



1. C++介紹

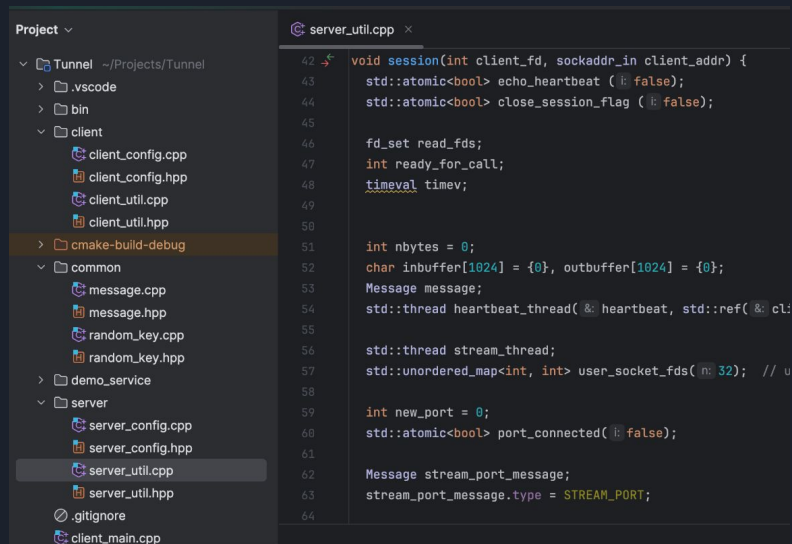
2024資訊研究社語法班
Made with ❤️ by jheanlee

我們要寫的程式

- 程式比賽導向 -> 解題目 快速寫完
- 軟(韌)體 -> 讓之後比較好懂 不注重省資源 清楚明白的程式 (可能逐漸被高階語言取代)

這學期要上的重點是第一個目標是要讓各位能掌握到**競速程式**的基礎
當然 這個基礎也能幫助之後C++軟體的學習

```
173
174 int main() {
175     fastio
176     int n, m;
177     cin >> n >> m;
178
179     int rtn = 0;
180     int v;
181     int v1, v2;
182
183     unordered_map<int, int> map; // location = key, num = val;
184
185     for (int run = 0; run < m; run++) {
186         cin >> v;
187         if (v == 1) {
188             cin >> v1 >> v2;
189             if (map.find(v1) != map.end() && map.find(v2) != map.end()) {swap(map[v1], map[v2]);}
190             else if (map.find(v1) != map.end()) {map[v2] = map[v1]; map[v1] = v2;}
191             else if (map.find(v2) != map.end()) {map[v1] = map[v2]; map[v2] = v1;}
192             else {map[v1] = v2; map[v2] = v1;}
193         } else if (v == 2) {
194             cin >> v1;
195             if (map.find(v1) != map.end()) {rtn += map[v1];}
196             else {rtn += v1;}
197         }
198     }
199
200     cout << rtn << endl;
201
202     return 0;
203 }
```



最最最基本的C++程式架構

可參照 basic.cpp

```
1  #include <iostream> // 標頭檔 取得電腦需要的code
2
3  using namespace std; // 將std的所有成員提到全域 (看不懂的話就把他當魔法) \
4  例如: std::string 可以直接用 string
5
```

#include <path_to_header> or #include "path_to_header"
告訴編譯器這個程式需要哪些功能 (對應的"標頭檔 header")

using namespace name_of_namespace;

*競速程式常用的做法 原本大部分的功能都被包在 "std" 的空間裡面
會讓原本 std底下的所有函式跑到全域

例如:

using namespace std;
std::string() 會變成 string()

using namespace std::chrono;
std::chrono::milliseconds 會變成 milliseconds

header

```
<string>
string()
append()
insert()
size()
```

```
<cmath> or <math.h>
cos()
sin()
tan()
```

```
"~/test/test.hpp"
f1()
f2()
```

test.cpp

```
#include <string>
#include <cmath>
#include " ~/test/test.hpp"
```

最最最基本的C++程式架構

註解

就是讓程式更好懂的東西

單行註解用//開頭

// 這後面的所有東西都是註解

多行註解用/*和*/包起來

/*

這中間

所有東西

都是註解

*/

```
12 // 這是註解
13
14 /* ← 這個是註解的開頭
15
16 這是貓貓
17    / |、
18    (°、 °7
19    |、~\
20    じし、)ノ
21
22 註解到這裡結束 → */
23
```

最最最基本的C++程式架構

C++的程式由上到下, 左到右執行

所有程式都由主程式開始

主程式就是 `main() {}`

```
25
26 // 你的程式會從 main() 開始跑
27 int main() { <- 主程式從這個大括號開始
28     cout << "Hello, World!" << endl;    // 每一行程式結束都有一個 分號(;)
29     cout << "Goodbye, World!" << endl;
30
31     return 0; // return 0; 是個好習慣 他告訴電腦: "我的程式很順利的跑完了"
    }
```

一直到這個大括號結束 ->