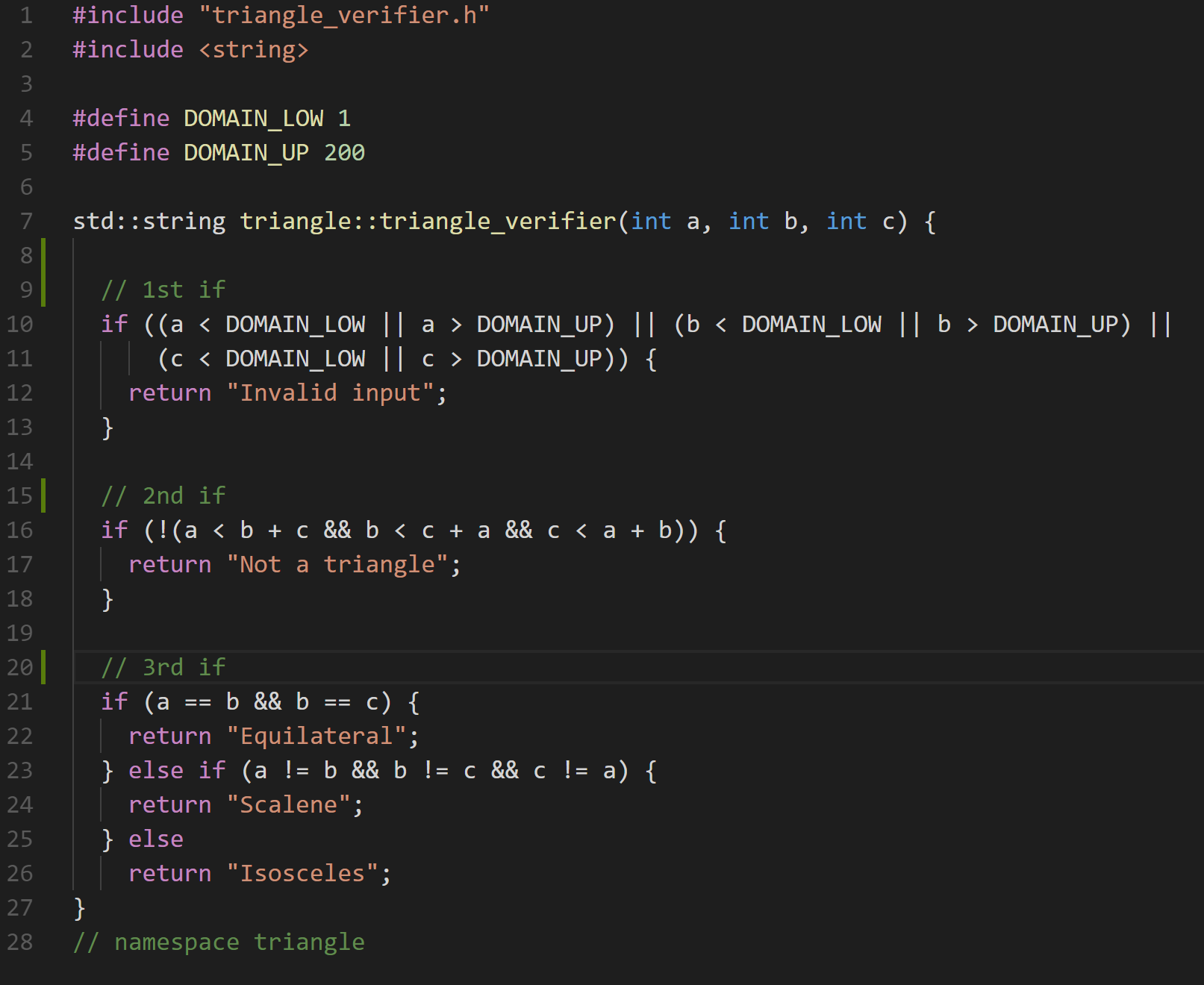
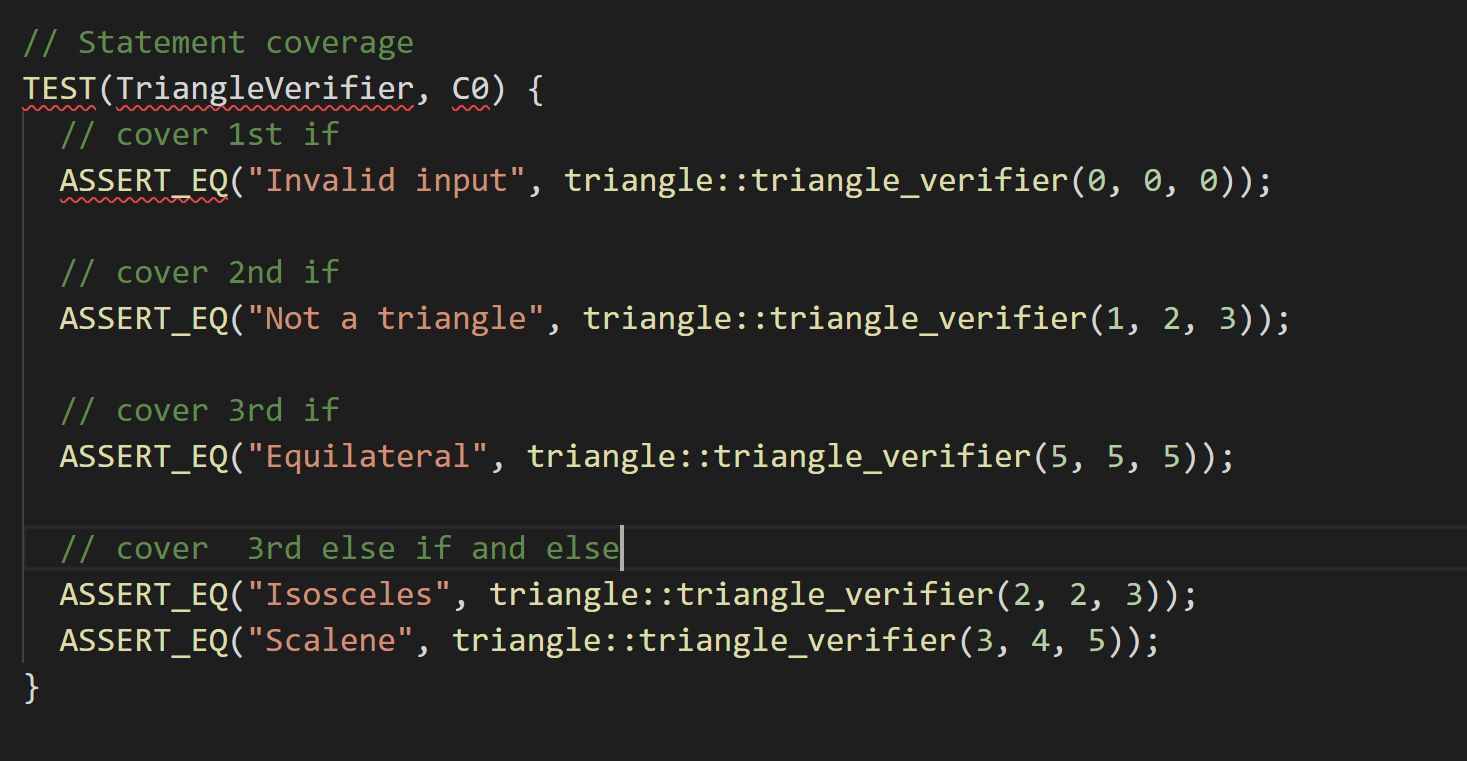
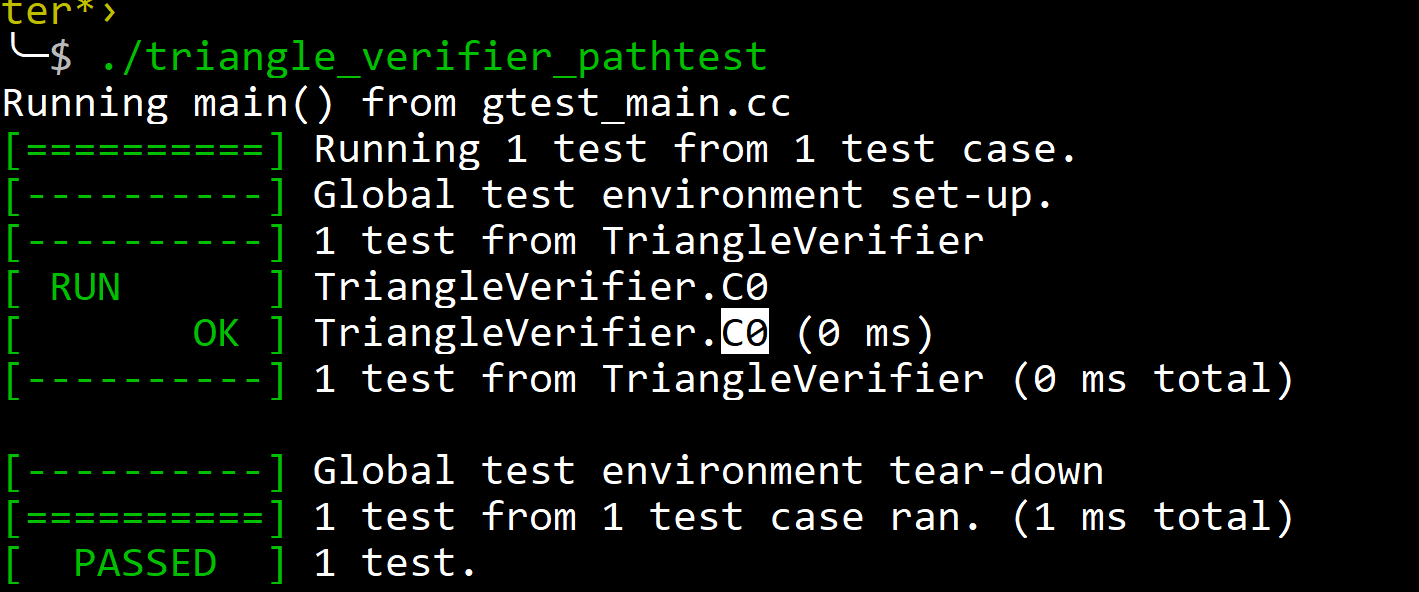
# **Software-Testing HW2**

