

實驗項目 – 九九乘法表

1. 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

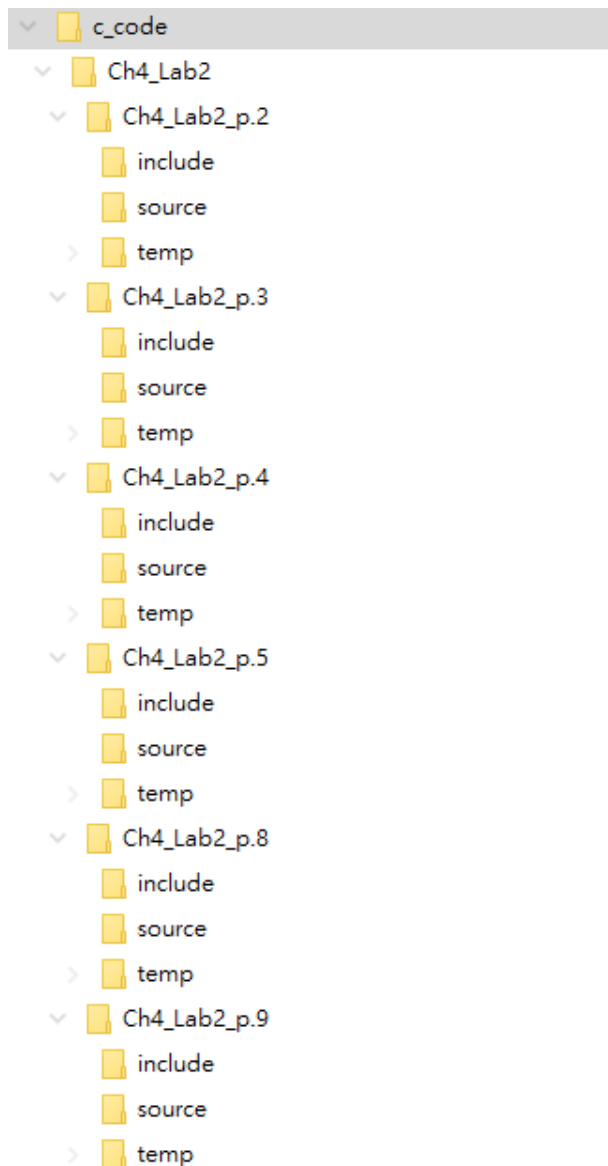
2. 設計重點：

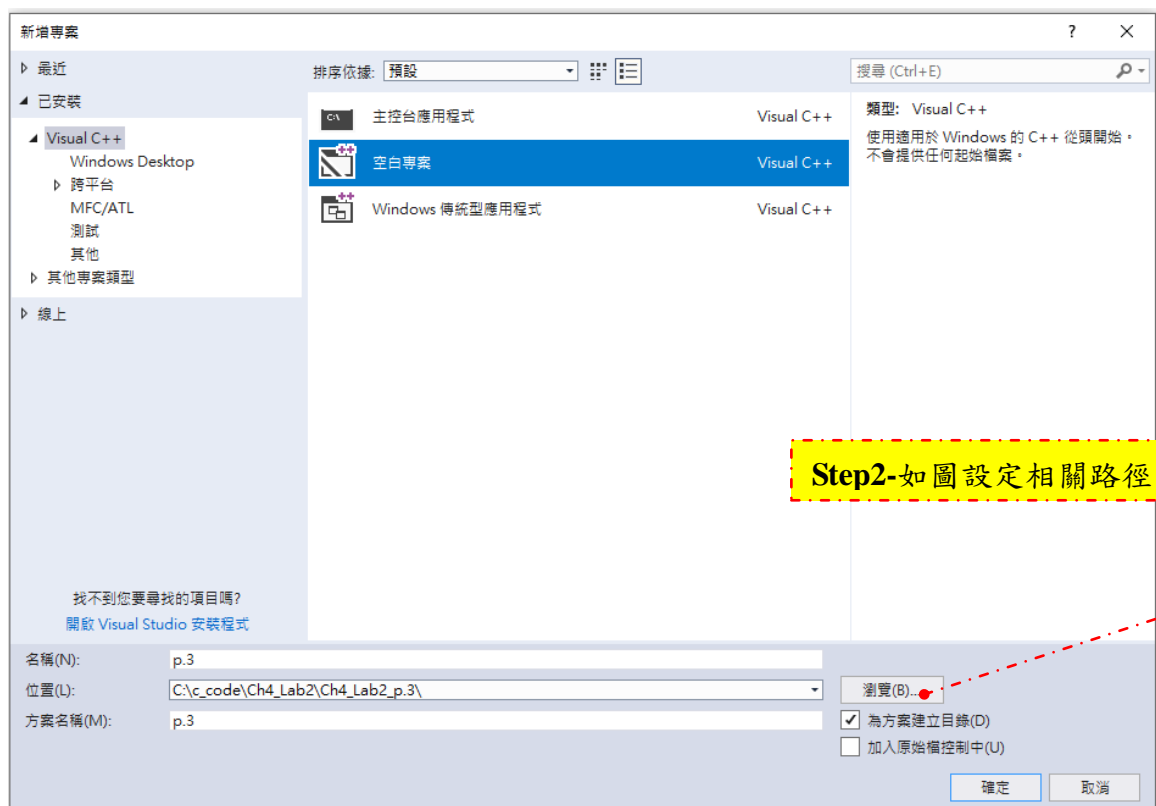
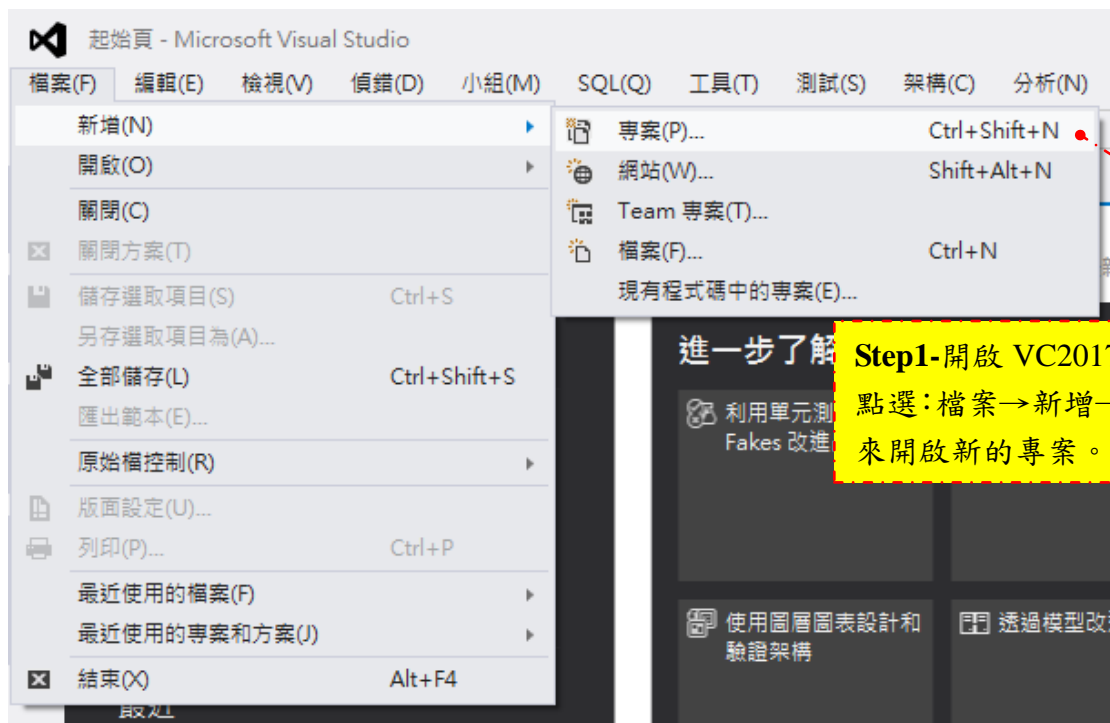
- C 語言的函數的參數傳遞

