**程式設計(二)-HW02**

Due to 04/14 PM 11:59／授課老師：紀博文

**一、基本資料**

姓名：林育辰

系級：資工111

學號：40771131H

**二、檔案有哪些？**

|  |  |
| --- | --- |
| **40771131H\_hw02 (主資料夾)** | **hw0204(次資料夾)** |
| **hw0201.c** | **test.h** |
| **hw0202.c** | **test.c** |
| **hw0203.c** | **script.txt** |
| **hw0204.c** | **readfile.h** |
| **hw0205.c** | **hw0204 (主資料夾make後產生)** |
| **hw0206.c** | **ac\_code.c(執行hw0204後產生)** |
| **readfile.c** | **ac\_code(次資料夾make後產生)** |
| **readfile.h** | **makefile(執行hw0204後產生)** |
| **a.txt (第一題)** |  |
| **key.txt (第一題)** |  |
| **JamieOliver.srt (第二題)** |  |
| **Dosbox.conf (第五題)** |  |
| **Makefile** |  |
| **README.pdf** |  |

◎每個.c檔皆有詳細註解！問題的回答寫在README每題詳細說明中！

◎ 助教應以相對應檔案，取代a.txt, key.txt, JamieOliver.srt, Dosbox.conf，hw0204資料夾內，應放置**test.h**, **test.c**, **script.txt**，**三者皆須**，在我所提供的壓縮黨內已先放置hw02所提供的程式檔！

