

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，在我所提供的壓縮檔內已先放置 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

1 Substitution Cipher (20 pts)

In cryptography, a substitution cipher is a method of encrypting by which units of plaintext are replaced with ciphertext, according to a fixed system. Simple substitution can be demonstrated by writing out the alphabet in some order to represent the substitution. For example, given the following mapping table:

Plaintext	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ciphertext	Z	E	B	R	A	S	C	D	F	G	H	I	J	K	L	M	N	O	P	Q	T	U	V	W	X	Y

A message of

FL EE AT ONCE. WE ARE DISCOVERED!

enciphers to

SIAA ZQ LKBA. VA ZOA RFPBLUAOAR!

◎**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”FL EE AT ONCE. WE ARE DISCOVERED!” 根據替代表加密為”SIAA ZQ LKBA. VA ZOA RFPBLUAOAR!”

Please implement the substitution cipher encryption and decryption function. The user will provide two files, one is the text file and the other is the key file. The key file format is as follows:

```
1 A: Z
2 B: E
3 C: B
4 ...
5 Z: Y
```

Undoubtedly, you need to check the key file format. You also need to take care the case issue. Uppercase remains uppercase and vice versa. The usage of the program is as follows:

```
1 $ ./hw0201
2 Please enter the input file: a.txt
3 Please enter the output file: b.txt
4 Please enter the key file: key.txt
5 Encryption (1) or Decryption (2): 1
6 Done!
```

→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：

```
└─yuchen0515@NTNUMATHLIN ~/pra/40771131H_hw02 <master*>
└─$ ./hw0201
Please enter the input key: a.txt
Please enter the output key: b.txt
Please enter the key file: key.txt
Encryption (1) or Decryption (2): 1
Done!
```

[illegible]

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)

1 | SIaA ZQ LkBA. VA ZoA RFPBLuAOAR!

◎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

2 Subtitle Shifter (20 pts)

When you watch a English movie, will you read Chinese subtitles. I hope no because it implies your English is good enough. Unfortunately, My English not good enough. So when I download a Movie from Internet, I need to download its subtitle too. However, sometimes the subtitle's timeline is not compatible with the movie. So please develop a program for a user to modify the subtitle timeline.

The SubRip file format is described on the Matroska multimedia container format website as "perhaps the most basic of all subtitle formats." SubRip (SubRip Text) files are named with the extension .srt, and contain formatted lines of plain text in groups separated by a blank line. Subtitles are numbered sequentially, starting at 1. The timecode format used is hours:minutes:seconds,milliseconds with time units fixed to two zero-padded digits and fractions fixed to three zero-padded digits (00:00:00,000). The fractional separator used is the comma, since the program was written in France.

First, you need to let a user to input a subtitle file name. Backup the file to a file named [filename].org. Then ask the user how many seconds he/she wants to shift forward or behind. The subtitle format is srt. I will not introduce what srt is. Please use a text to read it yourself. Of course, the modified subtitle should work successfully with video. If you do not know which video player can support the subtitle feature, you can try VLC player.

◎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

```
1 $ ./hw0202
2 Please enter the subtitle file: a.srt
3 Forward Shift (1) or Backward Shift (2): 1
4 Shift time (sec): 10
5 Done!
```

For your reference, I put a video and its subtitle on my website. The timeline of the subtitle is not compatible. You need to shift back 4.2 second. Note that you should not allow the user to shift cross zero, right?

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)

```
1 1
2 00:00:16,960 --> 00:00:18,936
3 Sadly,
```

平移後：(JamieOliver.srt)

```
1 1
2 00:00:12,759 --> 00:00:14,735
3 Sadly,
```

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) 一一掃過存到另一個字串，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

3 Simple Syntax Highlighter (20 pts)

When you coding, you always uses a text editor that supports the syntax highlight feature, right? Try to implement this function ... just kidding. I just want you to highlight all keywords of a C source code. For your convenience, I list the keywords you need to highlight:

break	case	char	const
continue	default	do	double
else	enum	extern	float
for	goto	if	
int	int8_t	int16_t	int32_t
int64_t	long	return	short
signed	sizeof	static	struct
switch	typedef	uint8_t	uint16_t
uint32_t	uint64_t	union	unsigned
void	while		

◎**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。

So the user inputs a C file, you need to print the source code on the screen with highlighting the keywords. You also need to highlight parentheses, square brackets, curly brackets with different colors.

→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

```
r-yuchen0515@NTNUMATHLIN ~/pra/40771131H_hw02 <master*>  
└─$ ./hw0203  
Please input a file name: hw0202.c|
```

編譯後，執行”\$./hw0203”，並輸入一個正常的.c 檔

輸出如下：

```
r-yuchen0515@NTNUMATHLIN ~/pra/40771131H_hw02 <master*>  
└─$ ./hw0203  
Please input a file name: hw0202.c  
#include "readfile.h"  
#include <stdio.h>  
#include <stdint.h>  
  
//字幕時間的格式  
struct time{  
    int32_t h;  
    int32_t m;  
    int32_t s;  
    double ms;  
}l,r;  
  
//時間進位處理  
int32_t carry(struct time *p){  
    if ((p->ms) < 0){  
        p->ms = 1000+(p->ms);  
        p->s -= 1;  
    }  
    (p->s)+= ((int32_t)p->ms)/1000;  
    p->ms = ((int32_t)p->ms)%1000;
```

◎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：

How to print colorful words? Please google "ansi color". You also need to answer one additional question: Why don't I list **define** and **include** here?

- 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 ansi-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

4 C Unit Test (20 pts)

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Now I want you to develop a simple unit test tool for C.