* 沿用 HW1 的三個問題的實作，故直接開始寫 test

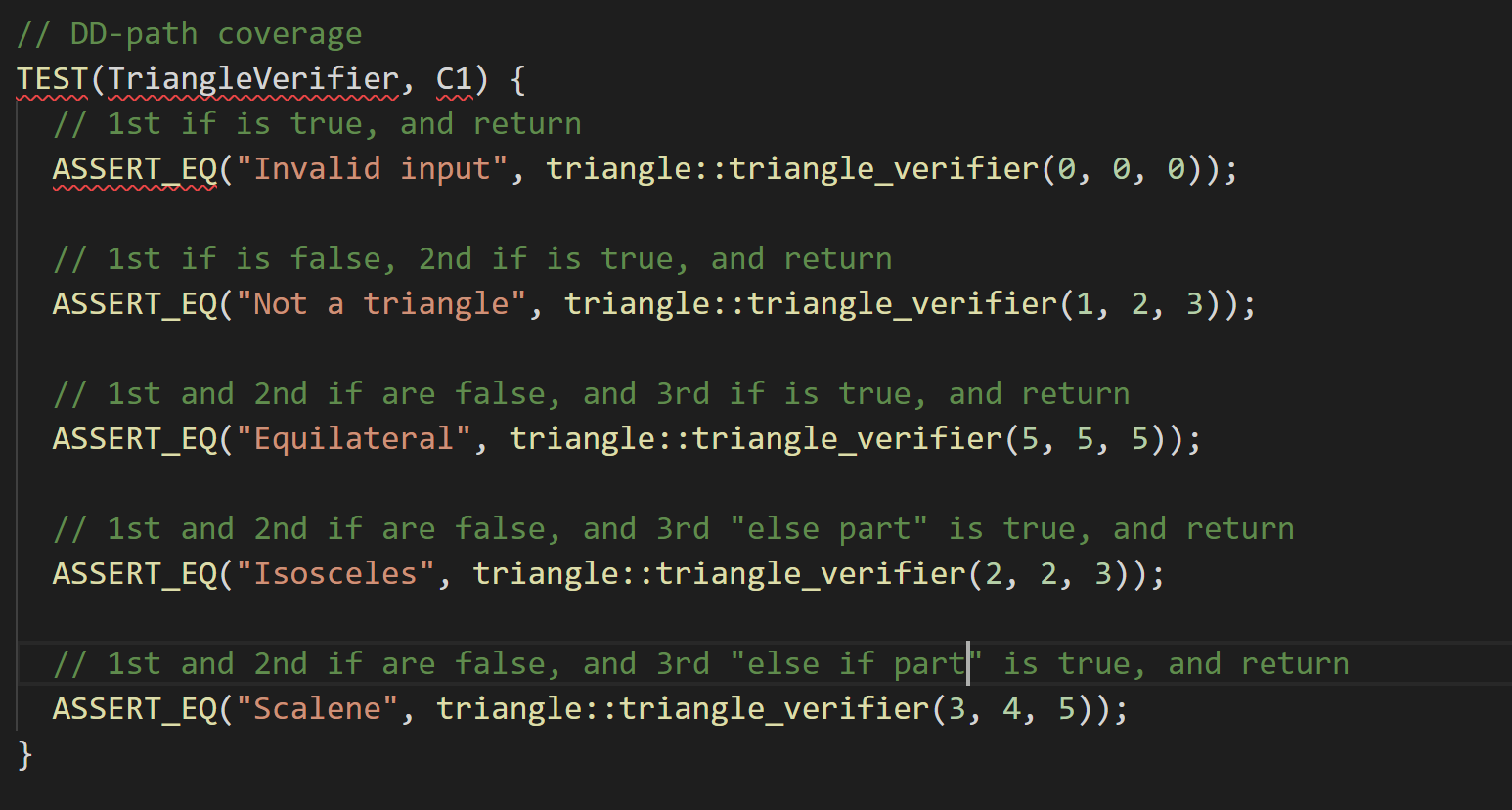
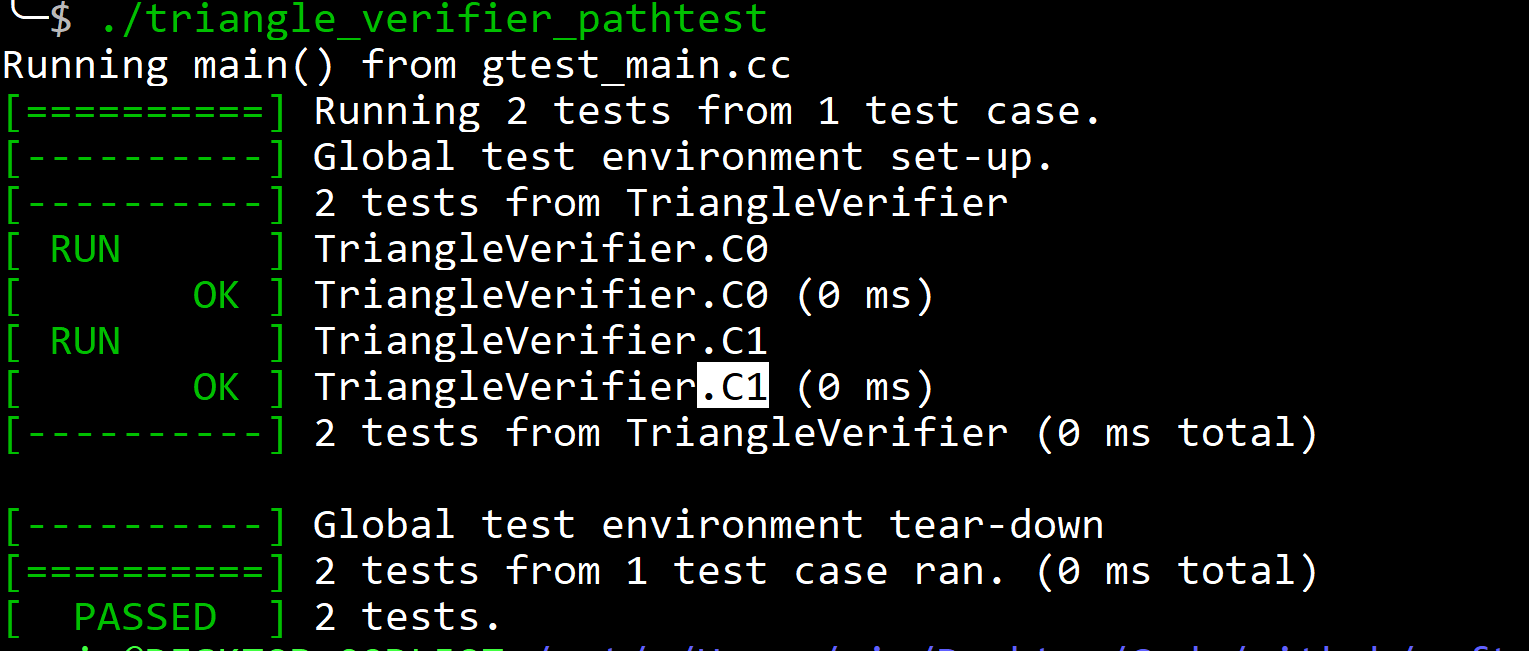
## **Triangle Problem**