◎ 第二、四、五題所提供之測試檔案**應須符合格式**

**三、如何執行？**

請輸入make→編譯hw0201.c~hw0206.c→產生hw0201~hw0206檔

指令如下：

$ make

$ ./hw0201

$ ./hw0202

$ ./hw0203

$ ./hw0205

$ ./hw0206

特別的是，hw0204.c編譯後，**可執行檔hw0204放置”hw0204”資料夾內**，

助教應輸入以下指令，以完成第四題之測試：

**$ cd hw0204 #切換至hw0204資料夾**

**$ ./hw0204 #執行”hw0204”的可執行檔→自動產生ac\_code.c, Makefile**

**$ make #編譯”test.c”, “ac\_code.c”為”ac\_code”**

**$ ./ac\_code #執行unit test程式**

然而要離開此資料夾，以便持續測試第五、六題，請輸入：

$ cd .. #回到主資料夾

如此，即可完成六題的測試。

**四、索引**

**第一題**──────────────────P.3-4

第二題──────────────────**P.5-6**

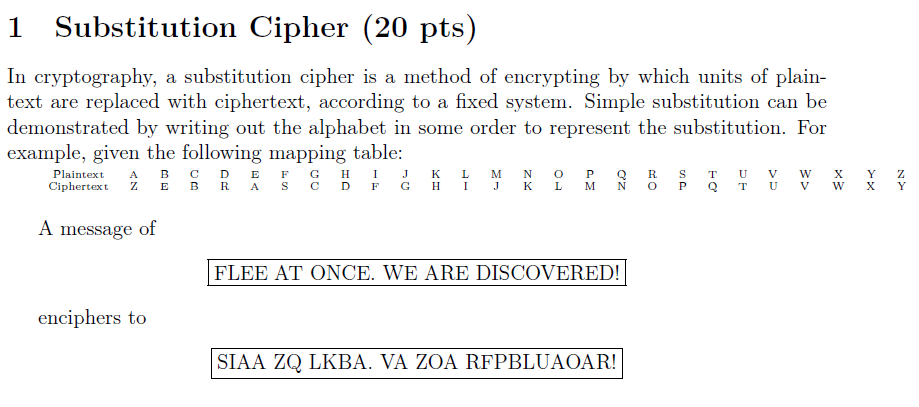
**第三題**──────────────────P.7-9

第四題──────────────────**P.9-12**

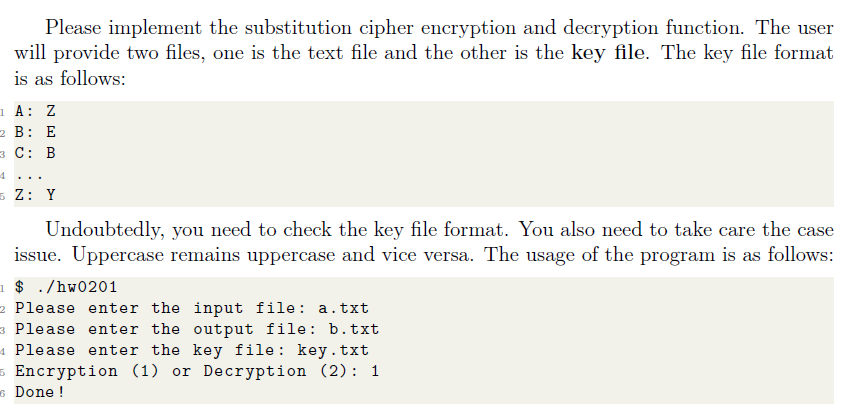
**第五題**──────────────────P.13-14

第六題──────────────────**P.15-16**

**說明**



**◎題意說明**→ 在密碼學中，替代字元是一個加密的方式，舉例來說，我們可以將”FLEE AT ONCE. WE ARE DISCOVERED!” 根據替代表加密為”SIAA ZQ LKBA. VA ZOA RFPBLUAOAR!”



→為了實作這樣的加密、解密系統，使用者會提供兩個檔案，一個是text file，另一個則為key file，即替代表，且格式如圖所示，你需要檢查key file的格式，也需要注意小寫、大寫問題，若其輸入小寫，你仍然會根據替代表將她更換為相對應的字母，但仍保持小寫，反之亦然。

**※注意**

1) 應自行提供a.txt, key.txt，其中無論大小寫均會加密、解密，但仍保持其大、小寫型態，其中key.file務必要有內容

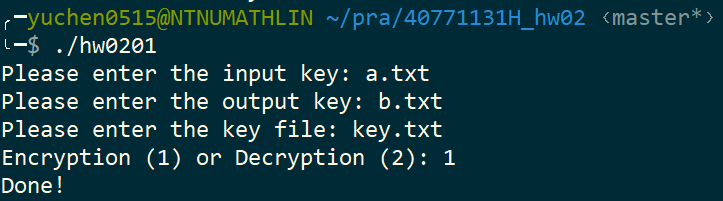
2) key file中，**必須保持一一對應**，否則執行後程式會告訴你key file錯誤！另格式請保持圖上所展示的，否則也會告訴您key file錯誤！

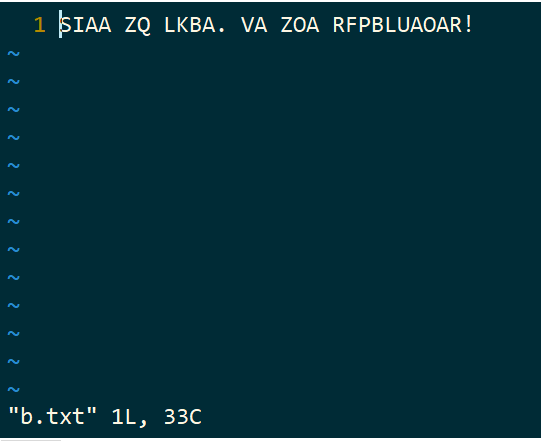
**◎輸出&輸出格式**

編譯後，執行”$ ./hw0201”

其結果依據助教所提供之a.txt, key.txt之內容而定

按照作業範例，則輸入如下：





以上則為輸出檔”b.txt”的檔案，若不全為大寫，仍會轉換，但不改變其大小寫型態：(此為b.txt)

