

Turnin作業6\c.hw6\

- Turnin Code: **c.hw6**
- Due Date: 2024/12/17(週二) 23:59 **Hard Deadline**
- 涵蓋範圍: chapter 8

作業說明

每次Turnin作業會包含多個程式題目:p1\p2\...\pN\每題將提供其題目、檔案命名規定、功能要求以及參考的執行結果。建議同學可以為每次的Turnin作業建立一個資料夾以便於管理，例如本次的作業可以使用mkdir hw6 指令來建立一個名為hw6的資料夾，並在該資料夾裡編寫此次作業的程式檔案。另外要注意的是，每一題都必須依據題目要求為程式檔案命名，例如此次turnin作業的p1\其檔案名稱要求命名為ReverseArray.c\請務必注意大小寫一致，任何錯誤的檔案命名都將不予計分。

注意：從此次作業開始，如題目有加註星號 (*)，該題分數將加倍計算。同理，如題目有加註兩個星號 (**)，該題分數將以四倍計算，餘依此類推。

注意：受限於自動批改程式的功能，同學作答時請不要在程式中使用\b作為輸出。

繳交方式說明（以下說明皆假設在hw6的目錄下操作）

請注意每次turnin作業都必須在截止時間前完成繳交，逾時不候且不得補交。以此次作業為例，其turnin code為c.hw6\當你完成p1的程式碼撰寫，並經編譯執行確認無誤後，可以使用下列指令完成繳交（請自行注意是否在正確的資料夾裡操作）：

```
[7:17 user@ws in hw6] turnin\c.hw6\ReverseArray.c\
```

後續還可以使用同樣的方式，將p2\p3\...\p10的程式檔案加以繳交。當然，你也可以等到所有小題都作答完成後，再整批地將資料夾下的所有作業上傳，請參考以下指令將目前資料夾\hw6\內所有副檔名為.c的檔案整批繳交：

```
[7:17 user@ws in hw6] turnin\c.hw6\*\.c\
```

註：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淺灰色方框顯示。

p1 反轉陣列[]ReverseArray.c[]

請設計一個C語言的程式 ReverseArray.c[]讓使用者輸入 10 個整數，並將這 10 個數字倒序輸出，其執行結果可參考以下輸出內容：

```
[7:17 user@ws in hw6] ./a.out
1▲2▲3▲4▲5▲6▲7▲8▲9▲10
original▲array:▲1▲2▲3▲4▲5▲6▲7▲8▲9▲10
reversed▲array:▲10▲9▲8▲7▲6▲5▲4▲3▲2▲1
[7:17 user@ws in hw6] ./a.out
10▲9▲8▲7▲6▲5▲4▲3▲2▲1
original▲array:▲10▲9▲8▲7▲6▲5▲4▲3▲2▲1
reversed▲array:▲1▲2▲3▲4▲5▲6▲7▲8▲9▲10
[7:17 user@ws in hw6] ./a.out
-5▲-4▲-3▲-2▲-1▲0▲1▲2▲3▲4
original▲array:▲-5▲-4▲-3▲-2▲-1▲0▲1▲2▲3▲4
reversed▲array:▲4▲3▲2▲1▲0▲-1▲-2▲-3▲-4▲-5
[7:17 user@ws in hw6] ./a.out
4▲3▲2▲1▲0▲4▲3▲2▲1▲0
original▲array:▲4▲3▲2▲1▲0▲4▲3▲2▲1▲0
reversed▲array:▲0▲1▲2▲3▲4▲0▲1▲2▲3▲4
[7:17 user@ws in hw6] ./a.out
3▲0▲2▲0▲1▲1▲0▲2▲0▲3
original▲array:▲3▲0▲2▲0▲1▲1▲0▲2▲0▲3
reversed▲array:▲3▲0▲2▲0▲1▲1▲0▲2▲0▲3
[7:17 user@ws in hw6]
```

p2 判斷稀疏矩陣[]IsSparseMatrix.c[]此處首字母為大寫的I(ㄞ))

