

# C++程式設計專題製作報告

## 題目:Bulls and Cows(1A2B) 遊戲製作

Project hosting: <https://github.com/FelicityTomato/Bulls-and-Cows/>

作者:臺北市私立復興實驗高級中學 十年信班 劉蕃熙

指導老師:宋千儀

# 一、專題簡介

在本學期初，我和幾位志同道合的同學們創立了資訊研究社，身為社長的我，課程的準備也成為我的責任。正當我思索著要給大家完成什麼遊戲時，我發現班上的同學們很喜歡玩的一個遊戲：1A2B，加上我本身也是1A2B的愛好者，便決定選擇這個遊戲。考慮社員的程式能力，所以僅製作了電腦出題給玩家猜的版本。

但在社課結束後，我便希望可以將它變成以如朋友一樣的功能，也就是互相出題給對方猜，先猜對者獲勝。這樣，當你想玩遊戲卻又找不到伴時，就可以和電腦對戰又可以享受和朋友玩的真實感。因而接續完成「電腦與人對玩」的版本，製作過程中，我發現我在這一個專案中學習到了很多技巧，也就製作了今天這一個人專案來記錄這一個遊戲的製作過程與收穫。相信經過這一次的學習後，可以幫助我未來遇到問題時的邏輯架構與程式技巧。

# 二、專題製作過程

## (一) 問題分析：

### 1.電腦出題：

隨機產出一組謎底：四位數的不重複數

輸入：一組四位數的不重複數

將輸入的數字與謎底比較並產出A,B

輸出：A與B的個數

### 2.電腦猜：

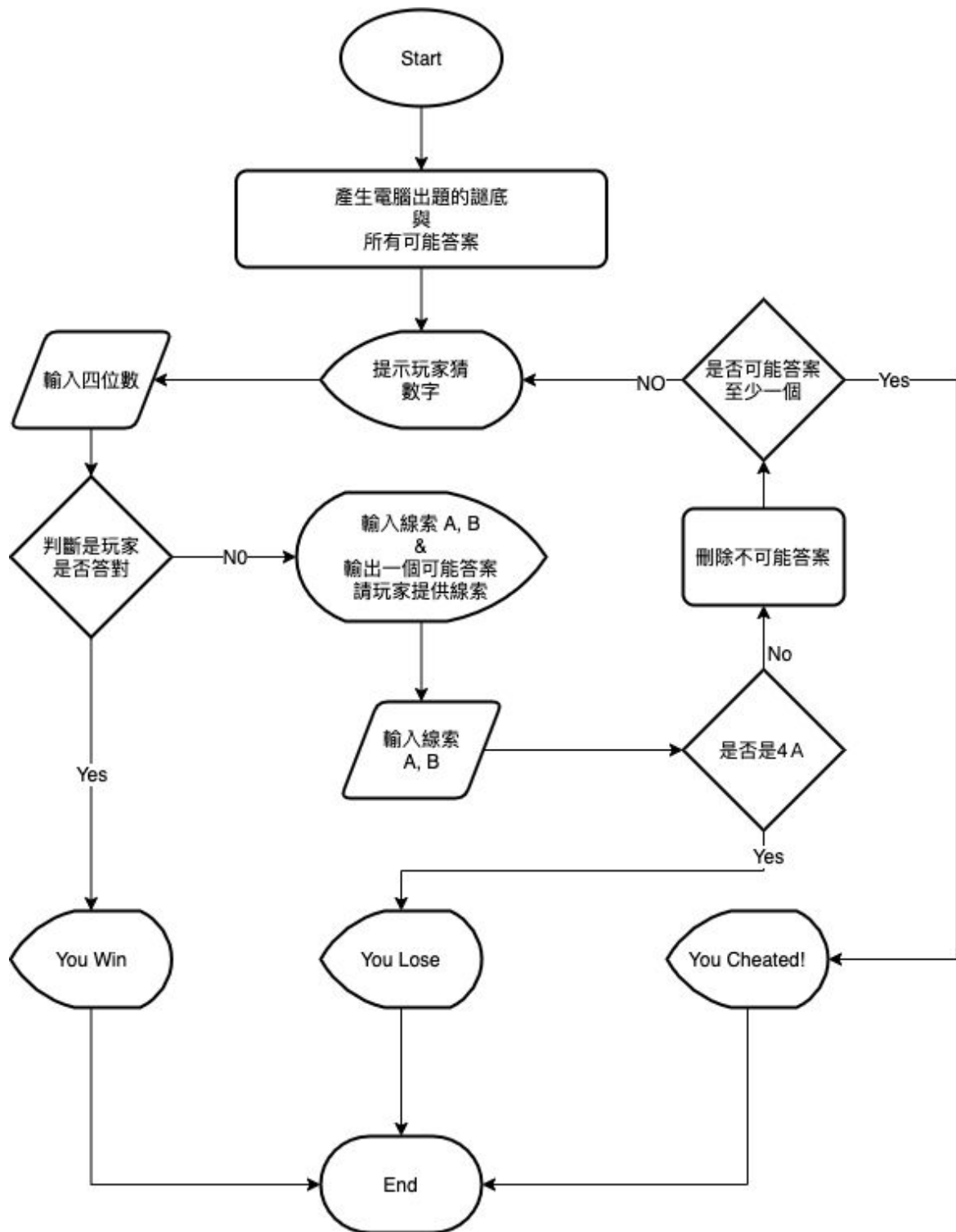
將所有可能縫入一個std容器的set裡 (size=  $P_4^{10} = 5040$ )