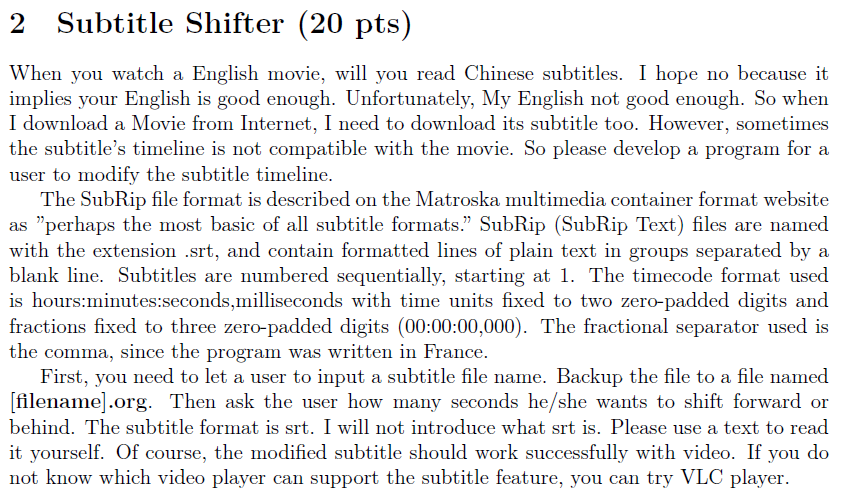


**◎程式設計思路**

1) 主要就是要會開檔、讀檔，寫好掃描格式的函式，並利用兩個26長度的陣列去存key file的內容，之後再讀取寫入就完成了

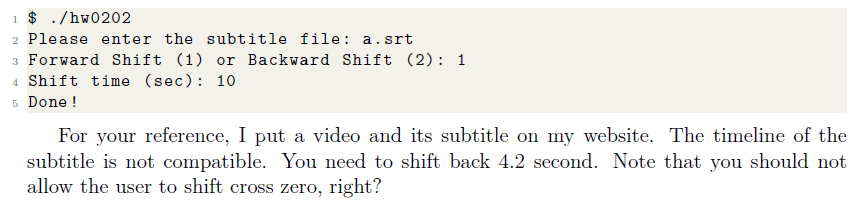
2) 之後才想到，可以利用上次作業一的第三題寫的函式來檢查格式，會更加簡單，不過這邊暫時先用比較暴力一些的方式檢查格式

**說明**

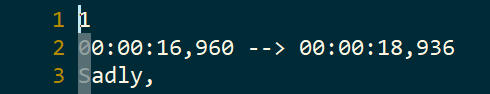


**◎題意說明**→內容很多，總之srt是一個字幕的格式，其中它的格式為hour:min:sec:msec，並且有固定的格式”00:00:00,000”，而且每段字幕間都會空一行做分隔。使用者會輸入一個srt檔，請先將此檔案備份為[filename].org，(在此我命名為[name].src.org)，然後對原始檔案修改。

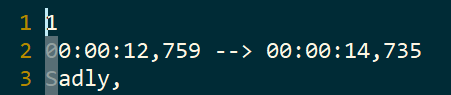
**◎輸入&輸出格式**



輸入格式大致如上，然助教所提供的.srt檔，**不應在windows底下執行過**(原因是會比原本\n多了/r/n兩個字元)，請使用老師所提供的字幕檔(壓縮檔內即是)執行，即可正常平移。

原始檔案：(JamieOliver.srt)

平移後：(JamieOliver.srt)



另原始檔案，備份為JamieOliver.srt.org

**注意：**

1. **Backward** shift為影片往**回**移(往過去移)
2. **Forward** shift為影片往**前**移(往未來移)
3. 您的**srt必須符合格式**
4. 若**時間平移後會產生負數，則不平移**，並提示使用者檔案將不被更動。