* 先貼 code 才有辦法看 test cover 到什麼程度:
  + 為 code 增加註解，比較好對照
* 

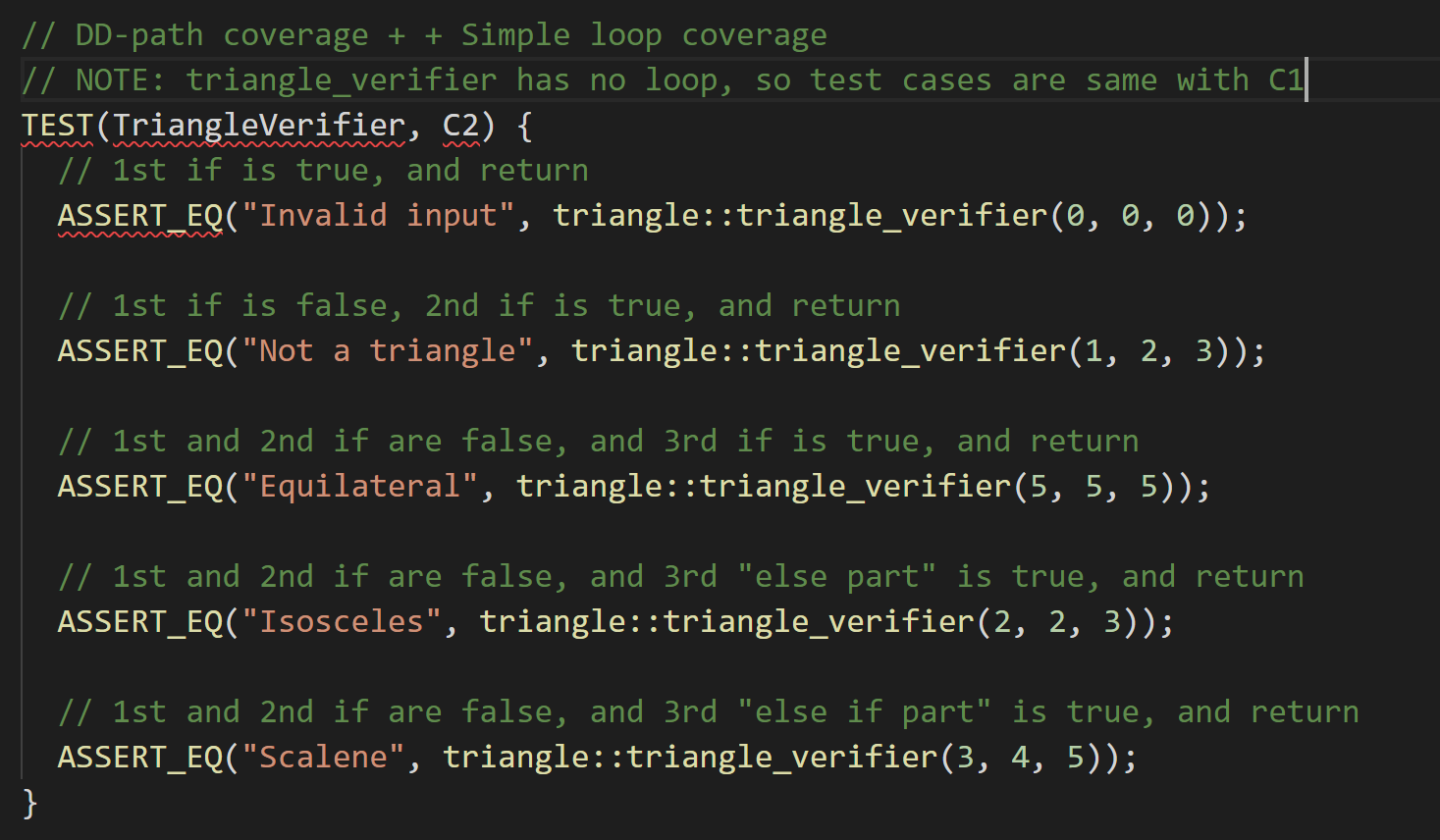
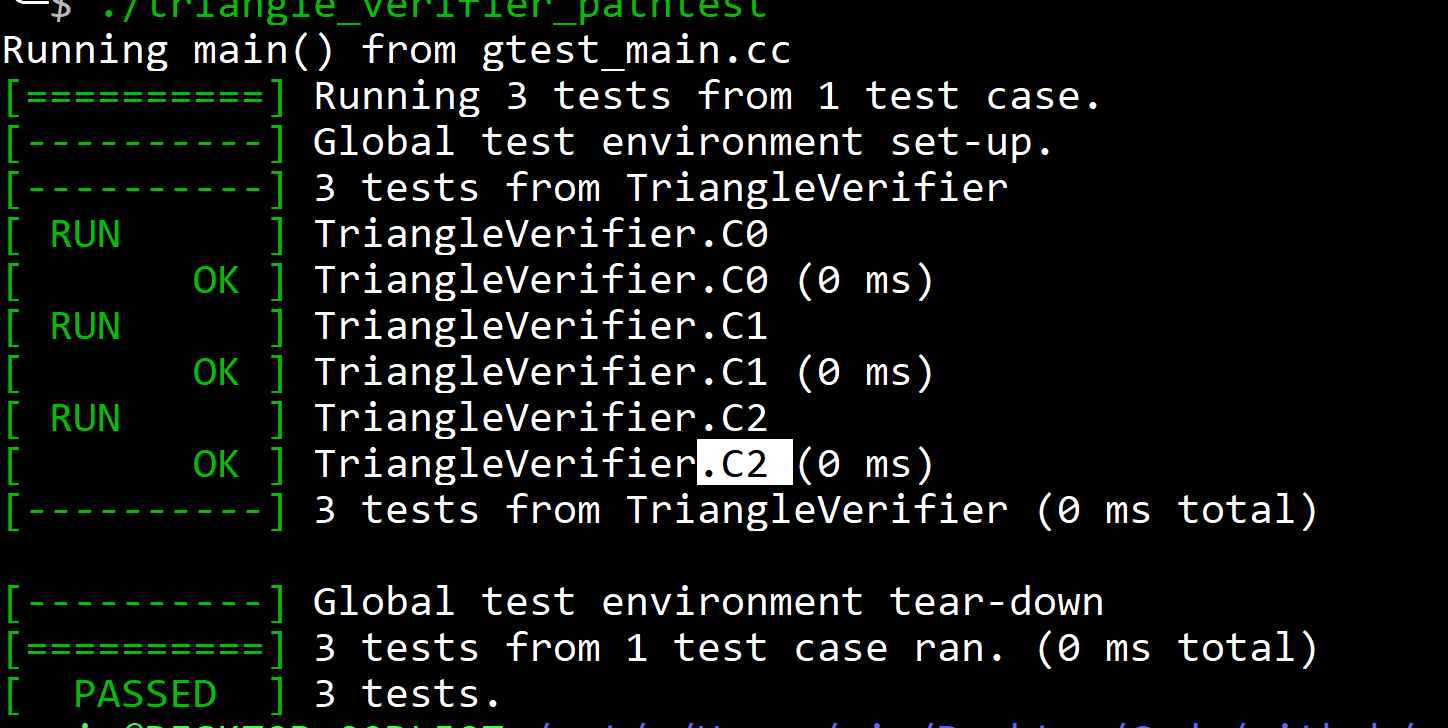
### **C0**