稀疏矩陣是一個矩陣中零元素的數量多於非零元素的矩陣。

請設計一個C語言的程式 IsSparseMatrix.c[]使用者將輸入 3×3 的矩陣且每個元素都是整數，最後輸出該矩陣是否為稀疏矩陣，其執行結果可參考以下輸出內容：

```
[7:17 user@ws in hw6] ./a.out
Enter▲data:↵
1▲2▲3
0▲0▲0
0▲0▲1
It▲is▲a▲sparse▲matrix!
[7:17 user@ws in hw6] ./a.out
Enter▲data:↵
```

```
1▲1▲1↵
0▲0▲0↵
1▲1▲1↵
It is not a sparse matrix!↵
[7:17 user@ws in hw6] ./a.out↵
Enter data:↵
1342▲3214▲543↵
547382▲2345▲1235↵
2315▲1032▲213978↵
It is not a sparse matrix!↵
[7:17 user@ws in hw6] ./a.out↵
Enter data:↵
678954▲237684▲783954↵
0▲0▲0↵
0▲0▲527390↵
It is a sparse matrix!↵
[7:17 user@ws in hw6] ./a.out↵
Enter data:↵
0▲0▲0↵
0▲0▲0↵
0▲0▲0↵
It is a sparse matrix!↵
[7:17 user@ws in hw6]
```

p3 找出數字\FindNumber.c\

請設計一個C語言的程式 FindNumber.c\讓使用者依序輸入 10 個整數並保留於程式當中；後續再讓使用者輸入一個整數，輸出該整數為之前所輸入的第幾個整數（若該整數出現兩次以上則請輸出最後一次出現的位置；若找不到則輸出錯誤訊息）。其執行結果可參考以下輸出內容：

```
[7:17 user@ws in hw6] ./a.out↵
Enter data:▲1▲2▲3▲4▲5▲6▲7▲8▲9▲10↵
Enter a number to find:▲5↵
5 is the 5th number.↵
[7:17 user@ws in hw6] ./a.out↵
Enter data:▲0▲1▲0▲1▲1▲0▲0▲1▲1▲0↵
Enter a number to find:▲1↵
1 is the 9th number.↵
[7:17 user@ws in hw6] ./a.out↵
Enter data:▲-4▲-3▲-2▲-1▲0▲1▲1▲2▲2▲3↵
Enter a number to find:▲-3↵
-3 is the 2nd number.↵
[7:17 user@ws in hw6] ./a.out↵
Enter data:▲-4▲-3▲-2▲-1▲0▲1▲1▲2▲2▲3↵
Enter a number to find:▲-2↵
```

```
-2 is the 3rd number.  

[7:17 user@ws in hw6] ./a.out  

Enter data: 123 414 23984 2352 12515 124 29957 -1235215 -12 51  

Enter a number to find: 123  

123 is the 1st number.  

[7:17 user@ws in hw6] ./a.out  

Enter data: 123 414 23984 2352 12515 124 29957 -1235215 -12 51  

Enter a number to find: -1  

-1 is not found  

[7:17 user@ws in hw6]
```

p4 找零 Cashier.c

有一臺收銀機，它的功能之一是會將要找零的硬幣數量還有紙鈔數量告訴收銀員。假設該國發行的貨幣有以下面額：

- 1 元
- 5 元
- 10 元
- 50 元
- 100 元
- 500 元
- 1000 元

請設計一個C語言的程式 Cashier.c 讓使用者輸入應付與已付金額（皆為正整數）後，輸出所需找零的最少紙鈔與硬幣數量（假設硬幣數量及紙鈔數量充足），其執行結果可參考以下輸出內容：

```
[7:17 user@ws in hw6] ./a.out  

Enter the total amount: 100  

Enter the amount paid: 500  

1>>0  

5>>0  

10>>0  

50>>0  

100>>4  

500>>0  

1000>>0  

[7:17 user@ws in hw6] ./a.out  

Enter the total amount: 263  

Enter the amount paid: 1000  

1>>2  

5>>1  

10>>3  

50>>0  

100>>2  

500>>1  

1000>>0
```

```
[7:17 user@ws in hw6] ./a.out
Enter the total amount:▲1000
Enter the amount paid:▲1000
1▲▲▲>▲0
5▲▲▲>▲0
10▲▲▲>▲0
50▲▲▲>▲0
100▲▲>▲0
500▲▲>▲0
1000▲>▲0
[7:17 user@ws in hw6] ./a.out
Enter the total amount:▲-1
Enter the amount paid:▲1000
Invalid value
[7:17 user@ws in hw6] ./a.out
Enter the total amount:▲1000
Enter the amount paid:▲-1
Invalid value
[7:17 user@ws in hw6] ./a.out
Enter the total amount:▲1000
Enter the amount paid:▲500
Invalid input
[7:17 user@ws in hw6]
```