**◎程式設計思路**

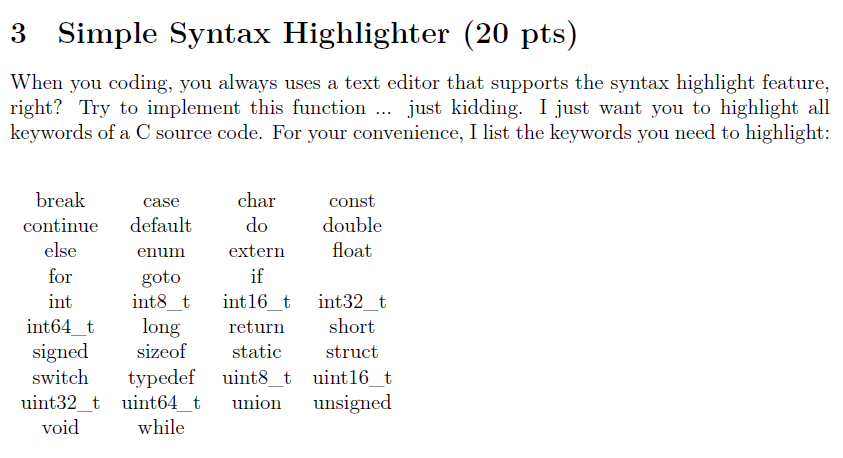
1) 先正常開檔，並將檔案存成字串，再開始尋訪

2) 一一掃過存到另個字串，遇到換行就判斷該行是否為時間格式，若是則進行更動，判斷方式則利用上次作業一第三題的函式處理

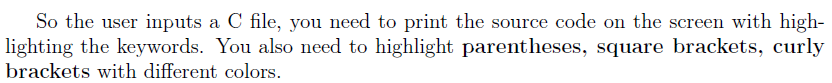
3) 若發生負數情況，則紀錄bool，然後不予更動

4) 此題會利用pattern檢查格式，並使用sscanf就可以處理(p.s. 進位函式是此題的靈魂)

**說明**



**◎題意說明**→通常coding時會使用文字編輯器，而這編輯器一般都支持syntax highlight的系統。



→因此你必須寫出一個程式，當使用者輸入一個cfile名稱，你需要將此source code利用highlighting的方式顯示在螢幕上，除了關鍵字以外，你必須以不同顏色去標明括號，如[], {}, ()。

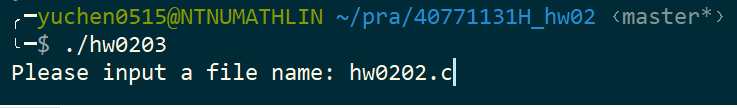
→P.S. 在此題中，我將變數類標成藍色，其他標成橘色，括號則標成淺藍，註解標成紫色(依據我自己使用的編輯器調配)

**※注意**

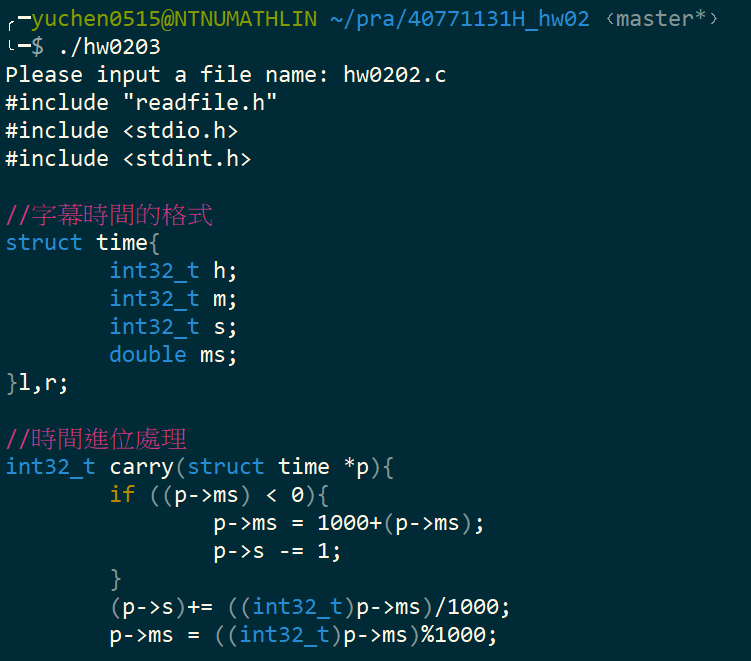
→題目內事實上並無要求註解、””內的文字highlight的問題，因此在我的程式中，如果出現 printf(“break”);，**在””內出現關鍵字，仍會標明**，題目並沒有要求這個要額外做處理。

