

## README

40947007s 張懷齡

### Hw0501

印出兩個陣列，一個是原本的，另一個是奇數在前，由小排到大，偶數在後，由大排到小。另外寫一個函式 `sorting` 放在 `sorting.c`，傳入 `array` 以及 `array_size`，先利用迴圈及取 2 的餘數來計算奇數與偶數個別有幾個，接下來分別設一個奇數陣列及一個偶數陣列，將傳入的陣列為奇數的數利用泡泡排序法由小到大存入新的奇數陣列，接著再印出來。再利用一次泡泡排序法，將傳入的陣列為偶數的數由大到小存入新的偶數陣列，接著再印出來。

執行：`make`

`./hw0501`

### Hw0502

矩陣乘法，輸入一個二維陣列，程式會印出這個二維陣列，及乘出來後的結果，另外寫一個函式 `determinant` 放在 `determinant.c`，傳入二維陣列，將公式帶入，即可算出答案。

執行：`make`

`./hw0502`

### Hw0503

輸入兩個多項式的次方、係數，會印出這兩個多項式以及相加、相減、相乘後的結果。另外寫了四個函式 `printPoly`、`addition`、`subtraction`、`multiplication` 放在 `polynomial.c`，`printPoly` 的功能是印出方程式，先處理正負號，如果係數是正的就印出 `+`，如果係數是負的就印出 `-`，因為前面已經處理過正負號，所以後面的係數印出來時都要加上絕對值，否則負的會有兩個負號，利用 `do while` 迴圈讓方程式隨著次方遞減印出。

`addition` 的功能是多項式的加法，傳入兩個矩陣，及矩陣的大小 `p1,p2`，分成三種情形，`p1>p2`、`p1<p2`、`p1=p2`，根據不同情形，新設的陣列 `add_pol` 的大小會有所不同，設完陣列後，先將傳入的陣列次方相同的相加並存入新的陣列 `add_pol`，再把較高次方沒有被加到的數字存入新的陣列 `add_pol`，陣列的索引值要處理好否則會覆蓋到。接著再呼叫 `printPoly` 把新的陣列 `add_pol` 印出來。

`subtraction` 的功能是多項式的減法，傳入兩個矩陣，及矩陣的大小 `p1,p2`，分成三種情形，`p1>p2`、`p1<p2`、`p1=p2`，根據不同情形，新設的陣列 `sub_pol` 的大小會有所不同，設完陣列後，先將傳入的陣列次方相同的相減並存入新的陣列 `sub_pol`，再把較高次方沒有被減到的數字存入新的陣列 `sub_pol`，陣列的索引值要處理好否則會覆蓋到。接著再呼叫 `printPoly` 把新的陣列 `sub_pol` 印出來。

`multiplication` 的功能是多項式的乘法，傳入兩個矩陣，及矩陣的大小 `p1,p2`，分

成三種情形， $p1 > p2$ 、 $p1 < p2$ 、 $p1 = p2$ ，新設的陣列 `mul_pol` 大小都是  $p1 + p2 - 1$  利用兩層 `for` 迴圈讓相同次方的係數相加存入新的陣列 `mul_pol`，接著再呼叫 `printPoly` 把新的陣列 `mul_pol` 印出來。

執行：`make`

`./hw0503`

#### Hw0504

讓使用者給定幾個點及幾個點的值，算出回歸直線方程式，設一個變數 `point` 存總共有幾個點，再設兩個陣列分別是 `coordinate_x` 存座標的  $x$  值，及 `coordinate_y` 存座標的  $y$  值，另外寫了一個函式 `linear` 放在 `linear.c`，傳入兩個陣列 `coordinate_x`、`coordinate_y` 和 `point`，假設方程式為  $y = bx + a$ ，先利用公式算出  $b$ ，再利用回歸直線必通過平均值的特點帶入  $y = bx + a$  求出  $a$ ，最後再印出方程式。

執行：`make`

`./hw0504`

#### Hw0505

踩地雷，在 `minesweeper.c` 寫了六個函數，`initiliseMine` 是要初始化 `isMine` 以及處理隨機產生的炸彈要放在哪個位置。`open` 是處理當使用者輸入打開的指令時，要把那格位置變成 `1`，這裡打開的時候會用到類似遞迴的概念，呼叫自己。`printMap` 是要印出地圖的函式。`initiliseFlag` 是初始化 `isFlaged`。

`initiliseNumbers` 是要初始化 `numberMine`。`initiliseOpens` 是要初始化 `isOpen`。

我在主函式設了一個無窮迴圈，要讓使用者選擇是要 `open` 還是 `flag`，根據輸入不同的選擇會需要不同的函式來處理，在這裡我設定獲勝的條件為，當我豎立的旗子和炸彈一樣多，就獲勝，並且印出地圖，遊戲結束。失敗的條件為開到炸彈，此時也會印出地圖，但這時印出的地圖只會顯示出哪裡有炸彈。

執行：`make`

`./hw0505`