

實驗項目 - C 語言的條件運算

一、本節目的：

- 學習開發 C 語言程式
- 實現在 Visual Studio 2017 系統設計平台上

二、設計重點：

- C 語言的條件式應用

三、設計步驟：

1. 建立專案

方法 A. 透過 Github Classroom 下載並開啟專案

注意：透過方法 A 建立專案後，直接跳至步驟 **3. 撰寫 C 語言程式**

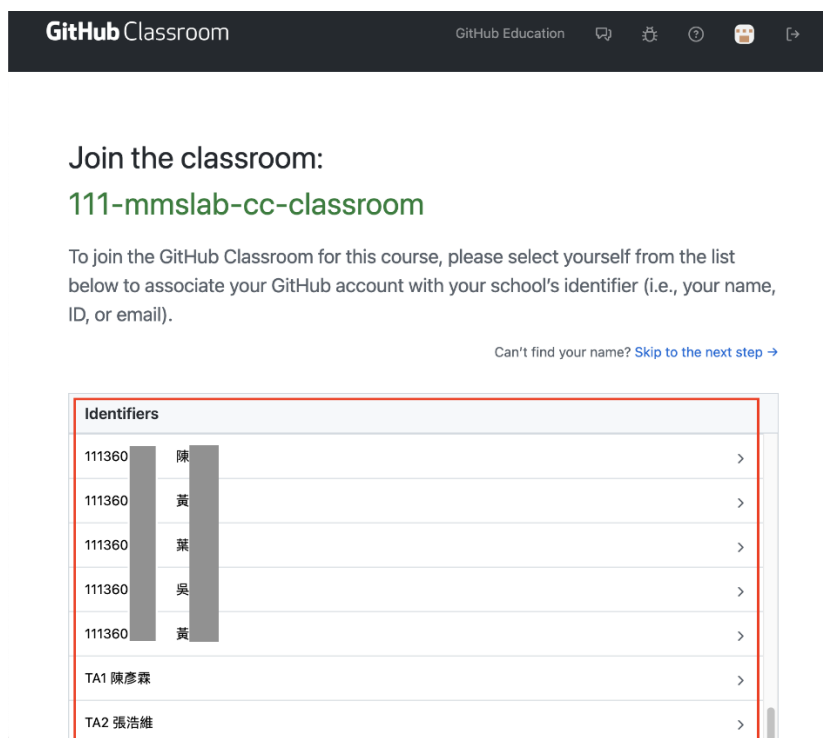
Step1. 點擊 Github Classroom 連結

- Ch1-Lab：<https://tools-api.italkutalk.com/cc/ch1-lab>

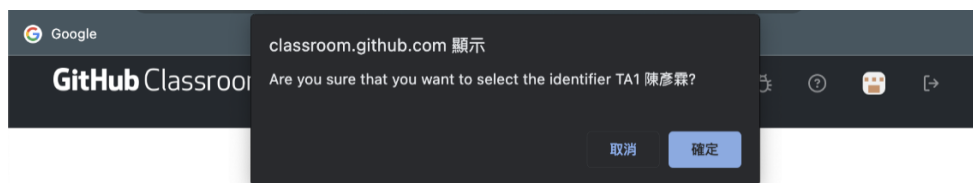
Step2. 將自己的 Github 帳號與 Classroom 學生連結

同一門課程 Github Classroom 的作業或實驗僅需連結一次 (若曾經連結過，可以略過此步驟)。

在學生清單中，會列出本門課程尚未被連結的學生，請找到並點擊自己的學號/姓名



點擊後會跳出確認提示，確認無誤點擊確定。

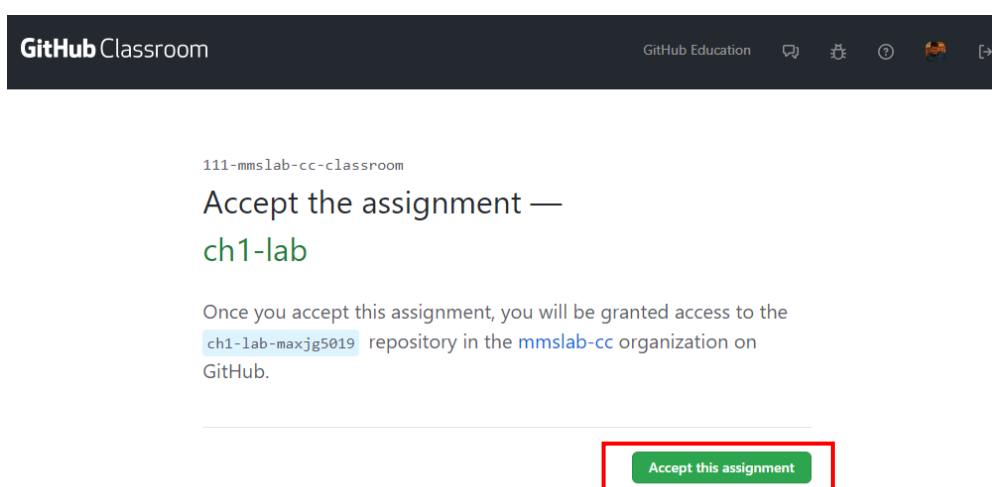


如果遇到以下問題：