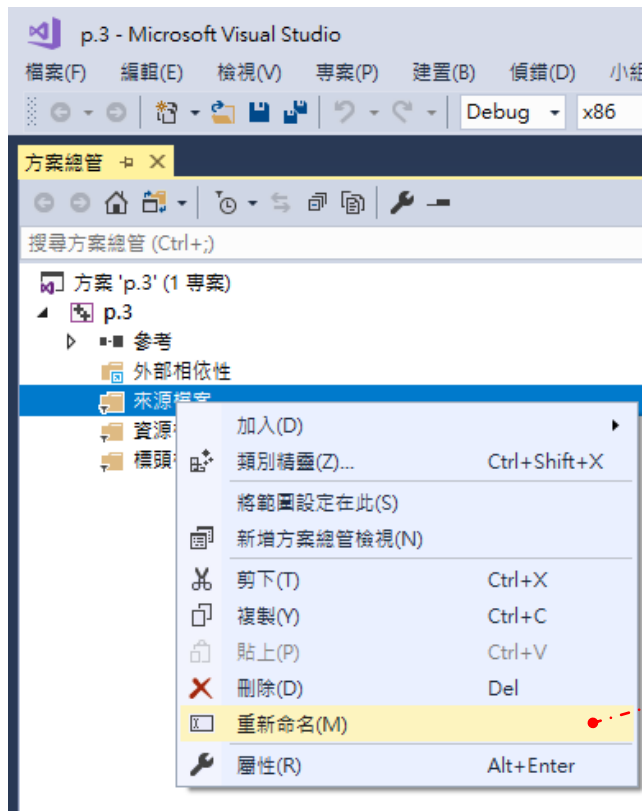
3. 設計步驟：

- 建立新的程式資料夾

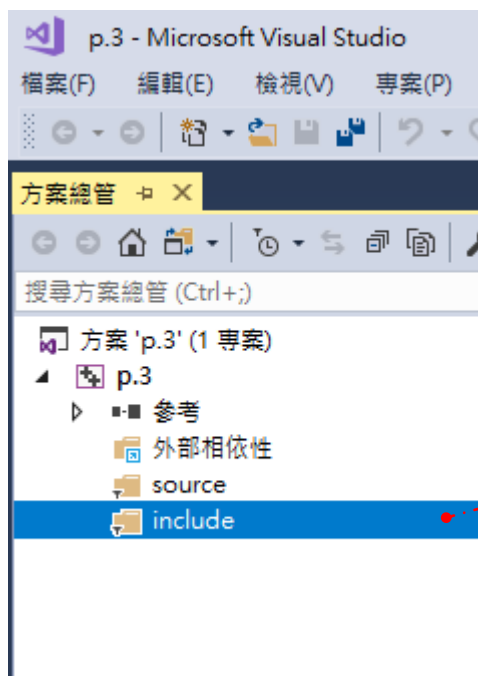
在 C:\c_code 資料夾內新增名為“Ch4_Lab2”的資料夾，再於 Ch4_Lab2 資料夾內建立 Ch4_Lab2_p.2、Ch4_Lab2_p.3...等 6 個資料夾，再分別建立 include、source、temp 等資料夾，建立完成後如下圖



