

## 實驗項目- 民意調查

### 一、 本節目的：

- 指標以及指標運算子

### 二、 設計重點：

- 利用指標來以傳參考呼叫的方式傳遞引數給函式

### 三、 實驗題目：

寫一個簡單的民意調查程式，讓使用者針對五個社會意識議題來評分，1 代表最不重要，10 代表最重要的議題。選擇五個對你來說重要的議(例如：政治議題、全球環境議題等等)。使用一個一維陣列 topics(型別為 char \*) 來儲存這五個議題。為了要整理調查結果，請你使用具有 5 列、10 行的二維陣列 responses(型別為 int)，每一列都對應到 topics 陣列的一個元素。當程式執行時，他應該要求使用者對每個議題進行評分。然後讓程式顯示整理過後的結果，包括：

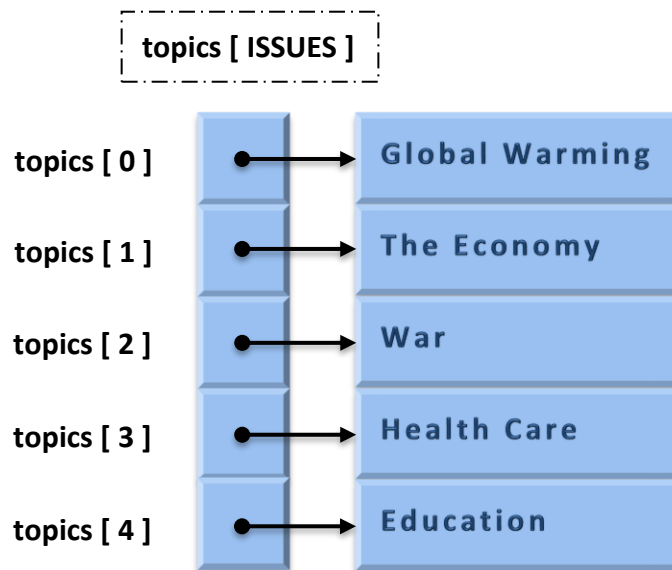
- a) 以表格顯示結果，將五個議題顯示在左邊，十個等級的評分方式放在上方在每一格列出每個議題在該評分等級的計數。
- b) 在每一列的最右邊，列出該議題的平均分數。
- c) 哪一個議題得到最高的總點數？印出該議題以及所得點數。
- d) 哪一個議題得到最低的總點數？印出該議題以及所得點數。

### 四、 程式解說：

- 此程式定義了一個指標陣列 topics (第 14~15 行)，topics 陣列內的 4 個字串分別為 Global Warming、The Economy、War、Health Care、Education

```
14  const char *topics[ISSUES] = { "Global Warming", "The Economy",  
15  [1] "War", "Health Care", "Education" };
```

- 指標陣列 topics 裡位址 0 放置對應的字串 Global Warming  
位址 1 放置對應的函式 The Economy  
位址 2 放置對應的函式 War  
位址 3 放置對應的函式 Health Care  
位址 4 放置對應的函式 Education (如下圖所示)

