

程式設計(一)-HW05

Due to 12/15 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？

- 1) hw0501.c
- 2) hw0502.c
- 3) hw0503.c
- 4) hw0504.c
- 5) hw0505.c
- 6) hw0505_header.c
- 7) hw0505_header.h
- 8) hw0506.c
- 9) Makefile
- 10) 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！

◎根據老師給的程式碼，hw0506 會有 warning，(因為宣告的是 uint8_t，他會提是你應該 scanf 的是"%hhu"，為了不要有 warning，Makefile 內**不編譯 hw0506.c**，但會在這 Makefile 寫清楚原因。

三、如何執行？

請輸入 make→編譯 hw0501.c~hw0505.c→產生 hw0501~hw0505 檔

指令如下：

```
$ make
```

```
$ ./hw0501
```

```
$ ./hw0502
```

```
...
```

以此類推，即可分別執行 hw0501~hw0505 [hw0506 不編譯]

四、索引

第一題	P.2-5
第二題	P.6-8
第三題	P.9-16
第四題	P.17-18
第五題	P.19-22
第六題	P.22-26

說明

1 Matrix Arithmetic Operation (20 pts)

I believe that you know what matrix is, right? If no, do not worry. You have Google and Wikipedia. We also have a class called Linear Algebra. Please develop a matrix arithmetic calculator.

If there is any input that is invalid, print an error message.

◎題意說明→ 寫出程式，讓使用者輸入兩個矩陣，若選擇為 1，則輸出矩陣相加結果，選擇為 2，輸出相減結果，選擇為 3，輸出矩陣乘法結果

※注意

- 1) 必須檢查兩個矩陣 row 跟 column 是否能夠相加、相減、相乘，如相加或相減，兩個矩陣 row 和 column 的數量要一樣(如大小 3x2 vs 3x2)
- 2) 相乘則第一個矩陣的 column 數應要跟第二個矩陣的 row 數量一致(如 3x3 vs 3x3, 3x4 vs 4x5)
- 3) 若不符合上述兩點，則輸出 error message.

◎輸入格式

```
1 $ ./hw0501
2 Please enter the operation choice (1: add; 2: sub; 3: mul): 1
3 Please enter the size of the 1st matrix: 3 2
4 Please enter the 1st row of the 1st matrix: 1 1
5 Please enter the 2nd row of the 1st matrix: 1 1
6 Please enter the 3rd row of the 1st matrix: 1 1
7 Please enter the size of the 2nd matrix: 3 2
8 Please enter the 1st row of the 2nd matrix: 2 2
9 Please enter the 2nd row of the 2nd matrix: 2 2
10 Please enter the 3rd row of the 2nd matrix: 2 2
11 Answer:
12 3 3
13 3 3
14 3 3
```

編譯後，執行”\$./hw0501”

→先選擇要相加、相減還是相乘

→再依序輸入矩陣大小(row,column)，輸入 row 次長度為 column 的數字

→再輸入第二個矩陣大小，再次輸入 row 次長度為 column 的數

※若選擇相加、相減，第二個矩陣大小不等於第一次矩陣大小

→提示使用者重新輸入

◎檢查

- 1) 選擇運算不再[1,3]內時→重新輸入
- 2) 相加相減，兩個矩陣大小不相等→重新輸入
- 3) 相乘，第一個的 column 不等於第二個的 row→輸出 error message

◎輸出格式

- 1) 若無法作運算→輸出 error message
- 2) 可運算→輸出 Answer:／矩陣的每個 entry

◎程式設計思路

- 1) 設兩個二維陣列存取使用者輸入的兩個矩陣
- 2) 如果選擇是 1 或 2，單純跑雙層迴圈，依序輸出相加結果就可以了
- 3) 如果是 3，跑三層迴圈(第一層跑 row1 次，第二層跑 column2 次，第三層則依序跑 column1 或 row2 次，將第一個矩陣的列跟第二個矩陣的行作內積，跳出迴圈後再輸出即可)

◎各情形範例

1) Ex: "4"

Tip: 輸入不在[1,3]範圍內=>提示使用者重新輸入

```
yuchen0515@NTNUMATHLIN ~/程式設計hw5
└─$ ./hw0501
Please enter the operation choice (1: add; 2: sub; 3: mul): 4
Error message! (about your type isn't on [1,3])
Please enter the operation choice (1: add; 2: sub; 3: mul): |
```

2) Ex: "1/3 2/1 1/1 1/1 1/3 2/2 2/2 2/2"

Tip: test case #1

```
Please enter the operation choice (1: add; 2: sub; 3: mul): 1
Please enter the sizeof the 1st matrix: 3 2
Please enter the 1st row of the 1st matrix (2 numbers): 1 1
Please enter the 2nd row of the 1st matrix (2 numbers): 1 1
Please enter the 3rd row of the 1st matrix (2 numbers): 1 1
Please enter the sizeof the 2nd matrix: 3 2
Please enter the 1st row of the 2nd matrix (2 numbers): 2 2
Please enter the 2nd row of the 2nd matrix (2 numbers): 2 2
Please enter the 3rd row of the 2nd matrix (2 numbers): 2 2
Answer:
3 3
3 3
3 3
```