* 跑過所有 statements
* 
* 

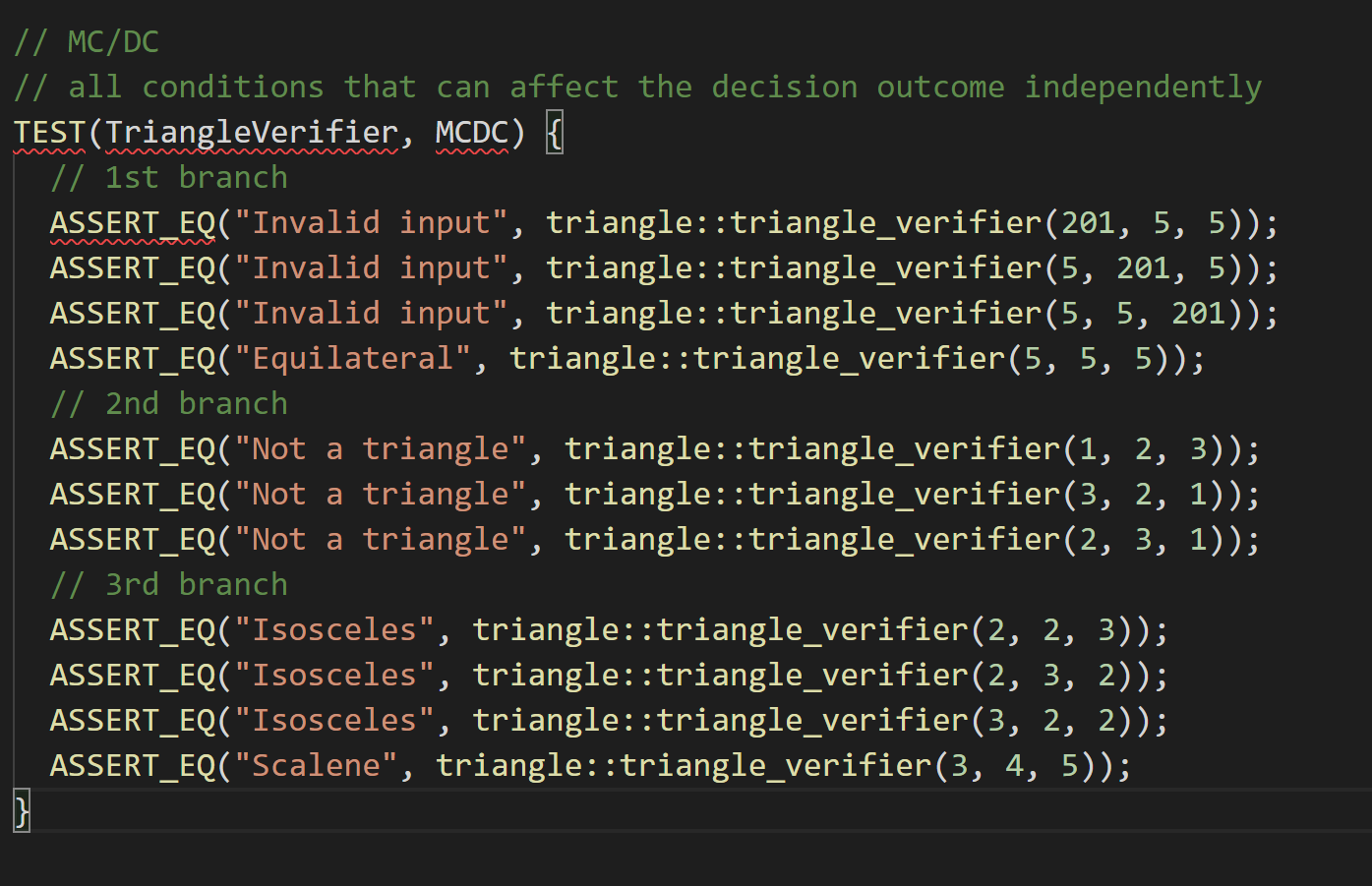
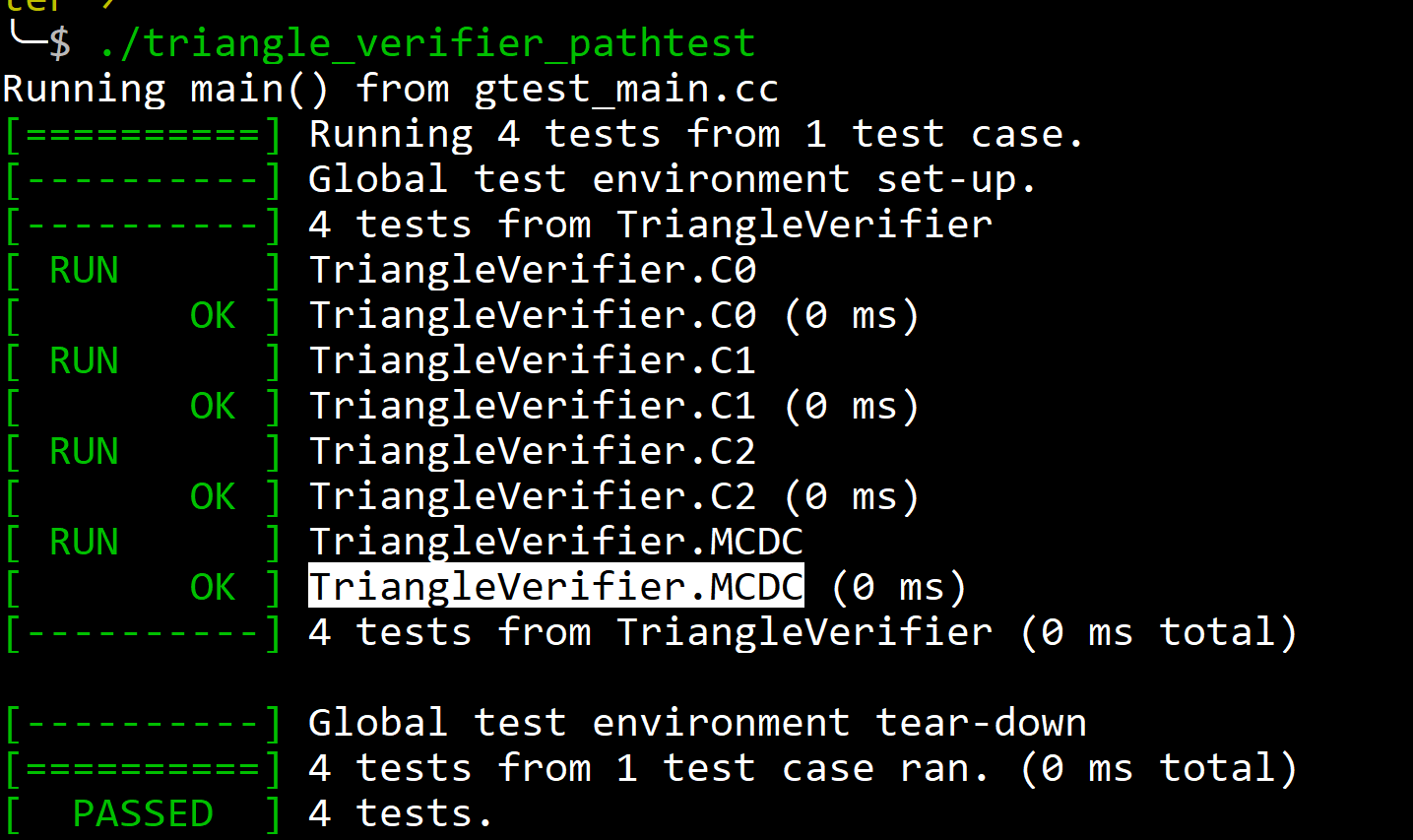
### **C1**

