Lab of Object-Oriented Programming:
I/O String

黃威、陳岳紘、邱彥翔

2022

Google Meet 連結（已更新在moodle）:

https://meet.google.com/ywz-roab-ajq（請使用學校信箱加入）

Outline

- I/O
- String
- Coding Style
- Exercise2

Standard Input & Output

C

#include <stdio.h>

Input :

getchar(), gets(), scanf() ...

Output :

putchar(), puts(), printf()…

```
#include <iostream>
using namespace std;

cout << " Something to display";
```

C++

#include <iostream>

using namespace std:

Output :

cout(Operator: <<)

```
#include <iostream>

// Use cout of std library
std::cout << "Something to display";
```

String

- Difference between C & C++
- Declaration
- Useful function

String in C & C++ ?

For C :

string 是 char 陣列的包裝, 所以函式本質上是以 char * 或 char[] 來操作, 並不是一個既有的資料型別 (一個一個字存)

For C++ :

string 是一個 class 位於標準函式庫中, 有豐富多元的 function 可供操作 (一個字串存)

#include

For C :

```
#include <string.h>
```

For C++ :

```
#include <string>
#include <cstring>  //C strings and arrays
```

宣告 String

```
string s1;   // 建立空字串物件

string s2 = "my string";
string s3("my string");     } 設置 string 初始值

string s4 = s2;
string s5(s4);              } 複製字串

string s6(s2, 3);        // 以 s2 第三個字元後的字串作為初始值
string s7(s2, 3, 3);     // 以 s2 第三個字元後的3個字元作為初始值

string s8[2];  // 建立字串陣列
s8[0] = "hello";
s8[1] = "world";
```

String 長度

size() : 回傳字串長度

```
s1.size();
```

length() : 回傳字串長度

```
s1.length();
```

empty() : 如果是空字串, 則回傳 true

```
s1.empty();
```

String 交換、串接

swap() : 兩字串內容互換

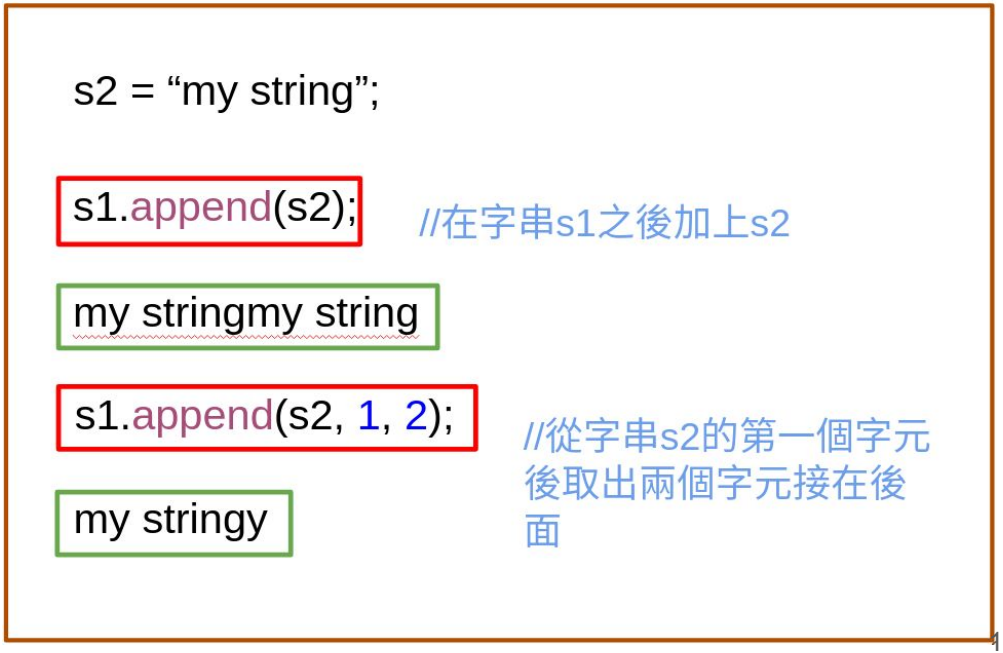
s1.swap(s2);

append() : 連接兩字串

s1.append(s2);

也可以直接使用 "+"

⇨ s2 = s1 + s2



String 修改

assign() : 指定字串

s1.assign(s2);

```
s1 = “my string”;

s1.assign(s2);
s2.assign(s3, 1, 2);

my string
y
```