3) EX: "1/3 2/1 1/1 1/1 1/3 3"

Tip: 相加、大小不同

```
Please enter the operation choice (1: add; 2: sub; 3: mul): 1
Please enter the sizeof the 1st matrix: 3 2
Please enter the 1st row of the 1st matrix (2 numbers): 1 1
Please enter the 2nd row of the 1st matrix (2 numbers): 1 1
Please enter the 3rd row of the 1st matrix (2 numbers): 1 1
Please enter the sizeof the 2nd matrix: 3 3
Error! your type is add or sub, the two matrix should be same as
Please enter the sizeof the 2nd matrix: |
```

4) EX: "2/3 4/1 2 3 4/5 6 7 8/9 10 11 12/3 4/13 12 11 10/9 8 7 6/5 4 3 2"

```
Please enter the operation choice (1: add; 2: sub; 3: mul): 2
Please enter the sizeof the 1st matrix: 3 4
Please enter the 1st row of the 1st matrix (4 numbers): 1 2 3 4
Please enter the 2nd row of the 1st matrix (4 numbers): 5 6 7 8
Please enter the 3rd row of the 1st matrix (4 numbers): 9 10 11 12
Please enter the sizeof the 2nd matrix: 3 4
Please enter the 1st row of the 2nd matrix (4 numbers): 13 12 11 10
Please enter the 2nd row of the 2nd matrix (4 numbers): 9 8 7 6
Please enter the 3rd row of the 2nd matrix (4 numbers): 5 4 3 2
Answer:
-12 -10 -8 -6
-4 -2 0 2
4 6 8 10
```

5) EX: "3/2 3/1 0 2/-1 3 1/3 2/3 1/2 1/1 0"

```
Please enter the operation choice (1: add; 2: sub; 3: mul): 3
Please enter the sizeof the 1st matrix: 2 3
Please enter the 1st row of the 1st matrix (3 numbers): 1 0 2
Please enter the 2nd row of the 1st matrix (3 numbers): -1 3 1
Please enter the sizeof the 2nd matrix: 3 2
Please enter the 1st row of the 2nd matrix (2 numbers): 3 1
Please enter the 2nd row of the 2nd matrix (2 numbers): 2 1
Please enter the 3rd row of the 2nd matrix (2 numbers): 1 0
Answer:
5      1
4      2
```

這一特性使得矩陣代數與常見的一些數體（有理數、實數、複數）以及環（多項式環、整數環）等，稱為 A 的交換子環。這些矩陣也構成 $M(n, \mathbb{R})$ 的一個子空間，稱為 A 的可交換空間^[17]。與矩陣，其中的 I_n 是單位矩陣，也就是主對角線上的元素為 1，其它元素為 0 的矩陣。任意矩陣 M

6) EX: "3/2 3/1 0 2/-1 3 1/4 2/3 1/2 1/1 0/5 8"

Tip: 矩陣相乘，無法進行運算

```
Please enter the operation choice (1: add; 2: sub; 3: mul): 3
Please enter the sizeof the 1st matrix: 2 3
Please enter the 1st row of the 1st matrix (3 numbers): 1 0 2
Please enter the 2nd row of the 1st matrix (3 numbers): -1 3 1
Please enter the sizeof the 2nd matrix: 4 2
Please enter the 1st row of the 2nd matrix (2 numbers): 3 1
Please enter the 2nd row of the 2nd matrix (2 numbers): 2 1
Please enter the 3rd row of the 2nd matrix (2 numbers): 1 0
Please enter the 4th row of the 2nd matrix (2 numbers): 5 8
Error message! (about your two array size don't product)
```

說明

2 Gaussian Elimination (20 pts)

Gaussian elimination, also known as row reduction, is an algorithm in linear algebra for solving a system of linear equations. If you do not know what it is, do not worry. I just want you to solve simultaneous equations. For example, given

$$\begin{cases} x + y + z = 3 \\ x + 2y + 2z = 5 \\ x + 3y + 5z = 9 \end{cases},$$

you should derive $x = 1, y = 1, z = 1$.

◎**題意說明**→讓使用者輸入方程式共要求解的數量(n)，並輸入 n 條方程式，每條有 n+1 個係數(包含常數項)，並進行高斯消去法輸出最終每個未知數的數值，若為無解則輸出無解，無限多組解則輸出哪幾個是 free variable，其餘答案正常輸出

In this problem, all answers are integers.