* 跑過所有 condition 的 branches
* 我的實作總共有七個 branches
  + 1st 有 2個，2nd 有 2個，3rd 有 3個
* 但我這個 test case 剛好會跟 C0 的一樣(C1 本來就 >=(subsume) C0)
* 
* 

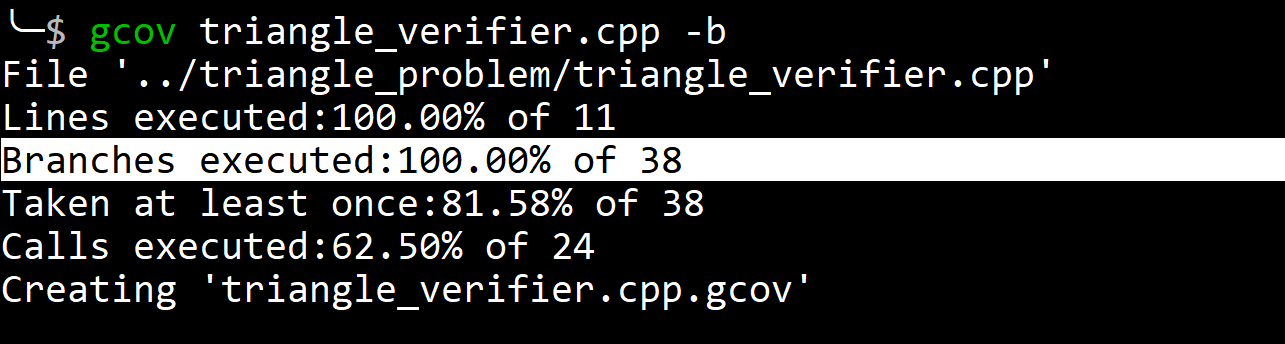
### **C2**

* 因為 triangle problem 沒有 loop，所以 C2 會跟 C1 一樣
* 
* 

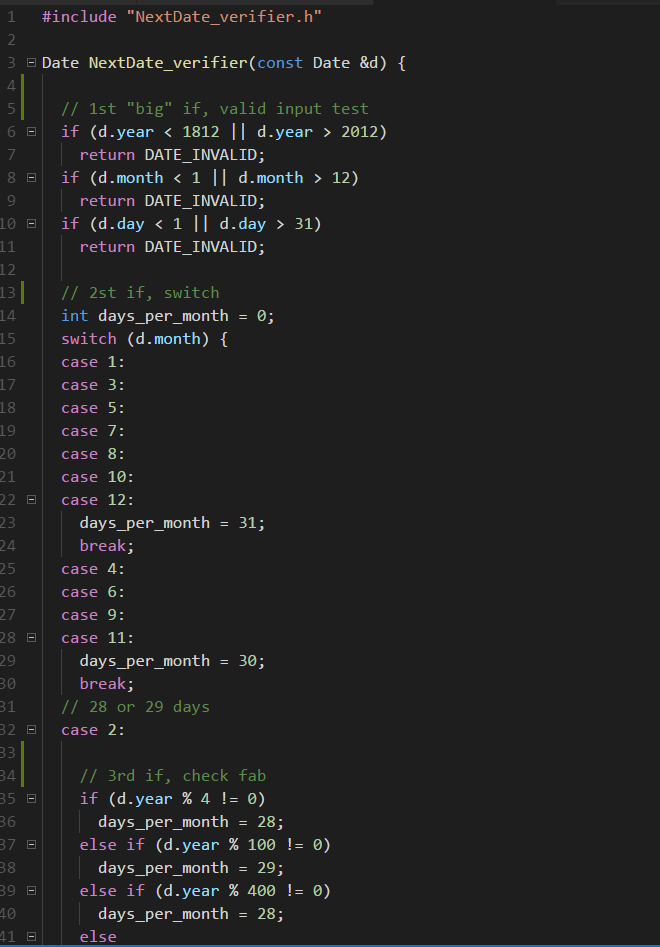
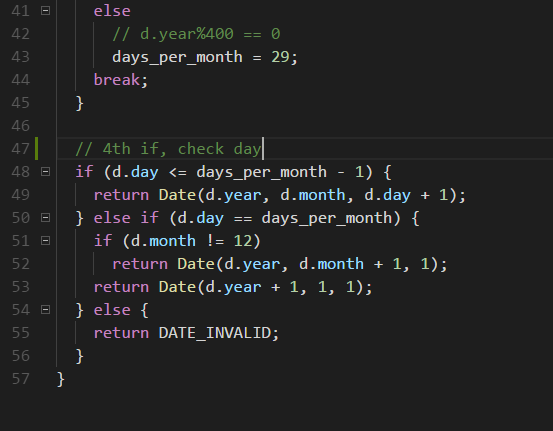
### **MCDC**

* 要把每個 branch 內的，會**獨立**影響 true/false 的 condition 都執行過一次
* 這裡比較麻煩的是三角形的三邊長彼此會互相依賴，例如 !(a < b + c && b < c + a && c < a + b) 這樣的 expression；
* 總之 code 是這樣:
  + 
  + 