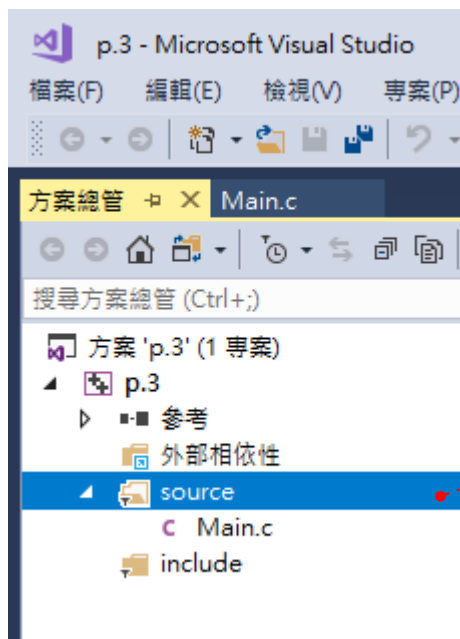
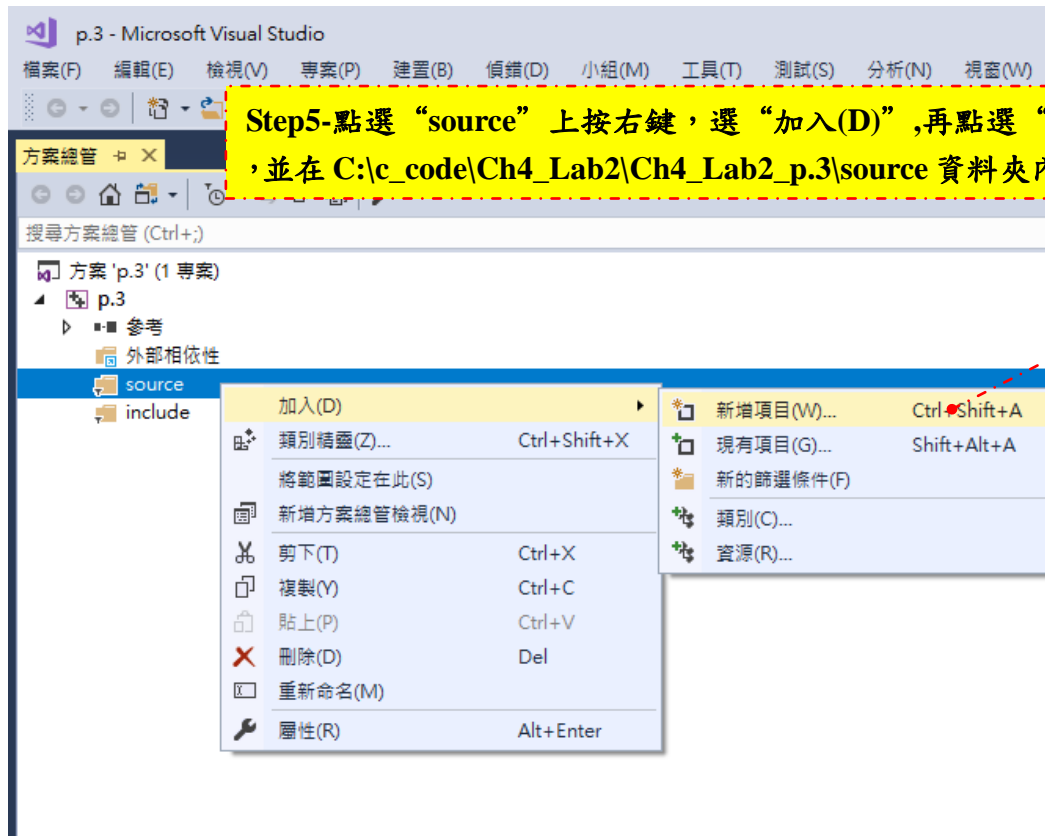




