Data Structure
Homework 5

書面報告:

procedure 功能：

此程式中，有以下 procedures-

```
void PrintList();                                   // 印出list的所有資料
void Push_front(int data, char shape, int code);// 在list的開頭新增node
void Push_back(int data, char shape, int code); // 在list的尾巴新增node
void Delete(int data_in, char shape, int code); // 刪除list中的data_in
void Clear();                                       //把整串list刪除
void Reverse();                                 // 把list 反轉
ListNode random_get();                              //隨機取牌
```

輸出入介面說明：(下方圖說明之)

1. 顯示以下文字請選擇模式：

```
Mode input
(1 for functions demo mode; 2 for game mode):
```

1為基本功能 2為額外功能（兩個玩家），輸入數字後按Enter。

2. 若輸入模式1則顯示以下畫面：

```
--------------------------------------------------------------------------------
Function choosing section:
1 for plusing n cards to list
2 for choosing a shape & listing in original order
3 for choosing a shape & listing in oppsite order
4 for choosing a shape & listing in value from smaller to bigger
5 for choosing a card & delete the minimum card among those bigger than the card you choose
others input for Exit
--------------------------------------------------------------------------------
```

選擇1-接著輸入幾張牌，則顯示使用者抽到的牌。

選擇2-接著輸入花色，則依照使用者抽到的順序顯示該花色牌。

```
you choose: 1
The number of the card: 5
Licensing......
S2(2)
D5(31)
C12(51)
C8(47)
S8(8)
Licensed!
```

```
--------------------------------
you choose: 2
The shape of the card: C
C12(51)
C8(47)
--------------------------------
Function choosing section:
```

選擇3-接著輸入花色，則依照相反順序（發牌）顯示該花色牌。

選擇4-接著輸入花色，則依照小至大順序顯示該花色牌。

```
------------------------------
you choose: 3
The shape of the card: C
C8(47)
C12(51)
------------------------------
Function choosing section:
```

```
------------------------------
you choose: 4
The shape of the card: C
C8(47)
C12(51)
------------------------------
```

選擇5-接著先輸入花色，再輸入牌色，則刪除比輸入牌中大但最小的牌。

```
you choose: 5
The shape of the card: C
The number of the card: 8
delete: C12
------------------------------------------------------------
Function choosing section:
1 for plusing n cards to list
2 for choosing a shape & listing in original order
3 for choosing a shape & listing in oppsite order
4 for choosing a shape & listing in value from smaller to
5 for choosing a card & delete the minimum card among tho
others input for Exit
------------------------------------------------------------
you choose: 2
The shape of the card: C
C8(47)
```

選擇其他鍵-程式結束。

P.S.依照作業說明牌色以1-13印出，而牌色大小為1>13>12>11>...>10。

3. 若輸入模式2則顯示以下畫面：

```
(1 for functions demo mode; 2 for game mode): 2
playerA function section:
------------------------------------------------------------------------------------------
Function choosing section:
1 for plusing n cards to list
2 for choosing a shape & listing in original order
3 for choosing a shape & listing in oppsite order
4 for choosing a shape & listing in value from smaller to bigger
5 for choosing a card & delete the minimum card among those bigger than the card you choose
6 for turn ending and switch to next section
others input for Exit
------------------------------------------------------------------------------------------
you choose:
```

4. 模式2操作與模式1相同，唯一差別在輸入6時則跳下一位玩家-

```
you choose: 6
change to next section!
playerB function section:
------------------------------------------------------------------------------------------
Function choosing section:
1 for plusing n cards to list
2 for choosing a shape & listing in original order
3 for choosing a shape & listing in oppsite order
4 for choosing a shape & listing in value from smaller to bigger
5 for choosing a card & delete the minimum card among those bigger than the card you choose
6 for turn ending and switch to next section
others input for Exit
------------------------------------------------------------------------------------------
you choose:
```

5. 操作與模式1相同，唯一差別在輸入6時則進入遊戲-(遊戲模式未完成)。

程式時間複雜度分析：

以n張牌為例：

`void PrintList();` O(n)，每張牌列一次。

```
void Push_front(int data, char shape, int code);
```
O(1)，共三個敘述。
```
void Push_back(int data, char shape, int code);
```
O(n)，依序找到牌的尾巴才插入一個node。
```
void Delete(int data_in, char shape, int code);
```
O(n)，依序尋找要刪除的牌，所以最差情況為O(n)。
```
void Clear();
```
O(n)，每張牌刪一次。
```
void Reverse();
```
O(n)，每張牌反轉一次。
```
ListNode random_get();
```
O(1)，每次抽一張。

再依照所選的功能和次數來總和時間複雜度，舉例來說：抽n張牌且五個功能都執行一次為O(n)+O(1)+O(n)+O(n)+O(n)+O(n)+O(1)=O(n)。