

# Computer Programming Final

## 程式設計期末考

Teacher: Po-Wen Chi 紀博文

2019.1.9

- You cannot access Internet when exam. After you complete the exam, please raise your hand and TA will plug the cable for you and submit your codes to moodle.  
考試過程中不可以連到網際網路。考試結束後請舉手，由助教幫你接上網路線，並上傳你的程式碼到 moodle。
- You file should be [id]\_final.zip .  
你的檔案應該要命名為 學號\_final.zip
- Please do not change the file name.  
請不要更改檔案的名稱。
- A Makefile is still required or I will not know which library you are using. test.c will be written by TA to test your code. If there is no Makefile or Makefile does not work, -20.  
你還是要提供一個 Makefile，畢竟我不知道你會使用哪個函式庫。test.c 會由助教撰寫來測試你的函式庫。如果沒有 Makefile 或是 Makefile 有問題的話，扣 20 分。
- I provide you the library template. You cannot change the function definition.  
我有提供你函式庫的範本，請不要更改函式的定義。
- If there is a problem you do not know how to write, please do not modify the template or make will be failed.  
如果有不會寫的題目，請不要改題目的範本。不然會 make 會有問題。
- In my opinion, problem 3.3 is the most difficult one. So you may not waste too much time on this problem.  
就我的意見，我認為 Problem 3.3 是最難的。所以也許你不應該花太多時間在這題上面。

## 1 Matrix Operation (25 pts)

This problem asks you to implement the following matrix operations:  
你需要實做下列的矩陣運算功能：

1. Addition. 加法 (5 pts)
2. Subtraction. 減法 (5 pts)
3. Multiplication. 乘法 (5 pts) 。  
If these two metrics cannot be multiplied, return -1 。  
如果這兩個矩陣是沒有辦法乘的話，回傳-1 。
4. Transpose. 轉置 (10 pts)

As talked in the class, when passing two dimensional array to a function, you need to make the row size clear. It is not convenient for users. So in this problem, I ask you to use one dimensional array to simulate a two dimensional array. The simulation way is as follows.

正如我們在課堂上提到的，當你把二維陣列輸入到一個函式裡面，你必須要給定每一行的大小。這對使用者來說不是很方便。所以這一題我要求你用一維陣列來模擬二維陣列。模擬的方式如下：

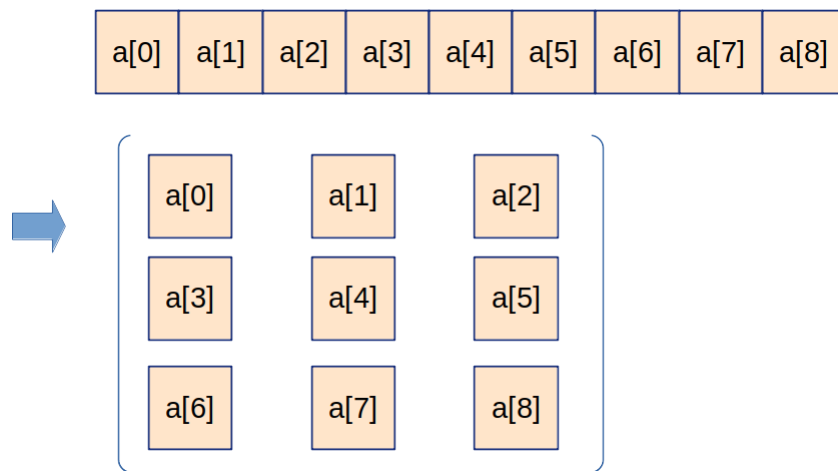


Figure 1: Use one dimensional array to simulate 2D array.

Hint 提示：

- You can write a function to transform the location of two dimensional arrays into the location of one dimensional array.  
你可以寫一個函式來將二維陣列的位置轉成一維陣列的位置。

## 2 Sorting Two Arrays (25 pts)

In this class, we have implemented a program to sort an array. Now give you two arrays, please combine them into one array and sort the array. Note, the given two arrays are not sorted. The two arrays' sizes may be different.

