

The background is a dark blue gradient with a subtle pattern of white dots. Overlaid on this are several white geometric elements: a large circular scale on the left with tick marks and numbers (160, 170, 180, 190, 200, 210, 230, 240, 250, 260); a smaller concentric circle with an arrow on the top left; another concentric circle with an arrow on the top right; and a dashed circle with an arrow on the bottom left.

NUMBER GUESSING

2015 NCKUEE Freshman Camp

2015/08/19

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

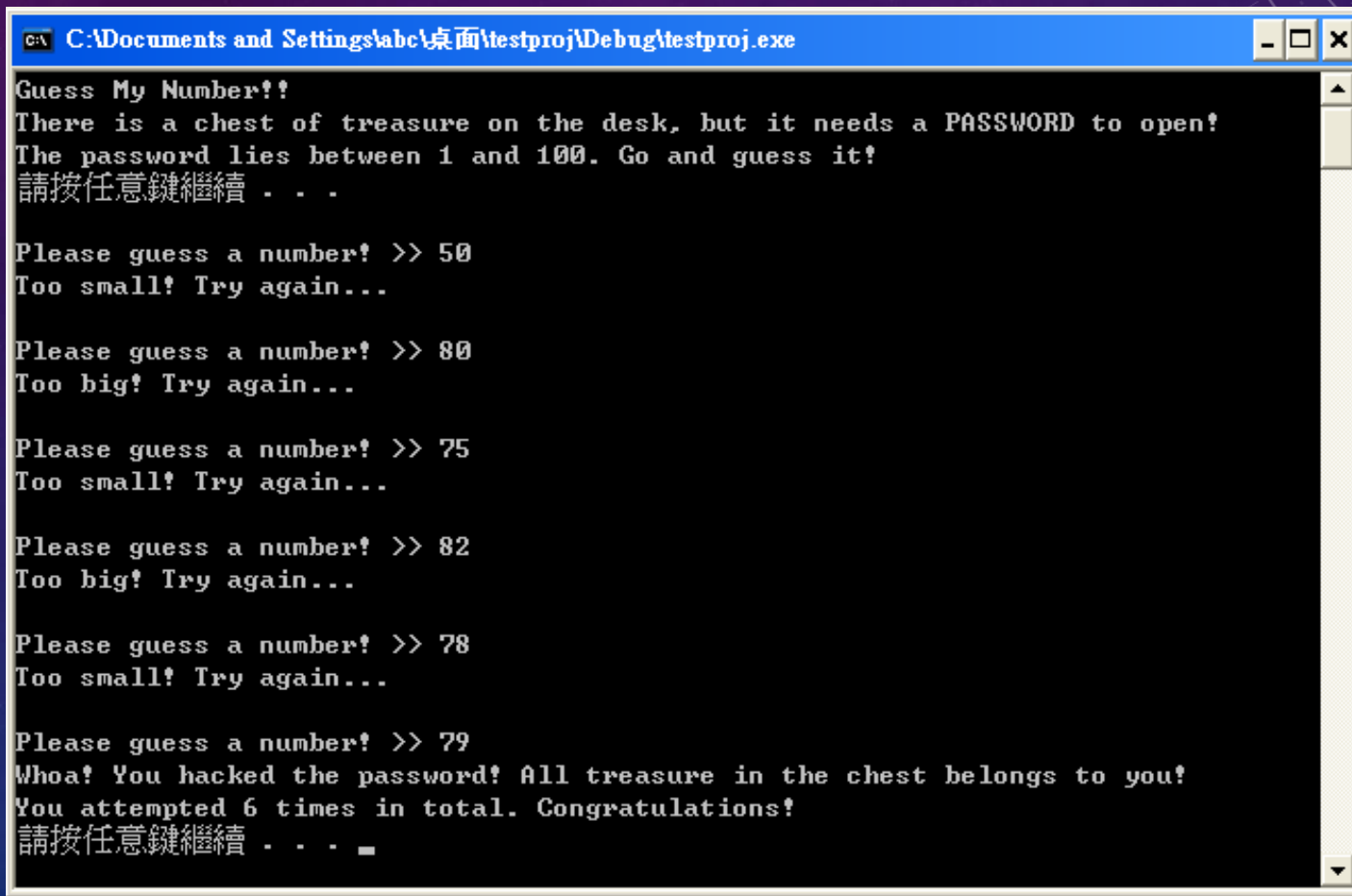
```
/* Start guessing!! */
while(!password_hacked) {
    attempts++;
    cout << endl << "Please guess a number! >> ";
    ?????????? // COMPLETE IT!!! ---> GET a input from keyboard, then stores it in variable "guess"

    /* determine the number entered 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

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。」

The background is a gradient of dark blue and purple, speckled with small white dots. On the left side, there are several concentric circles and a large circular scale with degree markings from 150 to 260. Some of the circles have arrows indicating a clockwise direction. The overall aesthetic is technical and futuristic.

QUESTIONS?

Just Ask!

The background is a gradient of dark blue and purple, speckled with small white dots. On the left side, there are several concentric circles and a large circular scale with degree markings from 160 to 260. Some of the circles have arrows indicating a clockwise direction. The text 'THANK YOU!' is centered in the middle-right area in a large, bold, white sans-serif font.

THANK YOU!

Enjoy Coding!