輸出：隨機選擇set內的一種輸出

輸入：A與B的個數

判斷並將不符合的從set裡刪除

(二) 程式設計流程圖：



## 三、程式碼與功能說明

### (一) 使用技巧：

1. rand函數
2. 遞迴
3. STL 容器: set,vector
4. 疊代器(iterator)
5. struct 結構

### (二) 使用工具：

1. git 版本控制--使用github
2. markdown 文件撰寫--使用hackmd.io

### (三) 程式碼說明：

#### 1. 電腦出題：

隨機產生一組解：

```
string trueans="abcd"; //定義global: trueans變數則無須回傳值
void gen_trueans(){
    srand(time(0)); // 以時間作為rand()的seed, 確保每次rand結果不同
    set<int> st; //set記錄以選到的數並避免重複
    int num;
    for (int i=0; i<4; i++) {
        do {
            num=rand()%10; //利用rand隨機產出並利用餘數控制範圍
        } while (st.count(num)!=0);
        st.insert(num);
        trueans[i]=num+'0';
    }
}
```

判斷A, B個數：

```
int A=0,B=0;
for (int i=0; i<4; i++){
    for (int j=0; j<4; j++){
        if (trueans[i]==guess[j]) {
            if(i==j) A++; //先比較A在比較B可以避免重複計算
            else B++;
            break;
        }
    }
}
```

## 2. 電腦猜題：

使用遞迴列出來所有可能解，並存在 set 中

```
set<string> st;
// 使用遞迴列出來所有可能解
string arr = "abcd";
void gen_allposible(int idx){
    for(int i='0'; i<='9'; i++){ // 產生 char '0' ~ '9'
        if(arr[0] != i && arr[1] != i && arr[2] != i ){
            arr[idx]=i;
            if(idx==3){
                st.insert(arr); //使用set記錄所有可能的解
            }
            else gen_allposible(idx+1); //遞迴
            arr[idx]=-1;
        }
    }
}
```

開始猜

```
// 電腦開始猜
set<string>::iterator it; // 定義一個 iterator
it = next(st.begin(), rand() % st.size()); // rand 指到set中的一個可能解
// 然後輸出等待玩家輸入
```

根據玩家輸入的Hint (xAyB)，從SET刪除不可能的解

```
// 參考：https://www.techiedelight.com/remove-elements-set-cpp/
for(it = st.begin() ;it!=st.end(); ){ // 從第一個元素，到set結束

    // call check(), 檢查是否與Hint相符
    bool check_ans= check( ans[0], ans[1], *it, gs);
    if (check_ans == false) {
        st.erase(it++); // 避免iterator指到set中已刪除的元素，所以要這樣寫
    } else {
        ++it;
    }
}
```

### 3. 儲存輸出：

```
set<string> st;
struct anss{
    string ta; //輸入值
    int a; //A個數
    int b; //B個數
};
vector <anss> Record; //曾經的輸入結果

void printout() {
    cout << "+-----+-----+\n"
         << "|  PLAYER      |      PC      |\n"
         << "+-----A-B-+-----A-B-+\n";
    for (int i =0; i<Record.size(); i++ ) {
        cout << "| " << Record[i].ta
              << " " << Record[i].a
              << " " << Record[i].b
              << " ";

        i++;
        if (i == Record.size()) {
            cout <<"|          |\n";
        } else {
            cout << "| " << Record[i].ta
                  << " " << Record[i].a
                  << " " << Record[i].b
                  << " |\n";
        }
    }

    cout << "+-----+-----+\n";
}
```



