

作業題目:

- 1.因應使用者需求能發給使用者 1~13 張牌，並可以重發，發出來牌需擁有其花色與數字兩項屬性且發出來的所有牌不得重複
- 2.在使用者確定不要重發後詢問使用者是否列出使用者所有的出牌順序出牌順序不考慮套牌的出法，一張一張的出，當總發牌數大於 6 張時，能詢問使用者，出牌順序總量繁多，是否要繼續陳列。

前置想法:

撲克牌總共為 52 張牌，所以應該事先把整副牌用陣列的方式模擬出來接者使用隨機的方式來洗牌，再利用判斷式和迴圈來完成題目要求。

作法:

I. 建立撲克牌排組:

先用定 `typedef struct` 定義 `Card` 的結構體，它有 `rank`、`suit` 兩個變數，前者用於表示數字；後者則為花色，透過這個結構體，可以創建表示撲克牌的對象，每個對象包含一個數字和一個花色，以便在程序中進行操作和處理。接著使用函數並用 `*deck` 作為 `Card` 的指針參數，利用兩個變數，一個為卡牌儲存地方，另

一個為儲存數字再搭配迴圈來為每個數字配上花色。所以

deck[0]是 S1、deck[1]是 H1、deck[2]是 D1、deck[3]是 C1 以此類推，就完成了一副牌。

II. 洗牌

聲明變量 `i` 和 `j`，分別用於迴圈和隨機選擇元素的索引。

聲明一個臨時變量 `temp`，用於交換牌組中的元素。並使用

`srand(time(NULL))` 初始化隨機數生成器，確保每次運行程序都會生成不同的隨機數序列。從最后一張牌開始，即索引為

`MAX_CARDS - 1`，遞減循環變量 `52` 直到 `1`。在每次迴圈中，生成一個隨機索引 `j`，範圍是從 `0` 到 `52`（包含 `52`）之間的隨機

數。將 `i` 的牌與 `j` 的牌進行交換，通過臨時變量 `temp` 進行交換操作。實現洗牌的效果

III. 發牌本身結構:

設置牌數為 `52` 張卡，並加入兩個判斷用的變數，和一個使用者所需的卡牌變數，一開始詢問使用者所需卡牌數量，因為要求卡牌需要一張一張發所以我用了按下 `enter` 鍵會發出一張牌且當超過 `6` 張牌時，利用判斷式來判斷是否繼續列出牌，此時第一個判斷變數判斷如果繼續則再按下 `enter` 鍵會發牌，如果不發牌

會詢問是否要重新洗牌，這時會利用到第二個判斷式，如果確定則重新洗牌，並再次詢問要發出多少牌數，如果否定則退出整個程式。裡面使用了 **while** 迴圈的方式來讓使用者只能輸入 1~13 的數字。

問題:

遊戲本身的判斷式和迴圈為了能迎合題目程式的判斷式所以寫得有點亂，且為了防止使用者輸入錯誤的值必須將判斷是寫的冗長，且再選取發牌數時當輸入完畢會按下 **enter** 鍵，這時會進入發牌的迴圈裡面，而剛才所按下的 **enter** 鍵就算第一次，所以會馬上發牌。

延伸:

因為題目本身為發牌遊戲所以我想到的延伸遊戲為 21 點、猜大小、抽鬼牌或者大老二

- I. 21 點:按下 **enter** 鍵則會發牌，並用一個變數計算抽取卡片的數字，且每次抽取石階會詢問是否繼續抽卡，再用判斷式來表示是否超過 21 點，或者抽到超過 5 張牌。
- II. 猜大小:也是使用和 21 點同樣的方法，但變數要刷新且判斷式以是否超過 7 來辨別大小，，用一個變數來紀錄對錯當猜

完 52 張牌會告訴使用者猜對多少題。

- III. 抽鬼牌與大老二:要將牌組分成四組或三組，我覺得可以利用上學期計概所講的 **client sever** 來連線，**sever** 負責將牌分成跟 **client** 一樣數量的牌組。當 **client** 端的玩家選取要出的牌，此時回傳到 **sever** 就會將牌組內的此儲存格清空，並透過判斷式來判斷玩家是否能出他選取的卡牌，且誰的牌組最線被清空就取得勝利並重新開始遊戲，但是尚未使用過如次複雜的程式加上 **client sever** 的方法也只有概念，所以上述所想的也只是大智的架構，大老二裡遊戲機制的問題也有實現的困難點。