程式設計(二)期末專題
--十三支企劃書

組員名單：

資傳1B　1072005　何信億

目錄

一、玩法概念

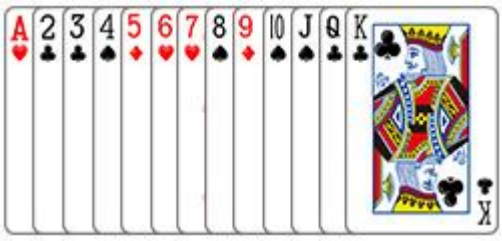
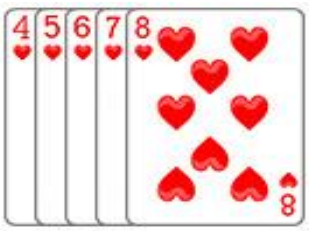
1.1 十三支介紹

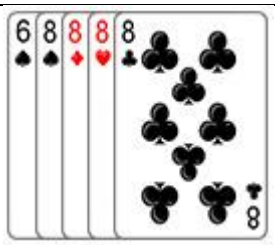
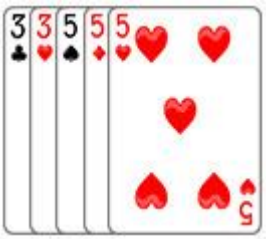
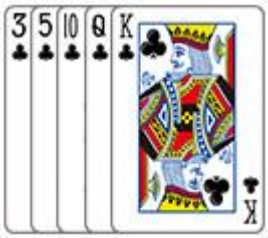
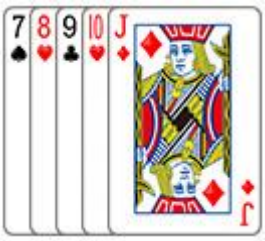
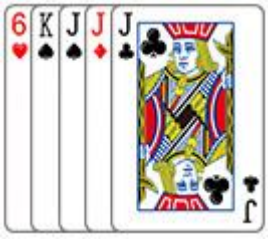
十三張，又稱十三支，是一種撲克遊戲。進行 13 支遊戲時，共有四位玩家互相對抗。每位玩家在發牌後會拿到 13 張牌，玩家可以依照自己的策略，把牌張分成前墩（3 張）、中墩（5 張）及後墩（5 張）。接著四家開牌，比較所有玩家之間的勝負關係。

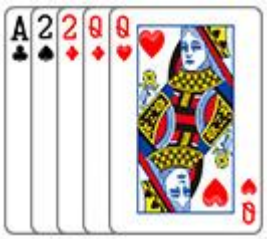
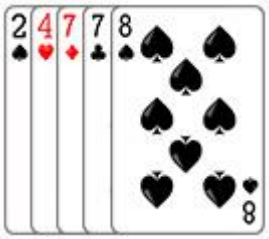
1.2 十三支玩法

1.牌型大小： 一條龍＞同花順＞鐵枝＞葫蘆＞同花＞順子＞三條＞兩對＞對子＞散牌

2.數字大小： A＞K＞Q＞J＞10＞9＞8＞7＞6＞5＞4＞3＞2

例圖	牌型組合	說明
	一條龍	玩家手牌中持有 A、2、3、4、5、6、7、8、9、10、J、Q、K 各一張。
	同花順	五張牌組成是順子且為同花色。

	鐵支	四張同樣數字的牌外加任一單張，以四張數字相同的牌為比較基準。
	葫蘆	三張數字相同的牌加兩張數字相同的牌，以三張相同數字的牌為比較基準。
	同花	五張牌為同一花色但數字順序並不連續。若兩位以上的玩家同一墩為同花，該墩數字較大者獲勝。
	順子	五張數字相連但不同花色。若兩位以上的玩家同一墩為順子，該墩數字較大者獲勝。
	三條	三張相同數字的牌加上兩張互不相同的牌。若兩位以上的玩家同一墩為三條，則三條數字較大者獲勝。

	兩對	兩個不同的對子加上任一單張。若兩位以上的玩家同一墩為兩對，則該墩有較大對子者獲勝；若兩位玩家該墩的較大對子相同，則比較次大的對子。
	對子	兩張相同數字的牌。若兩位以上的玩家同一墩為一對，則該墩有較大對子者獲勝；若兩位玩家該墩的對子相同，則比較第三張牌，有最大單張者獲勝。
	單張	以上所有牌型以外的單張牌，若兩位以上玩家同一墩為單張，則比較該墩最大的牌；如果玩家最大的牌相同，則比較次大牌。

1.3 邏輯思考(玩法)