## 四、 操作說明與測試結果

### 玩家勝

利用 `srand(1024)`，使每次電腦出題都是一樣

```
liufanxide-MacBook-Air:src felicity$ ./a.out

Hello! Welcome to 1A2B!
-----

Your turn. Please guess my number? 1829
+-----+-----+
|  PLAYER  |    PC    |
+-----A-B-+-----A-B-+
| 1829   0 3 |          |
+-----+-----+
The hint is 0A3B

My turn. I guess the number is 2039
Please give me a hint. (Input 'A' <space> 'B' <Enter> ): 0 2
+-----+-----+
|  PLAYER  |    PC    |
+-----A-B-+-----A-B-+
| 1829   0 3 | 2039   0 2 |
+-----+-----+

Your turn. Please guess my number? 8195
---- You Win. ----
liufanxide-MacBook-Air:src felicity$
```

## 電腦勝

```
liufanxide-MacBook-Air:src felicity$ ./a.out

Hello! Welcome to 1A2B!
-----

Your turn. Please guess my number? 1234
+-----+-----+
|  PLAYER  |    PC   |
+-----A-B-+-----A-B-+
| 1234  1 1 |      |
+-----+-----+
The hint is 1A1B

My turn. I guess the number is 9468
Please give me a hint. (Input 'A' <space> 'B' <Enter> ): 1 2
+-----+-----+
|  PLAYER  |    PC   |
+-----A-B-+-----A-B-+
| 1234  1 1 | 9468  1 2 |
+-----+-----+

Your turn. Please guess my number? 5678
+-----+-----+
|  PLAYER  |    PC   |
+-----A-B-+-----A-B-+
| 1234  1 1 | 9468  1 2 |
| 5678  0 2 |      |
+-----+-----+
The hint is 0A2B

My turn. I guess the number is 0489
Please give me a hint. (Input 'A' <space> 'B' <Enter> ): 4 0
---- You Lose ----
liufanxide-MacBook-Air:src felicity$
```

## 五、專題結果與討論

### (一) 所遇困難：

Set的刪除：如何避免iterator指到set中已刪除的元素

ex:

程式若這樣寫--當set內含有偶數個時

```
int main()
{
    set<int> st = { 1,2,3,4,5,6,7,8};

    for(auto it = st.begin() ;it!=st.end(); it++ ){
        if ( *it %2 == 0) {
            st.erase(it);
        }
    }

    for(auto it = st.begin() ;it!=st.end(); it++ ){
        cout << *it << " ";
    }

    cout << endl;
    return 0;
}
```

輸出：1 3 4 5 6 7 8

```
liufanxide-MacBook-Air:src felicity$ g++ -std=c++11 test.cpp
liufanxide-MacBook-Air:src felicity$ ./a.out
1 3 4 5 6 7 8
liufanxide-MacBook-Air:src felicity$
```

正確輸出應是：1 3 5 7

經上網搜尋

(<https://www.techiedelight.com/remove-elements-set-cpp/>)

與實做查證發現問題出在iterator的point錯誤

## (二) 可改進之處：

1. 根據測試統計，電腦猜題最多大約8次可達對，因此其小於  $O(\log_2 N)$  ( $\log_2 5040 \approx 12.3$ ) 因此希望未來可利用統計將剩餘可能的選項中的0-9數字出現在每個位置的數量，找出現機率最高的先猜，將電腦猜的版本答對率提高。
2. 可以利用function將程式碼變得容易閱讀與改進。
3. 因程式分階段完成（先寫1A2B\_questioner再寫1A2B\_guesser）造成類似部分使用不同寫法完成，雖然可以比較其差異，卻會使程式碼更難閱讀）。

## (三) 心得與探討：

在這次的1A2B專題報告中，我才真正的理解到程式結構清楚的重要性，程式結構清楚可以幫助我麼要修改、新增和除錯時更方便迅速。而要讓結構清楚有幾個方式，其中最重要的就是使用function並使用適當的命名。這樣當只想看程式結構時只要看main()就可以了。

而這一個程式專題研究中，我採用了版本控制--github存放我的專案，這樣我就很方便去檢視每一次的差異與回顧。另外，這一個專題研究我也把它分成兩個部分再統合，我歸納了這個方式的優缺：優點是結構相對清楚，且發生問題時也比較容易確定使哪一部份的問題又或者是統合時的錯誤；而缺點是因為是分開寫的因而寫法不一致，造成debug和新增功能或重新檢視時的不易。

而為了讓遊戲可以進行得更順利，我使用了自定義的struct結構(string,int,int)以及vector來儲存雙方已經猜過的答案與提示，可以幫助玩家整理以方便作答。

希望未來可以將目前不足之處改進，並利用github追蹤修正，可以將其變得稱有效率，答題速度更快。未來的進展將會放在我的github上。  
(<https://github.com/FelicityTomato/Bulls-and-Cows/>)