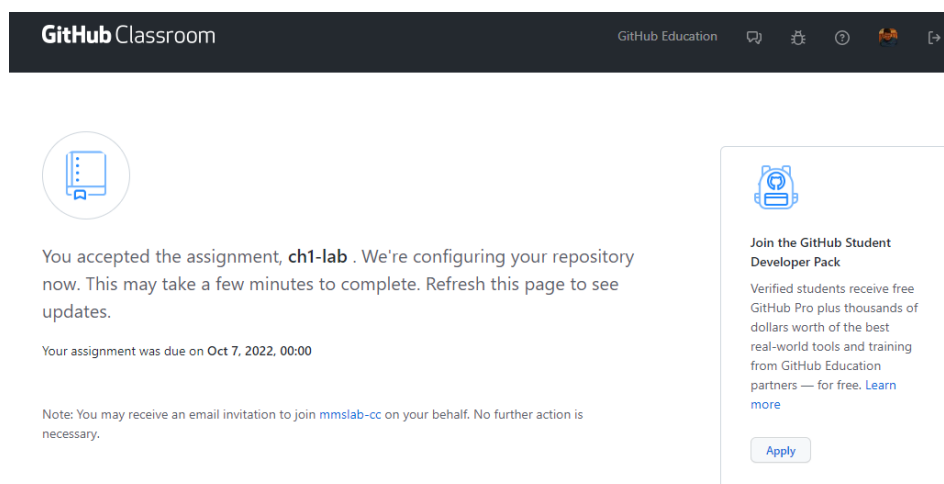
1. 名單中找不到自己的學號姓名
2. 選擇錯人
3. 學號姓名錯誤

請與課堂助教反應，助教將會協助處理。

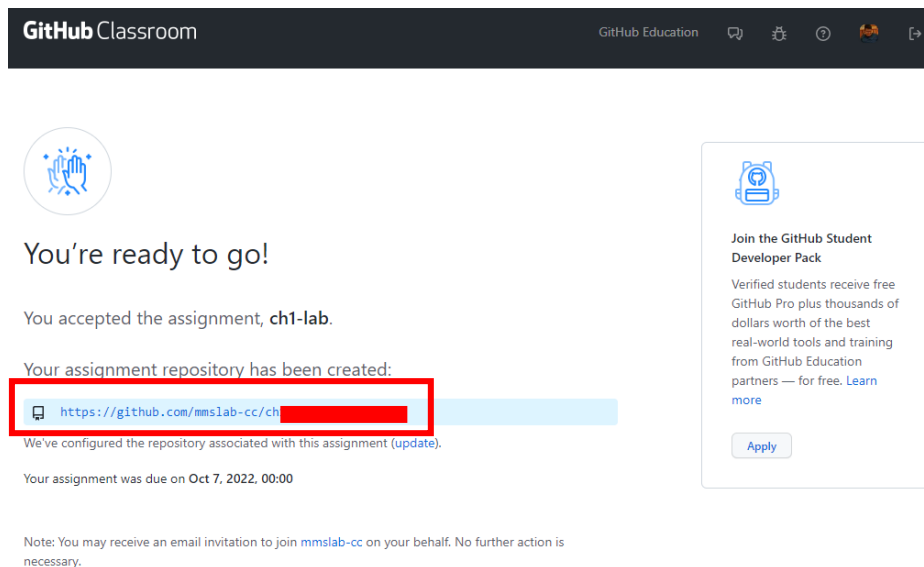
Step3. 接受 Assignment，點擊 Accept this assignment



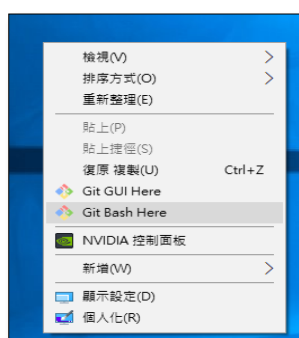
接受 Assignment 後，Github classroom 會幫你建立專屬的 repository，而建立專屬的 repository 需要一段時間，請等 10 秒左右刷新此頁面



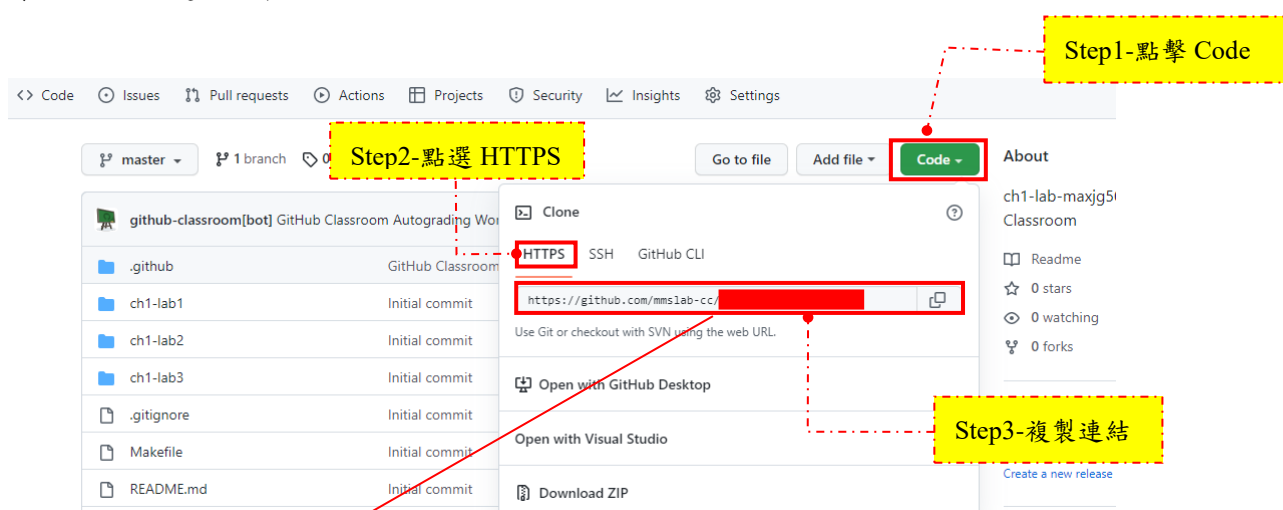
刷新頁面後，將會看到屬於自己的 repository 連結，並點擊該連結。



Step4. 將專案 Clone 到自己電腦
到桌面開啟 Git Bash



複製專案在遠端資料庫的位置，並在 Git bash 輸入指令進行下載專案(命令列點擊右鍵可以選貼上)



\$ git clone <https://github.com/xxx/xxx.git> (記得改成自己的資料庫連結網址)