1. 先看好牌， 然後思考自己的牌型。

2. 前墩要小於中墩要小於後墩。

二、對戰規則

2.1 說明想法(作法)

1.依序找出鐵支、葫蘆…，從大的找到小的排型，找到就先塞入後墩

2.最後將沒有 0 的地方，都塞單張進去。

2.2 說明規範

1. 後面的牌不能比前面的小

2. 只看牌型及數字大小，不看花色。

3. 牌型:

   5 張:鐵支>>葫蘆>>順子>>三條>>2pairs>>1pair>>雜牌

   3 張:3 條>>1pair>>雜牌

2.3 說明細節(程式)

```
int icard[13] = { 0 };//卡牌號碼(原始1~52)
int ivector[13] = { 0 };//卡牌號碼(轉換後)
int front[3] = { 0 };//(頭墩)
int mid[5] = { 0 };//(中墩)
int back[5] = { 0 };//(尾墩)
int finaly[13];//最終回傳的牌

//分類為 單張 對子 三條 鐵支(四張同數)
//開頭加X為原始號碼1~52
int a[13] = { 0 };//記傳來的陣列
int along[13] = { 0 };//記單張
int Xalong[13] = { 0 };//記原始單張

int pair1[13] = { 0 };//記對子
int Xpair[13] = { 0 };//記原始對子

int three[13] = { 0 };//記三條
int Xthree[13] = { 0 };//記原始三條

int four[13] = { 0 };//記鐵之四張
int Xfour[13] = { 0 };//記鐵之的四張
```

一開始先宣告記錄單張、對子、三條、鐵之、前三張(前墩)、中間五張、後面五張的陣列，前面有加 X 名稱的陣列為記 1-52 的數字，而沒加則是轉換成 1-13 的數字。

```
//ivector % 13 轉換為 1~13
for (int i = 0; i < 13; i++) {
    ivector[i] = ivector[i] % 13;
    if (ivector[i] == 0)
        ivector[i] = 13;
}
for (int i = 12; i >= 0; i--) {
    for (int j = 0; j < i; j++) {
        if (ivector[j] > ivector[j + 1]) {
            //ivector排序
            int change;
            change = ivector[j];
            ivector[j] = ivector[j + 1];
            ivector[j + 1] = change;
            //icard跟著ivector排序
            change = icard[j];
            icard[j] = icard[j + 1];
            icard[j + 1] = change;
        }
```

這邊將 ivector 陣列裡的數字轉換成 1-13，在用泡沫排序法將原本 1-52 跟 1-13 的陣列都由小排到大。

```
//找出雜牌紀錄到along &Xalong
int n = 0;
if (ivector[0] != ivector[1]) {
    along[n] = ivector[0];
    Xalong[n] = icard[0];
    n++;
}
for (int i = 1; i < 13; i++) {
    if ((ivector[i] != ivector[i + 1]) && (ivector[i] != ivector[i - 1])) {
        along[n] = ivector[i];
        Xalong[n] = icard[i];
        n++;
    }
}
```

這邊記錄雜牌，如果找到雜牌 n 就+1。

```
//找鐵支(四張相同)記錄到  four & Xfour
n = 0;
for (int i = 0; i < 13; i++) {
    if (ivector[i] == ivector[i + 1] && ivector[i + 1] == ivector[i + 2] && ivector[i + 2] == ivector[i + 3]) {
        for (int j = 0; j < 4; j++) {
            four[n] = (ivector[i + j]);
            Xfour[n] = (icard[i + j]);
            //紀錄後值改為0
            ivector[i + j] = 0;
            n++;
        }

    }
}
```

這邊是找鐵支，如果四個相同就找出來。

```
//找三條(三張相同)記錄到 three & Xthree
n = 0;
for (int i = 0; i < 13; i++)
{
    if (ivector[i] == ivector[i + 1] && ivector[i + 1] == ivector[i + 2] && ivector[i + 2] != ivector[i + 3]) {
        if (ivector[i] != 0) {
            for (int j = 0; j < 3; j++) {
                three[n] = (ivector[i + j]);
                Xthree[n] = (icard[i + j]);
                //紀錄後值改為0
                ivector[i + j] = 0;
                n++;
            }
        }
    }
}
//找對子
n = 0;
for (int i = 0; i < 13; i++) {
    if (ivector[i] == ivector[i + 1]) {
        if (ivector[i] != 0) { //鐵支與三條已讀取的值已改為0 若符合條件但為0者便不被記錄下
            for (int j = 0; j < 2; j++) {
                pairl[n] = (ivector[i + j]);
                Xpair[n] = (icard[i + j]);
                n++;
            }
        }
    }
}
```