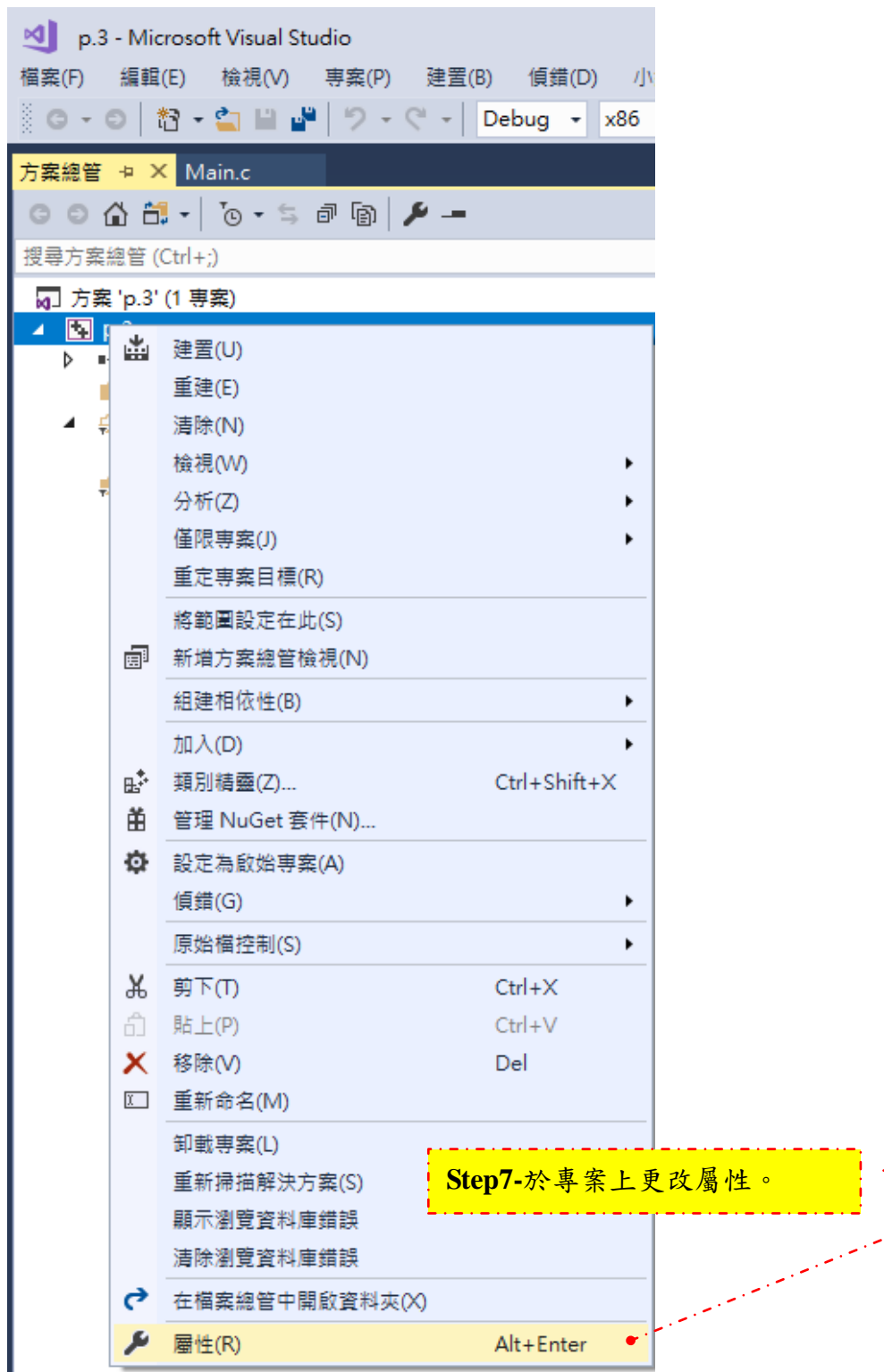
Step3-將原始程式檔重新命名為 source。

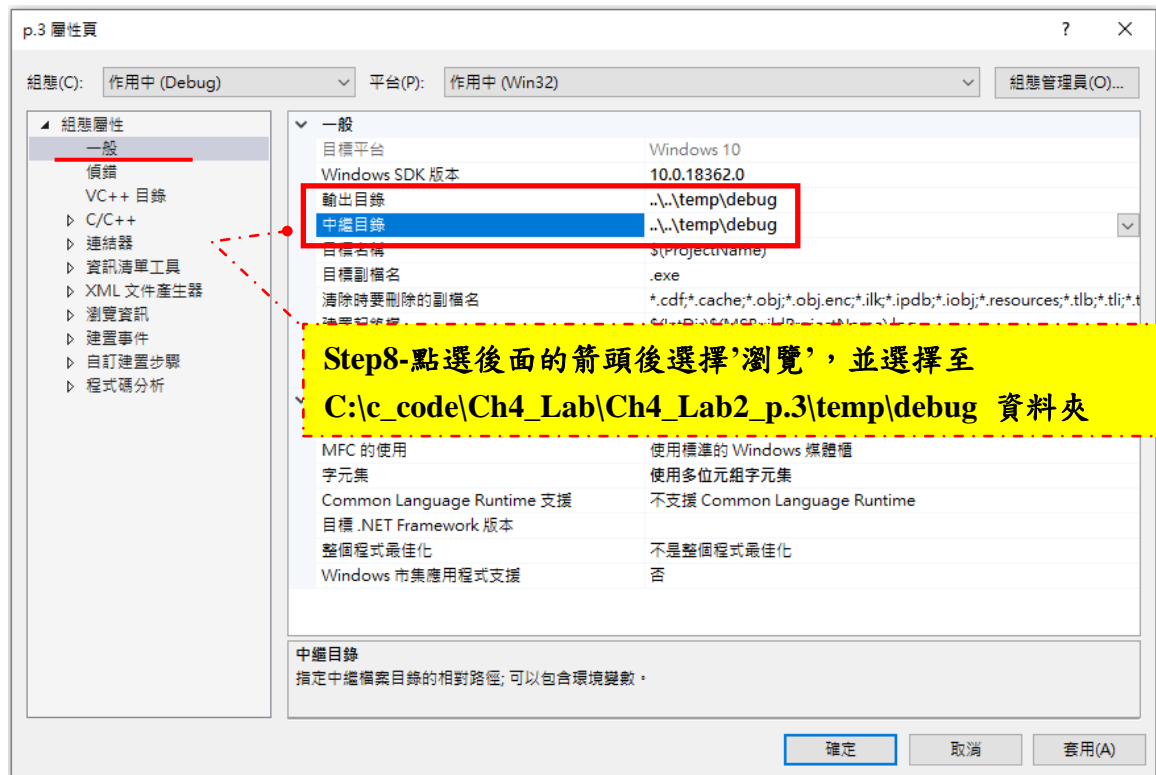


Step4-將標頭檔重新命名為 include，並將資源檔刪除，如圖所示。

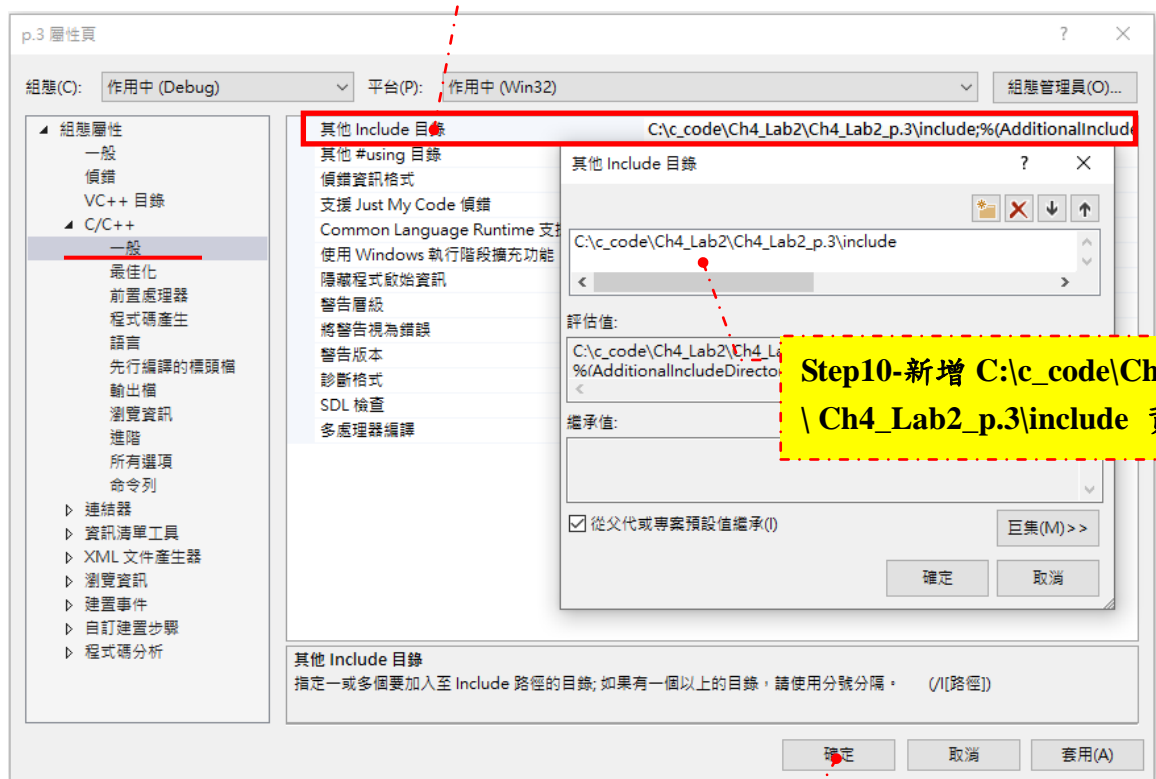