p5* 計算成績\ScoreCalculation.c\

吳教授的程式設計課程有非常多的作業要完成，假設這學期他將作業成績配分方式調整為如下：

- 每次作業分數滿分皆為100
- 若該次作業缺交，標記為 -1 並視為 0 分
- 若該次作業抄襲，標記為 -2 並視為 0 分。此外，作為懲罰，計算總平均時取最低三次作業（含缺交與抄襲所導致的 0 分）的平均當作總平均
- 作業的總成績計算方式為10次作業的平均，若該學生曾抄襲作業則使用抄襲的成績計算方式
- 若某學生所有作業皆大於等於 90 分（滿分 100 分），計算完平均之後額外加 5 分作為獎勵，但最高不能超過 100 分

請設計一個C語言的程式 ScoreCalculation.c\讓使用者輸入 10 個作業的成績輸入（輸入範圍\$ -2\le x \le 100 \$\先將取得的分數由大到小排序後依照上述規則完成計算並且輸出，如果處理完的成績有小數點，請四捨五入到整數位，其執行結果可參考以下輸出內容：

若有使用浮點數的需求，請使用 **double** 形態。

```
[7:17 user@ws in hw6] ./a.out
Enter the scores:▲-1▲46▲84▲93▲77▲10▲65▲78▲99▲100
homework score:▲65
```

```

highest:▲100↙
lowest:▲0↙
[7:17 user@ws in hw6] ./a.out↙
Enter the scores:▲0▲0▲72▲8▲-2▲66▲19▲82▲78▲98↙
homework score:▲0↙
highest:▲98↙
lowest:▲0↙
[7:17 user@ws in hw6] ./a.out↙
Enter the scores:▲90▲90▲90▲90▲90▲90▲90▲90▲90↙
homework score:▲95↙
highest:▲90↙
lowest:▲90↙
[7:17 user@ws in hw6] ./a.out↙
Enter the scores:▲1▲2▲3▲4▲5▲6▲7▲8▲9▲10↙
homework score:▲6↙
highest:▲10↙
lowest:▲1↙
[7:17 user@ws in hw6] ./a.out↙
Enter the scores:▲-3▲0▲0▲0▲0▲0▲0▲0▲0↙
Invalid score entered!↙
[7:17 user@ws in hw6] ./a.out↙
Enter the scores:▲101▲0▲0▲0▲0▲0▲0▲0▲0↙
Invalid score entered!↙
[7:17 user@ws in hw6]

```

p6 行列式 Determinant.c

假設有一個 3×3 的矩陣，如下所示

$$A = \left[\begin{array}{ccc} a & b & c \\ d & e & f \\ g & h & i \end{array} \right]$$

A 的行列式被記作“ $\det(A)$ ”且“ $\det(A)$ ”的值運算方法如下

$$\det(A) = aei + bfg + cdh - ceg - afh - bdi$$

請設計一個C語言的程式 Determinant.c 讓使用者輸入一個 3×3 的整數矩陣，並輸出其行列式的值。其執行結果可參考以下輸出內容：

```

[7:17 user@ws in hw6] ./a.out↙
Enter data:↙
1▲2▲3↙
4▲5▲7↙
2▲3▲4↙
det(A) result:▲1↙
[7:17 user@ws in hw6] ./a.out↙
Enter data:↙
3▲4▲7↙
12▲32▲5↙
11▲-11▲-11↙

```

```

det(A)►result:►-3531◄
[7:17 user@ws in hw6] ./a.out◄
Enter►data:◄
0►1►0◄
1►0►1◄
0►1►0◄
det(A)►result:►0◄
[7:17 user@ws in hw6] ./a.out◄
Enter►data:◄
44►78►59◄
-23►-100►-13◄
-32►0►0◄
det(A)►result:►-156352◄
[7:17 user@ws in hw6] ./a.out◄
Enter►data:◄
32►11►54◄
10►33►41◄
9►7►123◄
det(A)►result:►98975◄
[7:17 user@ws in hw6]

```

p7*** 發票對獎\Lottery.c\

在台灣買東西大家都會記得要索取統一發票，因為每奇數月份的 25 號會開獎一次，從統一發票號碼的末 3 碼開始到全部的 8 碼，只要與開獎號一致就可以領取從 200 元到 1000 萬元不等的獎金。