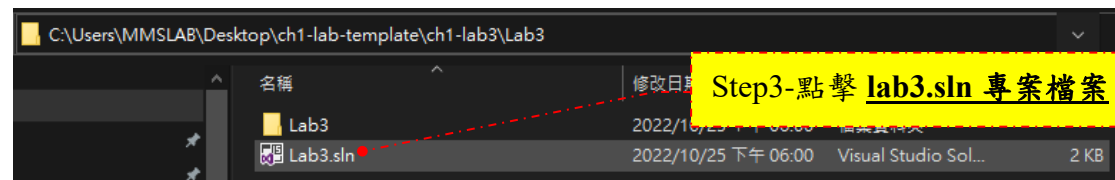
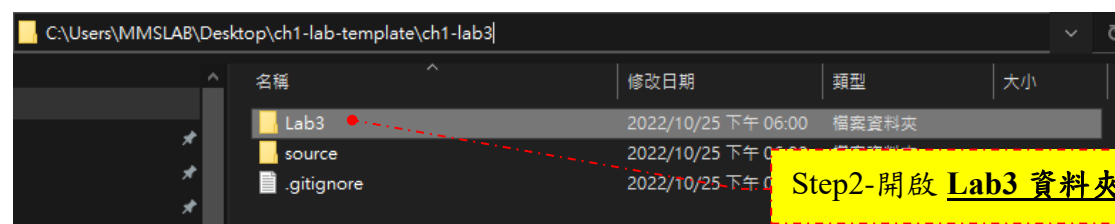
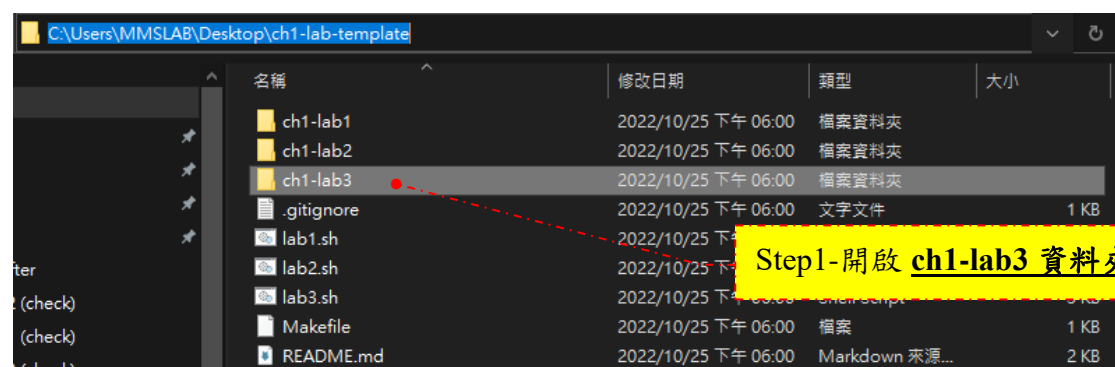
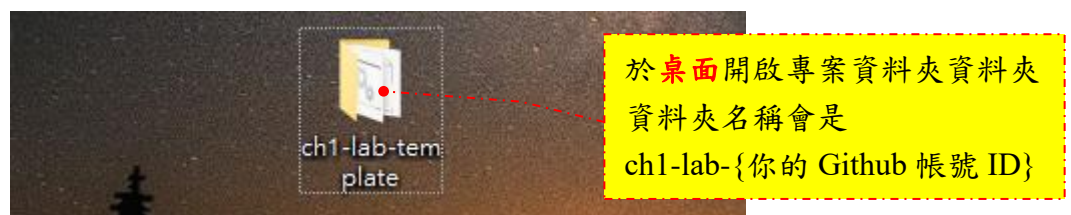
輸入指令情況：

```
Jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop
$ git clone https://github.com/your-github-id/ch1-lab3.git
Cloning into 'ch1-lab3'...
remote: Counting objects: 7, done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 7 (delta 0), reused 7 (delta 0), pack-reused 0
Unpacking objects: 100% (7/7), done.

Jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop
$ |
```

Step5. 完成下載專案後請到桌面開啟剛下載的專案資料夾，開啟專案檔案。

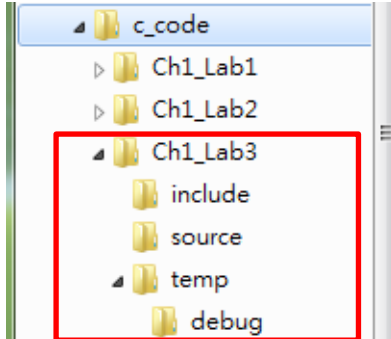
檔案路徑為：**Desktop\ch1-lab-{你的 Github 帳號 ID}\ch1-lab3\Lab3\Lab3.sln**



注意：透過方法 A 建立專案後，直接跳至步驟 3.撰寫 C 語言程式

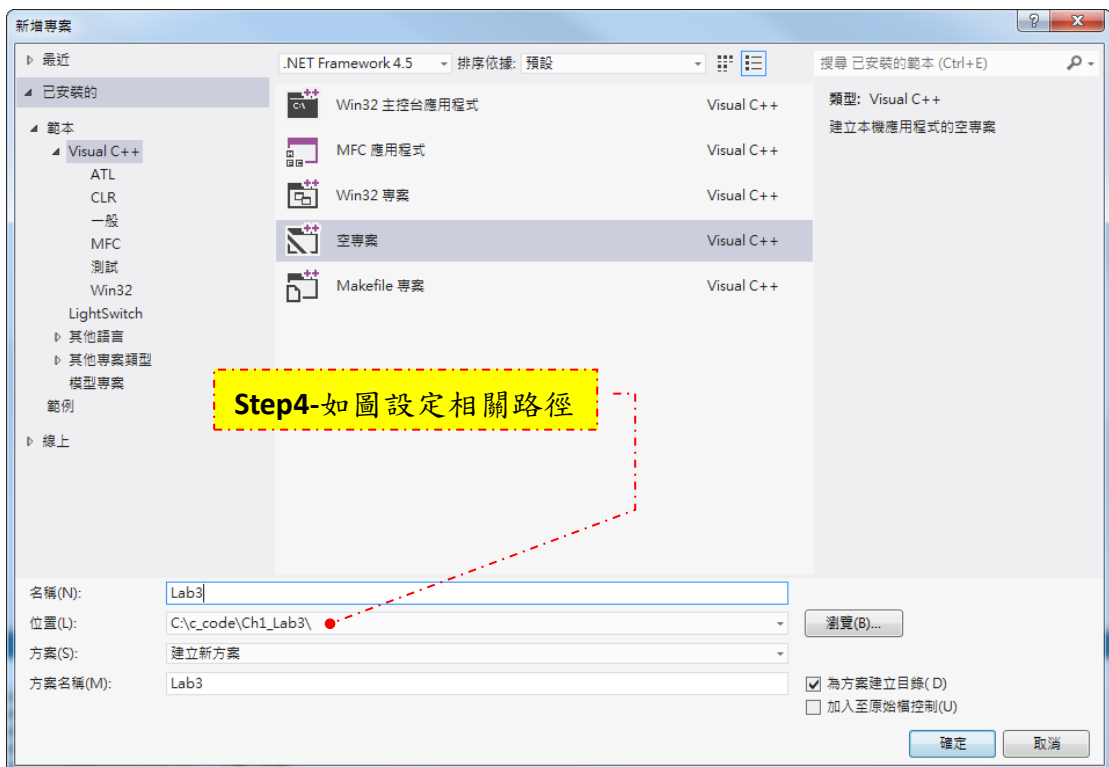
方法 B. 透過 Visual Studio 新建專案

Step1-在 C:\c_code 資料夾內新增名為“Ch1_Lab3”的資料夾，再於 Ch1_Lab3 資料夾內分別建立 include、source、temp 等資料夾，建立後需要在 temp 資料夾內新增名為“debug”的資料夾，建立完成後如下圖

