NUMBER GUESSING

2014 NCKUEE Freshman Camp

2014/08/13

Guess My Number!

- The system randomly generate a secret number, lies between 1 and 100
- We guess the number according to the hint (too big / too small) after each turn
- Until we guessed the secret number
- 有玩過「終極密碼」吧？

Main Steps

1. Randomly generate a secret number between 1 and 100
2. User inputs his/her guess via keyboard
3. Determine this number is illegal or not
   - If so, output the error message
4. Give the user feedback (too big/small)
   - If right, congratulations to the user and give him/her comments

Step.1 指引與隨機生成數字

```
/* Declaring variables (for "telling" the OS how much space of RAM should be distributed) */
int password; // stores the actual password
int guess; // stores the number user entered
int attempts = 0; // records the number of attempts
bool password_hacked = false; // whether the password of the chest is hacked or not

/* Welcome message & instructions */
cout << "Guess My Number!!" << endl;
cout << "There is a chest of treasure on the desk, but it needs a PASSWORD to open!" << endl;
cout << "The password lies between 1 and 100. Go and guess it!" << endl;
system("pause");

/* Generate a password */
password = (rand() % 100) + 1;
```

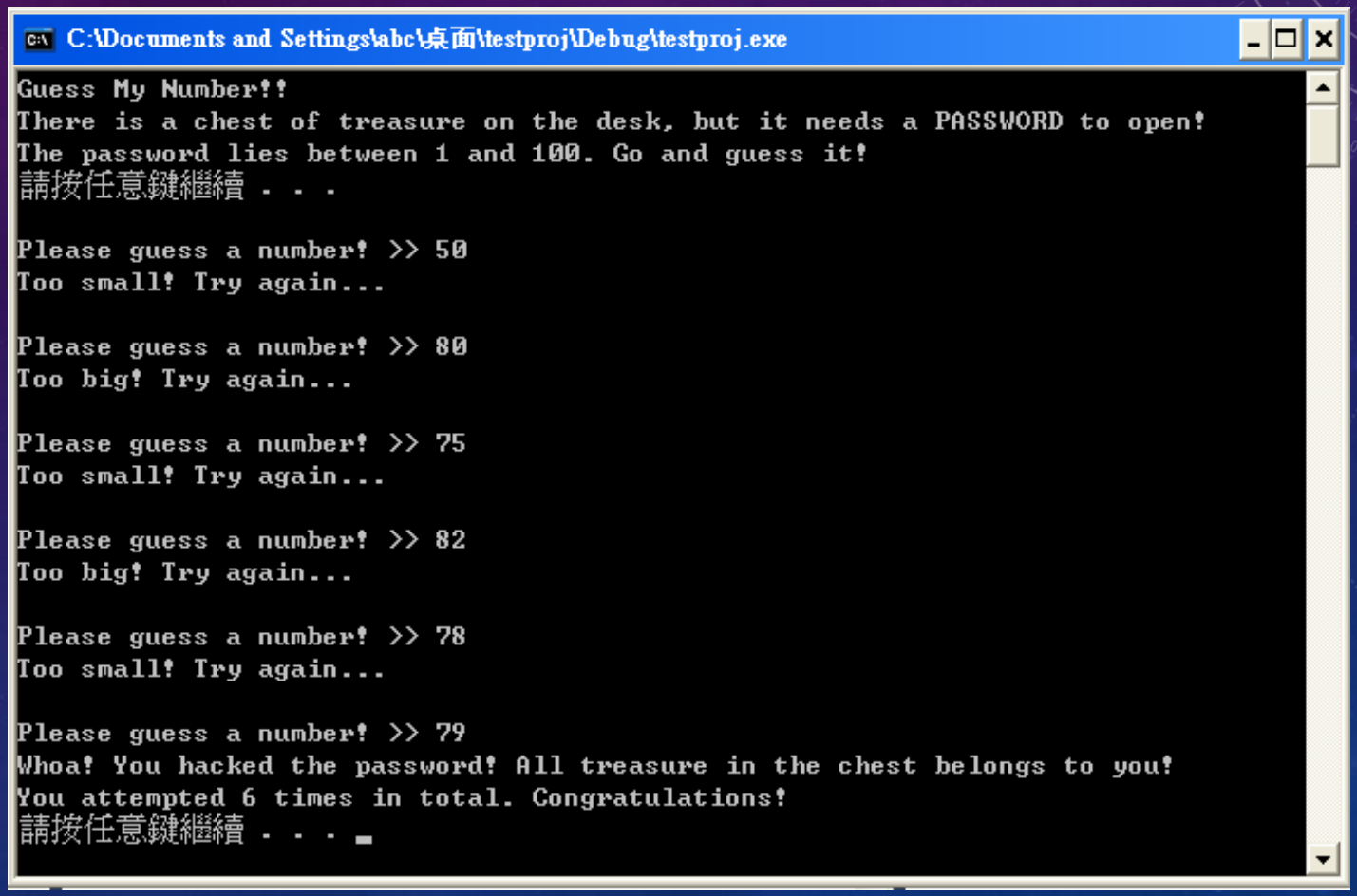
Step.2 開始猜、給回饋

```
/* Start guessing!! */
while(!password_hacked) {
    attempts++;
    cout << endl << "Please guess a number! >> ";
    ?????????? // COMPLETE IT!!! ---> GET a input from keyboard, then stores it in variable "guess"

    /* determine the number endtered by user is right or not */
    if(??????????) {
        cout << "Whoa! You hacked the password! All treasure in the chest belongs to you!" << endl;
        password_hacked = true;
    }
    else if(??????????) {
        cout << "The password lies between 1 and 100!" << endl;
    }
    else if(??????????) {
        cout << "Too small! Try again..." << endl;
    }
    else if(??????????) {
        cout << "Too big! Try again..." << endl;
    }

}
```

