以及其發行者名稱。

此題的執行結果可參考如下:

```
[3:23 user@ws p5]$ ./a.out↵
Please input your card number: 4013735633800642↵
The card number [4013-7356-3380-0642] is valid, and its card issuer is Visa.↵
[3:23 user@ws p5]$ ./a.out↵
Please input your card number: 5556394982015269↵
The card number [5556-3949-8201-5269] is valid, and its card issuer is Mastercard.↵
[3:23 user@ws p5]$ ./a.out↵
Please input your card number: 3543488430004932↵
The card number [3543-4884-3000-4932] is valid, and its card issuer is JCB.↵
[3:23 user@ws p5]$ ./a.out↵
Please input your card number: 6018539047509504↵
The card number [6018-5390-4750-9504] is valid, and its card issuer is none of Visa, Mastercard and JCB.↵
[3:23 user@ws p5]$ ./a.out↵
Please input your card number: 4485798329585303↵
Invalid Card!↵
[3:23 user@ws p5]$
```

---

---

請注意，從現在開始，之後的每一題同一資料夾可能需要繳交超過一個檔案。

# p6 分解浮點數

請同學們至系計中依據檔名旁的路徑取得程式碼。請參考下列的 main.cpp 程式:

```
#include "decompose.h"
#include <iostream>
using namespace std;

int main()
{
    double fp_num = 0;
    cout << "Please input a floating-point number: ";
    cin >> fp_num;

    cout << "integer part: " << int_part(fp_num) << endl;
    cout << "floating part: " << float_part(fp_num) << endl;
    return 0;
}
```

請完成名為 **decompose.cpp** 與 **decompose.h** 的 C++ 語言程式，其中分別包含 int_part() 以及 float_part() 函式的 **Implementation** 與其 **Prototype 宣告**。

本題的相關程式將使用以下的 Makefile 進行編譯：

```
all: main.cpp decompose.o
    c++ main.cpp decompose.o
decompose.o: decompose.cpp decompose.h
    c++ -c decompose.cpp
clean:
    rm -f a.out *.c~ *.o*
```

此題的執行結果可參考如下：

```
[3:23▲user@ws▲p6]▲./a.out↵
Please▲input▲a▲floating-point▲number:▲3.1415↵
integer▲part:▲3↵
floating▲part:▲0.1415↵
[3:23▲user@ws▲p6]▲./a.out↵
Please▲input▲a▲floating-point▲number:▲123↵
```

```
integer▲part:▲123↵
floating▲part:▲0↵
[3:23▲user@ws▲p6]▲
```



- 本題相關的程式碼路徑已註明於檔名右側，同學們可以透過路徑複製到自己的家目錄。
- 本題若有使用浮點數的需求，請使用 double 型態。
- 本題應繳交檔案如下（至於 main.c 與 Makefile 則不需繳交）：
  - decompose.h
  - decompose.cpp

# p7 最大公因數【遞迴】

請同學們至系計中依據檔名旁的路徑取得程式碼。請參考下列的 main.cpp 程式:

```
#include "gcd.h"
#include <iostream>
using namespace std;

int main()
{
    int a = 0, b = 0;
    cout << "Please input 2 numbers(a b): ";
    cin >> a >> b;

    cout << "gcd(" << a << ", " << b << ")" << " = " << gcd(a, b) << endl;
    return 0;
}
```

請完成名為 **gcd.cpp** 與 **gcd.h** 的 C++ 語言程式，其中分別包含 gcd() 函式的 **Implementation** 與其 **Prototype 宣告**本題限定使用「遞迴」方式撰寫，否則不予計分。

本題的相關程式將使用以下的 Makefile 進行編譯：

```
all: main.cpp gcd.o
    c++ main.cpp gcd.o
gcd.o: gcd.cpp gcd.h
    c++ -c gcd.cpp
clean:
    rm -f a.out *.c~ *.o*
```

此題的執行結果可參考如下：

```
[3:23▲user@ws▲p7]▲./a.out↵
Please▲input▲2▲numbers(a▲b):▲12▲15↵
gcd(12,▲15)▲=▲3↵
[3:23▲user@ws▲p7]▲./a.out↵
Please▲input▲2▲numbers(a▲b):▲70▲80↵
gcd(70,▲80)▲=▲10↵
[3:23▲user@ws▲p7]▲./a.out↵
Please▲input▲2▲numbers(a▲b):▲72▲60↵
gcd(72,▲60)▲=▲12↵
[3:23▲user@ws▲p7]▲
```



- 本題限定使用「遞迴」方式撰寫，否則不予計分。
- 本題相關的程式碼路徑已註明於檔名右側，同學們可以透過路徑複製到自己的家目錄。
- 本題應繳交檔案如下（至於 main.c 與 Makefile 則不需繳交）：
  - gcd.h
  - gcd.cpp