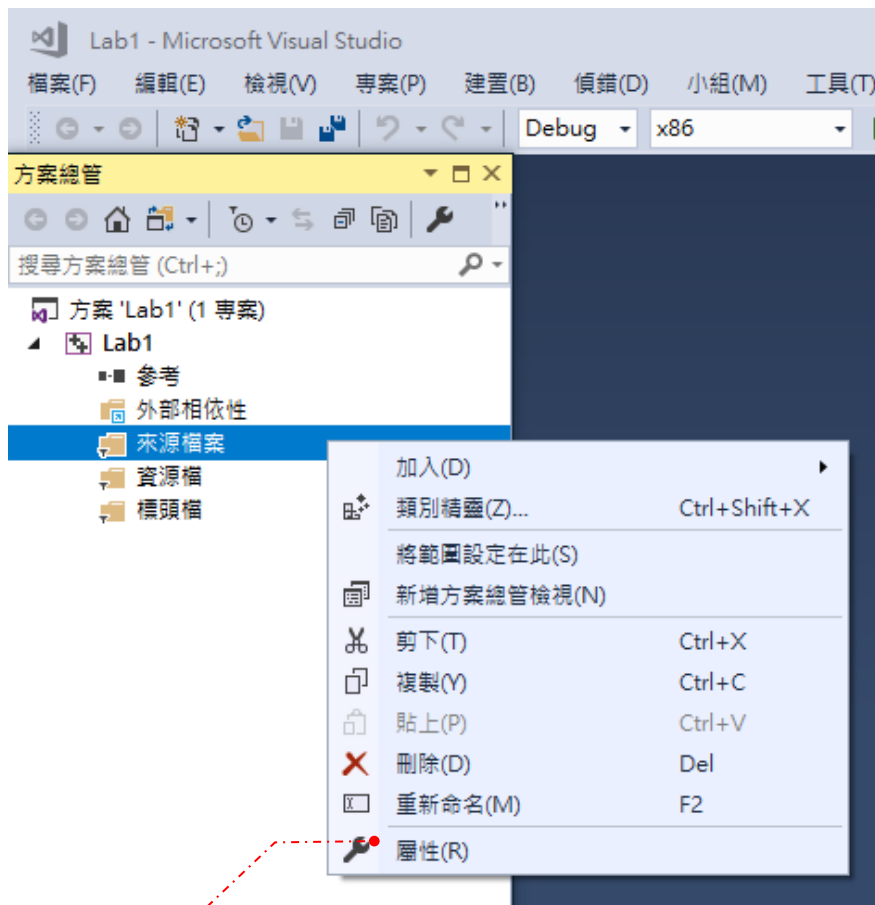


Step2-開啟 Microsoft Visual Studio 視窗畫面後點選左上角“開始(F)”

Step3-點選“新增(N)”再點選“專案(P)”