Step6-完成後，如圖所示。





Step9-點選後面的箭頭選擇編輯

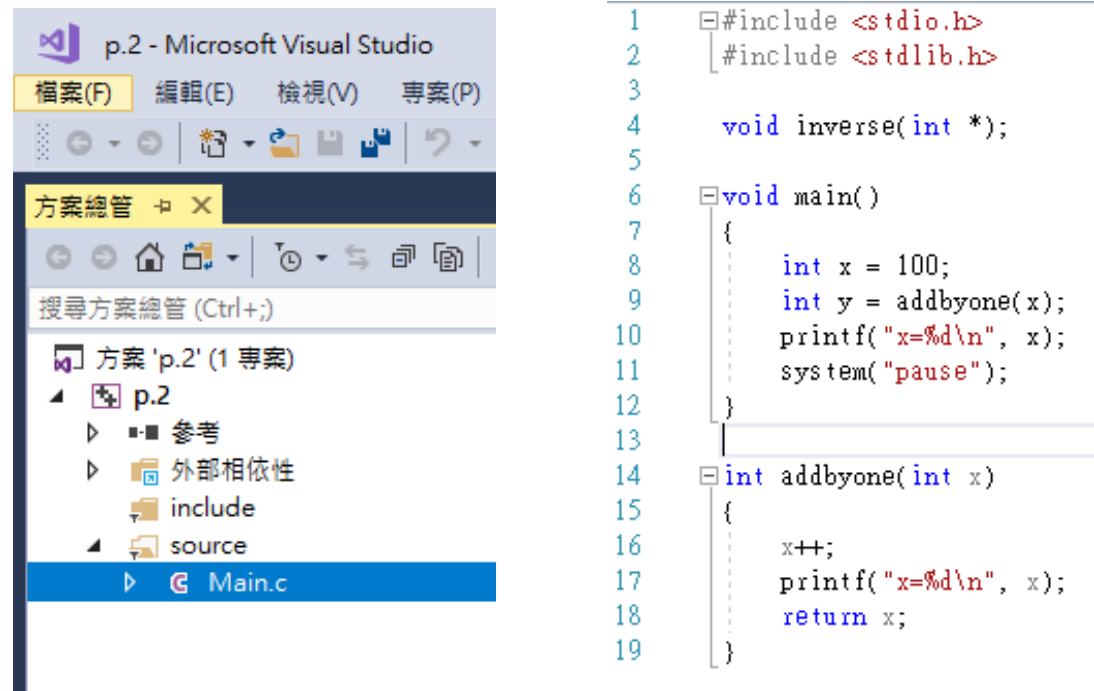


Step11-修改完成後按確定

– Call by value

- 函數呼叫: `function(a, b)`
- 函數定義: `void function(int x, int y)`
- 主要把數值拷貝到函示，函示與主程式的變數互不相干