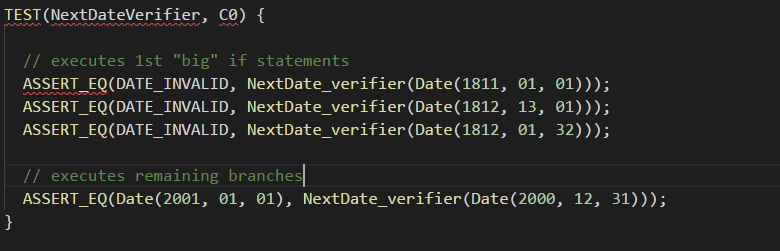
### **gcov 測試**

* statement coverage:
  + 
* branch coverage:
  + 

## **NextDate Problem**

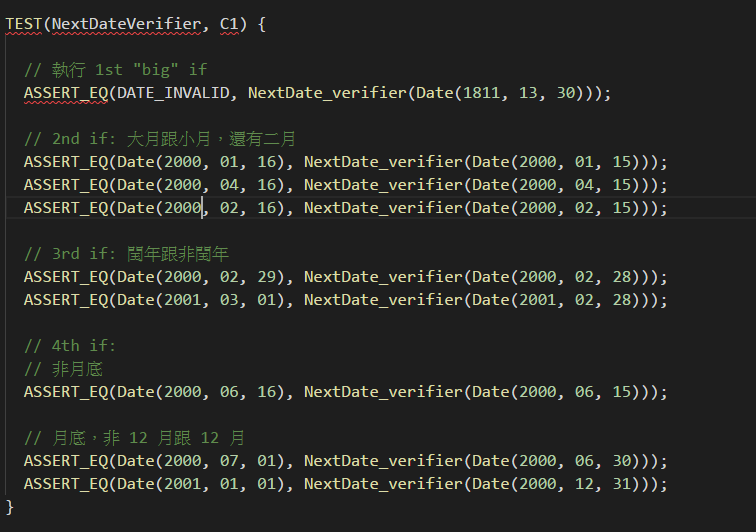
* 一樣先貼 code:
* 
* 

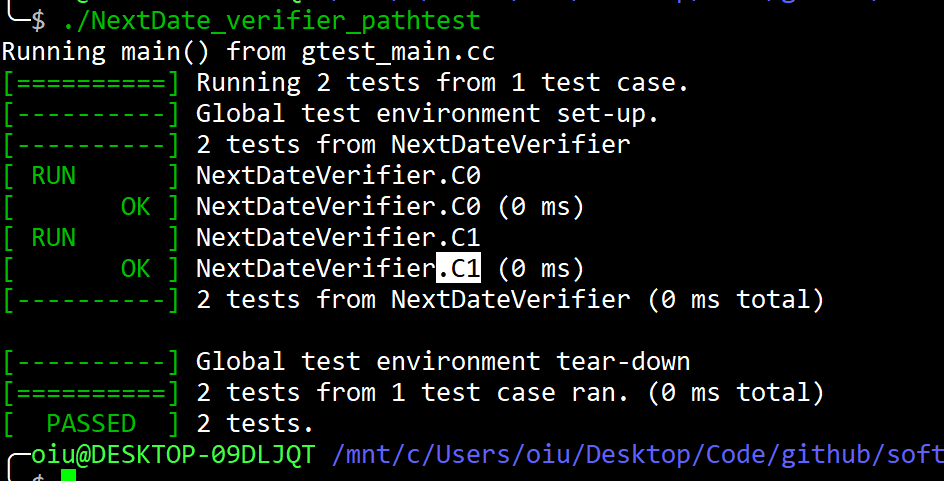
### **C0**

****

* 前面三個 tests 把 1st “big” if 都執行過，最後一個則是都會碰到剩下的 branches

### **C1**

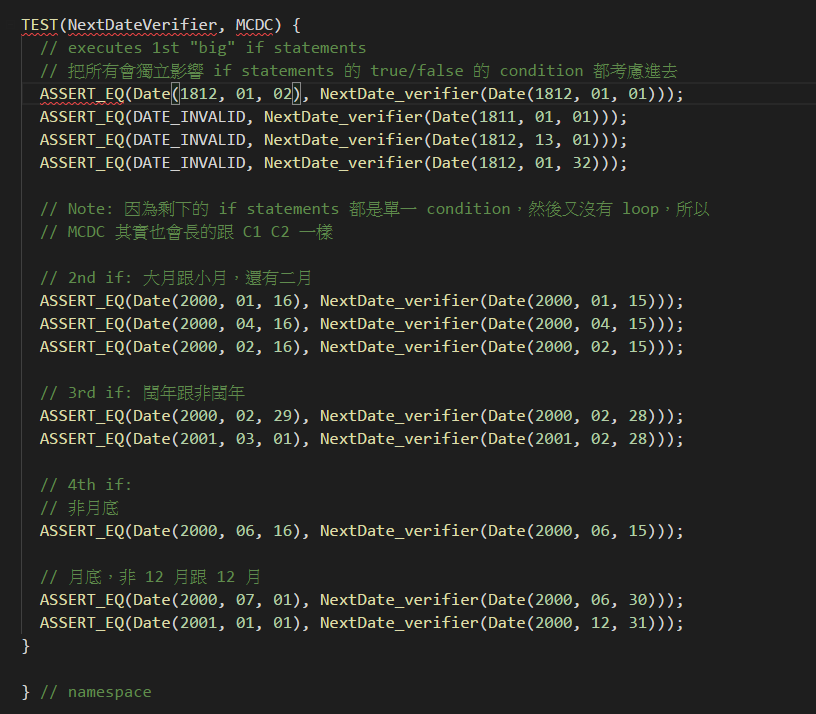
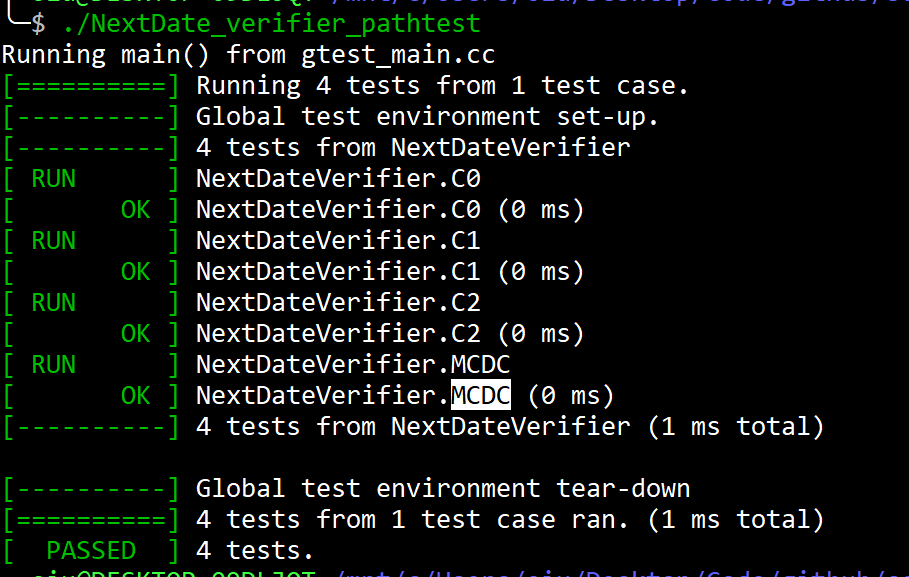
****