Step.3 若猜對、則恭喜並評分

```
/* Password hacked! Give the user comments */
cout << "You attempted " << attempts << " times in total. ";
if(attempts < 3) {
    cout << "Wow! You are so lucky!" << endl;
}
else if(attempts < 6) {
    cout << "You are a lucky guy!" << endl;
}
else if(attempts < 9) {
    cout << "Congratulations!" << endl;
}
else {
    cout << "It seems that you took lots of time to hack it..." << endl;
}
system("pause");
```

Demo


C:\Documents and Settings\abc\桌面\testproj\Debug\testproj.exe
Guess My Number!!
There is a chest of treasure on the desk, but it needs a PASSWORD to open!
The password lies between 1 and 100. Go and guess it!
請按任意鍵繼續 . . .
Please guess a number! >> 50
Too small! Try again...
Please guess a number! >> 80
Too big! Try again...
Please guess a number! >> 75
Too small! Try again...
Please guess a number! >> 82
Too big! Try again...
Please guess a number! >> 78
Too small! Try again...
Please guess a number! >> 79
Whoa! You hacked the password! All treasure in the chest belongs to you!
You attempted 6 times in total. Congratulations!
請按任意鍵繼續 . . .

From Ideas to Algorithms

- 進入更難一點的練習前，我們想先談談這件事
- 關於**「我有個想法，但不知道怎麼變成程式」**
- 許多初學者遭遇的第一關

From Ideas to Algorithms

- 我們大致能依照以下步驟：
  - 首先要先知道這個問題的要求，決定該提供什麼功能
  - 然後要滿足這些功能，應該需要輸入什麼東西、又應該輸出什麼、需要多少空間存這些變數
  - 把解決這個問題的步驟一一列出，並畫成流程圖
    - 哪時候要提示使用者輸入資料
    - 哪時程式要進行運算、如何運算
    - 哪時要把資料輸出
  - 再根據這些流程圖轉換成程式語言

The Meaning of Variables

- **Variable (變數)**
  - a symbolic name associated with a value and whose associated value may be changed
- 為什麼要「宣告」變數
  - E.g. `int i; double d; char c;`
  - 告訴作業系統要預先分配多少記憶體空間給這些值
  - C++ 是靜態型別語言，必須在編譯前就明確指出其型態

Bulls and Cows

- Another type of number guessing where the password is a 4-digit numeric value that are all different
- If the matching digits are on their right positions, they are "bulls" (A)
- If on different positions, they are "cows" (B)
- 就是有提示「幾A幾B」的版本

Bulls and Cows: An Example

- Answer: 4271
- If we give…
  - 9865 -> 0A0B
  - 1234 -> 1A2B
  - 7851 -> 1A1B
  - 2417 -> 0A4B
  - 4271 -> 4A0B (win!)