//複製s2裝到s1中

//將s3第1個字元起2個字元當作s2內容

insert() : 將字串插入指定index的位置

s1.insert(index, s2);

```
s2 = “my string”;

s1.insert(2, s2);

mymy string string
```

//將s2插入s1 從 s1的第2個字元

子字串、取代

substr() : 取得子字串

s1.substr(s2);

replace() : 取代字串

s1.replace();

```
s1 = “my string”;

s1.substr(3);          // 取得s1第3個字元之後的字串
s1.substr(0, 2);       // 取得s1第0個字元之後的2個字元

string
my
```

```
s1 = “my string”;
s2 = ”my string”;

s1.replace(2, 4, s2);  // 將s1第2個字元之後的4個字元
                          清除並插入s2

mymy stringing
```

String 搜尋

find() : 搜尋字串中目標出現的位置

```
s1.find(s2);
```

find_first_of() : 搜尋字串中目標第一次出現的位置

```
s1.find_first_of(s2);
```

find_last_of() : 搜尋字串中目標最後一次出現的位置

```
s1.find_last_of(s2);
```

*若找不到則回傳-1

String 比較

- 字串由char(字元)所構成, 依照ASCII碼的編號比較大小
- 比較方式
  - Equality (相等) 運算子 == 、!=
  - Relational (相對關係) 運算子　<、<=、>、>=
  - 使用函式 compare()

String 比較 - 運算子

若兩子串長度不同, 但 s1 的前段與 s2 相同時, 則 s1 > s2

s1 = abcde, s2 = abc ⇨ s1 > s2

若兩字串完全不相同, 用第一個不相同的字元進行比較

s1 = abcde, s2 = acdef ⇨ s1 < s2

s1 = bcdef, s2 = abcde ⇨ s2 < s1

String 比較 - compare()

compare() 輸出值有三種, 比較兩個指定的 String 物件, 傳回一個整數, 表示兩者在排序順序中的相對位置

s1.compare(s2)

- >0 ⇨ s1 > s2
- =0 ⇨ s1 = s2
- <0 ⇨ s1 < s2

Coding Style

While programming…….

- Appropriate indentation and space
- Meaningful names
- Comment
- Error messages

Appropriate indentation and space

- 好的排版留給自己寫Code好看 也給他人欣賞方便
- 方便Debug 例如括號or分號
- ~~看到亂亂的code心情就不好了~~

Meaningful Names

- 依照你要做的事情幫變數和函式命名，避免過於簡短或無意義的符號
- No “magic number”
  - 盡量避免直接使用數字，因為對於他人，甚至過了一段時間後你自己可能都會難以理解此數字的意義及功能
  - 當數值需要變動時，要改變的可能不止一處

#define -> const

```cpp
#include<iostream>
using namespace std;

int main(){

    double radius;
    double area, perimeter;

    cout << "Please enter the circle's radius: ";
    cin >> radius;

    area = 3.14 * radius * radius;
    perimeter = 3.14 * 2 * radius;

    cout << "The area is: " << area << endl;
    cout << "The perimeter is:" << perimeter << endl;

    return 0;
}
```

```cpp
#include<iostream>
using namespace std;

#define PI 3.1415926

int main(){

    double radius;
    double area, perimeter;
```

C++: const double PI = 3.1415923

C++11: constexpr double pi() { return std::atan(1)*4; }
C++20: std::numbers::pi

Comment

養成寫註解的習慣有助於日後或者他人理解程式碼的內容

- 描述程式的功能、演算法、邏輯
- 目前可能發生或已經發現的 BUG
- 開發進度

Error Message

練留看error message, 現在 compiler 很強大, 很多時候只要看錯誤訊息就可以初步排除bug了

```
week2.cc:8:2: error: use of undeclared identifier 'doule'; did you mean 'double'?
        doule radius;
        ^~~~~
        double
week2.cc:11:46: error: expected ';' after expression
        cout << "Please enter the circle's radius: "
                                                    ^
                                                    ;
2 errors generated.
```

Exercise 2

Exercise 2

- 從使用者選單 ->「參加課程」-> 輸入「課程代碼」:121017a87712d72db5d4b5afb5f73227

- **務必點擊「參加競賽」參與作業測驗**，否則助教會看不到你的上傳結果

Balance String

Content ：
Balanced strings are those that have an equal quantity of `'L'` and `'R'` characters.

Given a balanced string `s`, split it in the maximum amount of balanced strings.