※注意：輸入測資的答案都為整數，可不用考慮小數答案的情況

◎檢查

1) 未知數數量必須大於 0，否則重新輸入

◎輸入格式

```
1 $ ./hw0502
2 Please enter the variable number: 3
3 Please enter the 1st equation: 1 1 1
4 Please enter the 2nd equation: 1 2 2
5 Please enter the 3rd equation: 1 3 5
6 Answer:
7 x1 = 1
8 x2 = 1
9 x3 = 1
```

編譯後，執行“\$./hw0502”，輸入：

- 1) 未知數的數量 n(需大於 0)
- 2) 輸入 n 次，每次輸入 n+1 個係數

◎輸出格式

- 1) 輸出每個未知數的答案
- 2) 若為無解，輸出 no solution
- 3) 若為無限多組解，則照樣輸出，不過有下列要點：

EX: $x_1 + 2x_2 = 9$

$$3x_1 + 7x_2 = 16$$

$$2x_1 + 4x_2 = 18$$

$\Rightarrow x_1 = 31 / x_2 = -11 / x_3 = \text{free}$

※在資工系線性代數課內，將可為任意數的 free variable，寫為 free

◎程式設計思路

- 1) 寫出三個副函數，模擬 elementary row operations 的三個操作，exchange(列交換)/scaling(列乘以常數)/row operation(列運算)
- 2) 依序由上而下作 forward pass，再由下而上往回作 backward pass，最終即可得解
- 3) 實作上則跑雙層迴圈，外圈跑到 num-1，內圈由 i+1 跑到 num，若原本第 i 列的領導係數為 0，則往下找第一個領導係數不為 0 的和他做交換，再進行列運算，若遇到整行都為 0 的則在 many_sol 陣列中紀錄為 1，(代表有 free variable)，那麼我的 i 行應該要再跑一次，只是 column index 要往右移一格(領導係數往後移)，依此類推
- 4) 由上而下，由下而上消除完畢後，應該為 reduced row echelon form，在 diagonal 線上都為 1，若有為 0 的那列則要檢查常數項是否為 0，若不為 0 則是”無解”，(即 $0x = 8$ 的狀況)
- 5) 輸出時，檢查該變數紀錄上是否為 many_sol 的成員(為 1 則代表該變數為自由變數)，若是就輸出 free，不是就輸出答案(常數項)

◎各情形範例

1) Ex: "0"

Tip: 變數數量沒有大於 0

```
yuchen0515@NTNUMATHLIN ~/程式設計hw5
└─$ ./hw0502
Please enter the variable number: 0
Input Error! Your value should be over zero
Please enter the variable number: |
```

2) Ex: "3/1 1 1 3/1 2 2 5/1 3 5 9"

Tip: test case #1

```
Please enter the variable number: 3
Please enter the 1st equation (4 numbers): 1 1 1 3
Please enter the 2nd equation (4 numbers): 1 2 2 5
Please enter the 3rd equation (4 numbers): 1 3 5 9
Answer:
x1 = 1
x2 = 1
x3 = 1
```

3) Ex: "3/1 2 0 9/3 7 0 16/2 4 0 18"

Tip: 無限多組解的情況

```
Please enter the variable number: 3
Please enter the 1st equation (4 numbers): 1 2 0 9
Please enter the 2nd equation (4 numbers): 3 7 0 16
Please enter the 3rd equation (4 numbers): 2 4 0 18
Answer:
x1 = 31
x2 = -11
x3 = free
```

4) Ex: "3/1 2 0 9/2 4 0 19/0 0 0 1"

Tip: 無解

```
Please enter the variable number: 3
Please enter the 1st equation (4 numbers): 1 2 0 9
Please enter the 2nd equation (4 numbers): 2 4 0 19
Please enter the 3rd equation (4 numbers): 0 0 0 1
Answer:
The equation isn't solution.
```


說明

3 Mahjong (20 pts)

OK, OK, OK. I know you do not like poker games. So I change the topic. Let's play Mahjong. For your simplicity, figure 1 to see what Mahjong is.

There are **four** identical copies of each Honors tile. Each tile is encoded as figure 2. For your simplicity, I skip the season tile and the dragon tile.

Now I will show you what a winning hand is. Here we use the Taiwanese rule, which involves hands of sixteen tiles. A winning hand must consists of **five melds** and **a pair**. There are three kinds of melds:

◎**題意說明**→好啦好啦好啦，知道你們不喜歡撲克牌，那我改玩麻將。現在每種 tile 有四張，編碼如下：

		Numbers									
		1	2	3	4	5	6	7	8	9	
Simples	Dots										1 - 9
	Bamboo										10 - 18
	Characters										19 - 27

Winds				Dragons			
East	South	West	North	Red	Green	White	
28-31				32-34			

Figure 2: 麻將 Encoding °

我們的手牌是依據台灣規則(也就是十六張)進行，胡牌應該由 5 個 melds 和 1 個 pairs 所組成，而 meld 的定義如下：

1. **Pongs**: Pongs are a set of three identical tiles.

- 九筒、九筒、九筒
- 東、東、東

2. **Kong**: Kong is a complete set of four identical tiles.

- 九筒、九筒、九筒、九筒
- 東、東、東、東

3. **Chow**: Chow is a meld of three suited tiles in sequence.

- 七筒、八筒、九筒
- 一萬、二萬、三萬

Eyes are two identical tiles which are an essential part of a legal winning hand. Since a meld could be **Kong**, a winning hand may be more than 17 hands.