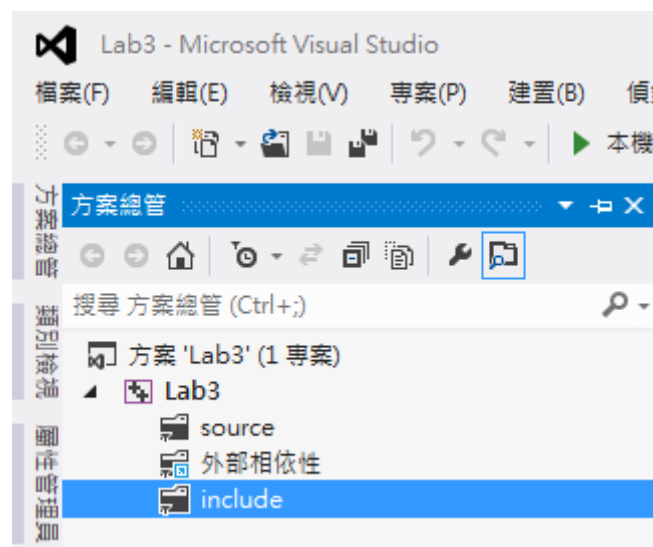


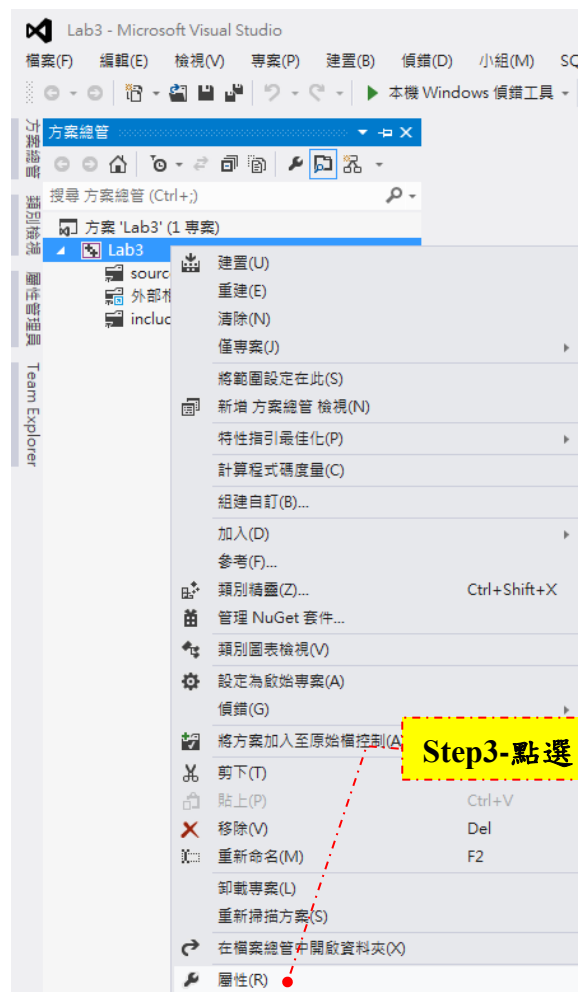
2. 路徑設定、新增 .c 檔

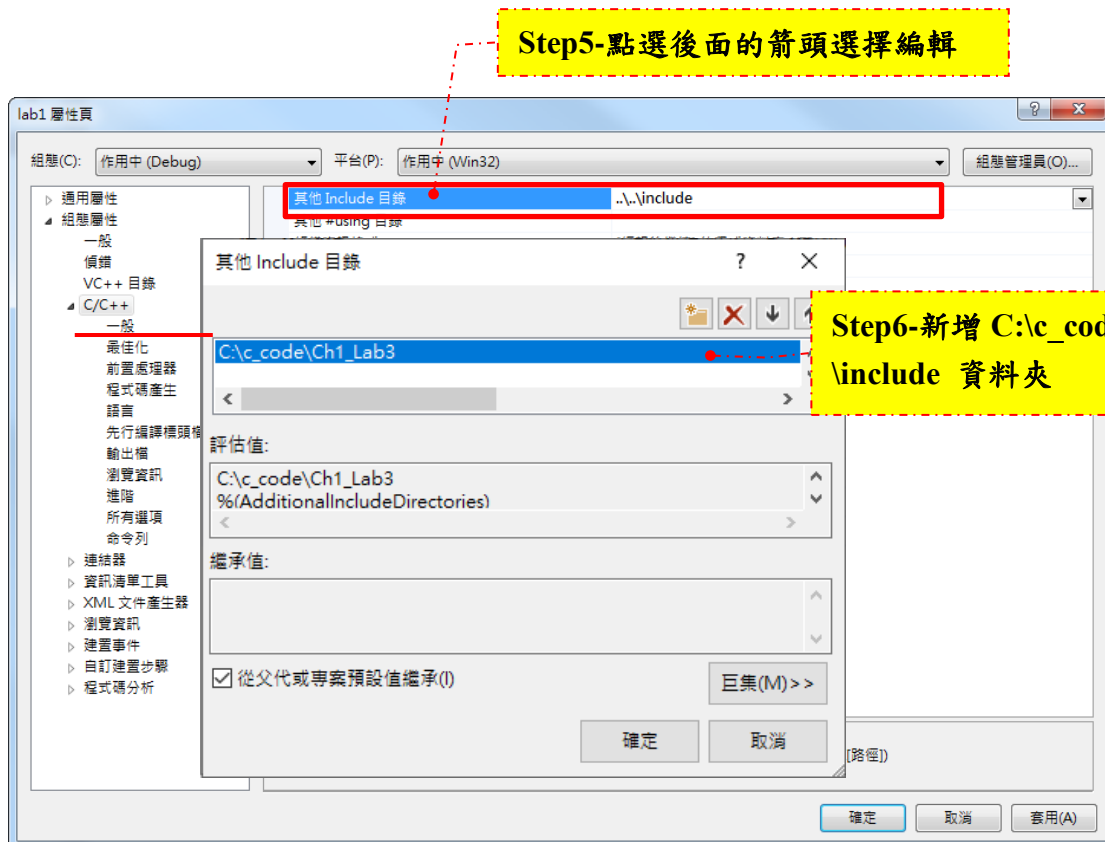
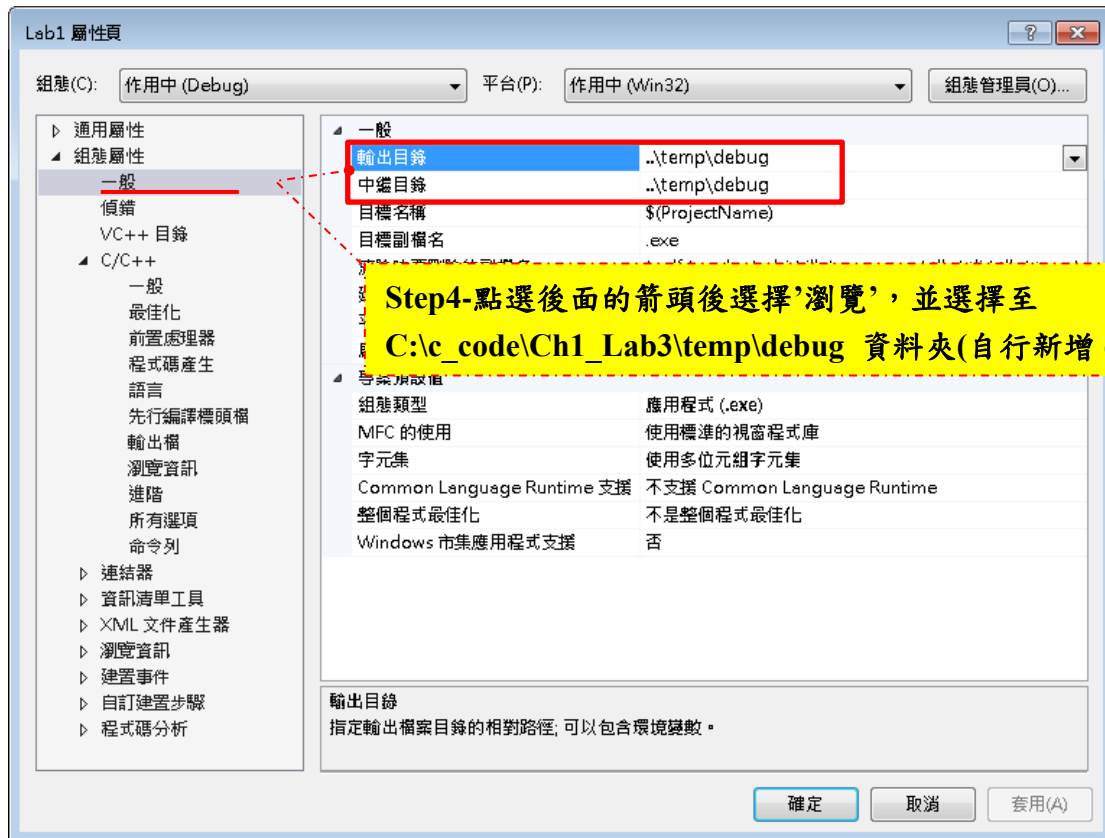