Return the maximum amount of split balanced strings.

Constraints:

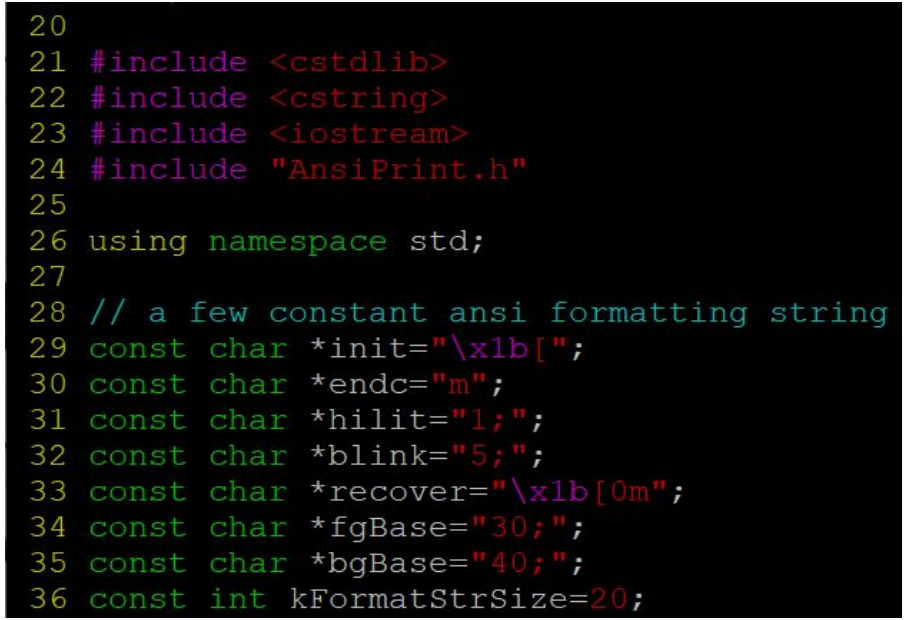
- `1 <= s.length <= 1000`
- `s[i]` is either `'L'` or `'R'`.
- `s` is a balanced string.

Input	Output	測資資訊：
• A string that have an equal quantity of `'L'` and `'R'` characters	• Maximum amount of split balanced strings	記憶體限制： 64 MB 公開 測資點#0 (100%): 1.0s , <1K

Sample Input	Sample Output
RLRRLLRLRL	4

為你的 vim 加上行號與顏色

```
[s110xx@oop]~# cp ~s10817/.profile .
[s110xx@oop]~# cp ~s10817/.vimrc .
[s110xx@oop]~# cp ~s10817/.bashrc .
```



```
20
21 #include <cstdlib>
22 #include <cstring>
23 #include <iostream>
24 #include "AnsiPrint.h"
25
26 using namespace std;
27
28 // a few constant ansi formatting string
29 const char *init="\x1b[";
30 const char *endc="m";
31 const char *hilit="1;";
32 const char *blink="5;";
33 const char *recover="\x1b[0m";
34 const char *fgBase="30;";
35 const char *bgBase="40;";
36 const int kFormatStrSize=20;
```

Assign1

Due: **10/02/2022 23:59:59**

遲交一天扣該 Assign 分數 **10%**

Assign 最遲繳交日: **10/09/2022 23:59:59**

超過最遲繳交日所繳交的 Assign 分數以 **0 分** 計算

Assign1

Topic: Printing a given number of cards from a deck in a nice format.

Objective: Review string, array, and pointer in C and practice new features in C++.

Description:

You are asked to draw poker cards from a deck of 52 cards and print them to the screen in a nice format. The input to the program is the number of cards to be printed, and the output is a graphical representation of the cards on a regular ASCII terminal. Each card is randomly drawn from a deck of 52 cards ordered by pip, (i.e. 1S 1H 1D 1C 2S 2H 2D 2C .....). The output of the cards will be formatted such that up to 5 cards are arranged in a row. Extra cards are automatically wrapped over to the next row. A drawn card should not reappear. Therefore, the number of cards that a user asks to display should not exceed 52. If it does, you should print an error message such as “Sorry, number of cards can not exceed 52.”