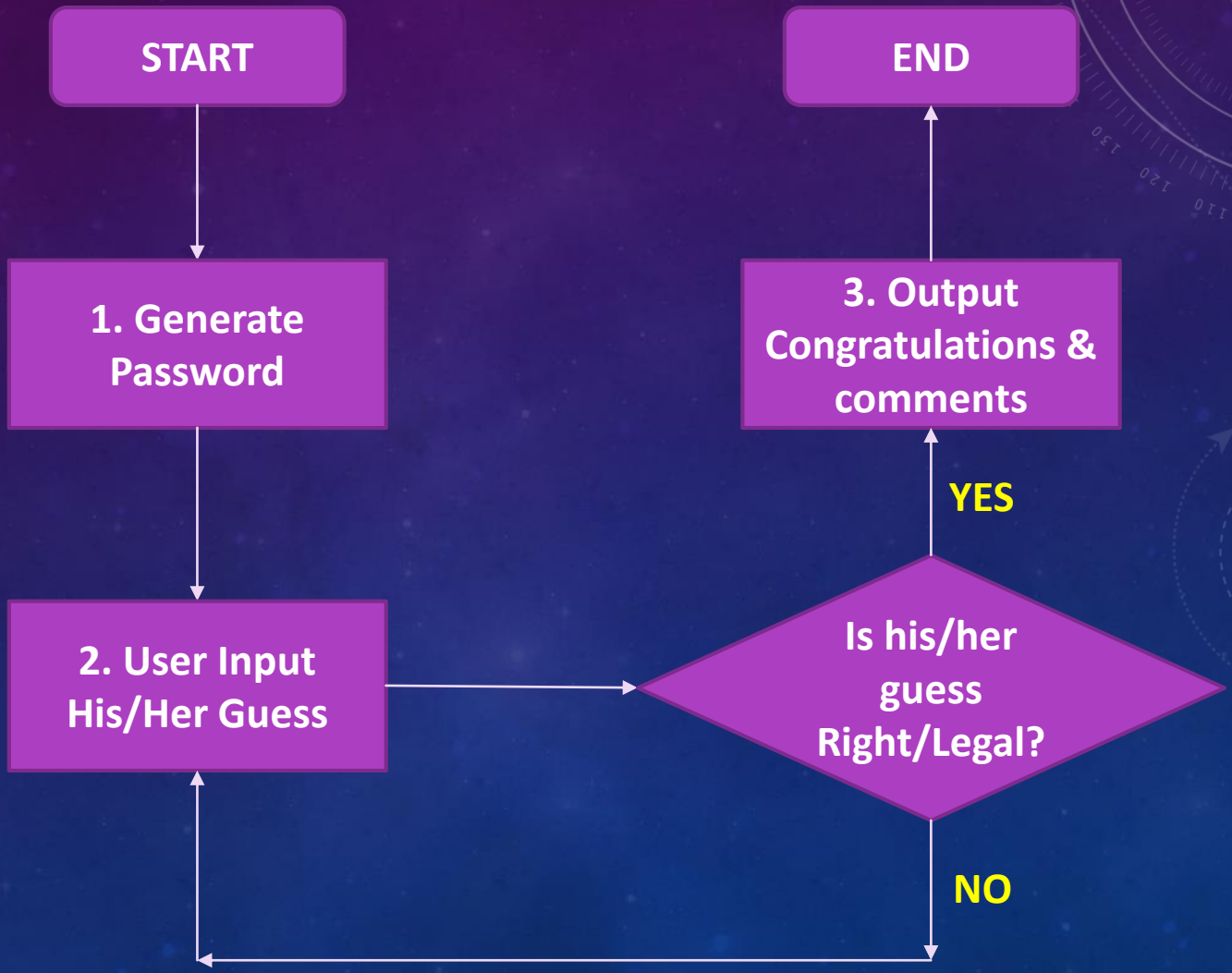
Problem Analysis

- 要輸入什麼？
  - User input
- 要輸出什麼？
  - How many bulls/cows
- 要存什麼？
  - The actual password
  - Number of attempts
  - …

Main Steps

1. Randomly generate 4 digits
   - They should be all different
2. The User inputs a 4-digit number
   - Are the numbers illegal?
   - If so, output error messages and start over
3. Determine if the user is win or not
   - Then give him/her feedbacks
   - If the user wins, output congratulation messages and comments