* 一樣把所有 branch 的 true/false 都走一次
* 

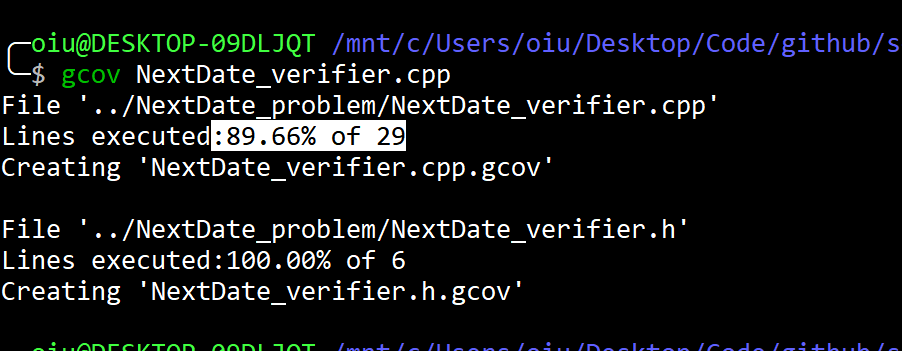
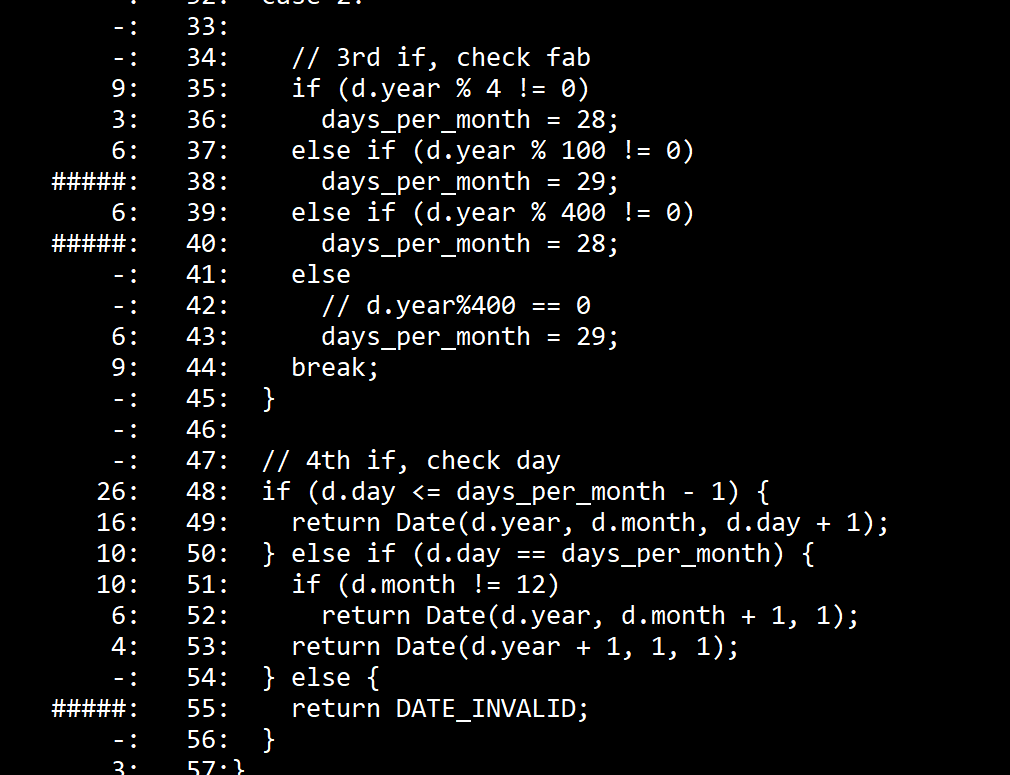
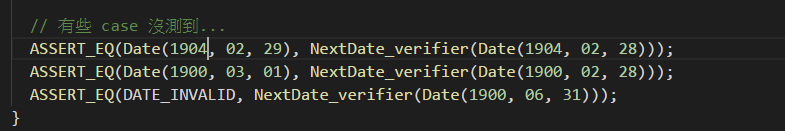
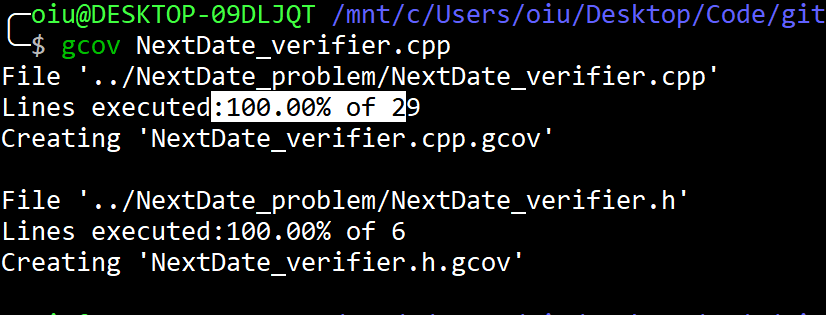
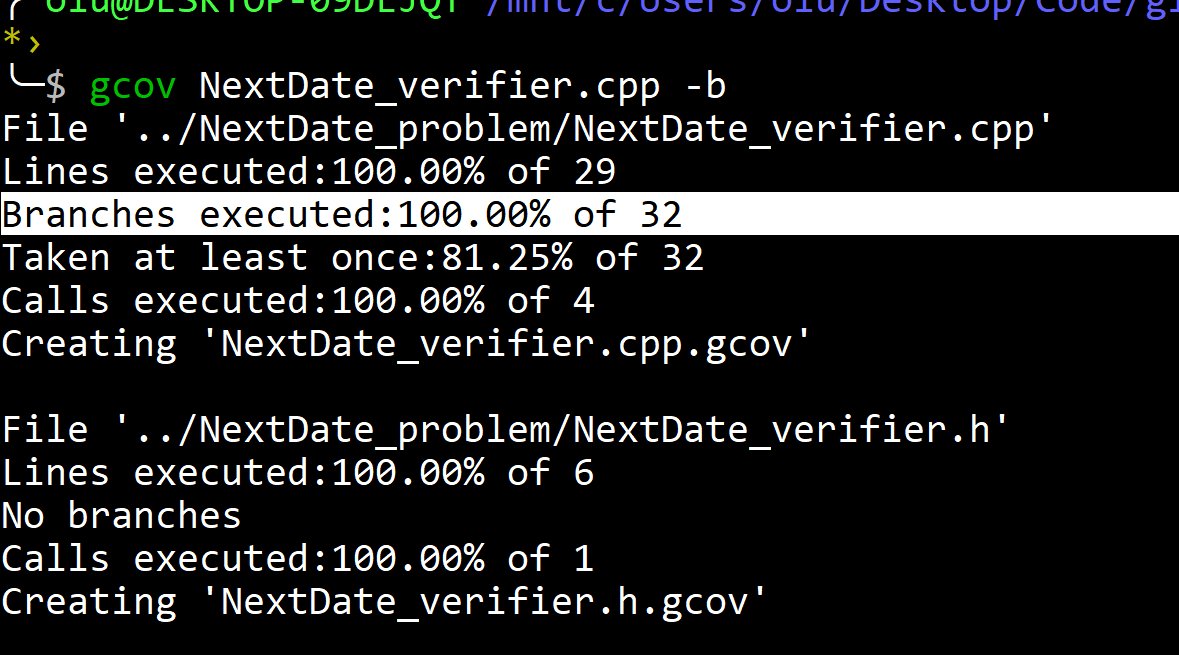
### **C2**