在課堂上，我們曾經寫過一隻程式來排序一個陣列。現在給你兩個陣列，請你把兩個陣列合成一個，並將這個陣列進行排序。請注意，給你的兩個陣列並沒有經過排序。這兩個陣列的大小可能是不一樣的。

1. I will prepare an output array for you. Do not worry! I promise the output size array is enough. Please sort the array in the descending order. (10 pts)  
我會準備一個輸出的陣列給你。別擔心，我保證輸出陣列的大小是足夠的。請由大排到小。
2. This time, you need to prepare an output array yourself. Please sort the array in the ascending order. (10 pts)  
這一次你要自己準備輸出的陣列。請由小排到大。
3. Remember your teacher likes to play poker game. Many poker games have the concept of trumps. Now we define trumps as numbers that are multiples of thirteen. Trumps are larger than other numbers. For example,  $13 > 14, 26 > 13$ . Please sort the array in the descending order. (5 pts)  
記得你的老師很喜歡玩撲克牌嗎？很多撲克牌的遊戲都有「王牌」的概念。現在我們定義王牌是 13 的倍數，王牌比其他數字都要大。舉例來說， $13 > 14, 26 > 13$ 。請將陣列由大排到小。

I know some of you already know how to use `qsort`. In this exam, **qsort is not allowed**. 我知道你們當中有些人已經知道如何使用 `qsort`，但在這次考試裡面，**不準使用 `qsort`**。

### 3 String to Number (25 pts)

We often use computers to do mathematical operations for us. However, user inputs are often be strings instead of numbers. Please write functions to transform a string to a integer. Note that in this problem, we only use unsigned integers which means that you do not need to consider the negative integers.

我們很常利用電腦來做數學運算，然而使用者的輸入往往都是字串而非數字。請實做一些函式來將字串轉成整數。請注意在這一題當中，測試的資料都會是正整數，也就是你不需要考慮負數的情況。

1. Please transform a string to an unsigned 32-bit integer. Do not worry! The test case will not over the size of 32-bit. (8 pts)  
請將一個字串轉成一個 32 位元的無號數。別擔心，測資裡面不會有大於 32 位元無號數的狀況。
2. Please transform a string to an unsigned 64-bit integer. Do not worry! The test case will not over the size of 64-bit. (8 pts)  
請將一個字串轉成一個 64 位元的無號數。別擔心，測資裡面不會有大於 64 位元無號數的狀況。
3. Please transform a string to an unsigned 512-bit integer. Oops, there is no type called `uint512_t` in C. So, we use `uint32_t [16]` to present this unsigned integer. The order is little endian. You can see an example in figure 2. Do not worry! The test case will not over the size of 512-bit (9 pts)  
請將一個字串轉成一個 512 位元的無號數。糟了，在 C 裡面我們沒有 `uint512_t`。

所以我們使用 `uint32_t` [16] 來代表這個無號數。排列順序為 little endian。圖 2 是一個範例。別擔心，測資裡面不會有大於 512 位元無號數的狀況。

- Bonus: If the input string is greater than 512-bit unsigned integer size, please return -1 (5 pts).  
加分題：如果輸入的字串的值大於 512-bit 無號整數，請回傳-1。

