STL

5/24, HYDAI

INTRODUCTION

- Standard Template Library
- Container
  - vector, list, string
- Iterator
- Algorithm
  - sort, next_permutation, random_shuffle

注意事項

- 需要 include 對應的 header
- STL 屬於 namespace std
  - 直接寫上 using namespace std;
  - 或是加上 std::

C++ 的輸出入

#include <iostream>

CIN, COUT

- 以前用的 scanf, printf 是屬於 C 的輸出入函式
- cin, cout 則是屬於 C++ 的輸出入方法
- 搭配上 STL ，用 cin, cout 有時候比較方便
- 不過 cin, cout 很慢，比 scanf, printf 慢很多

SYNTAX

- 輸入：`cin >> vars;`
- 輸出：`cout << vars;`
- 輸出換行：`cout << endl;`
- 不需要指定形態，C++ 會自動偵測

DEMO - CIN, COUT

C++ 的 String

#include <string>

STRING

- 神好用
- 有點慢
- 用太多小心 TLE

宣告

- `string str;`
- `string str = “”;`
- `string str = “hello, world”;`
- `string str = <char array>`

方法

- length(): 回傳字串長度
- clear(): 清除字串（等同于 str = “”）
- operator[index]: 和 C 字串一樣取 index 的值
- at(index): 等同于 operator[index]
- c_str(): 回傳 C 字串（const char array）

DEMO - STRING

可動態增長的陣列

#include <vector>

VECTOR

- 動態陣列
- 可以隨機存取，如果我們熟悉的陣列
- 隨機插入很慢，如果有這個需求，用 `list`

宣告

- `vector<type> V;`
- `type` 必須放你想要的形態
- 比如說一個 `int` 的陣列就填上 `int`
- 一個 `string` 的陣列就填上 `string`
- 一個 `struct` 的陣列就填上 `struct`

方法

- size():回傳 vector 大小
- clear(): 清除陣列
- operator[index]:行為同陣列的 []
- push_back(vars): 從尾端加入一個元素
- pop_back(): 從尾端拔掉一個元素

DEMO - VECTOR

鏈結串列

#include <list>

LIST

- 就是我們學過的 `linked list`
- 無法隨機存取
- 支援隨機插入

宣告

- `list<type> V;`
- `type` 必須放你想要的形態
- 比如說 `int` 的 `list` 就填上 `int`
- `string` 的 `list` 就填上 `string`
- `vector<int>` 的 `list` 就填上 `vector<int>`

方法

- size():回傳 list 大小
- clear(): 清空整條串列
- push_front(vars): 從前端加入一個元素
- push_back(vars): 從尾端加入一個元素
- pop_front(): 從前端拔掉一個元素
- pop_back(): 從尾端拔掉一個元素

DEMO - LIST

迭代器

ITERATOR

- 我要怎麼拜訪 STL 包裝好的結構？？
- 有沒有一個統一的方法呢？？
- 答案就是 `iterator`

宣告

- `container<type>::iterator iter;`
- `container<type>::` 放要迭代的容器形態
- 對應的 STL 通常會有 `begin()`, `end()` 代表 `iterator` 的開頭與結尾

用法

- 感覺很像指標
- `string::iterator iter = str.begin();`
- `*iter = str[0];`
- `iter++;`
- `*iter = str[1];`

DEMO - ITERATOR

演算法

#include <algorithm>

ALGORITHM

- 一堆神好用的東西ＯＡＯ
- 由儉入奢易，由奢入儉難，切記，切記

SORT

- 對 `int arr[100];`
- `sort(arr, arr+100);`
- 對 `vector<int> V;`
- `sort(V.begin(), V.end());`

NEXT_PERMUTATION

- 下一個排列
- 要用前請先排序好
- next_permutation(begin, end);
- 要用 do {
- }while(next_permutation(…));

RANDOM_SHUFFLE

- 打亂原本的東西
- 先塞一堆東西進容器，利用它來打亂
- 看起來有隨機變數的效果
- `random_shuffle(begin, end);`

DEMO - ALGORITHM

練習、大樂透開獎

- 總共有 52 顆球，分別從一號到五十二號

- 你的任務是每次都要能跑出六個普通號 & 一個特別號，同一個號碼不能重複

練習、遞減的梵蒂岡聖光

- 129
- 用今天學的東西試試看吧！

作業、文字轉轉轉

- 153
- 跟排列有關ＯＡＯ～

作業、二十一點

- 設計雙人對戰的二十一點
- 規則：
  - 先拿到二十一點者贏，但如果超過則為輸
  - 兩人都沒到二十一點，則比誰的分數高
  - A 可當作是 1 或 11
  - 2~9 就是 2~9
  - 10, J, Q, K 當作是 10

作業、二十一點（CON'T）

- 牌堆裡共有三副牌（52*3張牌）
- 每張牌只會有三張，不能多也不能少
- 發牌需要隨機，不能每次玩都固定發某些牌

作業、二十一點（CON'T）

- 繳交方式：
- 請壓縮後寄到 z54981220@gmail.com
- 信件名稱：資訊之芽 – 21 – 你的名字
- 請撰寫一份 readme 告訴應該怎麼操作
- 無法進行操作者，作業分數保留

參考

- http://www.cplusplus.com/reference/stl/
- 有些功能需要更多的功能，請參考上面連結

LINKED LIST OLD

- 180

- 182