First, a programmer writes the following code:

```
1 #include "test.h"
2
3 int fibonacci(int n)
4 {
5     if (n <= 1)
6         return n;
7     return fibonacci(n - 1) + fibonacci(n - 2);
8 }
9
10 int factorial(int n) {
11     int result = 1;
12     for (int i = 1; i <= n; i++) {
13         result *= i;
14     }
15
16     return result;
17 }
```

◎**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。

To test this code, the programmer can write down the test script as follows:

```
1 INCLUDE( test.h )
2 REQUIRE( test.c )
3
4 EXPECT_EQ(-5, fibonacci(-5));
5 EXPECT_EQ(-1, fibonacci(-1));
6 EXPECT_EQ(0, fibonacci(0));
7 EXPECT_EQ(1, fibonacci(1));
8 EXPECT_EQ(1, fibonacci(2));
9 EXPECT_EQ(2, fibonacci(3));
10 EXPECT_EQ(55, fibonacci(10));
11
12 EXPECT_EQ(1, factorial(-5));
13 EXPECT_EQ(1, factorial(-1));
14 EXPECT_EQ(1, factorial(0));
15 EXPECT_EQ(1, factorial(1));
16 EXPECT_EQ(2, factorial(2));
17 EXPECT_EQ(6, factorial(3));
18 EXPECT_EQ(40320, factorial(8));
```

Based on this script, your program needs to generate a C test source code and a Makefile to make your test code. The test source code can be named as you wish.

→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：

```
yuchen0515@NTNUMATHLIN ~/pra/40771131H_hw02/hw0204 <master*>  
└─$ ./hw0204  
Please enter the test script: script.txt  
ac_code.c and Makefile are generated.
```

此為產生的 ac_code.c：

```
1 #include "test.h"  
2 #include <stdio.h>  
3 #include <stdint.h>  
4  
5 int main(){  
6  
7     printf("Running 14 test cases:\n");  
8  
9     printf("1) fibonacci(-5): ");  
10    if ( fibonacci(-5) == -5 ) printf("PASS\n");  
11    else printf("Return: %d, Wanted: -5\n", fibonacci(-5));  
12  
13    printf("2) fibonacci(-1): ");  
14    if ( fibonacci(-1) == -1 ) printf("PASS\n");  
15    else printf("Return: %d, Wanted: -1\n", fibonacci(-1));  
16  
17    printf("3) fibonacci(0): ");  
18    if ( fibonacci(0) == 0 ) printf("PASS\n");  
19    else printf("Return: %d, Wanted: 0\n", fibonacci(0));  
20  
21    printf("4) fibonacci(1): ");  
22    if ( fibonacci(1) == 1 ) printf("PASS\n");  
"ac_code.c" 66L, 1976C
```

以及產生的 Makefile：

```
1 all:  
2     gcc ac_code.c test.c -c  
3     gcc ac_code.o test.o -o ac_code
```

3) 在子資料夾 make，輸入./ac_code，如圖：

```
r-yuchen0515@NTNUMATHLIN ~/pra/40771131H_hw02/hw0204 <master*>
└─$ make
gcc ac_code.c test.c -c
gcc ac_code.o test.o -o ac_code
r-yuchen0515@NTNUMATHLIN ~/pra/40771131H_hw02/hw0204 <master*>
└─$ ./ac_code
Running 14 test cases:
1) fibonacci(-5): PASS
2) fibonacci(-1): PASS
3) fibonacci(0): PASS
4) fibonacci(1): PASS
5) fibonacci(2): PASS
6) fibonacci(3): PASS
7) fibonacci(10): PASS
8) factorial(-5): PASS
9) factorial(-1): PASS
10) factorial(0): PASS
11) factorial(1): PASS
12) factorial(2): PASS
13) factorial(3): PASS
14) factorial(8): PASS
```

◎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

5 Dosbox Configuration (20 pts)

DOSBox is an emulator program which emulates an IBM PC compatible computer running a DOS operating system. I will give you a dosbox configuration file. Please develop a

program to configure the dosbox configuration file.

```
$ ./hw0205
Please enter the configuration file: dosbox.conf
Option: fullscreen
Value: true
Done!
```

The fullscreen option in dosbox.conf will be set to true.

◎題意說明→DOS 是一個在 linux 玩遊戲的方式，我將給你一個 dosbox 的 configuration 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

7) 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)

```
yuchen0515@NTNUMATHLIN ~/pra/40771131H_hw02 <master*>  
└─$ ./hw0205  
Please enter a file name: dosbox.conf  
Option: fullscreen  
Value: false  
Done!
```

修改後的狀況

```
21 # mapperfile: File used to load/save the key/event mappings from. Resetmapper only works with  
    the default value.  
22 # usescancodes: Avoid usage of symkeys, might not work on all operating systems.  
23  
24 fullscreen=false  
25 fulldouble=false  
26 fullresolution=original  
27 windowresolution=original  
  
26,1 4%
```

若輸入的 option 沒找到則：

```
Please enter a file name: dosbox.conf  
Option: full  
Value: false  
Your option don't be found!
```

說明

6 Bit Operation (5 pts)

In this class, I have written a code to display all bits of a 32-bit integers as follows:

```
1 int32_t number = 0;
2
3 scanf( "%d", & number );
4
5 int32_t bit = 1;
6 bit = bit << 31;
7
8 for( int i = 0 ; i < 32 ; i++ )
9 {
10     if( bit & number )
11         printf( "1" );
12     else
13         printf( "0" );
14     bit = bit >> 1;
15 }
```

Please explain the problem of this code and show how to fix it.

◎**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 `uint32_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 $[2^{31} - 1, -2^{31}]$ ，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 -2^{31}** ，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