The Flow Chart

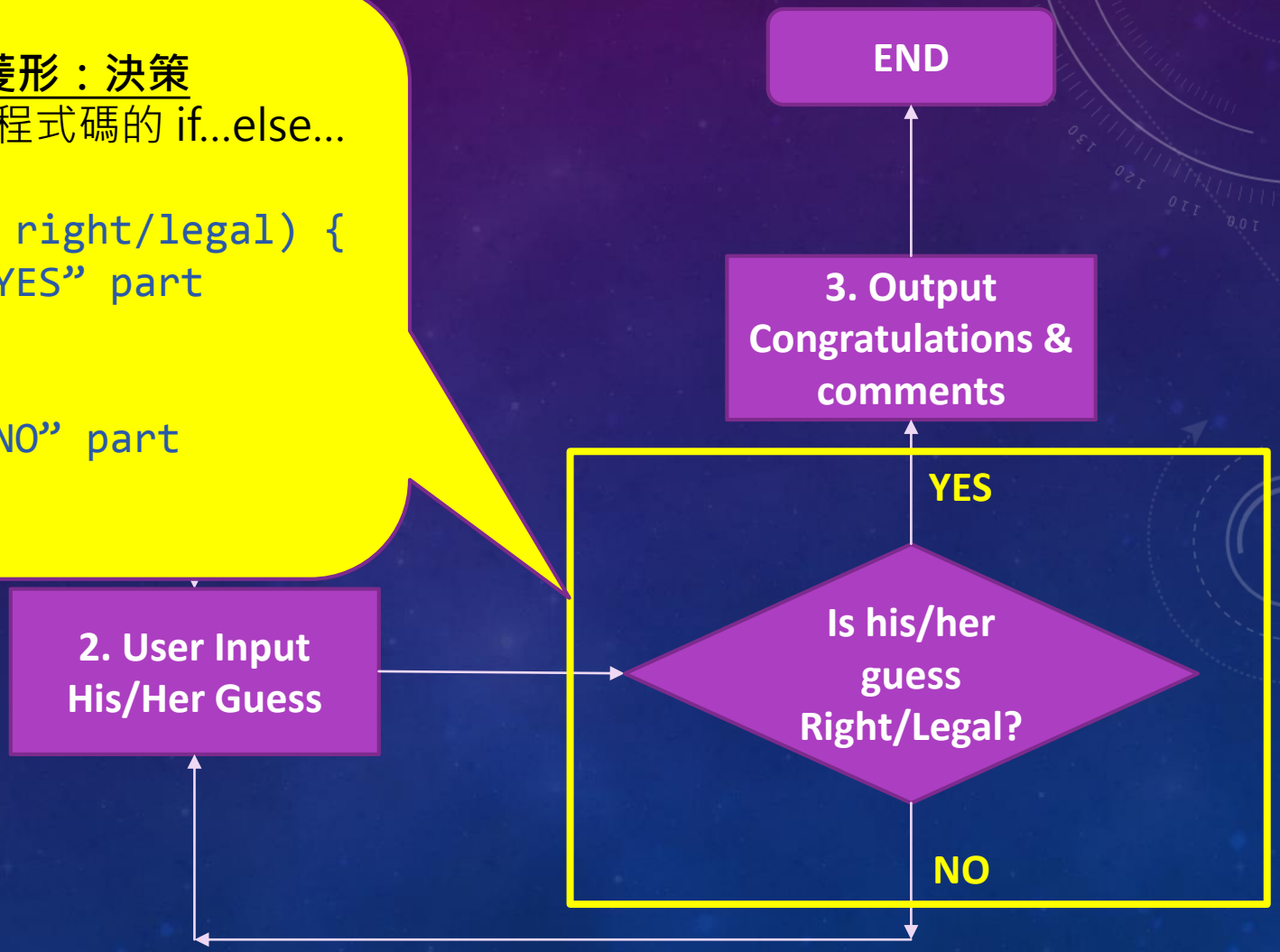


The Flow Chart

菱形：決策

可以轉成程式碼的 if...else...

```
if(guess right/legal) {
    // “YES” part
}
else {
    // “NO” part
}
```