Please develop a program to check if a given hand is a winning hand. The user can input tiles continuously until -1. Note that the input sequence is unordered. If the hand is a winning hand, print **Winning Hand**; Otherwise, print **Not a Winning Hand**.

有槓、刻、順三種方式可以形成 melds，你的任務是讓使用者不斷輸入牌，並判斷該牌可不可胡。

※注意

- 1) 不用考慮大四喜
- 2) 五個槓，一個對仍然可以胡牌(這和正確的麻將玩法不符，但符合作業定義)

◎檢查

- 1) 每張 tile 都應該只有 4 張，超過要提示使用者重新輸入
- 2) 超出編碼範圍要讓使用者重新輸入

◎輸入格式

```
1 $ ./hw0503
2 Please enter the tile: 1
3 Please enter the tile: 2
4 Please enter the tile: 3
```

```
5 Please enter the tile: 4
6 Please enter the tile: 5
7 Please enter the tile: 6
8 Please enter the tile: 7
9 Please enter the tile: 8
10 Please enter the tile: 9
11 Please enter the tile: 12
12 Please enter the tile: 12
13 Please enter the tile: 12
14 Please enter the tile: 31
15 Please enter the tile: 31
16 Please enter the tile: 31
17 Please enter the tile: 34
18 Please enter the tile: 34
19 Please enter the tile: -1
20 Winning Hand!
```

Of course, input check is necessary.

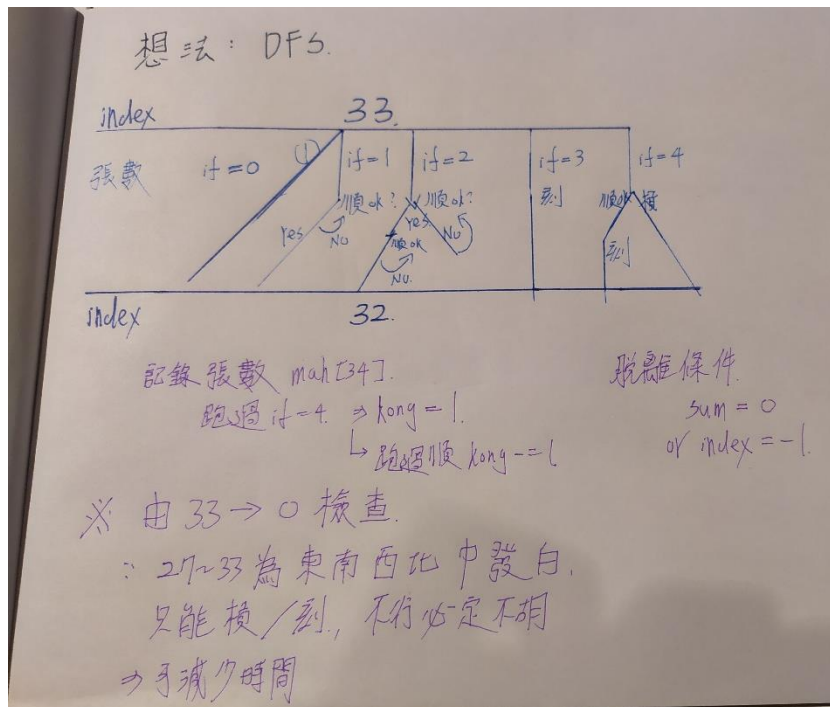
編譯後，執行”\$./hw0503”

輸入手牌編碼，直到輸入-1 為止

◎輸出格式

1) 輸出可不可胡牌

◎程式設計思路



- 1) 首先寫出複製的副函數、當門的副函式(將手牌可當眼的依序跑過一次)、取掉對後判斷該 tile 為順、刻、槓可不可胡的函數
- 2) 想法如上圖，每次傳入 mah[34](各種牌有幾張的陣列)，每次跑都判斷，如果為 0，就索引值扣 1(由後往前掃，因為東南西北中刻白不可為順，由後往前可以節省時間[明明只有 1-2 張，知道不可胡，卻從前面跑等於白跑])，如果為 1 就檢查是否能成為順，不能就傳回不可，可就繼續跑(index-1)／如果為 2 一樣檢查能不能為順，可就跑(index)[因為還剩一張]／如果為 3 就-3，index-1(繼續跑)／如果為 4 則檢查可不可成為順子，可就繼續跑(剩下 3 張，跑 index)，不可就當槓，跑(index-1)，其中在順子的判斷要檢查目前是不是有 kong==1 的狀況(也就是跑過手牌==4 的判斷式)
- 3) 實現上面的函式後，在輪流當門副函式中，如果手牌恰只有一組為對，則跑這組(如果他不能胡，必定不能胡，因為胡牌必須要有對才行)，當然如果她不當門，把有三張或四張的拿來當門，這該 tile 後面要有兩組順才行消掉，不過如果這樣的話，就不只一組為對了，因此超過一組為對，是有機會胡牌的，一樣跑遞迴
- 4) 如果牌數總和已經扣為 0，就回傳可以胡牌，index== -1 就跳出去
- 5) 就像圖上畫的，像是樹狀結構，在檢索能不能胡牌時是使用"深度優先搜尋法"(DFS)會比較快
- 6) 排序就用長度為 34 的陣列去存每個的張數，就能自然達成排序
 \rightarrow 輸入"\$ vim hw0503.c"，也可觀看詳細的註解