→不要輸入可執行檔，請輸入正常的程式，此外程式並不會幫忙檢查”()”括號的配對情形。

**◎輸入&輸出格式**



編譯後，執行”$ ./hw0203”，並輸入一個正常的.c檔

**輸出如下：**

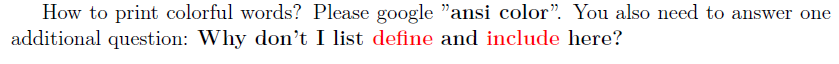
**◎程式設計思路**

1) 寫一個keyword function，去strcmp(檢查字串)是否為特定關鍵字，根據回傳值決定它的顏色

2) 一行一行讀，遇到關鍵字就前後塞上ansi color的跳脫字元即可

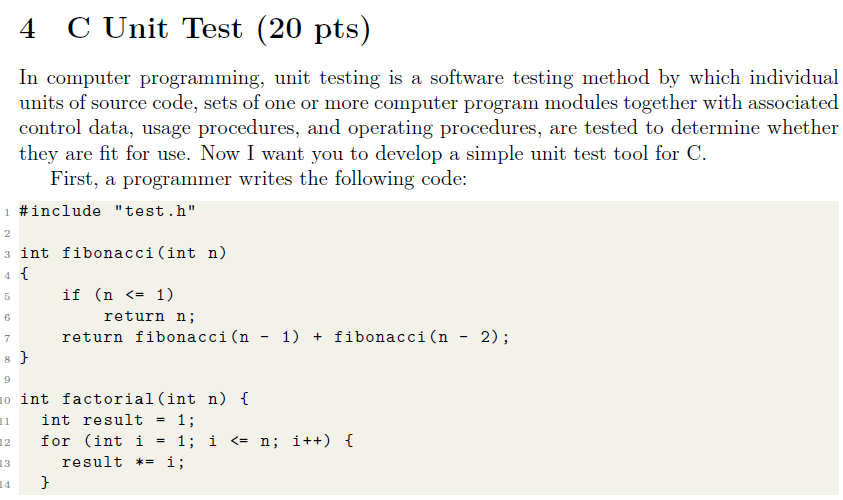
3) 事實上此題只有字串的比較會比較麻煩

**額外問題：**



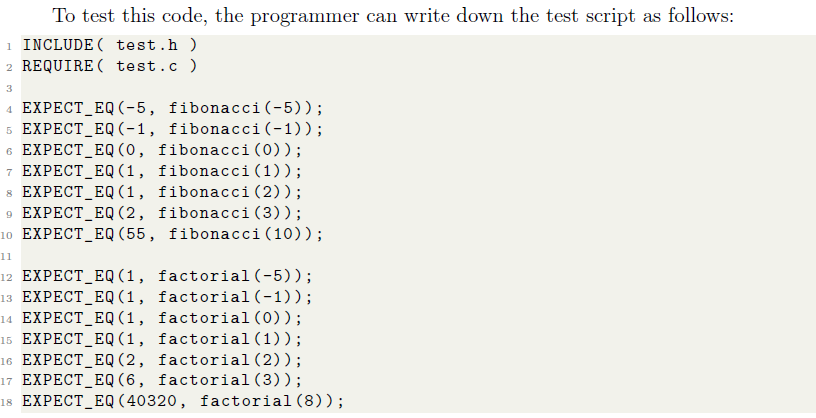
1. 我認為在程式撰寫中，必須盡可能少用define，他是預先處理的，直接把後面的程式碼全數代換掉，甚至連其後的註解都一起換掉
2. 當然硬要define和include去處理ansi color我認為實務上都是可行的，但問題在於ansi color有很多種顏色，如果我利用define去一一定義會非常沒效率，我們知道ansi color在printf中可以拆開寫，在數字處可以用%d依照想要的數值取代，然也許我們可以寫include的標頭檔
3. Define不能正確的指定型態
4. 若系統不支持anci-color插件，define and include的方式會導致程式出事
5. 就自己實際測試好幾次後，若#define break “\x1[;;m”…此類方式，會導致關鍵字已跳脫字元取代(在此程式)，因此雖可運行(編譯可過)，但gcc完全不知道程式已經被前置處理器代換過了，”編譯時不會出問題”，在實際運行後，以此例子來說，會陷入無限迴圈。
6. 倘若寫成 #define red …….／#define end ….，直接安插在printf前後，此方法基本上實務上都是沒有什麼問題的，然若用define指定define或include似乎也不會出事