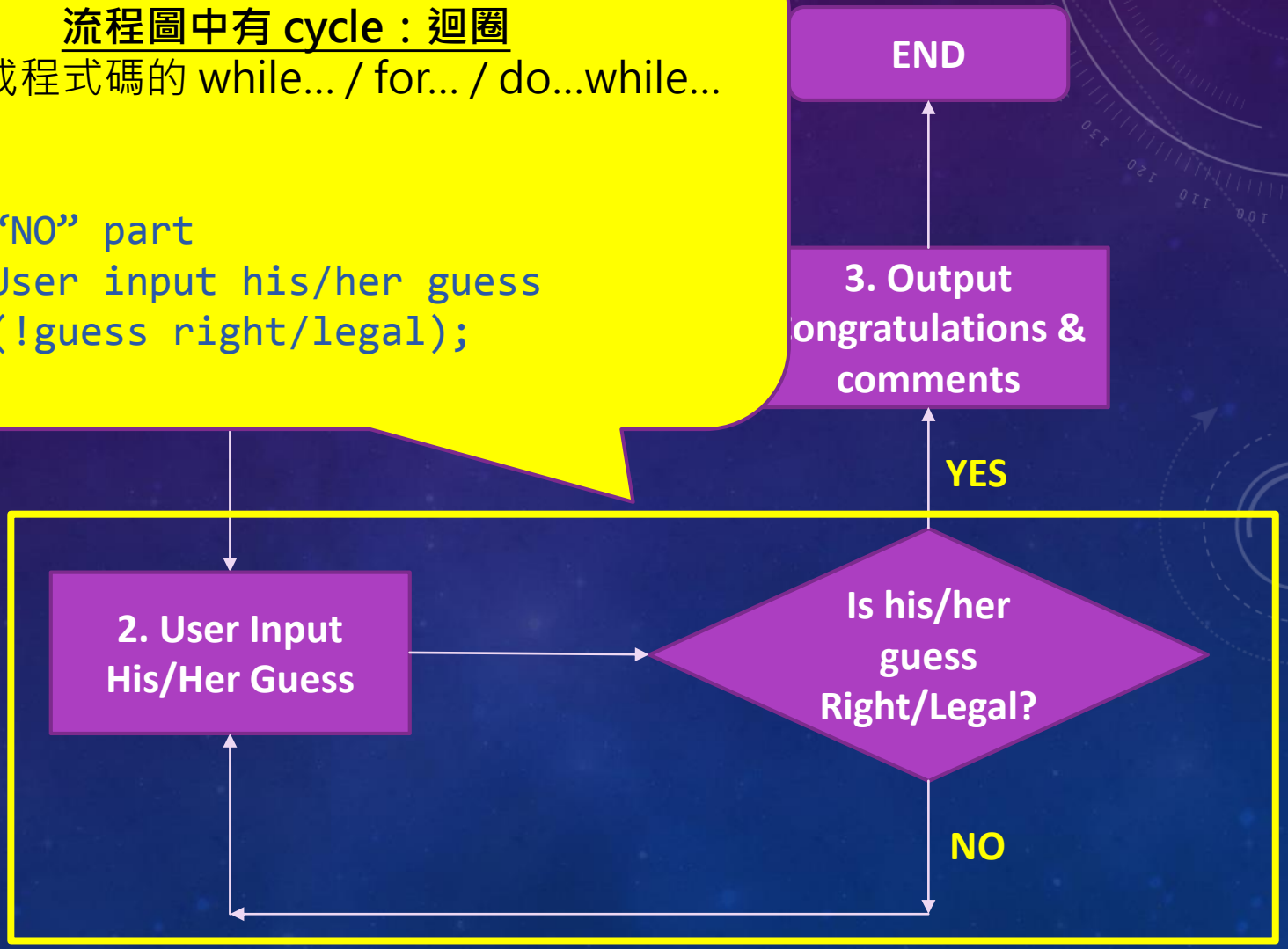


The Flow Chart

流程圖中有 cycle：迴圈

可以轉成程式碼的 while… / for… / do…while…

```
do {
    // “NO” part
    // User input his/her guess
} while(!guess right/legal);
```



Array (陣列)

- A data structure consisting of a collection of elements (with the same type), each identified by at least one array index or key

- Code example:

```
int ary[6];
ary[0] = 1;
ary[1]++;
```

Step.0 宣告變數與指引

```
/* Declaring variables (for "telling" the OS how much space of RAM should be distributed) */
int password[4], guess[4]; // stores the actual password and user's guess in array form
int guess_original; // stores the user input from keyboard
int i, j; // for loop counting
int bulls_A, cows_B; // records A and B
int count = 0; // for counting the generated digits of password
int attempts = 0; // records the number of attempts
bool password_hacked = false; // whether the password of the chest is hacked or not
bool guess_duplicated; // whether the user's guess has digit duplication

/* Welcome message & instructions */
cout << "Bulls and Cows!" << endl;
cout << "There is a chest of treasure on the desk, but it needs a PASSWORD to open!" << endl;
cout << "The password lies between 0000 and 9999." << endl;
cout << "The digits of password are all different. Go and guess it!" << endl;
system("pause");
```

Step.1 隨機產生密碼

```
/* Generate 4 non-duplicated numbers: password */
while(count < 4) {
    password[count] = rand() % 10;
    for(i = 0; i < count; i++) {
        if(password[count] == password[i]) {
            count--;
        }
    }
    count++;
}

/* DEBUG: Tests whether the password is legal or not */
/*
for(i = 0; i < 4; i++) {
    cout << password[i];
}
cout << endl;
*/
```

Step.2 使用者輸入與處理

```
/* reset/inititial variables */
guess_duplicated = false;
bulls_A = 0;
cows_B = 0;
attempts++;

/* Get the user input */
cout << endl << "Please guess the password! >> ";
cin >> guess_original;

/* preprocess the user input */
guess[0] = guess_original / 1000;
guess[1] = (guess_original % 1000) / 100;
guess[2] = (guess_original % 100) / 10;
guess[3] = guess_original % 10;
```

Step.3 檢測使用者輸入

```
/* Check the number that user entered is legal or not */
/* If so, output error message */
if((guess_original < 100) || (guess_original > 9999)) { // is a 4-digit password?
    cout << "The password has 4 digits!" << endl;
    continue;
}

/* check if the password user guessed has digit-duplication */
/* if so, set guess_duplicated as true */
//------------- Q1. COMPLETE IT!! -------------//
```

