Planning

* 花色大小寫均可
* 可能輸入狀況:

1. 未輸入
2. 花色不符
3. 數字不符
4. 數字輸如其他東西

* Function:

1. 輸入花色(string with space)及數字(string with space)，輸出撲克牌(一行n張)
2. 輸入不合規定則輸出鬼牌(一張)
3. 可以選擇隨機取牌，有可能出現1~13、joker、blank
4. 有機率跑出非撲克牌的牌
5. 輸出全部牌(選擇輸出數量?)
6. 左右切牌

* Input:

1. (1) spade(黑桃)

(2) heart(紅心)

(3) diamond(方塊)

(4) club(梅花)

(5) random

Input the card suit: (number)

1. Input the card value: (1~13)
2. 輸入\*則結束輸入

* Output:

1. 輸出單張牌

* Class:

1. Private: number、suit、line(static)
2. Public function:

☑[黑桃、紅心、方塊、梅花]

☑King、Queen、Soldier

☑joker

☑數字

1. Random:

。for choice : 1~15 (1~13、14: joker)

。for surprise: 1~5

1. call( top, bottom, endl and call print function)

☑process

→ process輸入即拆分資料

→ call function: 呼叫數字function 再呼叫花色function ，除了第一行外其餘先endl再繼續，用巢狀迴圈，外層為n行，內層為m張牌

→ 數字function: cout 頂+數字、數字+底

→花色function: cout 花色

* Function:

1. Judge: 判斷花色數字: 在範圍則判斷call which function，不在則輸出joker
2. Process: 輸入資料即拆分資料

void splitInput(const vector<string>& list, vector<char>& letters, vector<int>& numbers)

* Random:

#include<ctime>

#include<cstdlib>

srand(time(NULL));

cout << 下限 + rand() % (上限 – 下限 + 1)

→ 0~14 (0 for joker, 14 for blank)

* Vector:

#include<vector>

處理資料輸入

* Main:

1. 輸入
2. Random: 正常輸出則繼續執行程式，否則輸出彩蛋

→ ex: random範圍1~5，!1則可以繼續，彩蛋機率1/5

1. Call judge function → call suit class function → cout

* Poker card: (10 lines -> 8 lines without top and bottom) (space – 6)

|  |  |  |  |
| --- | --- | --- | --- |
| ----------------------  | 2 |  | /\ |  | / \ |  | / \ |  | \ / |  | \ / |  | \/ |  | 2 |  ---------------------- | ----------------------  | 7 |  | \_ \_ |  | / \\_/ \ |  | \ / |  | \ / |  | \ / |  | \/ |  | 7 |  ---------------------- | ----------------------  | 8 |  | / \ |  | / \ |  | / \ |  | I I |  | `--‘||`--‘ |  | /\_\ |  | 8 |  ---------------------- | ----------------------  | 5 |  | \_\_\_ |  | / \ |  | \_\_\ /\_\_ |  | / \ |  | \\_\_/||\\_\_/ |  | /\_\ |  | 5 |  ---------------------- |
| ----------------------  | J (diamond) |  | \_\_ |  | / \ |  | \ / |  | < > |  | / \ |  | \\_\_/ |  | (diamond) J |  ---------------------- | ----------------------  | Q (heart) |  | /\ |  | /\\//\ |  | \//\\/ |  | \/ |  | \*\ || /\* |  | || |  | (heart) Q |  ---------------------- | ----------------------  | K (spade) |  | /\ |  | | | |  | | | |  | | | |  | ,-====-‘ |  | || |  | (spade) K |  ---------------------- | ----------------------  | JOKER |  | / \_\\_/\_ \ |  | / /\_\_\_\_\_ \ \ |  | | | + + | ||  | 0 | O | 0|  | | ‘\\_\_/` | |  | `---------‘ |  | JOKER |  ---------------------- |
| ----------------------  | |  | |  | |  | |  | |  | |  | |  | |  ---------------------- | ----------------------  |\_\_\_\_\_ |  | ] [ ] | \ O / |  | ] [ ] | / |  | ] [ ] | ) ) |  | ] [ ] | |  | ] [ ] | |  | calm down !!! |  | ] [ ] | |  ---------------------- |  |  |

#include <iostream>

#include <vector>

using namespace std;

class MyClass {

private:

vector<int> data; // 在類別中定義一個 vector 作為成員

public:

// 成員函數：向 vector 添加元素

void addData(int value) {

data.push\_back(value);

}

// 成員函數：顯示 vector 中的所有元素

void displayData() const {

for (int value : data) {

cout << value << " ";

}

cout << endl;

}

// 成員函數：獲取 vector 的大小

size\_t getSize() const {

return data.size();

}

};

int main() {

MyClass obj;

// 向 vector 中添加元素

obj.addData(10);

obj.addData(20);

obj.addData(30);

// 顯示 vector 中的元素

obj.displayData();

// 顯示 vector 的大小

cout << "Size of vector: " << obj.getSize() << endl;

return 0;

}

#include <iostream>

#include <vector>

#include <string>

#include <cctype>

using namespace std; // 引入 std 命名空間

int main() {

// 原始資料

vector<string> list = {"A1", "D12", "C5"};

// 用來儲存拆分後的兩個list

vector<char> letters; // 儲存字母部分

vector<int> numbers; // 儲存數字部分

// 遍歷每一個元素

for (const auto& str : list) {

string letter\_part;

string number\_part;

// 拆解字母和數字

for (char ch : str) {

if (isalpha(ch)) {

letter\_part += ch; // 若是字母，加入到字母部分

} else if (isdigit(ch)) {

number\_part += ch; // 若是數字，加入到數字部分

}

}

// 將字母部分與數字部分儲存到對應的vector中

letters.push\_back(letter\_part[0]); // 假設每個字母部分只有一個字母

numbers.push\_back(stoi(number\_part)); // 轉換數字部分為整數

}

// 顯示結果

cout << "Letters: ";

for (const char& letter : letters) {

cout << letter << " ";

}

cout << endl;

cout << "Numbers: ";

for (const int& number : numbers) {

cout << number << " ";

}

cout << endl;

return 0;

}