1. 開始撰寫 C 語言程式



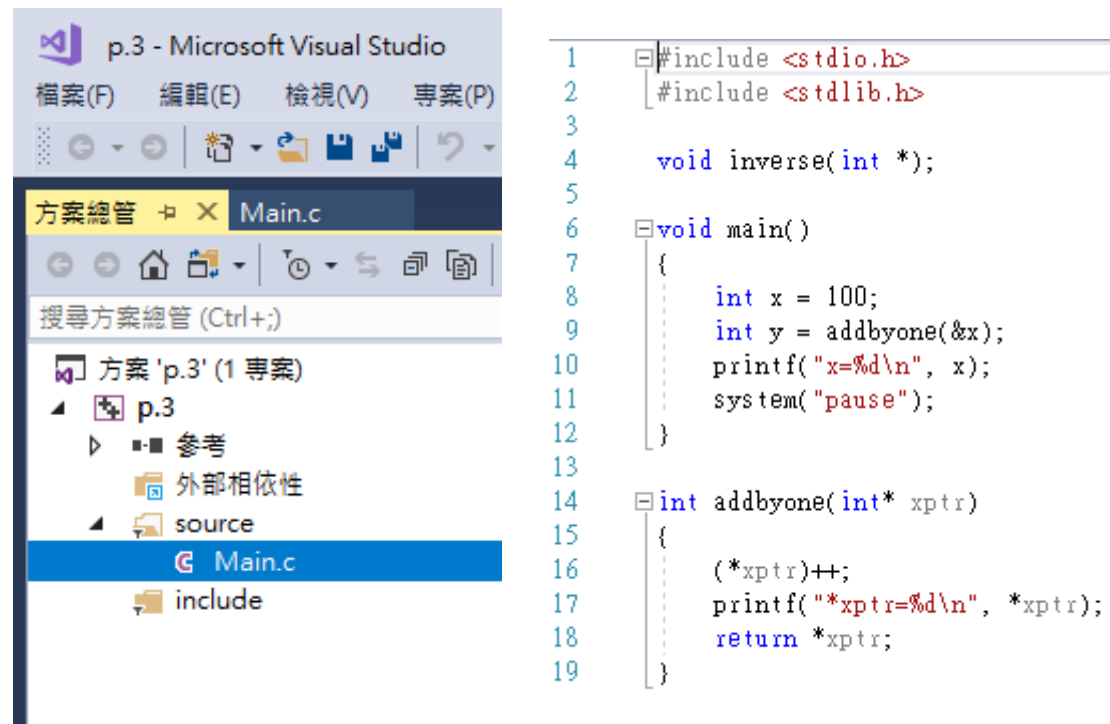
● 執行測試結果



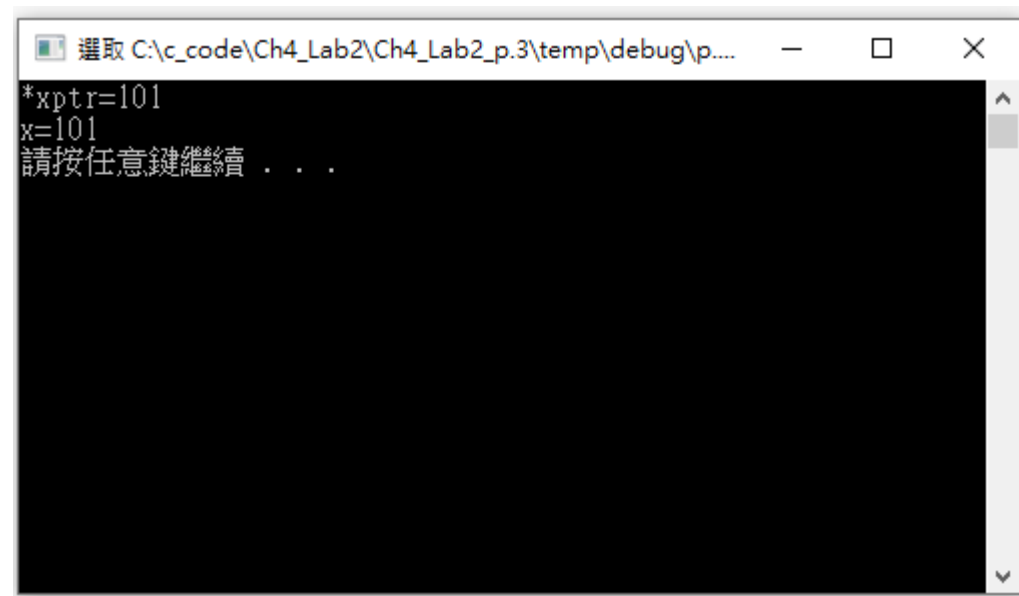
– Call by address

- 函數呼叫: function(&a, &b)
- 函數定義: void function(int * x, int *y)
- 呼叫函數主要傳給函數位址(&x)，函數則以指標指導相對應的變數 (*xptr)，函數運算會更改相對應的變數內容

1. 開始撰寫 C 語言程式

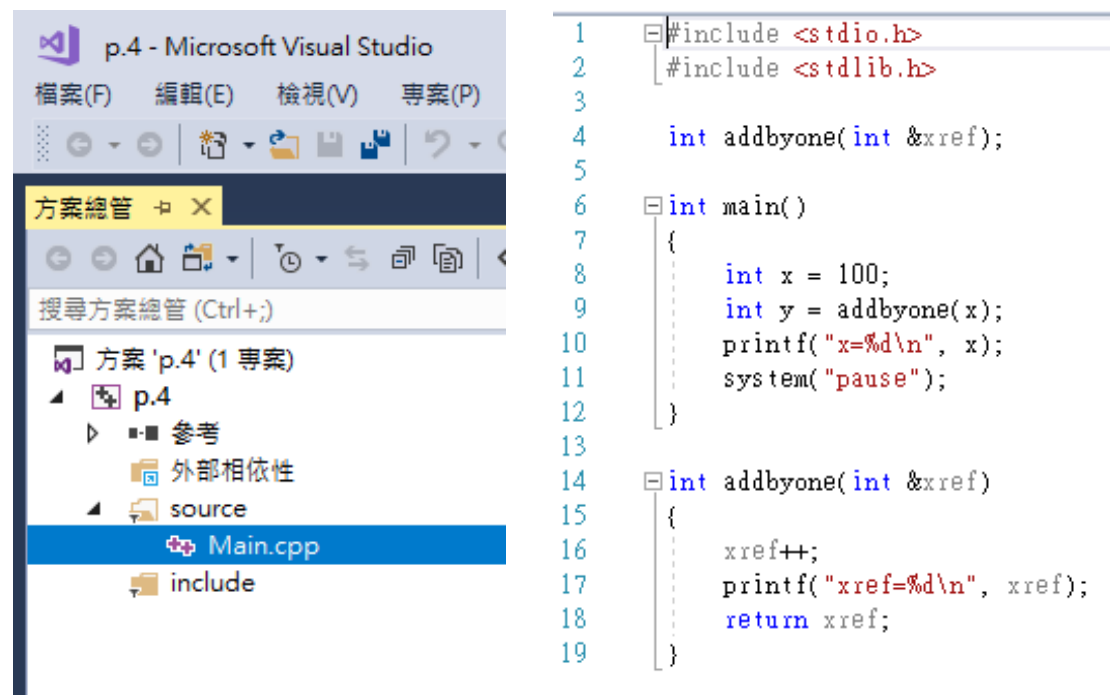


2. 執行測試結果



- Call by reference • 函數呼叫: function(a, b)
 - 函數定義: void function(int &x, int &y)
 - 呼叫函數主要傳給函數參考變數或物件(x), 函數會以 位址(&xref)建立起相連等號, 並表示使用相同記憶體空間, 函數運算會會更改相對應的變數內容
 - 因為 C 沒有支援, 需要把 main.c 改成 main.cpp