PRACTICE!

```
if(guess_duplicated) {
    cout << "The digits of password are ALL DIFFERENT!" << endl;
    continue;
}
```

Check Digit-Duplication

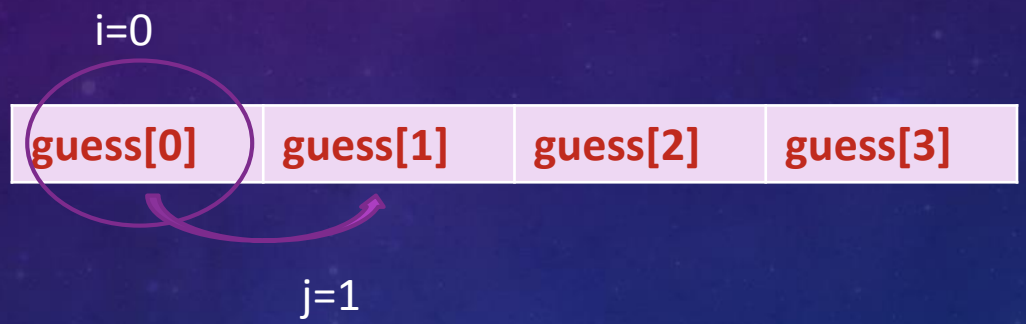
- 目的：檢測某組數值有無重複數字
- 觀念：測試數值中每位數兩兩是否相同
- 問題：如何找出所有位數組合並比較？