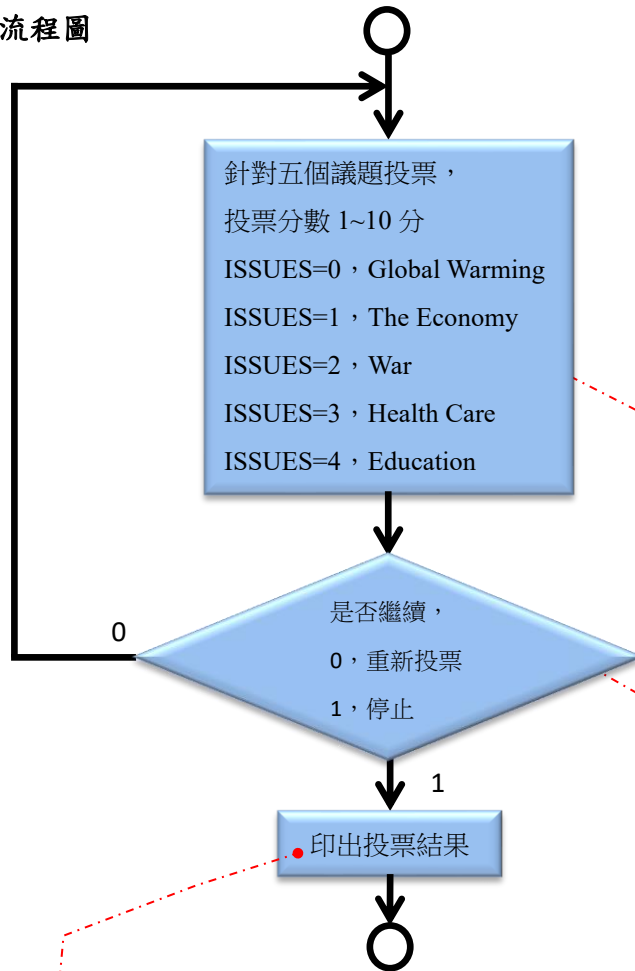


- 程式到第 26 行，執行 for loop，到了第 27 行時，  
i=0，即 topics[0]對應到字串 Global Warming 印出  
i=1，即 topics[1]對應到字串 The Economy 印出  
i=2，即 topics[2]對應到字串 War 印出  
...以此類推

```
24  |  
25  |  
26  |  
27  |  
28  |  
29  |  
30  |  
31  |  
32  |
```

```
for (i = 0; i < ISSUES; i++) {  
    do {  
        printf("%s? ", topics[i]);  
        scanf_s("%d", &response);  
    } while (response < 1 || response > 10);  
    recordResponse(i, response);  
}
```

## 五、 程式流程圖



39      `displayResults();`

```

21      printf("Please rate the following topics on a scale from 1 - 10"
22            "\n 1 = least important, 10 = most important\n");
23
24      for (i = 0; i < ISSUES; i++) {
25
26          do {
27              printf("%s? ", topics[i]);
28              scanf_s("%d", &response);
29              } while (response < 1 || response > 10);
30
31              recordResponse(i, response);
32          }
  
```

```

19      do
20      {
    ...
34      printf("Enter 1 to stop. Enter 0 to rate the issues again.");
35      scanf_s("%d", &response);
36
37      } while (response != 1);
  