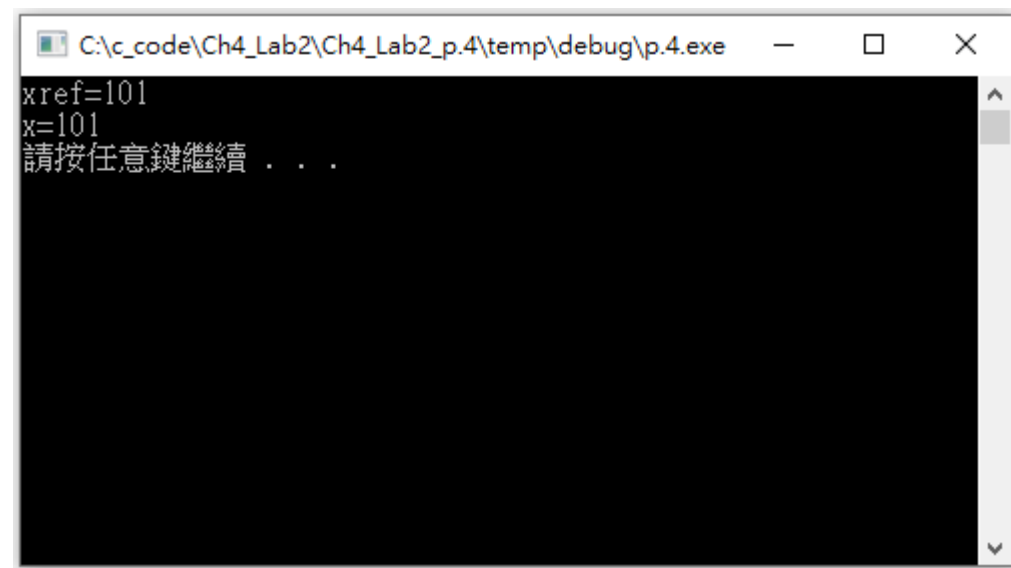
1. 開始撰寫 C 語言程式



The screenshot shows the Microsoft Visual Studio interface. On the left, the 'Solution Explorer' (方案總管) displays the project 'p.4' with a sub-project 'p.4' containing a 'source' folder with 'Main.cpp' and an 'include' folder. The main editor window shows the following C++ code:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int addbyone(int &xref);
5
6  int main()
7  {
8      int x = 100;
9      int y = addbyone(x);
10     printf("x=%d\n", x);
11     system("pause");
12 }
13
14 int addbyone(int &xref)
15 {
16     xref++;
17     printf("xref=%d\n", xref);
18     return xref;
19 }
```

2. 執行測試結果

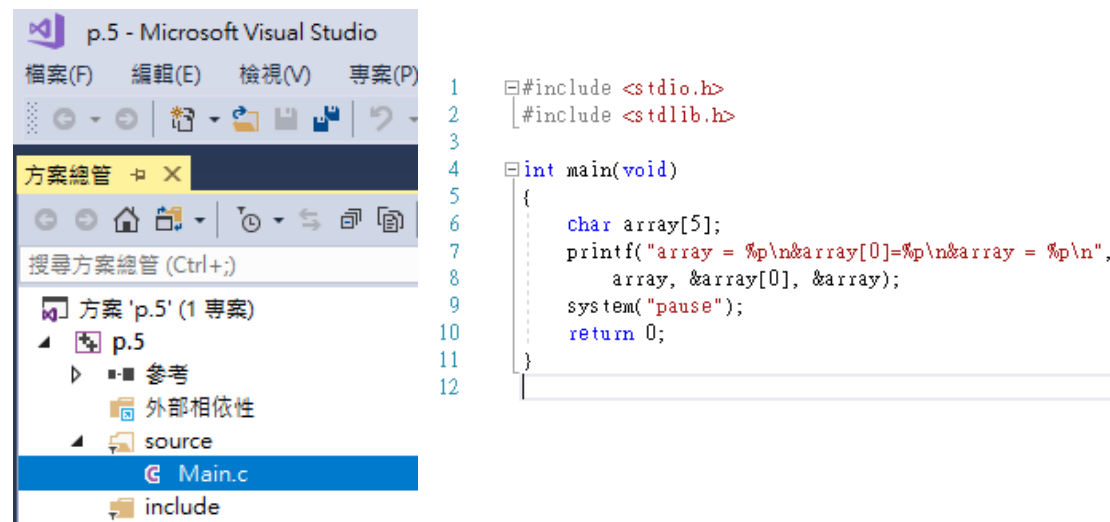


The screenshot shows a command prompt window titled 'C:\c_code\Ch4_Lab2\Ch4_Lab2_p.4\temp\debug\p.4.exe'. The output of the program is as follows:

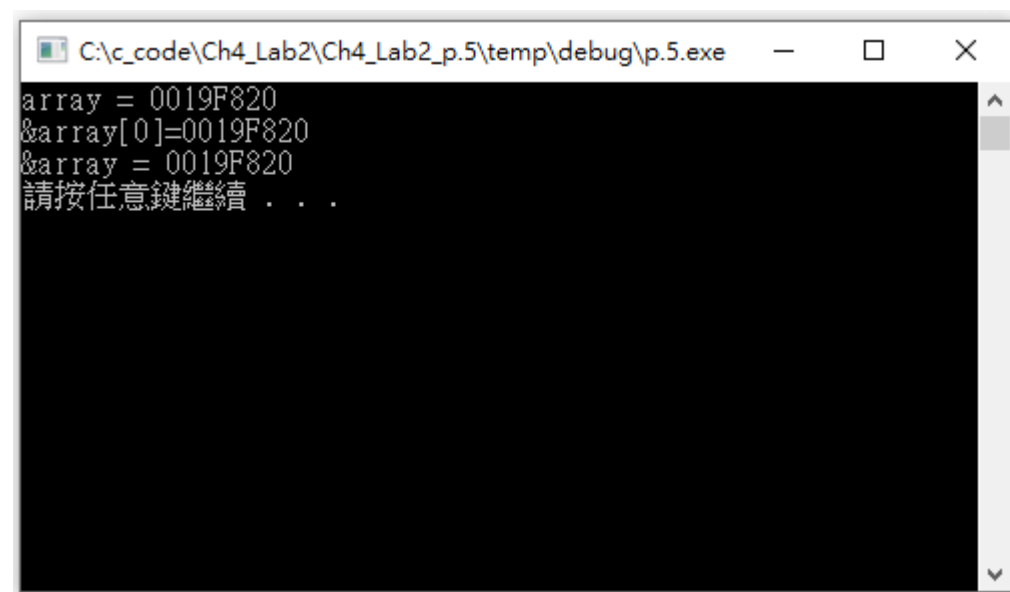
```
xref=101
x=101
請按任意鍵繼續 . . .
```

- 程式利用%p 轉換指定詞（一個用來列印位址的特殊轉換指定詞）印出 array，&array[0]和&array，來驗證陣列名稱確實是此陣列第一個元素所在的位址。
- %p 轉換指定詞通常會將位址以十六進制數的形式印出來。

1. 開始撰寫 C 語言程式




2. 執行測試結果



- 傳遞陣列引數給函式
- 陣列(a[5])自動以 Call by reference (傳參考) 來呼叫傳遞
 - 函數呼叫: modifyArray(a)
 - 函數定義: void modifyArray(int b[])
- 參數 b 接收一個整數陣列
- 陣列的中括號裡不需要指定陣列的大小

1. 開始撰寫 C 語言程式