◎各情形範例

1) Ex: "1 2 3 4 5 6 7 8 9 12 12 12 31 31 31 34 34 -1"

Tip: test case #1

```
~yuchen0515@NTNUMATHLIN ~/程式設計hw5
~$ ./hw0503
Please enter the tile: 1
Please enter the tile: 2
Please enter the tile: 3
Please enter the tile: 4
Please enter the tile: 5
Please enter the tile: 6
Please enter the tile: 7
Please enter the tile: 8
Please enter the tile: 9
Please enter the tile: 12
Please enter the tile: 12
Please enter the tile: 12
Please enter the tile: 31
Please enter the tile: 31
Please enter the tile: 31
Please enter the tile: 34
Please enter the tile: 34
Please enter the tile: -1
Winning Hand.
```

2) Ex: "1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5 6 6"

Tip: 五槓+一對

```
Please enter the tile: 1
Please enter the tile: 1
Please enter the tile: 1
Please enter the tile: 1
Please enter the tile: 2
Please enter the tile: 2
Please enter the tile: 2
Please enter the tile: 2
Please enter the tile: 3
Please enter the tile: 3
Please enter the tile: 3
Please enter the tile: 3
Please enter the tile: 4
Please enter the tile: 4
Please enter the tile: 4
Please enter the tile: 4
Please enter the tile: 5
Please enter the tile: 5
Please enter the tile: 5
Please enter the tile: 5
Please enter the tile: 6
Please enter the tile: 6
Please enter the tile: -1
```

Winning Hand.

3) Ex: "1 1 2 2 3 3 4 4 5 5 6 6 13 13 13 18 18 18 34 34 -1"

Tip: 共 7 組對卻能形成胡牌的情況

```
Please enter the tile: 1
Please enter the tile: 1
Please enter the tile: 2
Please enter the tile: 2
Please enter the tile: 3
Please enter the tile: 3
Please enter the tile: 4
Please enter the tile: 4
Please enter the tile: 5
Please enter the tile: 5
Please enter the tile: 6
Please enter the tile: 6
Please enter the tile: 13
Please enter the tile: 13
Please enter the tile: 13
Please enter the tile: 18
Please enter the tile: 18
Please enter the tile: 18
Please enter the tile: 34
Please enter the tile: 34
Please enter the tile: -1
Winning Hand.
```


4) Ex: “1 2 3 4 5 6 -1”

Tip: 手牌數少於 17 必定不胡

```
Please enter the tile: 1
Please enter the tile: 2
Please enter the tile: 3
Please enter the tile: 4
Please enter the tile: 5
Please enter the tile: 6
Please enter the tile: -1
Not a Winning Hand.
```

5) Ex: “1 2 3 4 5 6 7 8 9 13 14 15 18 18 18 34 33 -1”

Tip: 雖然是 17 張，四順一刻，但沒有對

```
Please enter the tile: 1
Please enter the tile: 2
Please enter the tile: 3
Please enter the tile: 4
Please enter the tile: 5
Please enter the tile: 6
Please enter the tile: 7
Please enter the tile: 8
Please enter the tile: 9
Please enter the tile: 13
Please enter the tile: 14
Please enter the tile: 15
Please enter the tile: 18
Please enter the tile: 18
Please enter the tile: 18
Please enter the tile: 34
Please enter the tile: 33
Please enter the tile: -1
Not a Winning Hand.
```

說明

4 Marquee (20 pts)

Please write a Number Marquee program. You should let the user to input the time interval (unsigned integer). Figure 3 is an example.



- Remember that we have taught how to use `sleep`, right? Unfortunately, `sleep` needs seconds instead of micro-seconds. I will give you another tool called `usleep`.
- You can use `system("clear");` to clear your screen. I will teach you this function next semester.

◎**題意說明**→讓使用者輸入毫秒(單位)，代表每幾毫秒就跑一次(跑馬燈)，要做出跑馬燈的效果，並且往左遞移。

◎檢查

- 1) 毫秒輸入必須在[1,1000]內，否則題是重新輸入

◎輸入格式

```
1 $ ./hw0504
2 Please enter the time interval (1-1000 msec): 50
```

編譯後，執行”\$./hw0504”

首先，輸入毫秒數，若要退出跑馬燈請按 ctrl+c

◎輸出格式

- 1) 依照輸入的毫秒，進行跑馬燈

◎程式設計思路

- 1) 長度為 10 的陣列依序存 0 1 2 3 4 5 6 7 8 9
- 2) 因為要製作往左遞移的效果，用指標達到效果，每次傳入想要開始的記憶體位置，在函數內會依序把箭頭往後，直到到了陣列最末段，就將指標回到 arr 開始的位置，直到指標指向與原本的位置相同就退出
- 3) 每次跑都讓程式 usleep
- 4) 因為 printf 是採用 stdout 的方式，即輸出不會直接輸出，會進行一些處理再輸出，所以如果直接這樣跑迴圈，螢幕清掉後會一直沒顯示東西，過一陣子會突然跑出一大串，但如果使用換行，可以強迫他把 buffer 內的東西 dump 出來，因此可以達到效果，另若使用 fprintf 並用 stderr，則會讓數字直接輸出，因此不會有這樣的問題