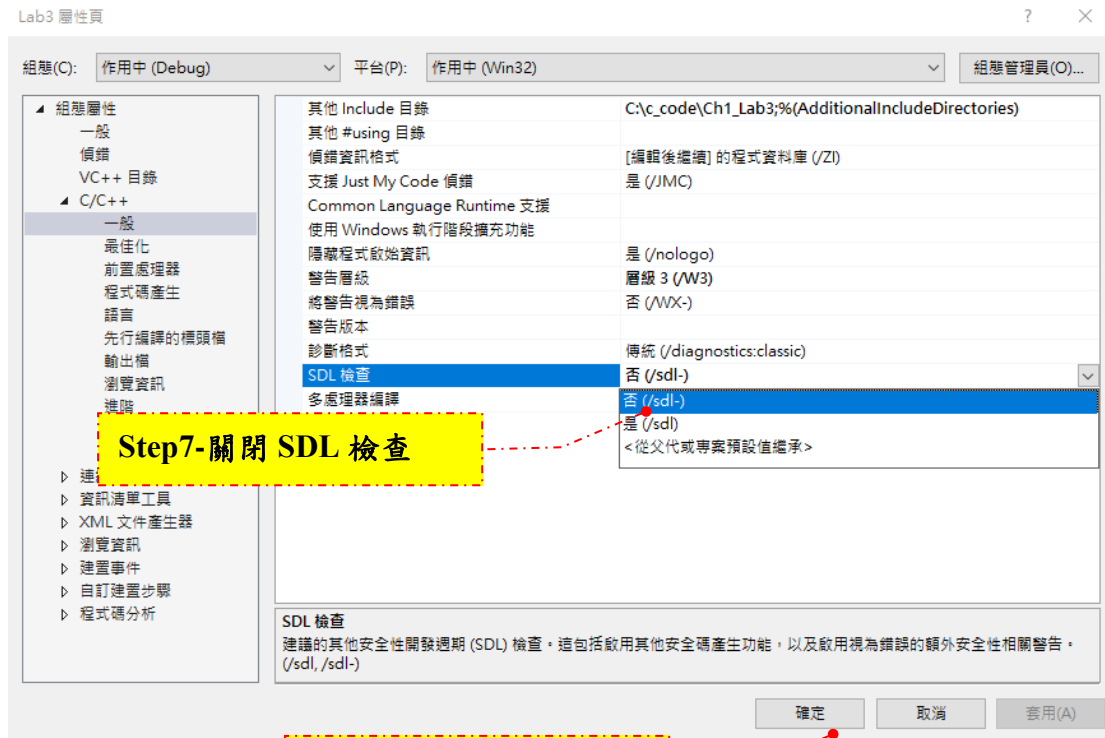
Step1-在左上角“方案總管”欄 Lab3 裡，“來源檔案”上按右鍵，選“重新命名(M)”，命名為“source”，再將“標頭檔”重新命名為“include”