```

## 六、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**

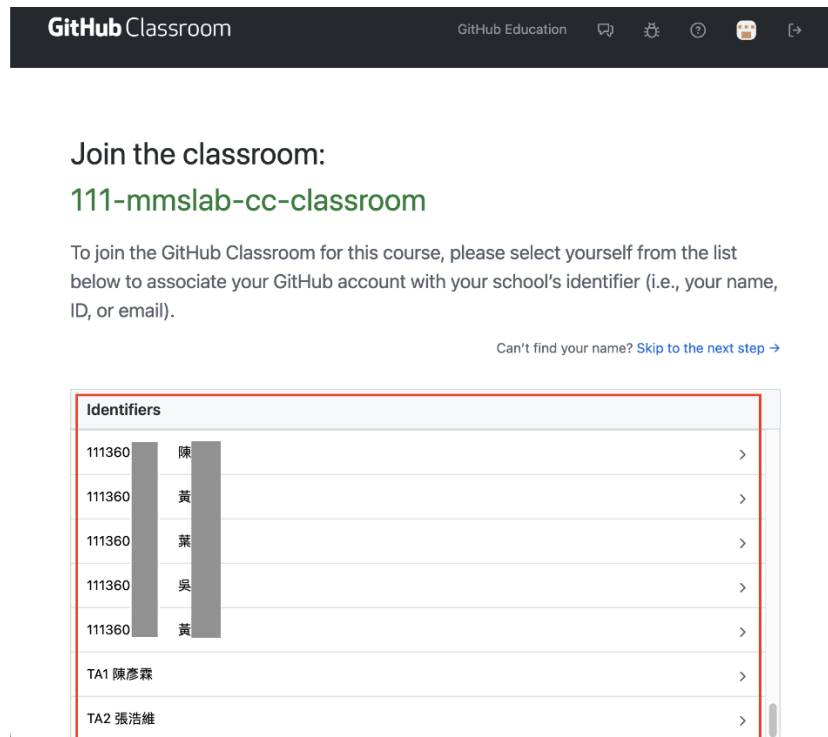
Step1. 點擊 Github Classroom 連結

- Ch5-Lab：<https://tools-api.italkutalk.com/cc/ch5-lab>

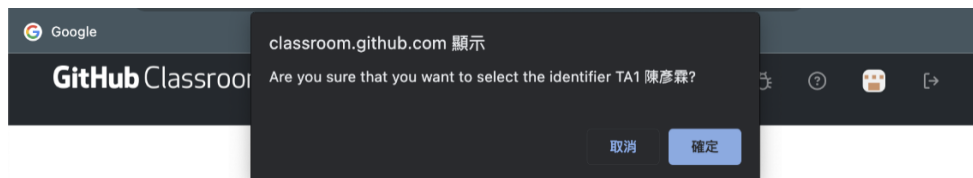
Step2. 將自己的 Github 帳號與 Classroom 學生連結

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。

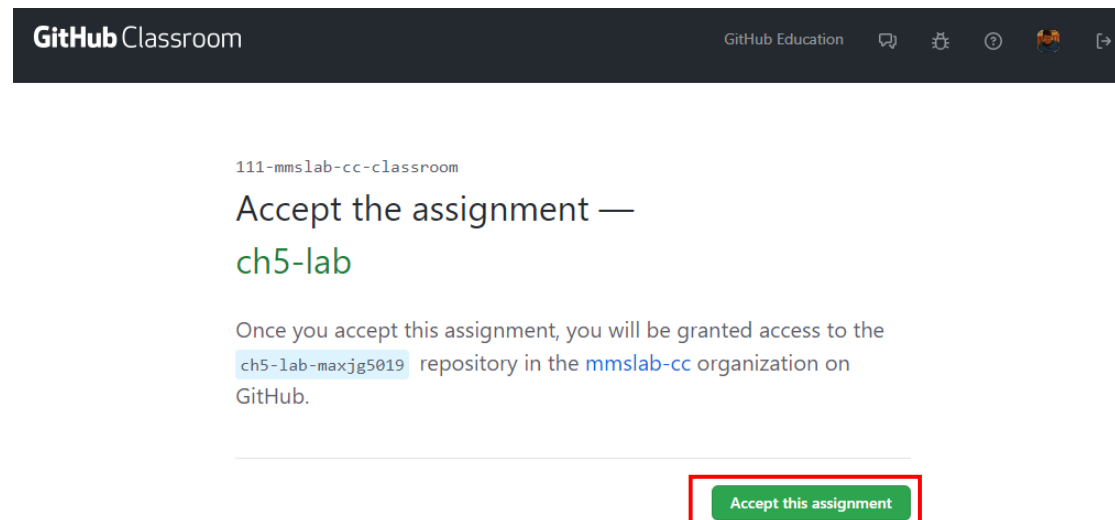


如果遇到以下問題：

