

## CPP Turnin作業3

國立屏東大學 資訊工程系 物件導向程式設計

# Turnin作業3(選擇、迴圈、陣列、函式)

- Turnin Code: [cpp.hw3](#)
- Due Date: 4/3 00:00 **Hard Deadline**

## 繳交方式說明

本次Turnin作業包含多個程式題，請為每一題建立一個資料夾，並將該題所要上傳的檔案放置其中後，再使用turnin指令上傳作業。請同學先為本次作業建立一個資料夾hw3，然後在hw3裡分別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 hw3] turnin▲cpp.hw3▲p1▲
```

當然，你也可以等到所有題目都完成後，回到hw3資料夾，使用以下指令將所有題目都加以上傳：

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

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灰階方式顯示。

## p1 完美數

若一個整數所有真因數(除了本身以外的因數)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(Perfect Number)後加以輸出。此題的執行結果可參考如下：

```
<WRAP caution> [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:▲"-2" \ Error \ [3:23 user@ws hw] \ </WRAP> 請注意本題只需繳交perfect.cpp，其它檔案則不需繳交。===== p2 阿財的故事 ===== 請設計一個C++語言的程式chaistory.cpp，讓使用者輸入「何時？」、「在哪裡？」、「做什麼？」等三個選擇，然後由程式輸出產生一個簡短小故事。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。最後再輸入代表「做什麼」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: 本題的執行結果可參考如下

```
<WRAP
caution> [9:19 user@ws hw] "./a.out<"\\ "MGS<"\\ 阿財在早上到花園睡覺<"\\ [9:19 user@ws hw]
"./a.out<"\\ "ESR<"\\ 阿財在晚上到學校看書<"\\ [9:19 user@ws hw] "./a.out<"\\ "aRW<"\\ 阿財在下午
到餐廳工作<"\\ [9:19 user@ws hw] "./a.out<"\\ "aSw<"\\ 阿財在下午到學校工作<"\\ [9:19 user@ws hw]
"./a.out<"\\ "XSZ<"\\ 阿財在錯誤的時間到學校做錯誤的事<"\\ [9:19 user@ws hw] "./a.out<"\\
"mxy<"\\ 阿財在早上到錯誤的地方做錯誤的事<"\\ [9:19 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<"\\ [9:19 user@ws hw] \\ </WRAP> ===== p3 繪製長條圖 ===== 請設
計一個C++程式barchart.cpp此程式讀入使用者輸入的7個整數, 並依據其值輸出長條圖。注意: 使用者
所輸入的整數都是介於0至10之間的整數(包含0與10)。本題的執行結果可參考如下
```

<WRAP caution> [9:19 user@ws hw] "./a.out<"\\ "3▲2▲0▲4▲1▲6▲7<"\\ </WRAP> {{:cpp:2024spring:p3-output.png?120}}

[9:19 user@ws hw]

請注意本題只須上傳barchart.cpp檔案 ===== p4 10點半遊戲點數計算 ===== 請參考下面的main.cpp 以及value.h程式:

```
#include <iostream>
#include <iomanip>
#include "10andhalf.h"
using namespace std;

int main()
{
    getCards();
    showCards();
    cout.precision(1);
    cout << fixed << calculatePoint() << " Points" << endl;
}
```

```
extern int card[5];

void getCards();
void showCards();
void showACard(int);
double calculatePoint();
```

你必須完成名為10andhalf.cpp的C語言程式, 其中包含相關的函式實作。此程式完成後, 可讓使用者輸入至多5張撲克牌(以-1代表沒有), 請依據十點半遊戲規則計算並輸出其點數。

注意: 此題如同過去我們所使用過的方法(和C課本上的撲克牌遊戲程式設計一樣), 使用整數0 ~ 51做為一副52張的撲克牌, 其中0 ~ 12為 、13 ~ 25為 、26 ~ 38為 、39 ~ 51為 ; 代表每張撲克牌的整數除以13的餘數再加1, 即為其牌面點數。

十點半的點數計算規則很簡單, 每張牌以其牌面點數計算, 但J/Q/K算半點。例如牌型A♦3♦10♦的點數為14.5點♦3♦Q♦K為4點。

本題的相關程式將使用以下的Makefile進行編譯：

```
all: main.cpp 10andhalf.o
      c++ main.cpp 10andhalf.o

10andhalf.o: 10andhalf.cpp 10andhalf.h
      c++ -c 10andhalf.cpp

clean:
      rm -f *.o *~ *.*~ a.out
```

此題的執行結果如下：

```
[3:23 user@ws hw] ./a.out
13▲17▲27▲40▲25
[♥A][♥5][♦2][♣2][♥K]10.5▲Points
[3:23 user@ws hw] ./a.out
0▲20▲-1▲-1▲-1
[♠A][♥8]9.0▲Points
[3:23 user@ws hw] ./a.out
-1▲12▲28▲-1▲51
[♠K][♦3][♣K]4.0▲Points
[3:23 user@ws hw]
```

注意1:本題只需繳交10andhalf.cpp，其它檔案則不需繳交。

注意2:本題所有測試檔所輸入的撲克牌皆為正確的數值(0-51，或使用-1表示沒有)。

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: 243972



Permanent link:

<https://junwu.nptu.edu.tw/dokuwiki/doku.php?id=cpp:2024spring:hw3>

Last update: 2024/03/26 14:49