這邊則是找三條跟對子，用的方法跟鐵知相同。

```
int s = 0;//紀錄單張拿到第幾張
int p = 0; //紀錄對子拿到第幾張

//若兩份鐵支 拿第二支 放後墩;
if (Xfour[4] > 0) {
    for (int i = 0; i < 4; i++) {
        back[i] = Xfour[i + 4];
        Xfour[i + 4] = 0;
    }
    if (Xalong[s] > 0) {//放單張到後墩第五張
        back[4] = Xalong[s];
        Xalong[s] = 0;
        s++;
    }
    else {
        back[4] = Xpair[p];//沒有單張 放對子到後墩第五張
        Xpair[p] = 0;
        p++;
    }
}
//若兩份鐵支或只有一支 拿第一支 ;
if (Xfour[0] != 0) {
    if (back[0] == 0) {//判斷尾墩為空 放入

        for (int i = 0; i < 4; i++) {
            back[i] = Xfour[i];
            Xfour[i] = 0;
        }
        if (Xalong[s] != 0) {
            back[4] = Xalong[s];
            Xalong[s] = 0;
            s++;
        }
        else {
            back[4] = Xpair[p];
            Xpair[p] = 0;
            p++;
        }

    }
```

這邊記錄若有兩份鐵隻的話，就先拿第二副鐵支放到第三墩，在放單張到第三墩的第五張，沒單張則放對子到第五張。在拿第一副鐵支方法相同。

```
//若兩份葫蘆 拿第二支;
if (Xthree[3] > 0 && Xpair[3] > 0)
{
    if (back[0] == 0) {//判斷尾墩為空 放入
        for (int i = 0; i < 3; i++) {
            back[i] = Xthree[i + 3];
            Xthree[i + 3] = 0;
        }
        for (int i = 0; i < 2; i++) {
            back[i + 3] = Xpair[i + 3];
            Xpair[i + 3] = 0;
        }
    }

    else if (mid[0] == 0) {//判斷中墩為空 放入
        for (int i = 0; i < 3; i++) {
            mid[i] = Xthree[i + 3];
            Xthree[i + 3] = 0;
        }
        for (int i = 0; i < 2; i++) {
            mid[i + 3] = Xpair[i];
            Xpair[i] = 0;
        }
    }
}
```

葫蘆的用法則跟鐵支一樣。

```
//三條
//若兩份三條 塞第二份三條
if (Xthree[3] > 0)
{
    if (back[0] == 0) {//判斷尾墩為空 放入
        for (int i = 0; i < 3; i++) {//放三條到後墩第123張
            back[i] = Xthree[i + 3];
            Xthree[i + 3] = 0;
        }

    }
    else if (mid[0] == 0) {//判斷中墩為空 放入

        for (int i = 0; i < 3; i++) {//放三條到中墩第123張
            mid[i] = Xthree[i + 3];
            Xthree[i + 3] = 0;
        }

    }

}
```

三條用法也相同。

```
//對子
//將對子方向顛倒 改為大到小排列
int b[13];
for (int i = 0; i < 13; i++) {
    b[i] = Xpair[12 - i];//複製顛倒陣列
}

for (int i = 0; i < 13; i++) {
    Xpair[i] = b[i];//將顛倒陣列貼上
}

//顛倒後前方數值為0 此步驟將對子往前推回最前方
while (Xpair[0] == 0) {
    for (int i = 0; i < 12; i++) {
        Xpair[i] = Xpair[i + 1];
    }
    Xpair[12] = 0;
}

//用t來記錄 目前有幾組對子
int t = 0;
for (int i = 0; i < 10; i = i + 2) {
    if (Xpair[i] > 0) {
        t++;
    }
}
```

一開始將對子改為從大到小排序，這樣拿對子時才會由最大開始拿，

在用 t 來記住目前幾組對子。

```
if (t == 5)//五組對子的情況
{
    if (back[0] == 0)//尾墩為空
    {
        for (int i = 0; i < 4; i++) {//後墩塞兩副對子
            back[i] = Xpair[p];
            p++;
        }
        for (int i = 0; i < 4; i++) {//中墩塞兩副對子
            mid[i] = Xpair[p];
            p++;
        }
        for (int i = 0; i < 2; i++) {//前墩塞一副對子
            front[i] = Xpair[p];
            p++;
        }
    }

}
```