**說明**





**◎題意說明**→軟體工程中，unit test是一個很常見的方式去檢驗你的函式是否出錯，因此，現在我需要你用C開發一個簡單的unit test tool，首先，剛開始的”test.c”檔如圖所示。





→而腳本內容如圖所示，然根據這個腳本，你必須使程式根據該內容去自動產生一個測試test.c的.c檔，和產生一個Makefile，使這個.c檔, test.c能夠被編譯，然而這個產生的檔案，**你可以自己命名**。

**◎注意**

1) **應額外將檔案放置在一個資料夾，裡面包含腳本、test.c、test.h(助教須提供)**

2) 為了助教方便，我將hw0204.c放在主資料夾，在主資料夾make過後，可執行檔會放置在hw0204資料夾內，請輸入以下指令：

$ cd hw0204

$ ./hw0204

$ make

$ ./ac\_code

即可執行，其中ac\_code.c為我所產生的.c檔名稱，可執行檔名稱為ac\_code，而不是a.out，若要回到主資料夾直接”cd ..”即可

3) 腳本內容應完全符合格式，且**只接受以下指令**：**(1) INCLUDE (2) REQUIRE (3) EXPECT\_EQ**，其中程式不負責檢查括號是否配對、合理，應與上圖完全一致，若有不合理的指令會輸出腳本錯誤，此外(1)和(2)必須出現恰好一次，且在呼叫(3)以前出現。

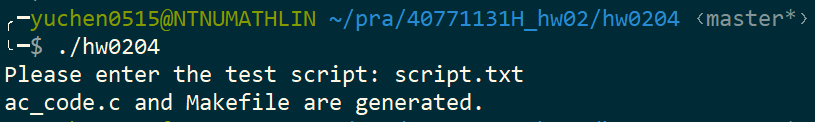
4) **不檢查回傳值**，預設一律為整數，理由是同學寄信問助教，助教表示在此題中還不需檢查回傳變數型態，但如果真的要檢查可以用sscanf