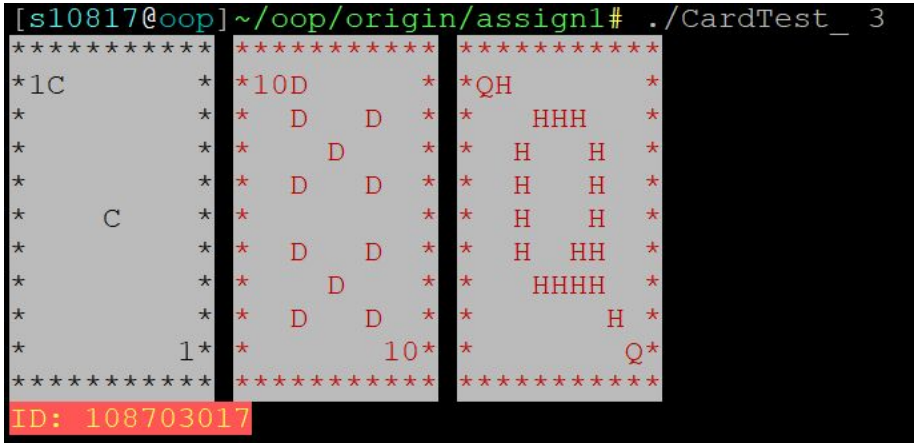
Assign1

Sample exec.

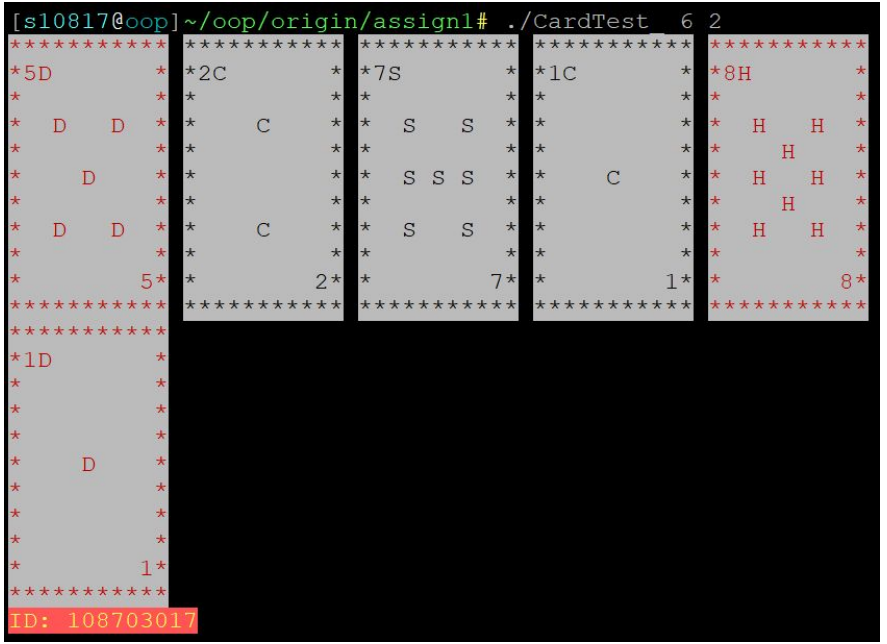
```
mtchi@oop 10:34am [86] assign1> CardTest_x64-sample
Usage: CardTest_x64-sample NCards [Seed]
mtchi@oop 10:34am [87] assign1> CardTest_x64-sample 3
**********  **********  **********
*6D      *  *8C      *  *1S      *
*         *  *         *  *         *
*   D   D   *  *   C   C   *  *         *
*         *  *     C     *  *         *
*   D   D   *  *   C   C   *  *     S     *
*         *  *     C     *  *         *
*   D   D   *  *   C   C   *  *         *
*         *  *         *  *         *
*        6*  *        8*  *        1*
**********  **********  **********
ID: u1234567
```

Your Student ID

Assign1



Your Student ID



Assign1

What are given:

You are given the following files to start with:

- a sample Makefile: Makefile
- a partial testing file: CardTest.cc
- a sample header file: Cards.h
- a pair of files for printing nice looking string on an ANSI-enabled terminal: AnsiPrint.cc and AnsiPrint.h. (See the comments in the files for how to use the given functions to print a string with special ANSI control characters.)

Assign1

What to hand in:

You are asked to write two C++ source files: CardTest.cc and Cards.cc.
If you name your source files differently, remember to modify the Makefile accordingly and hand it in with other source files.

A sample executable program compiled for the server (CardTest_x64-sample) is also included in the assignment directory for your reference. You must submit all program source code electronically.

Any questions?