◎各情形範例

- 1) 因是動態的效果，無法以截圖方式呈現
-

說明

5 Rational Number Arithmetic Operation (20 pts)

A rational number is maintained as a ratio of two integers, e.g. $\frac{1}{2}$, $\frac{31}{40}$. I believe that you know how to do rational number arithmetic operations, including addition, subtraction, multiplication and division. Please implement these four functions as follows:

Notes:

- All functions need to be implemented in another C file.
- The output must be in the **reduced form**.
- Some checks are undoubtedly necessary.

◎題意說明→輸入兩個分數，並輸出他們依序加減乘除的結果，並以最簡分數輸出

※請使用函式並以標頭檔實作

◎輸入格式

```
1 $ ./hw0505
2 Please enter 1st rational number: 1 2
3 Please enter 2nd rational number: 1 3
4 1/2 + 1/3 = 5/6
5 1/2 - 1/3 = 1/6
6 1/2 * 1/3 = 1/6
7 1/2 / 1/3 = 3/2
```

編譯後，執行"\$./hw0505"

依提示輸入一個模數(modulus)，再輸入一個整數(number)

◎檢查

1) 分母若為 0=>提示使用者重新輸入

◎輸出格式

1) 引出四個加減乘除的結果，若無法運算(分母為 0)就輸出不行

◎程式設計思路

- 1) 用輾轉相除法檢查，若分子分母的公因數不為 1，而且分子分母依序除以該公因數都能整除，則可以同除，變成最簡分數
- 2) 若為負數，則丟入 gcd 函數時，要丟入絕對值後的數值，出來後再乘以-1 即可，在分子為 0 的狀況時，要特別處理
- 3) 該題不難但較為繁瑣，詳細註解在鍵入"\$ vim hw0505.c"可以看到

◎各情形範例

1) Ex: "1 2 / 1 3"

Tip: test case #1

```
yuchen0515@NTNUMATHLIN ~/程式設計hw5
$ ./hw0505
Please enter 1st rational number: 1 2
Please enter 2st rational number: 1 3
1/2 + 1/3 = 5/6
1/2 - 1/3 = 1/6
1/2 * 1/3 = 1/6
1/2 / 1/3 = 3/2
```

2) Ex: "1 2 / 1 6"

Tip: 有理數為可以寫成分數的形式，所以在這 3/1 一樣輸出 3/1，不用轉為 3

```
Please enter 1st rational number: 1 2
Please enter 2st rational number: 1 6
1/2 + 1/6 = 2/3
1/2 - 1/6 = 1/3
1/2 * 1/6 = 1/12
1/2 / 1/6 = 3/1
```

3) Ex: "0 2 / 1 1"

Tip: 分子有 0 的情況(最簡分數的處理)

```
Please enter 1st rational number: 0 2
Please enter 2st rational number: 1 1
0/2 + 1/1 = 1/1
0/2 - 1/1 = -1/1
0/2 * 1/1 = 0/1
0/2 / 1/1 = 0/1
```

4) Ex: "0 1 / 0 2"

Tip: 分子都為 0(除法無法計算)

```
Please enter 1st rational number: 0 1
Please enter 2st rational number: 0 2
0/1 + 0/2 = 0/1
0/1 - 0/2 = 0/1
0/1 * 0/2 = 0/1
division don't operate since the denominator is zero
```

5) Ex: "0 2 / 4 8"

Tip: 輸入有分子為 0，第二個分數不為最簡分數的情況

```
Please enter 1st rational number: 0 2
Please enter 2st rational number: 4 8
0/2 + 4/8 = 1/2
0/2 - 4/8 = -1/2
0/2 * 4/8 = 0/1
0/2 / 4/8 = 0/1
```

6) Ex: “-8 -6 / -3 -4”

Tip: 輸入有負數，分子分母為-時轉為正，為負數則統一放在分子(如 $2/-2 \Rightarrow -2/2$)

```
Please enter 1st rational number: -8 -6
Please enter 2st rational number: -3 -4
-8/-6 + -3/-4 = 25/12
-8/-6 - -3/-4 = 7/12
-8/-6 * -3/-4 = 1/1
-8/-6 / -3/-4 = 16/9
```

說明

6 Bonus: What Happens? (10 pts)

Your teacher assign you a homework that you need to develop a program to make a user input 5 8-bits unsigned integers and print 5 number in the reverse order. For example, the user inputs 5 4 3 2 1 and your program should output 1 2 3 4 5. Now, some student write a code like this:

◎題意說明→

(1) 就如同之前我所要你做的，印出相反數的方式，現在下列程式碼也做一樣的事情，但出了點差錯，請閱讀以下程式碼，解釋他發生了什麼事：

```
1 #include <stdio.h>
2 #include <stdint.h>
3
4 int main()
5 {
6     uint8_t array[5] = {0};
7
8     for( int32_t i = 4; i >= 0 ; i-- )
9     {
10         printf( "Please enter a number: " );
11         scanf( "%u", &( array[i] ) );
```

```

12     }
13
14     for( int32_t i = 0 ; i < 5 ; i++ )
15     {
16         printf( "%u ", array[i] );
17     }
18
19     printf( "\n" );
20
21     return 0;
22 }

```

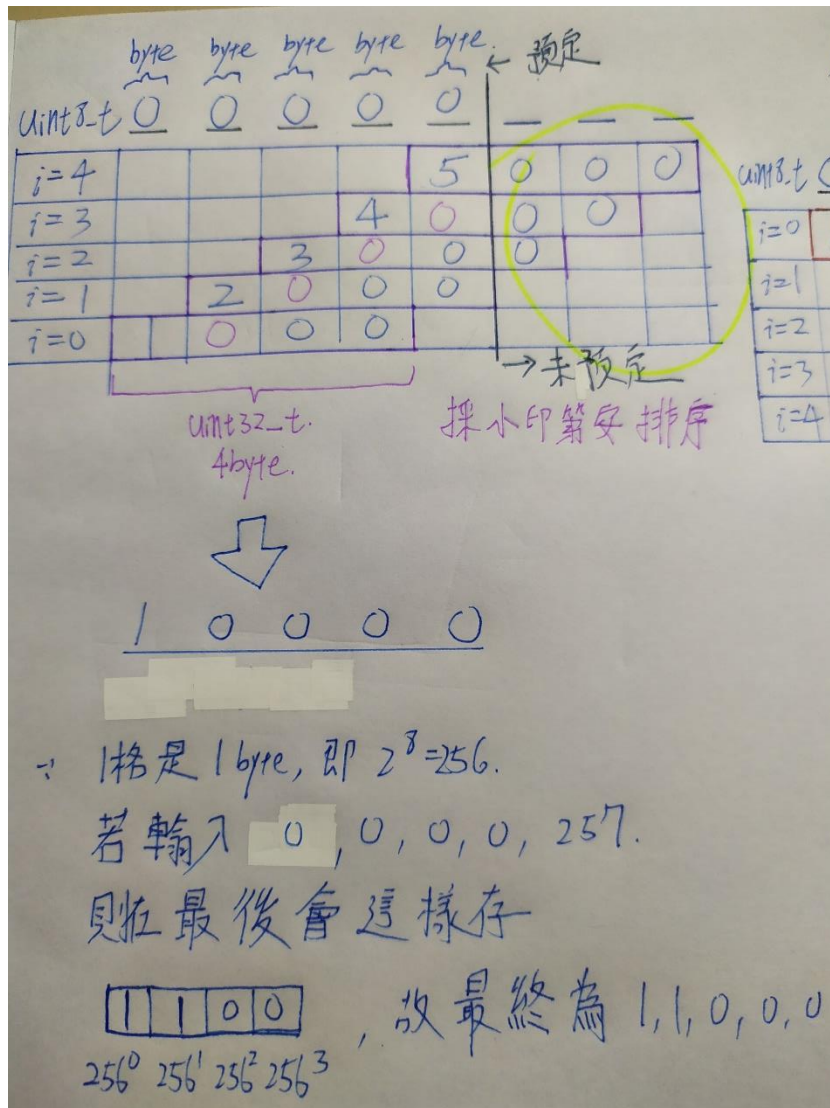
輸出應該為 5 4 3 2 1，卻為 1 0 0 0 0

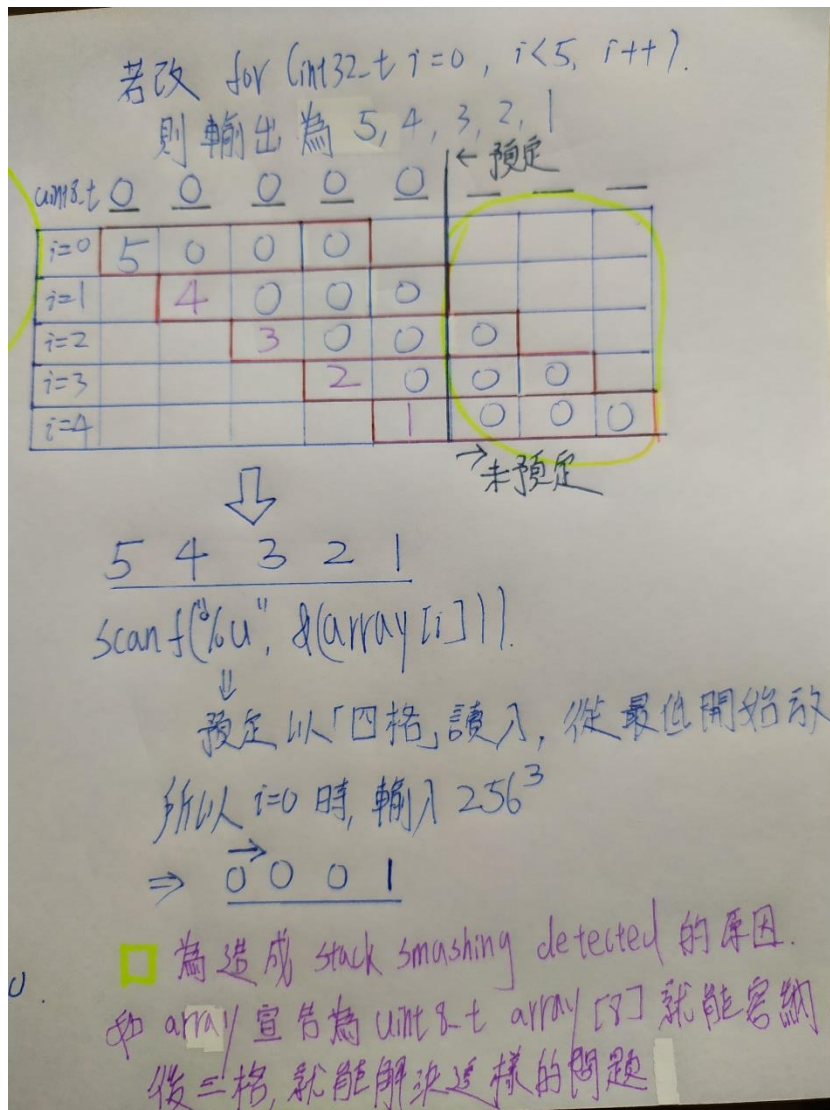
```

1 $ ./a.out
2 Please enter a number: 5
3 Please enter a number: 4
4 Please enter a number: 3
5 Please enter a number: 2
6 Please enter a number: 1
7 1 0 0 0 0

```

◎解釋





- 1) 在執行這隻程式時，除了建議你 scanf 從 %u 改 %hhu，還有 stack smashing detected，這種情況通常出現在原本你跟電腦要一塊記憶體，結果你卻使用到超過範圍的記憶體，就會發生
- 2) 當然詳細解釋如上圖，畫圖會比較清楚，而簡單概略則如下：
剛開始陣列是 uint8_t 的大小，每格 1byte，但輸入卻是以 uint32_t 的方式讀取，因此在讀的時候會用第一格+額外的三格去讀取他，電腦在讀數值時是採用”小印地安排序”的方式存取(組合語言有說明到，而網路則常常使用”大印地安排序”)，因此當你輸入 257 時，因為一格為 8bit=>2 的 8 次方為 256，但他讀取時是使用 4byte 去存，所以 257 因為超過 256 會進一位，剩下 1，因此存取時依序是 1 1 0 0 0