**◎執行步驟**

1) 主資料夾make過

2) cd至hw0204

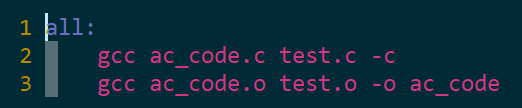
執行”$ ./hw0204”，並輸入腳本名稱，如下：



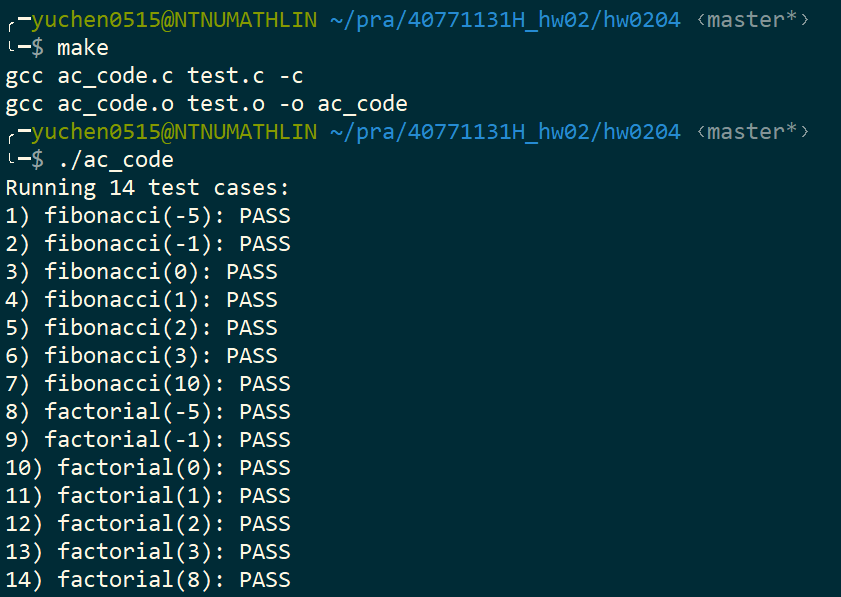
此為產生的ac\_code.c：



以及產生的Makefile：



3) 在子資料夾make，輸入./ac\_code，如圖：



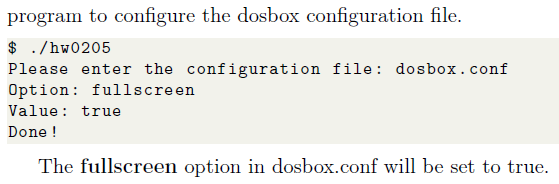
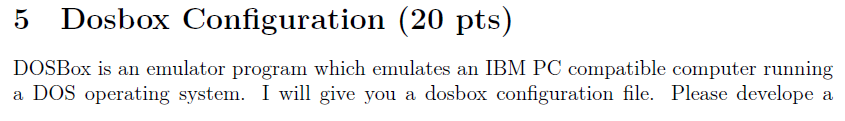
**◎程式設計思路**

1) 利用作業一第三題去檢查樣式，並使用sscanf去讀取資料，甚至是函式名稱

2) 只要能夠順利取得內容，產生.c檔跟Makefile就只是純粹fprintf而已

3) 比較困難的是，如何事前知道有幾筆測資？我的作法是先讀檔一次，檢查有幾個EXPECT\_EQ再存下來，之後再重新讀檔做後續的事情即可

**說明**



**◎題意說明**→DOS 是一個在linux玩遊戲的方式，我將給你一個dosbox的confguration file，請寫一個程式，讓我們能夠修改內容。

**◎輸入&輸出方式**

編譯後，執行”$ ./hw0205”