# p8 計算圖形面積

請同學們至系計中依據檔名旁的路徑取得程式碼。請參考下列的 main.cpp 程式:

```
#include "area.h"
#include <iostream>
using namespace std;
int main()
{

    int select = 0;

    cout << "Please choose a shape to calculate its area (1: Square, 2:
Triangle): ";
    cin >> select;

    switch (select)
    {
        case 1:
        {
            double side_length = 0;
```

```
            cout << "[Square] Please input the side length: ";
            cin >> side_length;
            cout << "area = " << area(side_length) << endl;
            return 0;
        }
        case 2:
        {
            double base = 0, height = 0;
            cout << "[Triangle] Please input the base and height(a b): ";
            cin >> base >> height;
            cout << "area = " << area(base, height) << endl;
            return 0;
        }
        default:
            cout << "Wrong Input!" << endl;
            return -1;
    }
    return 0;
}
```

請完成名為 **area.cpp** 與 **area.h** 的 C++ 語言程式，其中分別包含 area() 函式的 **Implementation** 與其 **Prototype 宣告**

本題的相關程式將使用以下的 Makefile 進行編譯：

```
all: main.cpp area.o
    c++ main.cpp area.o
area.o: area.cpp area.h
    c++ -c area.cpp
clean:
    rm -f a.out *.c~ *.o*
```

此題的執行結果可參考如下：

```
[3:23▲user@ws▲p8]▲./a.out↵
Please▲choose▲a▲shape▲to▲calculate▲its▲area▲(1:▲Square,▲2:▲Triangle):▲1↵
[Square]▲Please▲input▲the▲side▲length:▲10↵
area▲=▲100↵
[3:23▲user@ws▲p8]▲./a.out↵
Please▲choose▲a▲shape▲to▲calculate▲its▲area▲(1:▲Square,▲2:▲Triangle):▲1↵
[Square]▲Please▲input▲the▲side▲length:▲3.14↵
area▲=▲9.8596↵
[3:23▲user@ws▲p8]▲./a.out↵
Please▲choose▲a▲shape▲to▲calculate▲its▲area▲(1:▲Square,▲2:▲Triangle):▲2↵
[Triangle]▲Please▲input▲the▲base▲and▲height(a▲b):▲3▲4↵
area▲=▲6↵
[3:23▲user@ws▲p8]▲./a.out↵
```

```
Please▲choose▲a▲shape▲to▲calculate▲its▲area▲(1:▲Square,▲2:▲Triangle):▲2↵
[Triangle]▲Please▲input▲the▲base▲and▲height(a▲b):▲5▲3↵
area▲=▲7.5↵
[3:23▲user@ws▲p8]▲
```



- 本題相關的程式碼路徑已註明於檔名右側，同學們可以透過路徑複製到自己的家目錄。
- 本題若有使用浮點數的需求，請使用 double 型態。
- 本題應繳交檔案如下（至於 main.c 與 Makefile 則不需繳交）：
  - area.h
  - area.cpp

# p9 分數處理

請同學們至系計中依據檔名旁的路徑取得程式碼。請參考下列的 main.cpp 程式:

```
#include "grade.h"
#include <iostream>
using namespace std;

int main()
{
    int arr_i[10], data_type = 0;
    double arr_d[10];

    cout << "Should the grades be entered as integers(1) or floats(2)? ";
    cin >> data_type;

    if ((data_type < 1) || (data_type > 2))
    {
        cout << "Wrong Input!" << endl;
        return -1;
    }

    for (int i = 0; i < 10; i++)
    {
        cout << "Student " << i + 1 << ": ";
        if (data_type == 1)
            cin >> arr_i[i];
        else
            cin >> arr_d[i];
    }
```

```
    cout << "The highest score: " << ((data_type == 1) ? max(arr_i) :
max(arr_d));
    cout << endl;
    cout << "The lowest score: " << ((data_type == 1) ? min(arr_i) :
min(arr_d));
    cout << endl;
    cout << "average score: " << ((data_type == 1) ? average(arr_i) :
average(arr_d));
    cout << endl;

    return 0;
}
```

使用者會輸入 10 個成績，但是必須先選擇使用哪一個型態的陣列去儲存這些成績（使用者輸入的分數會符合所選擇的型態），最後找出這些分數中最高分與最低分，並計算這 10 個成績的平均。請完成名為 **grade.h** 的 C++ 語言程式，其中分別包含 max()、min() 以及 average() **函式模板**的 **Implementation** 與其 **Prototype 宣告**。本題限定使用「函式模板」方式撰寫，否則不予計分。

本題的相關程式將使用以下的 Makefile 進行編譯（Makefile 檔案已更新，請各位同學重新複製）：

```
all: main.cpp
    c++ main.cpp
clean:
    rm -f a.out *.c~ *.o*
```