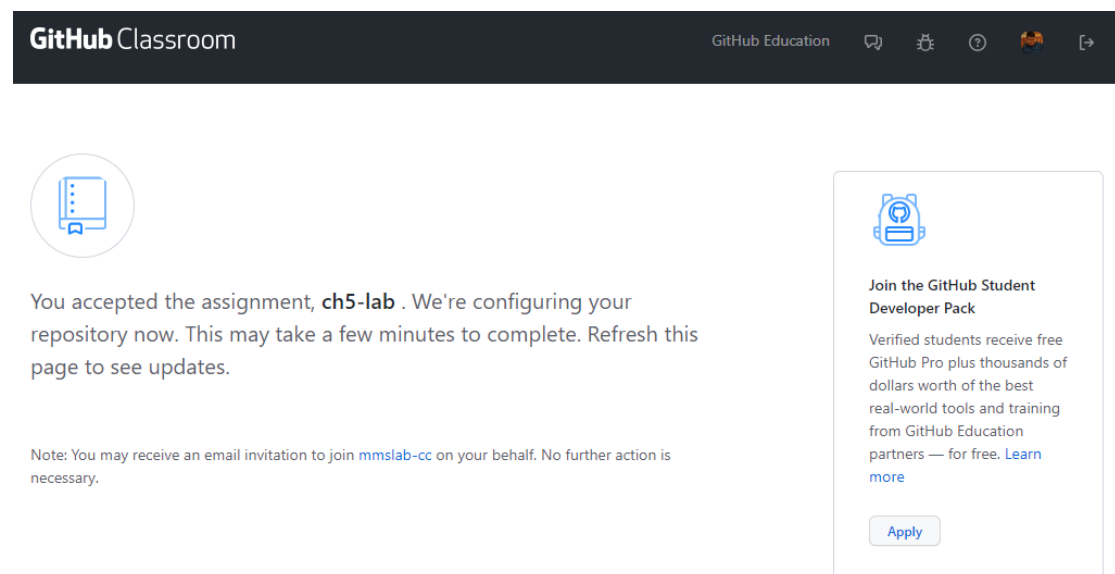
1. 名單中找不到自己的學號姓名
2. 選擇錯人
3. 學號姓名錯誤

請與課堂助教反應，助教將會協助處理。

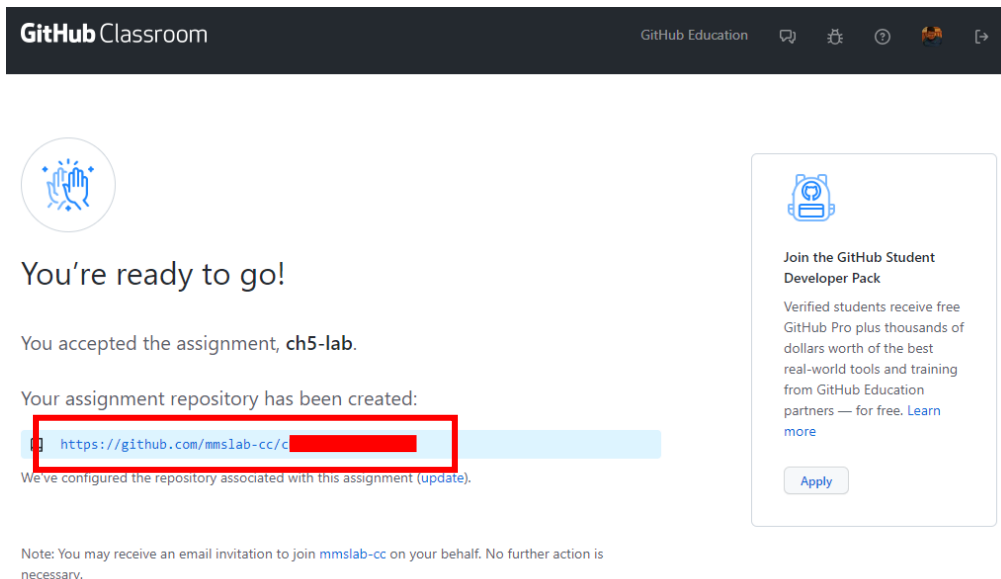
Step3. 接受 Assignment，點擊 Accept this assignment



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



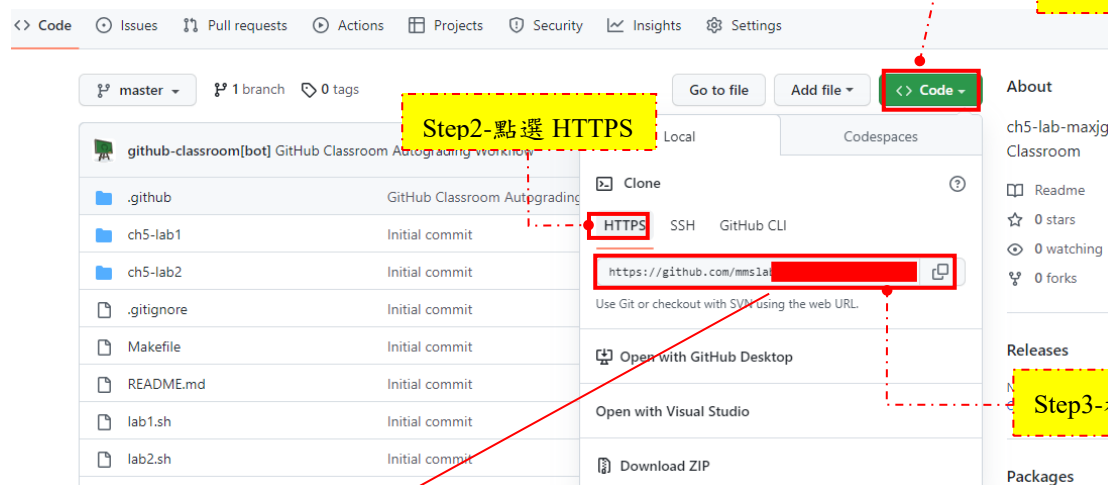
刷新頁面後，將會看到屬於自己的 repository 連結，並點擊該連結。



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)