點選“資源檔”，按右鍵選“刪除(D)”，完成後方案總管如下圖



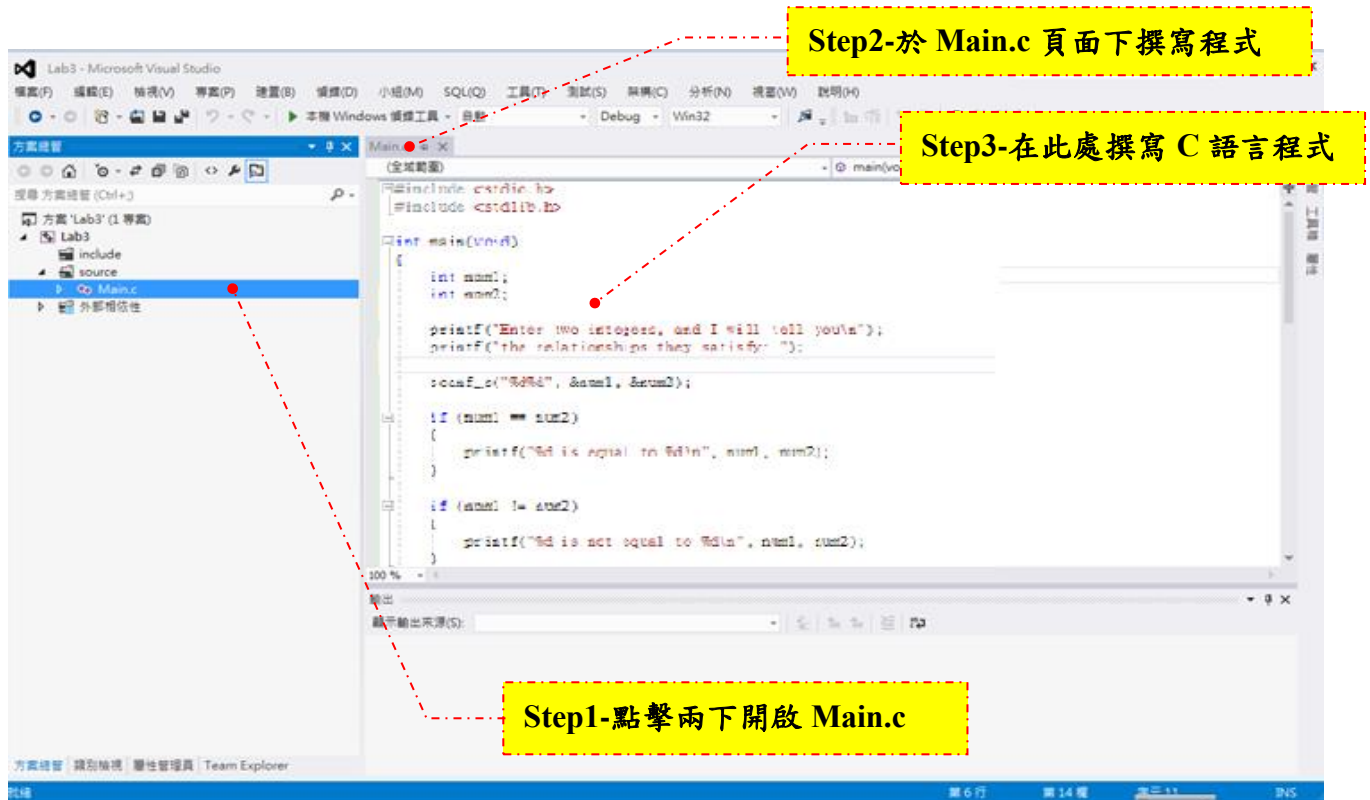






3. 撰寫 C 語言程式

如果透過「方法 A. 透過 Github Classroom 下載並開啟專案」建立專案，直接從此步驟繼續實驗。

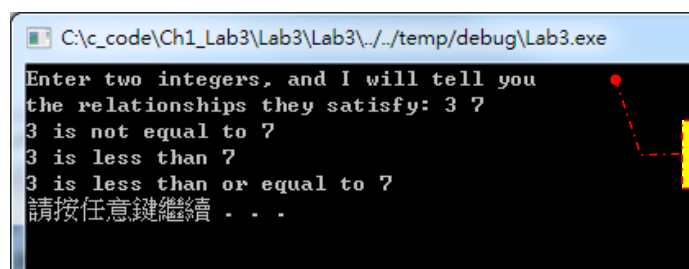
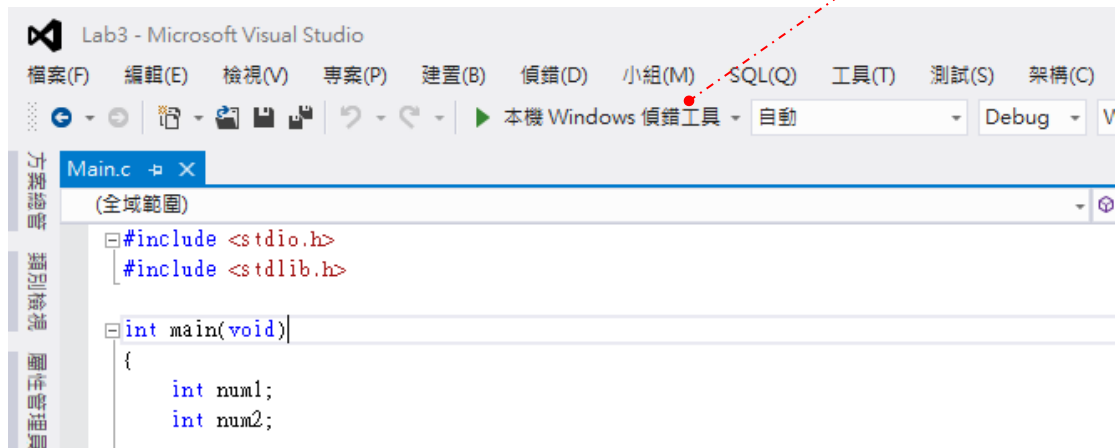


Main.c 程式碼：

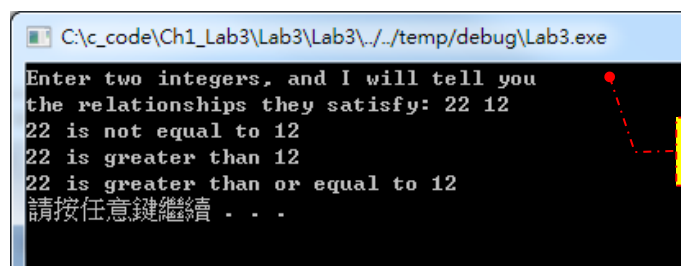
```
main.c ✕
Lab3 (全域範圍)
1  #include <stdlib.h>
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Enter two integers, and I will tell you\n");
7      printf("the relationships they satisfy:\n");
8
9
10     int num1;
11     int num2;
12
13     scanf("%d%d", &num1, &num2);
14
15     if (num1 == num2)
16     {
17         printf("%d is equal to %d\n", num1, num2);
18     }
19     if (num1 != num2)
20     {
21         printf("%d is not equal to %d\n", num1, num2);
22     }
23     if (num1 < num2)
24     {
25         printf("%d is less to %d\n", num1, num2);
26     }
27     if (num1 > num2)
28     {
29         printf("%d is greater to %d\n", num1, num2);
30     }
31     if (num1 <= num2)
32     {
33         printf("%d is less than or equal to %d\n", num1, num2);
34     }
35     if (num1 >= num2)
36     {
37         printf("%d is greater than or equal to %d\n", num1, num2);
38     }
39
40     system("pause");
41     return 0;
42
43 }
```