- 高中排列組合問題
  - **假設一聚會共有 n 人，聚會開始前彼此握手致意，試問共有多少種可能的握手組合？ A: n(n-1)/2 種**

四位數 -> 4(4-1)/2 -> 6 種

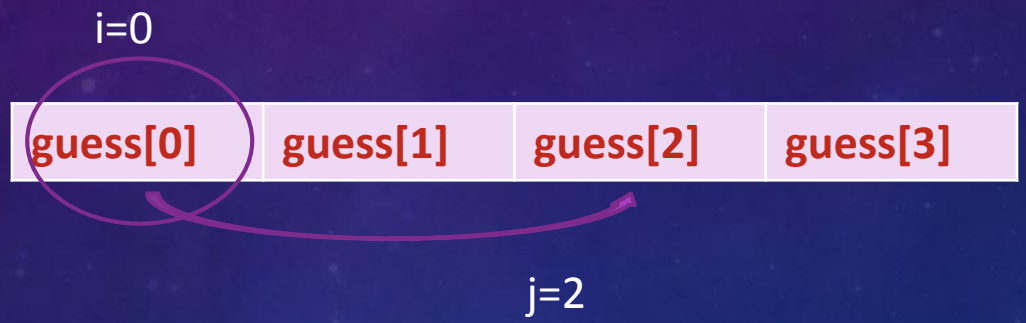
Check Digit-Duplication

NO.1



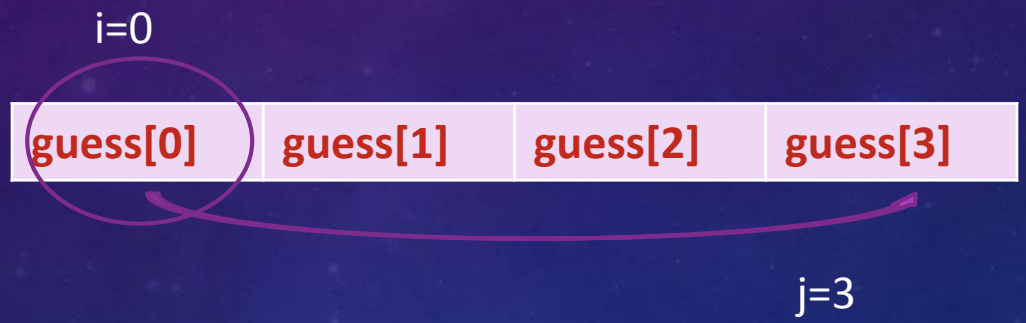
Check Digit-Duplication

NO.2



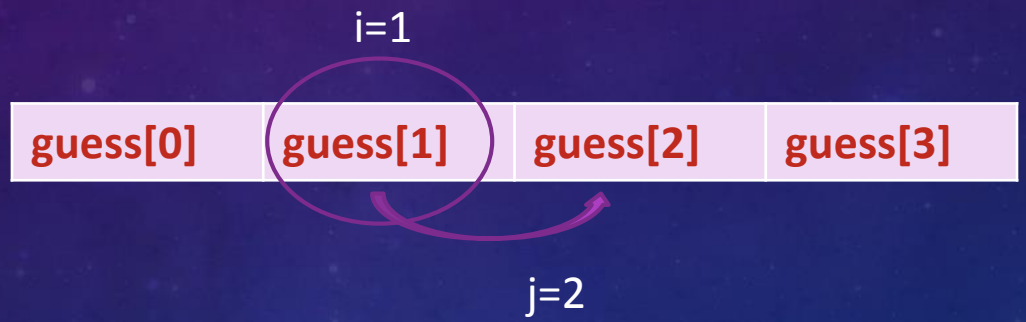
Check Digit-Duplication

NO.3



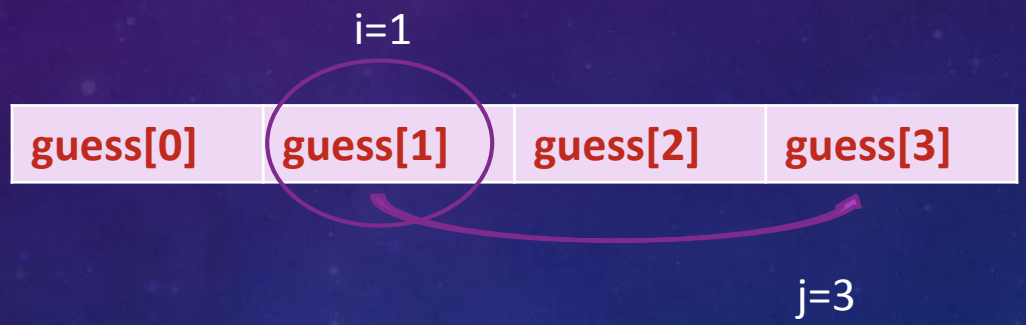
Check Digit-Duplication

NO.4



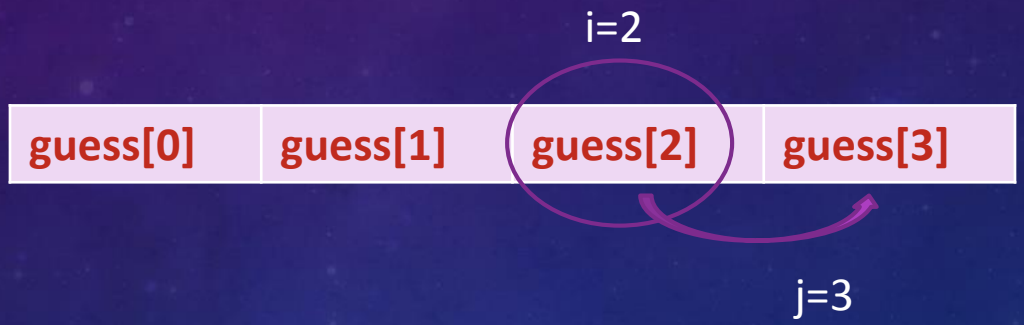
Check Digit-Duplication

NO.5



Check Digit-Duplication

NO.6



Step.4 給予提示(?A?B)

```
/* Get the value of "bulls"(A) and "cows"(B) */
/* performs all possible pairwise comparisons across digits */
//------------- Q2. COMPLETE IT!! -------------//
```

PRACTICE!

```
/* determine the password is hacked or not */
if(bulls_A == 4) { // the user wins!
    cout << "Whoa! You hacked the password! All treasure in the chest belongs to you!" << endl;
    password_hacked = true;
}
else {
    cout << "You got " << bulls_A << " bull(s) and " << cows_B << " cow(s) (" << bulls_A << "A" <<
}
```