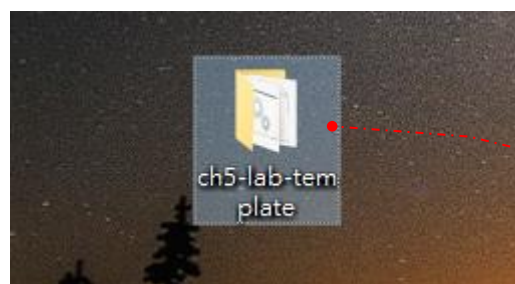
輸入指令情況：

```
Jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop
$ git clone https://github.com/your-github-id/ch5-lab2.git
Cloning into 'ch5-lab2'...
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.

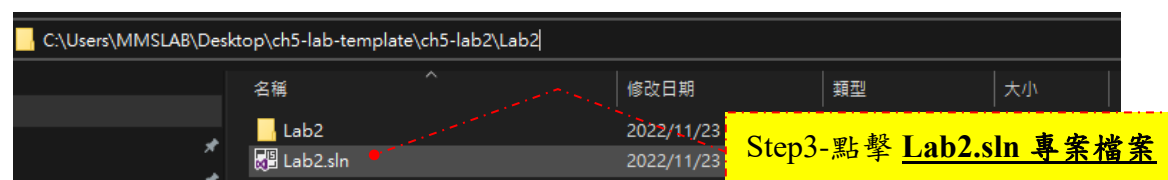
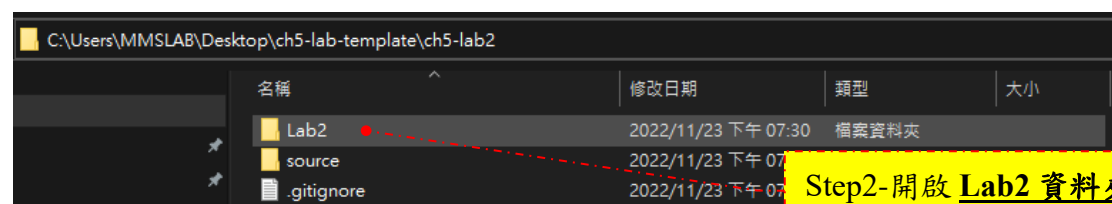
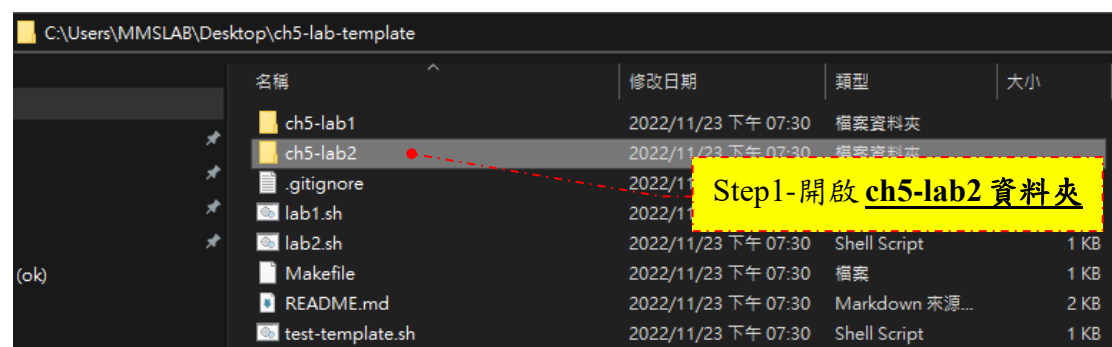
Jdway@DESKTOP-GBM49C1 MINGW64 ~/Desktop
$ |
```

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\ch5-lab-{你的 Github 帳號 ID}\ch5-lab2\Lab2\Lab2.sln



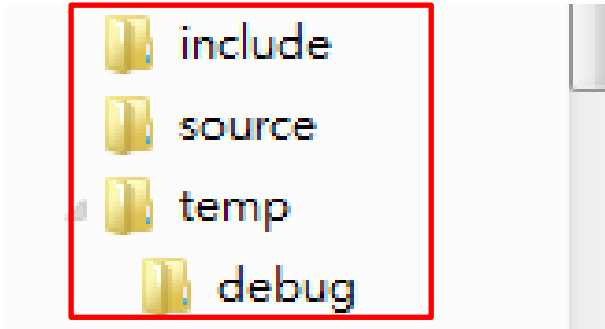
於**桌面**開啟專案資料夾資料夾  
資料夾名稱會是  
ch1-lab-{你的 Github 帳號 ID}



**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 “Ch5\_Lab2” 的資料夾，再於 Ch5\_Lab2 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