→輸入.conf的名稱

→輸入option

→輸入想改的數值

**◎注意**

1) 遇到”#”其後字元一律不處理

2) 若該option出現兩次，會一併修改

3) 您的conf必須完全符合格式如：

fullscreen=true

也許你不該隨意空格，可參考老師的dosbox.conf檔案，**應符合其格式**

4) 若該option沒找到，會輸出找不到此option

1. 考慮有些value有比較多數值如wid,len，在輸入時**中間不應有任何空格**。

**◎程式設計思路**

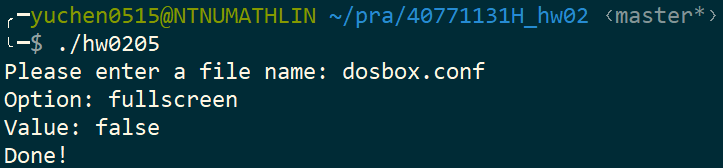
1) 將檔案所有內容讀入一個字串，並逐行開始檢查

2) 檢查時暫存進一個陣列，不斷比較是否等於options，若不是就直接吐出來原始的資料

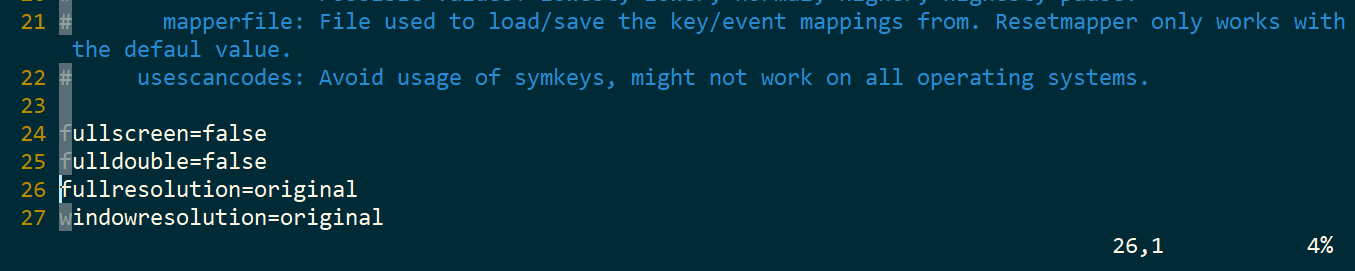
3) 遇到等號，空白就檢查目前陣列等不等於option，等於就直接讀掉其後的數值，直接fprintf指定的values

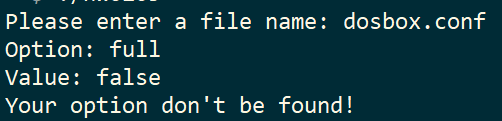
4) 遇到#，直接單純輸出後面的字元就好

實際範例如下：(輸入)

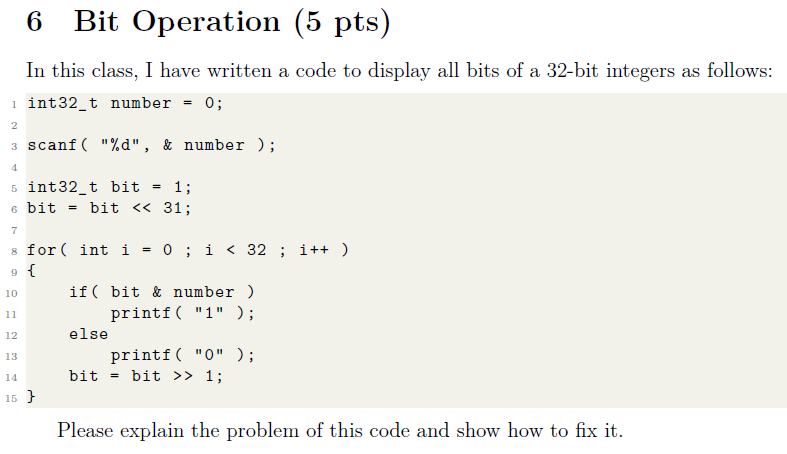


修改後的狀況



若輸入的option沒找到則：

**說明**



**◎題意說明**→ 在課堂上，我曾經寫過類似這樣的code，在這code中有一點小問題，請你修正這個code並說明這個程式碼發生了什麼事。

**◎修改方式**

→如我的hw0206.c內，直接將這裡的第五行改成**u**int32\_t bit =1; 即可

**◎解釋**

→這裡做的事，只是單純印出一個十進位數字的二進制數值，然int32\_t範圍為**[, ]**，因此bit <<31明顯恰高於上限”1”，因此根據計算機概論所學，他會變成10000……0000，然而現在電腦一般使用二補數的方式解讀，也就是說，**這串數字會變成**，為負的最大值(也就是最負)

→但這樣有什麼問題呢，他一樣是100000.0000呀?

→不，即使他**「目前」是符合你的期待的樣式**，但電腦已經將它當作負數解讀了，當執行”bit >>=1”時，**電腦會儲存其最高位(去記錄他為負數)**，然後確實你的1順利往後移一位，但他仍在最高位記錄著他是負數(也就是最高位還是有1)，變成**1100000…0000**…，再進行一次，則變成11100000….0000。

→電腦會這樣解讀不是無道理的，原本已經溢位，數值為-2147483648，我對這個數值除以二，就如同電腦所解讀的，我應該記錄他的負號，再除以二，因此會變成-1073741824

→也就是說，在這個for迴圈，我們的bit不會如預期地只是純粹一個1往後移動，而是他的所到之處，都會一併變為1，導致結果出現錯誤

→因此我只要讓bit不要發生溢位即可，最簡單的方式就是uint32\_t