Alan 有被害妄想症，他認為使用財政部的載具肯定會有人透過他的消費記錄還有時間推測他的足跡。因此 Alan 總是索取紙本的發票，並且在每個奇數月的 25 號對發票的時候都是用條碼掃描器將每一筆的發票號碼存進一個文字檔，然後再寫一個程式來檢查目前所持有的發票是否有中獎。

在 Alan 的儲存發票的文字檔裏面，[第一行](#)與[第二行](#)分別是當期的**特別獎**與**特獎**的號碼，第三行到第五行則為三筆頭獎的中獎號碼，接著則是連續 100 筆要對獎的發票號碼（關於統一發票中獎號碼相關說明可參照：[財政部](#)）

```

[7:17 user@ws in hw6] cat►testfile_1.txt◄
28630525◄
90028580◄
27435934◄
39666605◄
02550031◄
LE73703676◄
FX18075605◄
XP88939976◄
QN36281314◄
BN60741801◄

```

... 以下省略95筆發票號碼

發票的對獎規則如下：

- 特別獎：全部 8 碼與特別獎中獎號碼相符，可得獎金 1千萬 元。
- 特獎：全部 8 碼與特獎中獎號碼相符，可得獎金 2百萬 元。
- 頭獎：全部 8 碼與任一組頭獎中獎號碼相符，可得獎金 20萬 元。
- 二獎：末 7 碼與任一組頭獎中獎號碼末 7 碼相符，可得獎金 4萬 元。
- 三獎：末 6 碼與任一組頭獎中獎號碼末 6 碼相符，可得獎金 1萬 元。
- 四獎：末 5 碼與任一組頭獎中獎號碼末 5 碼相符，可得獎金 4千 元。
- 五獎：末 4 碼與任一組頭獎中獎號碼末 4 碼相符，可得獎金 1千元。
- 六獎：末 3 碼與任一組頭獎中獎號碼末 3 碼相符，可得獎金 2百 元。

不幸的是 Alan 寫程式的能力很菜，所以請幫助 Alan 請設計一個C語言的程式 Lottery.c[]透過輸入/輸出重導向[]I/O Redirect[]的方式將儲存發票號碼的文字檔餵給執行檔，幫助他完成對獎後依中獎金額由高至低加以輸出：

[]Alan 已準備兩個測試文字檔供同學參考，其路徑如下：

- /home/stu/public/c2024f/testfile/c.hw6/Lottery/testfile_1.txt
- /home/stu/public/c2024f/testfile/c.hw6/Lottery/testfile_2.txt

同學可使用以下指令進行輸入重導向：

```
[7:17 user@ws in hw6] ./a.out<testfile_1.txt
```

或

```
[7:17 user@ws in hw6] ./a.out<testfile_2.txt
```

本題的執行結果可參考如下：

```
[7:17 user@ws in hw6] ./a.out<testfile_1.txt
```

```
UH28630525▲10000000
QL29666605▲▲▲40000
RC34435934▲▲▲10000
GK61935934▲▲▲4000
CN07195934▲▲▲1000
CQ98820031▲▲▲1000
RK11170031▲▲▲1000
FX18075605▲▲▲▲200
W042553031▲▲▲▲200
```

[7:17 user@ws in hw6] ./a.out<<testfile_2.txt (測試檔2已更新 , 再一次從系計中路徑複製測試檔2)

```
ZP72110911▲40000↵
ZP74289408▲40000↵
ME28289408▲10000↵
NL38310911▲▲4000↵
RX80346901▲▲4000↵
GS45596901▲▲1000↵
ZY77828911▲▲200↵
```

[7:17 user@ws in hw6]

【提示】：同學可以參考以下的提示，透過輸入重導向取得在測試文字檔中的中獎號碼及發票號碼。假設我們在程式中宣告一個 100×10 的char型態二維陣列來儲存100張發票(每張發票號碼皆為10碼，其中前兩碼為英文字母，後8碼為數字)：

```
char invoices[100][10];
```

我們可以使用迴圈透過scanf()來取得100張的發票號碼，並保存於陣列中：

```
for(int i=0;i<100;i++)
{
    scanf(" %c%c%c%c%c%c%c%c",
&invoices[i][0],&invoices[i][1],&invoices[i][2],
&invoices[i][3],&invoices[i][4],&invoices[i][5],
&invoices[i][6],&invoices[i][7],&invoices[i][8],
&invoices[i][9]);
}
```

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: 250141



Permanent link:

<https://junwu.nptu.edu.tw/dokuwiki/doku.php?id=c:2024fall-hw6>

Last update: 2024/12/05 06:42