當然，剛開始我們是從 i=4 跑，所以當你輸入數值時，會一直跑到該記憶體+3byte 的位置，因此會超出原先預定的位置，導致系統有 stack smashing，因此如果宣告從 uint8_t array[5] 改成 uint8_t array[8]，就能把多

的 3byte 存下來，因此可避免掉這個錯誤如圖：

```
r-yuchen0515@NTNUMATHLIN ~/程式設計hw5
└─$ ./hw0506
Please enter a number: 5
Please enter a number: 4
Please enter a number: 3
Please enter a number: 2
Please enter a number: 1
1 0 0 0 0
```

原本的狀況如圖：

```
Please enter a number: 5
Please enter a number: 4
Please enter a number: 3
Please enter a number: 2
Please enter a number: 1
1 0 0 0 0
*** stack smashing detected ***: <unknown> terminated
[1] 1275 abort (core dumped) ./hw0506
```

剛開始輸入的 5，會從陣列最末的記憶體位置存 5 0 0 0 進去，而後面的 0 0 0 則依序存在陣列最末段記憶體+1,2,3bytes 的位置(因此造成 stack smashing detected)，而之後輸入的 4，則在陣列倒數第二的位置存 4 0 0 0 進去，其中後面三個 0 0 0 依序蓋掉原本的 5 0 0，之後依此類推，到輸入 1 時，存 1 0 0 0 進去，最終變成 1 0 0 0 0 (如手寫的圖 1)

當然，如果熟稔這樣的原理，我輸入 5 4 3 2 257，輸出就為 1 1 0 0 0，如果我把原本 scanf 的迴圈從 0 到 4，輸入 5 4 3 2 1，那麼會輸出 5 4 3 2 1，因為 5 0 0 0，4 0 0 0，3 0 0 0，2 0 0 0，1 0 0 0，剛好不會蓋到存的數值(個位數，如果你放 256 以上的數字還是會被蓋到)，但在 1 0 0 0 的地方因為又超出預定的空間了，他一樣會 stack smashing。

最好的辦法當然是不要這樣寫，要用 uint8_t 就用 %hhu 去存，不過熟悉原理也是好事。