* 因為一樣沒有 loop，所以跟 C1 一樣

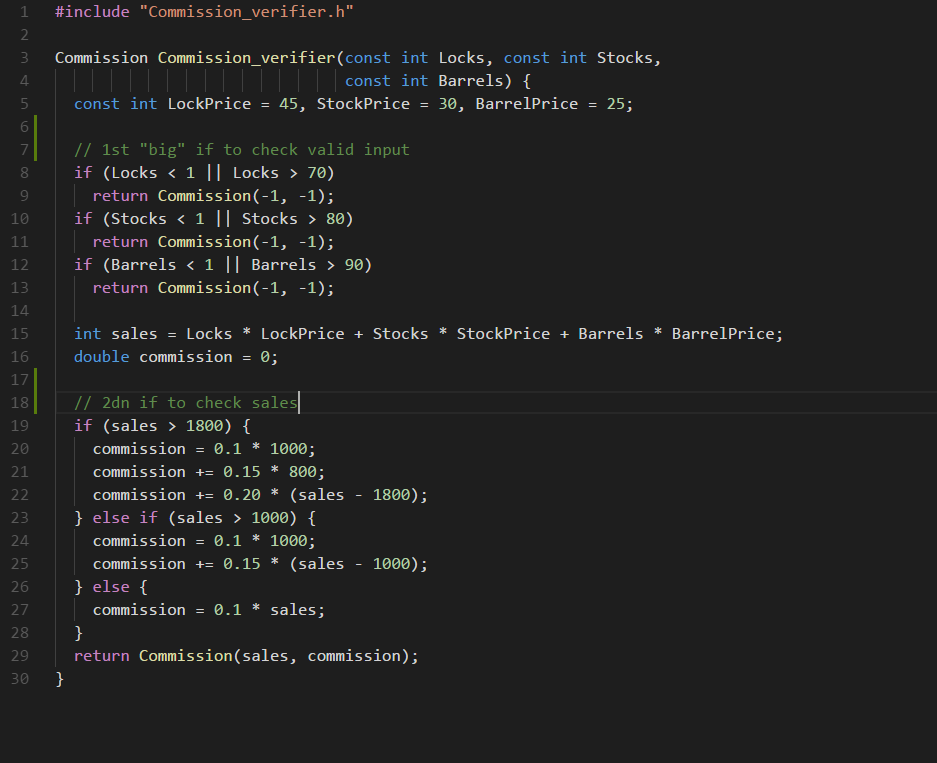
### **MCDC**

* 跟 C1 C2 差不多，因為大部分的 if statements 都是單一一個 condition(或者說只有一個變數)，然後又沒有 loop
* 
* 

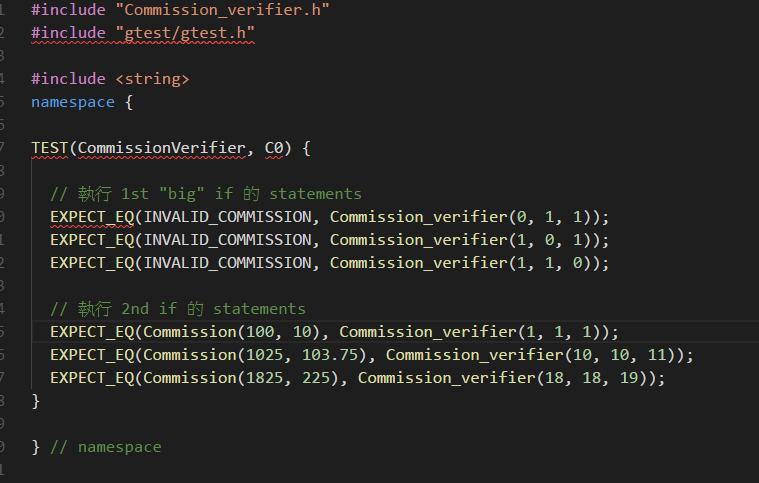
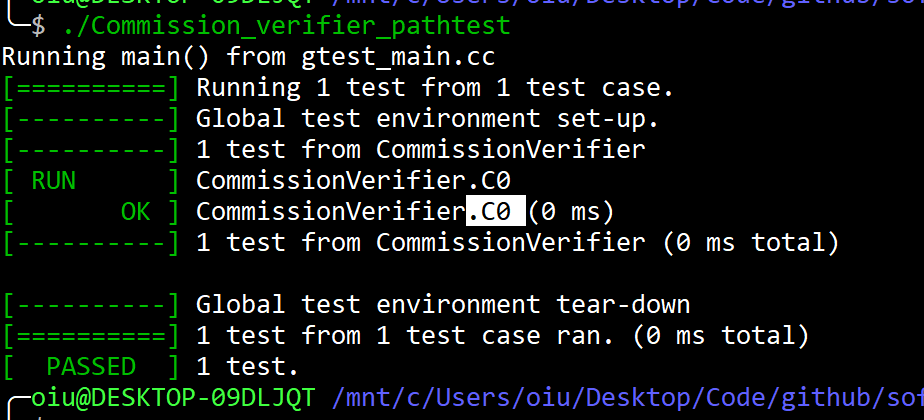
### **gcov 測試**

* statement coverage:
  + 
    - 沒 100%…
    - 看了一下，就有一些 corner case 沒確認
      * 
    - 為它們補上一些測資:
      * 
  + 
    - 這樣就 100% 了
* branch coverage:
  + 

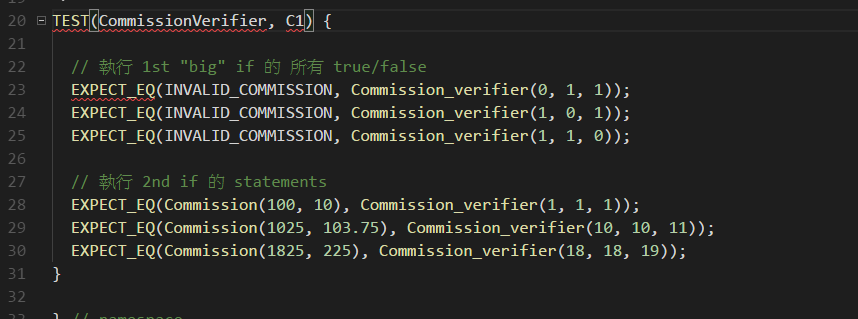
## **Commission Problem**

* 這題的 decision 就更單純了，而且一樣每個 decision 都只有一個 condition(或者說只有一個變數)，然後一樣沒有 loop
* code:
* 

### **C0**

* ****
* ****

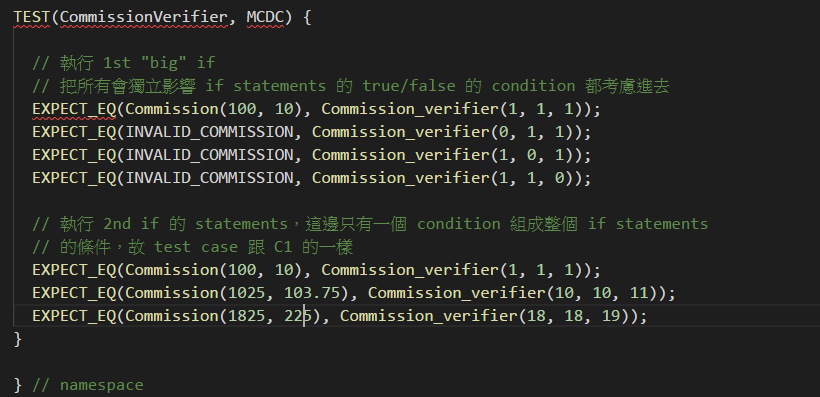
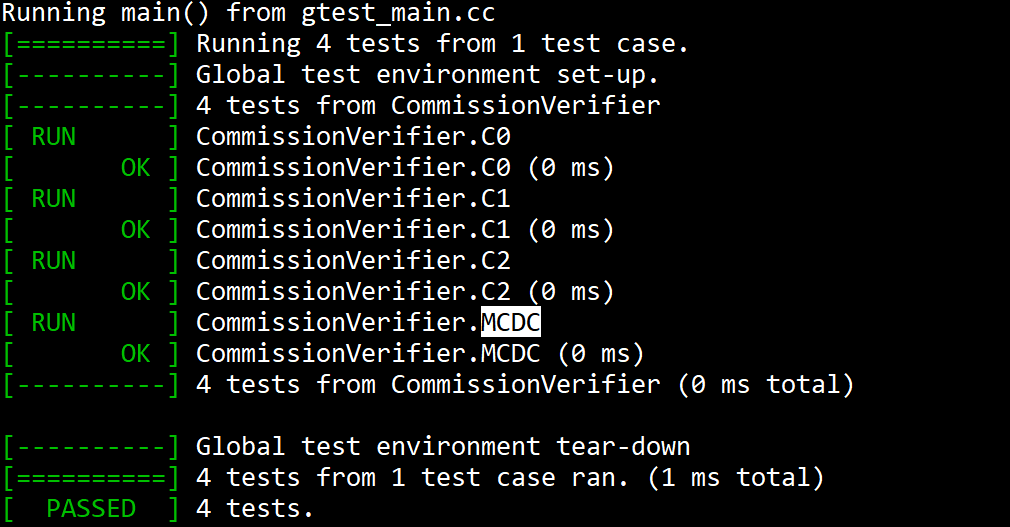
### **C1**

* 在這個題目跟實作之下，C1 剛好會跟 C0 一樣(C1 本來就 >=(subsume) C0)
* 

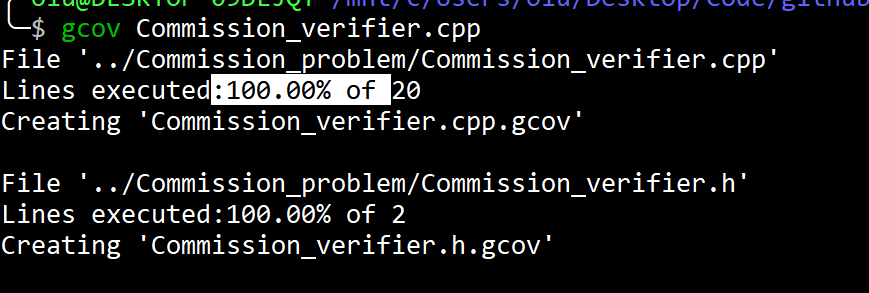
### **C2**

* 沒 loop，跟 C1 一樣

### **MCDC**

* 因為這個題目很單純，所以 MCDC 幾乎跟 C1 一樣…
* 
* 

### **gcov 測試**

* statement coverage:
  + 
* branch coverage:
  + 