4. 執行與測試程式結果

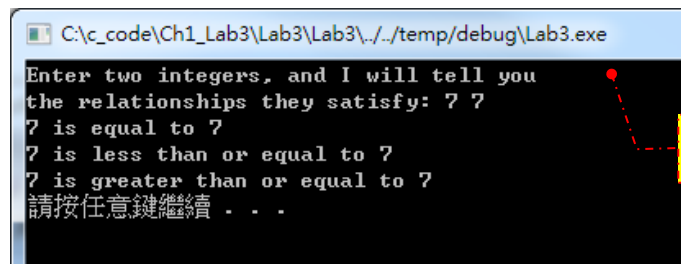
Step1-點選開始偵測，進行偵測



Step2-輸入樣本：3 和 7



Step3-輸入樣本：22 和 12

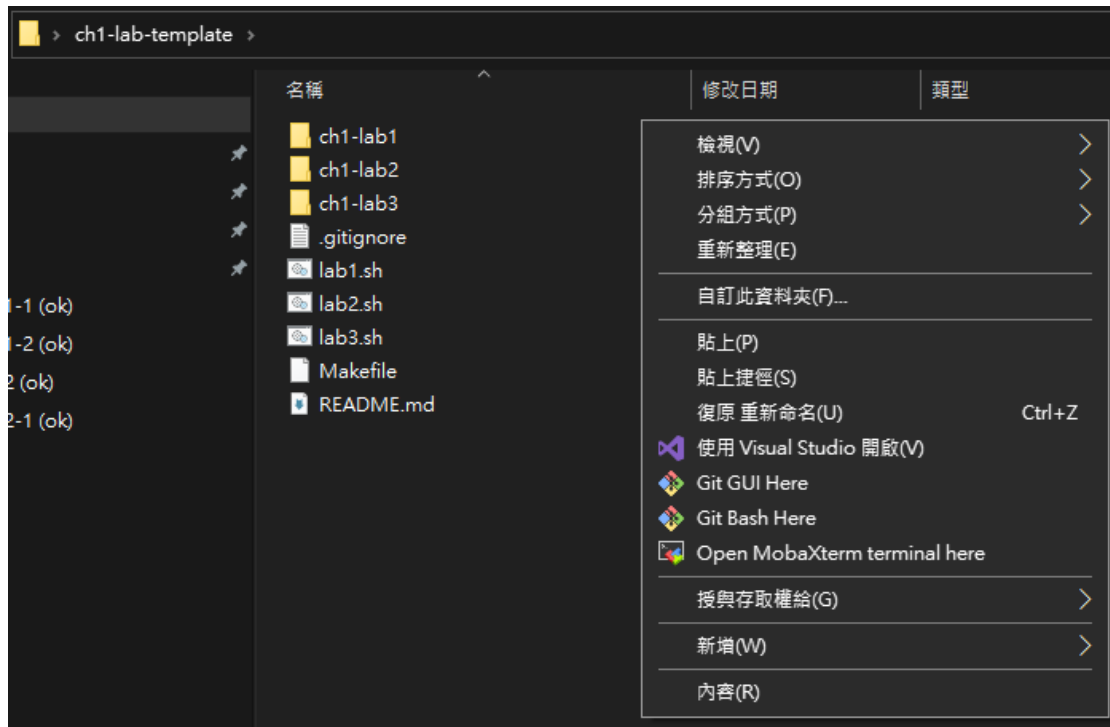


Step4-輸入樣本：7 和 7

5. 上傳實驗至 Github Classroom

請參考從下方的**上傳專案**說明，將專案透過 Git 指令 push 到 Github classroom

1. 專案編輯完成後，於本周課程專案的資料夾點選**滑鼠右鍵**，選擇 **Git Bash Here** 開啟 Git 終端機。



2. 於終端機中**依序**輸入以下指令，將程式上傳至遠端資料庫。

```
git add .
```

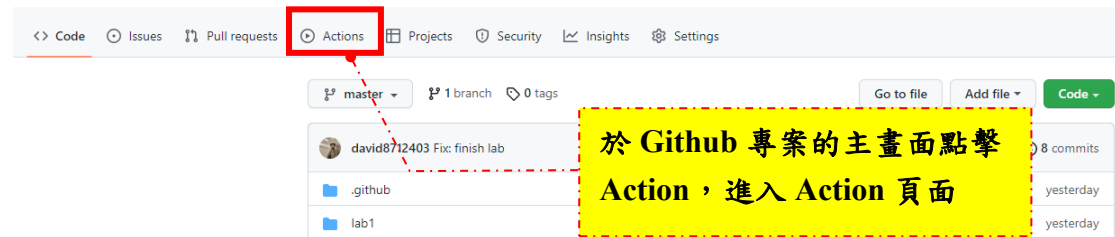
```
git commit -m "[修改的內容描述]"
```

```
git push origin master
```

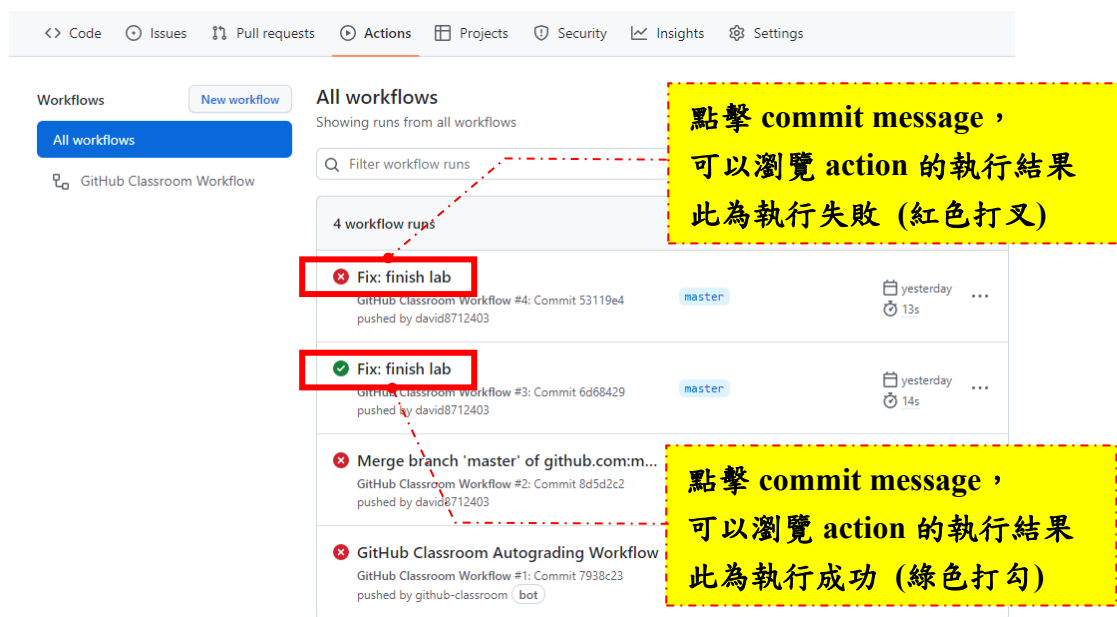
6. 觀察 Github Action 評分

在每次有新的 commit push 到 Github Classroom 時，會觸發定義好的 action 流程，會自動將程式編譯後執行，判斷是否執行正確。

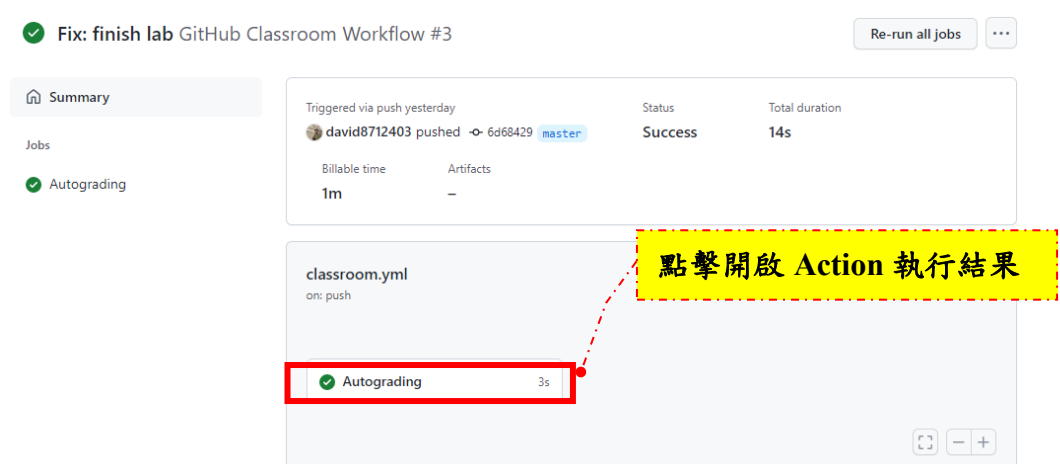
Step1. 進入 Github classroom 實驗的 repository，點擊 Action



Step2. 觀察自己每次 commit 時，action 的輸出及批改結果



Step3. 瀏覽 Action 執行結果



自動評分執行成功情況。本次 Lab 有 3 個 test case，全數通過會得到 100 分

The image displays two screenshots of the Autograding interface, showing successful test results for two lab tasks.

Top Screenshot (lab1):

- The task name "Run education/autograding@v1" is highlighted with a red box. A yellow callout points to it with the text "展開 Run education/autograding@v1".
- The test details for "lab1" are highlighted with a red box. A yellow callout points to it with the text "編譯及輸出結果".
- The test result "lab1" is highlighted with a red box. A yellow callout points to it with the text "✅ lab1 代表檢查通過".

Bottom Screenshot (lab2):

- The task name "Run education/autograding@v1" is highlighted with a red box. A yellow callout points to it with the text "展開 Run education/autograding@v1".
- The test details for "lab2" are highlighted with a red box. A yellow callout points to it with the text "編譯及輸出結果".
- The test result "lab2" is highlighted with a red box. A yellow callout points to it with the text "✅ lab2 代表檢查通過".