由如果有五組對子開始，寫到如果有第一組隊子的假設，五個對子就後墩塞兩個對子，中墩塞兩個，前墩塞一個四組對子則以此類推下去。

```
//單張
//塞前墩
for (int i = 0; i < 3; i++)
{
    if (front[i] == 0) {
        front[i] = Xalong[s];
        s++;
    }
}
//塞中墩
for (int i = 0; i < 5; i++)
{
    if (mid[i] == 0) {
        mid[i] = Xalong[s];
        s++;
    }
}
//塞後墩
for (int i = 0; i < 5; i++)
{
    if (back[i] == 0) {
        back[i] = Xalong[s];
        s++;
    }
}
```

塞單張，只要剩下前墩中墩後墩有 0 就塞 Xalong(單張)進去。

```
for (int i = 0; i < 3; i++)
{
    finaly[i] = front[i];
}
for (int i = 3; i < 8; i++)
{
    finaly[i] = mid[i - 3];
}
for (int i = 8; i < 13; i++)
{
    finaly[i] = back[i - 8];
}
//合併
```

最後合併成一個陣列完成。

三、心得

這次期末作業我原本想用 vector 寫，寫完後卻不太確定是否能用 vector，因此我又全部改回用 array 寫，我用 array 跑助教的網站可以執行，但是我用 vector 測試卻無法跑，之後我在改用原本一模一樣的 array 的 dll 跑又不能跑了，我煩惱很久,最後改 dll 的名子變成我隊友的學號又能跑了?我也不太清楚為什麼。雖然我最後用 array 寫，但是中間我也學到了 vector 的許多用法，之後若還有空希望我能把 vector 改成也能跑。

File is an DLL.
檔名符合
The file yzu1072035.dll has been uploaded.
CMD output :
----------Show Distribute Card----------
♦ 8 ♠ 10 ♣ J ♦ Q ♥ J ♦ 6 ♣ 3 ♥ 3 ♥ 5 ♦ 2 ♠ 2 ♣ Q ♣ 9

----------Show Your Card Type----------
♠ 2 ♦ 2 ♥ 5 ♥ 3 ♣ 3 ♦ 6 ♦ 8 ♣ 9 ♣ Q ♦ Q ♥ J ♣ J ♠ 10

----------Same Distribute Card----------
♠ 2 ♠ 10 ♣ 3 ♣ 9 ♣ J ♣ Q ♥ 3 ♥ 5 ♥ J ♦ 2 ♦ 6 ♦ 8 ♦ Q

----------Same Your Card Shot----------
♠ 2 ♠ 10 ♣ 3 ♣ 9 ♣ J ♣ Q ♥ 3 ♥ 5 ♥ J ♦ 2 ♦ 6 ♦ 8 ♦ Q

無更動發牌排組
----------RULECHECK----------
♠ 2♦ 2♥ 5
♥ 3♣ 3♦ 6♦ 8♣ 9
♣ Q♦ Q♥ J♣ J♠ 10

一對

一對

2pairs

----------Same Distribute Card----------
♦ 8 ♠ 10 ♣ J ♦ Q ♥ J ♦ 6 ♣ 3 ♥ 3 ♥ 5 ♦ 2 ♠ 2 ♣ Q ♣ 9

----------Same Your Card Shot----------
♦ 8 ♠ 10 ♣ J ♦ Q ♥ J ♦ 6 ♣ 3 ♥ 3 ♥ 5 ♦ 2 ♠ 2 ♣ Q ♣ 9

一樣的 dll 檔只是名子不同

File is an DLL.
檔名符合

Warning: move_uploaded_file(upload\yzu1072005.dll): failed to open stream: Permission denied in **C:\xampp\htdocs\Debug\upload.php** on line **45**

Warning: move_uploaded_file(): Unable to move 'C:\xampp\tmp\phpAF7A.tmp' to 'upload\yzu1072005.dll' in **C:\xampp\htdocs\Debug\upload.php** on line **45**
Sorry, there was an error uploading your file.

Dll 檔相同，我的學號無法跑我隊友的學號就可以跑。

四、參考資料

1.13張玩法說明 第 1 頁 :: 綜合討論區 :: 牌老匯 討論區 :: 遊戲基地 gamebase http://www.gamebase.com.tw/forum/6653/topic/72664816/1#ixzz5pa4MxIDS

2.13支|多米樂麻將局 https://www.mjweb.com.tw/htm/game_13pokerrule.htm

3.元智資傳程式設計二2019 https://www.facebook.com/groups/1965029646900131/?epa=SEARCH_BOX

4.vector用法 https://mropengate.blogspot.com/2015/07/cc-vector-stl.html

5.vector詳細用法 http://dangerlover9403.pixnet.net/blog/post/98733652-%5B%E6%95%99%E5%AD%B8%5Dc++-vector%E8%A9%B3%E7%B4%B0%E7%94%A8%E6%B3%95

6.線上測試網站

http://140.138.147.47/Debug/