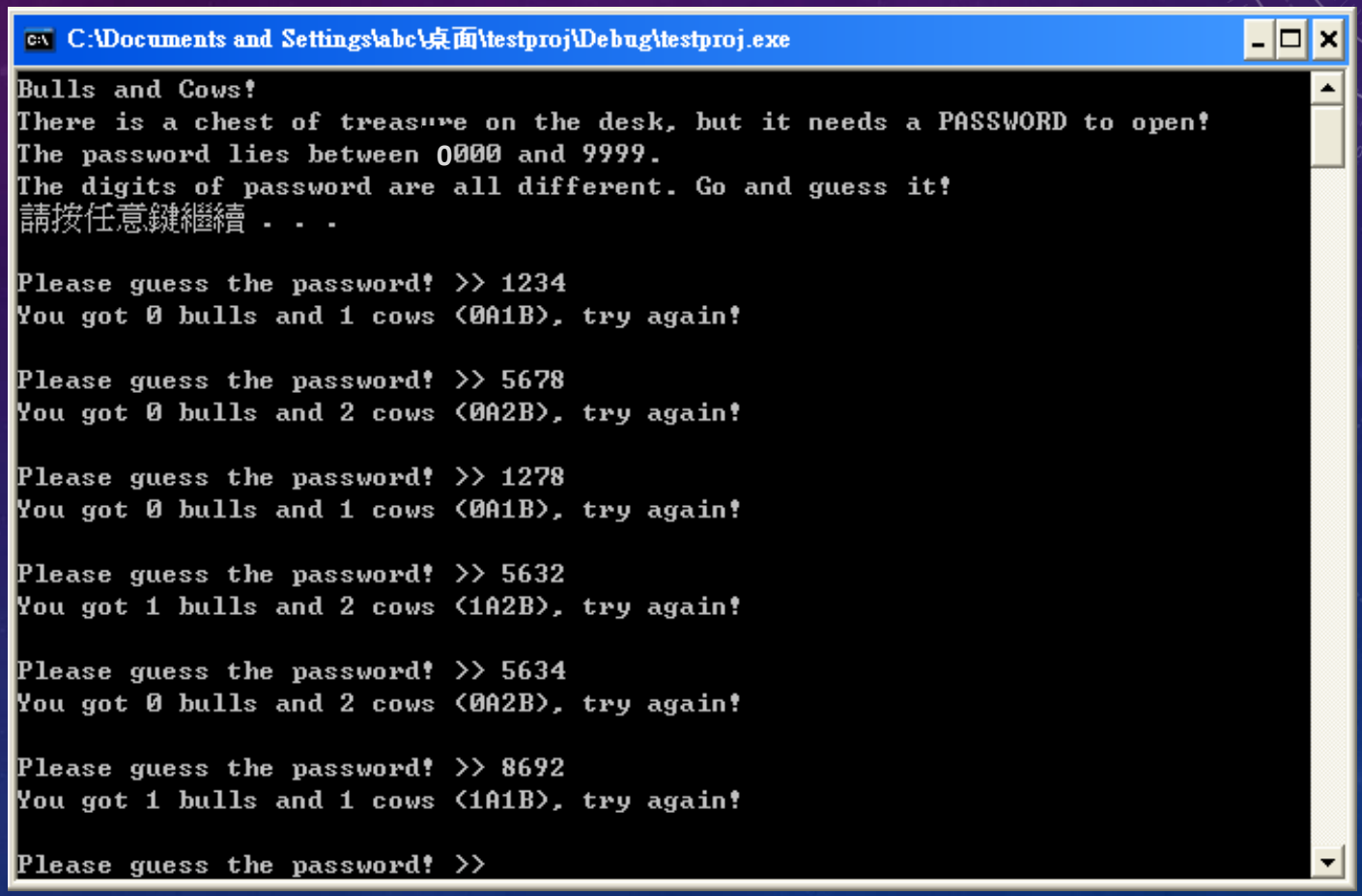
Step.2~4

```
while(!password_hacked) {
    // Step.2 使用者輸入與處理
    // Step.3 檢測使用者輸入
    // Step.4 給予提示(?A?B)
}
```

Step.5 恭喜與評價

```
/* Password hacked! Give the user comments */
cout << "You attempted " << attempts << " times in total. ";
if(attempts < 4) {
    cout << "Wow! You are so lucky!" << endl;
}
else if(attempts < 6) {
    cout << "You are a lucky guy!" << endl;
}
else if(attempts < 8) { // Trivia: It is proven that any number could be solved up to 7 turns!
    cout << "Congratulations!" << endl;
}
else {
    cout << "It seems that you took lots of time to hack it..." << endl;
}
system("pause");
```

Demo


C:\Documents and Settings\abc\桌面\testproj\Debug\testproj.exe
Bulls and Cows!
There is a chest of treasure on the desk, but it needs a PASSWORD to open!
The password lies between 0000 and 9999.
The digits of password are all different. Go and guess it!
請按任意鍵繼續 . . .
Please guess the password! >> 1234
You got 0 bulls and 1 cows (0A1B), try again!
Please guess the password! >> 5678
You got 0 bulls and 2 cows (0A2B), try again!
Please guess the password! >> 1278
You got 0 bulls and 1 cows (0A1B), try again!
Please guess the password! >> 5632
You got 1 bulls and 2 cows (1A2B), try again!
Please guess the password! >> 5634
You got 0 bulls and 2 cows (0A2B), try again!
Please guess the password! >> 8692
You got 1 bulls and 1 cows (1A1B), try again!
Please guess the password! >>

Tips: 寫好程式的小訣竅

- 勤寫註解(非常重要！)
- 使用有意義的變數名稱
- 保持各層級縮排一致

- *「傻瓜都能寫出電腦能理解的程式；*
- *優秀的工程師寫出的是人類能讀懂的程式。」*

QUESTIONS?

Just Ask!

THANK YOU!

Enjoy Coding!