此題的執行結果可參考如下：

```
[3:23 user@ws p9] ./a.out
Should the grades be entered as integers(1) or floats(2)? 1
Student 1: 59
Student 2: 35
Student 3: 85
Student 4: 67
Student 5: 95
Student 6: 46
Student 7: 84
Student 8: 73
Student 9: 57
Student 10: 84
The highest score: 95
The lowest score: 35
average score: 68.5
[3:23 user@ws p9] ./a.out
Should the grades be entered as integers(1) or floats(2)? 2
Student 1: 64.5
Student 2: 95.4
Student 3: 68
```

```
Student▲4:▲87.5↵
Student▲5:▲98.6↵
Student▲6:▲94.6↵
Student▲7:▲34.5↵
Student▲8:▲60.7↵
Student▲9:▲93.8↵
Student▲10:▲54.1↵
The▲highest▲score:▲98.6↵
The▲lowest▲score:▲34.5↵
average▲score:▲75.17↵
[3:23▲user@ws▲p9]▲
```



- 本題相關的程式碼路徑已註明於檔名右側，同學們可以透過路徑複製到自己的家目錄。
- 本題限定使用「函式模板」方式撰寫，否則不予計分。
- 本題應繳交檔案如下（至於 main.c 與 Makefile 則不需繳交）：
  - grade.h

# p10 趨橙釋的跨國分店

請同學們至系計中依據檔名旁的路徑取得程式碼。請參考下列的 main.cpp 程式:

```
#include "branch.h"
#include <iostream>
using namespace std;

int main()
{
    int select = 0;
    double subtotal = 0;
    cout << "Welcome to Matsons!" << endl;
    cout << "Choose a region(1: Taiwan, 2: Hong Kong): ";
    cin >> select;

    if (select < 1 || select > 2)
    {
        cout << "Unknown region!" << endl;
        return -1;
    }

    cout << "How much should the customer pay(excluding tax)? ";
    cin >> subtotal;
```

```
    cout << "Customer should pay $";
    switch (select)
    {
        case 1:

            cout << tw::with_tax(subtotal);
            break;
        case 2:
            cout << hk::with_tax(subtotal);
            break;
        default:
            cout << "Wrong Input!" << endl;
            return -1;
    }
    cout << endl;

    return 0;
}
```

趨橙釋的總部在香港，目前台灣也有分店。但是兩地的稅制計算方式不同：

- 台灣加上 5% 的營業稅（並四捨五入到整數位）
- 香港沒有營業稅（不需額外處理）

參考主程式，請完成名為 **branch.cpp** 與 **branch.h** 的 C++ 語言程式，其中分別包含同名的 with_tax() 在兩個命名空間tw與「hk的 **Implementation** 與其 **Prototype 宣告**

本題的相關程式將使用以下的 Makefile 進行編譯：

```
all: main.cpp branch.o
    c++ main.cpp branch.o
branch.o: branch.cpp branch.h
    c++ -c branch.cpp
clean:
    rm -f a.out *.c~ *.o*
```

此題的執行結果可參考如下：

```
[3:23▲user@ws▲p10]▲./a.out↵
Welcome▲to▲Matsons!↵
Choose▲a▲region(1:▲Taiwan,▲2:▲Hong▲Kong):▲1↵
How▲much▲should▲the▲customer▲pay(excluding▲tax)?▲100↵
Customer▲should▲pay▲$105↵
[3:23▲user@ws▲p10]▲./a.out↵
Welcome▲to▲Matsons!↵
Choose▲a▲region(1:▲Taiwan,▲2:▲Hong▲Kong):▲2↵
How▲much▲should▲the▲customer▲pay(excluding▲tax)?▲100↵
```

```
Customer▲should▲pay▲$100↵
[3:23▲user@ws▲p10]▲./a.out↵
Welcome▲to▲Matsons!↵
Choose▲a▲region(1:▲Taiwan,▲2:▲Hong▲Kong):▲1↵
How▲much▲should▲the▲customer▲pay(excluding▲tax)?▲24.5↵
Customer▲should▲pay▲$26↵
[3:23▲user@ws▲p10]▲./a.out↵
Welcome▲to▲Matsons!↵
Choose▲a▲region(1:▲Taiwan,▲2:▲Hong▲Kong):▲2↵
How▲much▲should▲the▲customer▲pay(excluding▲tax)?▲24.5↵
Customer▲should▲pay▲$24.5↵
[3:23▲user@ws▲p10]▲
```



- 本題相關的程式碼路徑已註明於檔名右側，同學們可以透過路徑複製到自己的家目錄。
- 本題應繳交檔案如下（至於 main.c 與 Makefile 則不需繳交）：
  - branch.h
  - branch.cpp

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: 334353

Permanent link:
**https://junwu.nptu.edu.tw/dokuwiki/doku.php?id=cpp:2026spring:hw4**

Last update: **2026/03/23 18:50**