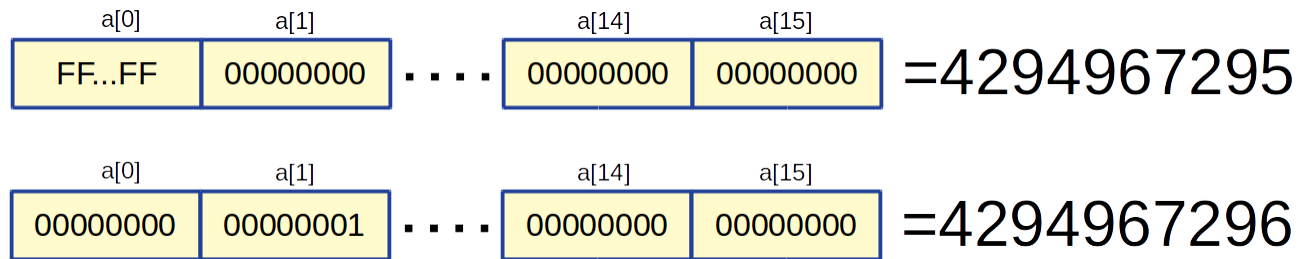


Figure 2: The above figure is an example of 255 and the below figure is an example of 256.

Note:

- Actually, C standard library has some functions can do the previous two works, like `atoi`, `sscanf`, `strtol`. In normal cases, you should use them directly instead of develop your own functions. However, this is an exam. So **you are not allowed to use existing functions**.  
事實上，C 的標準函式庫有一些函式可以作到前兩題的工作，像是 `atoi`, `sscanf`, `strtol` 等。一般來說，你應該直接使用它們而不是自己開發。然而，這是考試，你不准用這些現成的函式。
- You can use calculator in your computer to help you to verify your answer.  
你可以使用電腦的計算機來驗證你的答案。

## 4 Sudoku Solver (25 pts)

You know how to play sudoku, right? If no, let's me explain this game. Sudoku is a logic puzzle in which you are given a  $9 \times 9$  square of numbers, divided into rows, columns, and 9 separate  $3 \times 3$  sectors. In each row, column, and sector, the numbers 1-9 must appear. At the start of the puzzle, just enough of the grid's elements are filled in to guarantee only one unique solution based on these rules.

你知道如何玩數獨，對吧？如果不知道的話，讓我稍微解釋一下。數獨是一種數學邏輯遊戲，遊戲由  $9 \times 9$  個格子組成，每個格字的數字是由 1-9，玩家需要根據格子提供的數字推理出其他格子的數字。一個數字在同一行、同一列、自己所在的九宮格式不能重複的。在遊戲開始前，會提供你一些初始的數字供你推論，遊戲保證答案是唯一的。

It sounds hard, right? So I will give you some helps (I am a good guy). First, I prepare a function called **isValid**. This function is used to check if a number in some location with current matrix is valid. Second, I will give you a hint. You can use a recursive function to solve this problem. Of course, you can use your own approach.

聽起來很難，是嗎？所以我決定給你們一些幫助（我真是個好人）。第一。我準備了一個函式叫做 **isValid**，這隻函式可以幫你檢查在目前的矩陣下，某個數字放在特定的位置是否合法。第二，我要給你們一些提示，你可以用遞迴的函式來解決它。當然，你可以使用自己的解決方法。

For your reference, general sudoku problem is a NP-complete problem, which means it is very hard. You will learn the formal definition in other classes. You may ask why you can solve such a hard problem by hand. The reason is that the matrix is small.

給你一個參考，數獨這個遊戲的一般化形式的難度是 NP-complete，代表它非常難。正式的定義你在其他課程會學到。你可能會問說根本不難，我用手就可以計算了，那是因為你解決的數獨問題盤面太小了。

## 5 Bonus: Binary Complement (5 pts)

Given a 32-bit integer  $n$ , undoubtedly you know how to get it's binary form, right? Please write a function to output  $n$ 's binary complement. For example, if an integer is  $11100101_2$ , you should output an integer which has a binary form of  $00011010_2$ . That is, the output should be 26. Note the output type is **int32\_t**.

給定一個 32 位元的整數  $n$ ，毫無疑問你應該知道如何用二進未來表示它，對嗎？請寫一個函式來輸出  $n$  的二進位補數。舉例來說，有一個整數是  $11100101_2$ ，你應該輸出一個整數，它的二進位表示式是  $00011010_2$ ，也就是 26。輸出的格式為 **int32\_t**。