

# README

資工 113 40947007s 張懷齡

## Hw0601

給一個 64 位元的整數，讓 8 個 byte 分別被印出來，再讓使用者輸入位置編輯成別的數字。

先設一指標 `ptr` 存入使用者輸入整數的位置，再利用 `for` 迴圈利用 `*(ptr + i)` 把 8 個 byte 的位置以十六進位印出來，接著設一個無窮迴圈，在無窮迴圈裡先讓使用者輸入位置(`position`)，在使用者輸入為 0 的時候才會跳出來，接著讓使用者輸入那個位置所要更改的值 (`newValue`)，接著把新的值利用指標更改 `*(ptr + position - 1) = newValue`，最後再利用 `for` 迴圈印出新的位置。

執行: `make`

`./hw0601`

## Hw0602

處理上一步下一步的問題，再主函式設一個 `while` 迴圈讓使用者輸入數字，每輸入一個數字就會傳入 `redoAndundo`，另外寫兩個函式 `full` 及 `redoAndundo` 放在 `redoAndUndo.c`，`full` 沒有傳入值，功能是當陣列存入的值大於十個，就要把所有的數往前移一個，`redoAndundo` 傳入的是使用者每次輸入的數字，分成四種情況，情

況一：當輸入為 0 的時候，程式會印出結果並結束，情況二：當輸入為-1 且陣列 `save` 裡是有東西的時候，會將-1 前的那個數字存入另外一個 `undo` 的陣列裡，則此時 `save_index` 會-1，`undo_index` 會+1，情況三：當輸入為-2 且 `undo` 陣列裡是有東西的時候，會把 `undo` 裡的數字再丟回 `save` 陣列，情況四：當輸入的不是 0、-1、-2 時，會先呼叫 `full` 函數，確認陣列是否已滿，並把 `undo` 陣列裡的數字清空，設為 0，`undo_index` 也設為 0，最後把使用者輸入的數字存入 `save` 陣列。

執行: `make`

`./hw0602`

### Hw0603

旋轉角度求更改後的座標，另外寫一個函式 `rotate` 放在 `rotation.c`，傳入 `x` 的位置，`y` 的位置，以及旋轉的角度，利用指標指到 `x` 與 `y` 所在的位置利用公式更改旋轉後的值，最後在主函式印出 `x` 與 `y` 分別為多少。

執行: `make`

`./hw0603`

### Hw0604

印出平面方程式，再讓使用者輸入點座標，最後印出投影在平面的座標。另外寫一個函式 `project` 放在 `projection.c`，傳入 `x` 的位置，`y` 的位置，`z` 的位置，以及平面方程式的四個係數，利用投影點的公式算出答案，分別存入指標指到 `x`、`y`、`z` 的位置，最後在主函式印出答案。

執行: `make`

`./hw0604`

#### Hw0605

利用三種不同的排法來排手排，第一種是 **A** 最大，第二種是 **2** 最大，第三種是先按照花色排，花色相同再以 **A** 最大來排。

另外寫兩個函式 `print_card` 及 `sort_card` 放在 `poker.c`，`print_card` 傳入手排的陣列，功能是利用 `for` 迴圈用來印出手排的，`sort_card` 傳入手排陣列，及一個 `function pointer`，功能是利用泡泡排序法交換手排，接著在主函式寫三個函式 `func01`、`func02`、`func03`，分別代表三種排法，每個函式都會傳入兩個數值來比較，`func01` 的寫法是，先將傳入的兩個數字取 **13** 的餘數，當餘數等於 **1** 的時候，把值設為 **14**，這樣在比較的時候排面為 **A** 的手排就會最大，`func02` 的寫法和 `func01` 相同，但當餘數等於 **2** 的時候要把值設為 **15**，這樣排面為 **2** 的手排就會最大。而 `func03` 的寫法是設兩個變數 `a_suit`、`b_suit`

來代表兩張牌的花色，當數字為 1~13 時，`a_suit` 設為 4，當數字為 14~26 時，`a_suit` 設為 3，當數字為 27~39 時，`a_suit` 設為 2，當數字為 40~52 時，`a_suit` 設為 1，如此一來，手排就會按照花色大小排列，`b_suit` 也是。接下來，當 `a_suit = b_suit` 時，再用 `func01` 方法比數字大小，最後再利用 `print_card` 把手排印出來。

執行: `make`

`./hw0605`