Step2-參照 Ch1\_Lab3 中 “1.建立新的空專案” Step2~Step4，設定相關路徑位置為 C:\c\_code\ Ch5\_Lab2

### 2. 路徑設定、新增 .c 檔

Step1-參照 Ch1\_Lab3 中 “2. 路徑設定、新增 .c 檔” Step1~Step8，新增 main.c 檔與設定相關屬性設定。

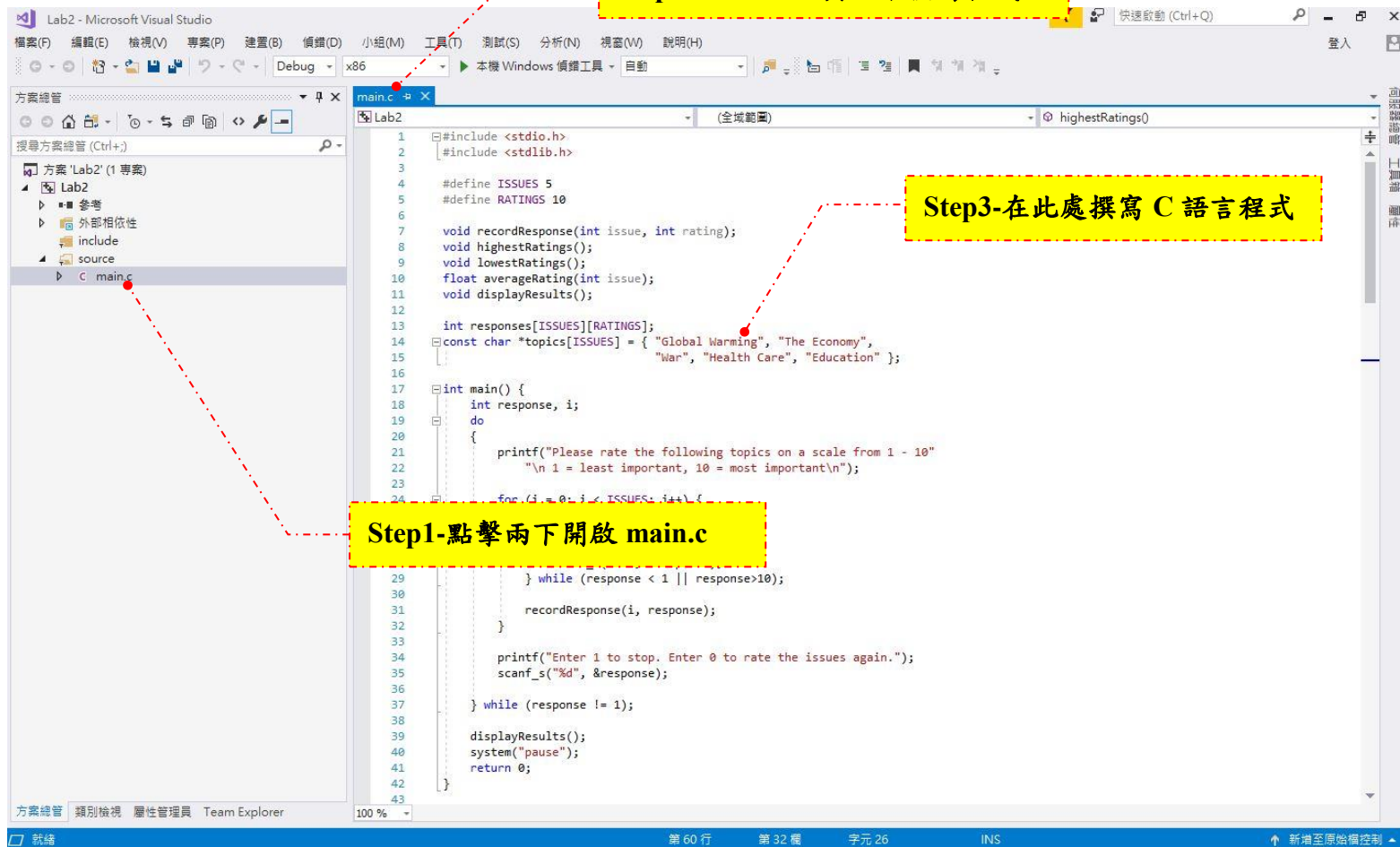


### 3. 撰寫 C 語言程式

Step2-於 main.c 頁面下撰寫程式

Step3-在此處撰寫 C 語言程式

Step1-點擊兩下開啟 main.c



main.c 程式碼：

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define ISSUES 5
5  #define RATINGS 10
6
7  void recordResponse(int issue, int rating);
8  void highestRatings();
9  void lowestRatings();
10 float averageRating(int issue);
11 void displayResults();
12
13 int responses[ISSUES][RATINGS];
14 const char *topics[ISSUES] = { "Global Warming", "The Economy",
15                                "War", "Health Care", "Education" };
16
17 int main() {
18     int response, i;
19     do
20     {
21         printf("Please rate the following topics on a scale from 1 - 10"
22              "\n 1 = least important, 10 = most important\n");
23
24         for (i = 0; i < ISSUES; i++) {
25
26             do {
27                 printf("%s? ", topics[i]);
28                 scanf_s("%d", &response);
29             } while (response < 1 || response > 10);
30
31             recordResponse(i, response);
32         }
33
34         printf("Enter 1 to stop. Enter 0 to rate the issues again.");
35         scanf_s("%d", &response);
36
37     } while (response != 1);
38
39     displayResults();
40     system("pause");
41     return 0;
42 }
43
```

```

44 void recordResponse(int issue, int rating) {
45     responses[issue][rating - 1]++;
46 }
47
48 void highestRatings() {
49
50     int highRating = 0;
51     int highTopic = 0;
52     int i, j;
53
54     for (i = 0; i < ISSUES; i++) {
55         int topicRating = 0;
56         for (j = 0; j < RATINGS; j++) {
57             topicRating += responses[i][j] * (j + 1);
58         }
59
60         if (highRating < topicRating) {
61             highRating = topicRating;
62             highTopic = i;
63         }
64     }
65
66     printf("The highest rated topic was ");
67     printf("%s", topics[highTopic]);
68     printf("with a total rating of %d\n", highRating);
69
70
71 }

```

```

72
73 void lowestRatings() {
74
75     int lowRating = 0;
76     int lowTopic = 0;
77     int i, j;
78
79     for (i = 0; i < ISSUES; i++) {
80         int topicRating = 0;
81
82         for (j = 0; j < RATINGS; j++) {
83             topicRating += responses[i][j] * (j + 1);
84         }
85
86         if (i == 0) {
87             lowRating = topicRating;
88         }
89         if (lowRating > topicRating) {
90             lowRating = topicRating;
91             lowTopic = i;
92         }
93     }
94
95     printf("The lowest rated topic was ");
96     printf("%s", topics[lowTopic]);
97     printf("with a total rating of %d\n", lowRating);
98
99 }

```

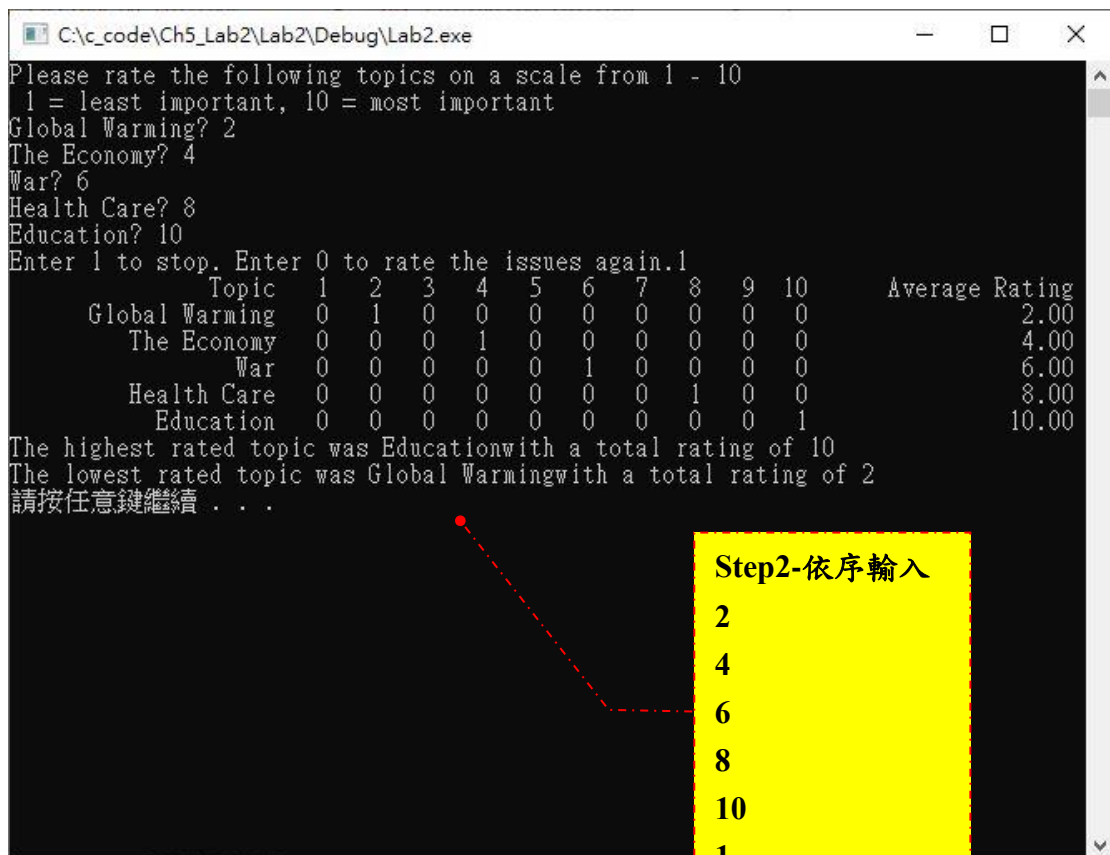
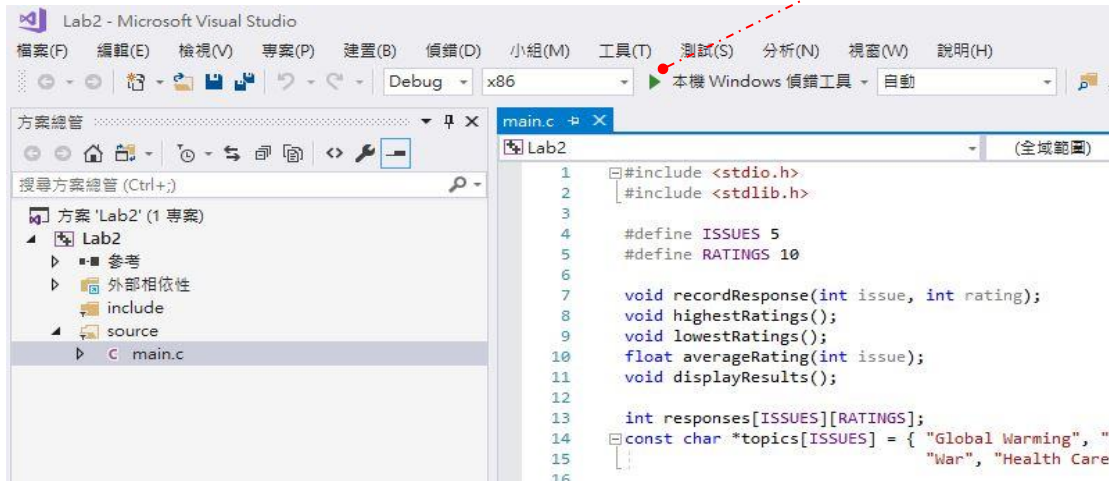
```

101 float averageRating(int issue) {
102     float total = 0;
103     int counter = 0;
104     int i;
105
106     for (i = 0; i < RATINGS; i++) {
107
108         if (responses[issue][i] != 0) {
109             total += responses[issue][i] * (i + 1);
110             counter += responses[issue][i];
111         }
112     }
113     return total / counter;
114 }
115
116 void displayResults() {
117
118     int i, j;
119     printf("%20s", "Topic");
120
121     for (i = 1; i <= RATINGS; i++) {
122         printf("%4d", i);
123     }
124
125     printf("%20s\n", "Average Rating");
126
127     for (i = 0; i < ISSUES; i++) {
128
129         printf("%20s", topics[i]);
130
131         for (j = 0; j < RATINGS; j++) {
132             printf("%4d", responses[i][j]);
133         }
134
135         printf("%20.2f\n", averageRating(i));
136     }
137
138     highestRatings();
139     lowestRatings();
140
141 }
142

```

#### 4. 執行與測試程式結果

Step1-點選開始偵測，進行偵測



#### ● 將結果呈現給助教檢查



## ● 檢查通過後上傳實驗至 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
```

- 本周課程 Lab 到此結束，謝謝各位同學參閱