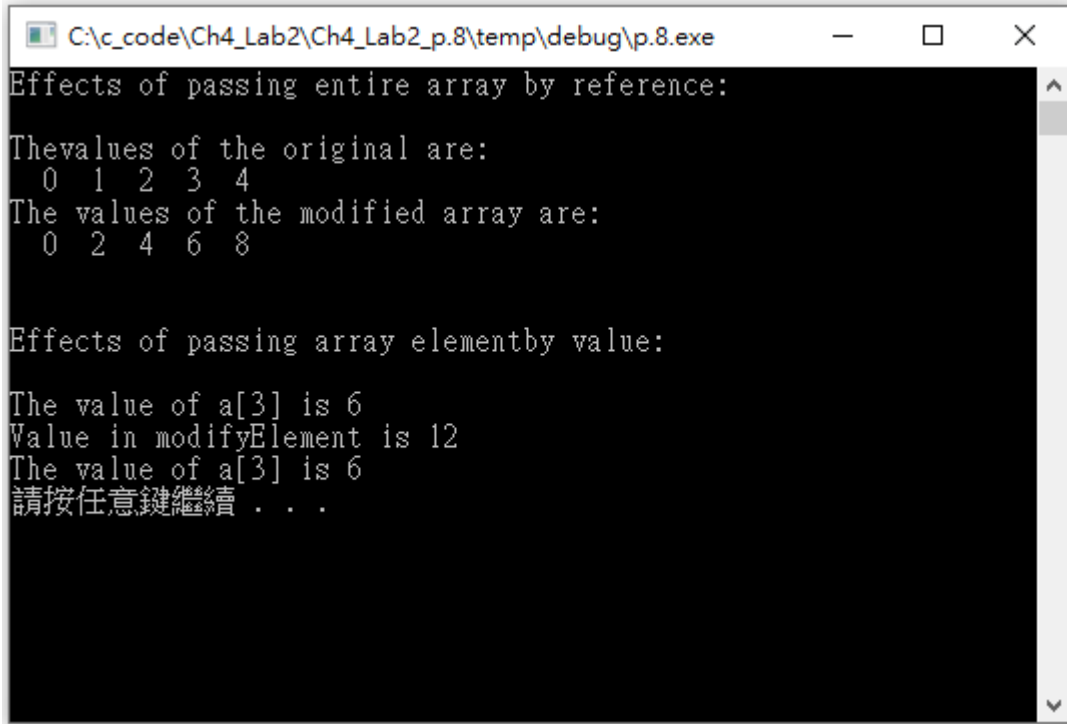


```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #define SIZE 5
4
5  void modifyArray(int b[], int size);
6  void modifyElement(int e);
7
8  int main(void)
9  {
10     int a[SIZE] = { 0,1,2,3,4 };
11     int i;
12
13     printf("Effects of passing entire array by reference:\n\nThe"
14           "values of the original are:\n");
15
16     for (i = 0; i < SIZE; i++)
17     {
18         printf("%3d", a[i]);
19     }
20     printf("\n");
21
22     modifyArray(a, SIZE);
23     printf("The values of the modified array are:\n");
24     for (i = 0; i < SIZE; i++)
25     {
26         printf("%3d", a[i]);
27     }
28
29     printf("\n\nEffects of passing array element"
30           "by value:\n\nThe value of a[3] is %d\n", a[3]);
31
32     modifyElement(a[3]);
33     printf("The value of a[3] is %d\n", a[3]);
34
35     system("pause");
36     return 0;
37 }
38
39 void modifyArray(int b[], int size)
40 {
41     int j;
42
43     for (j = 0; j < size; j++)
44     {
45         b[j] *= 2;
46     }
47 }
48
49 void modifyElement(int e)
50 {
51     printf("Value in modifyElement is %d\n", e *= 2);
52 }
53
54

```

2. 執行測試結果



```
C:\c_code\Ch4_Lab2\Ch4_Lab2_p.8\temp\debug\p.8.exe
Effects of passing entire array by reference:

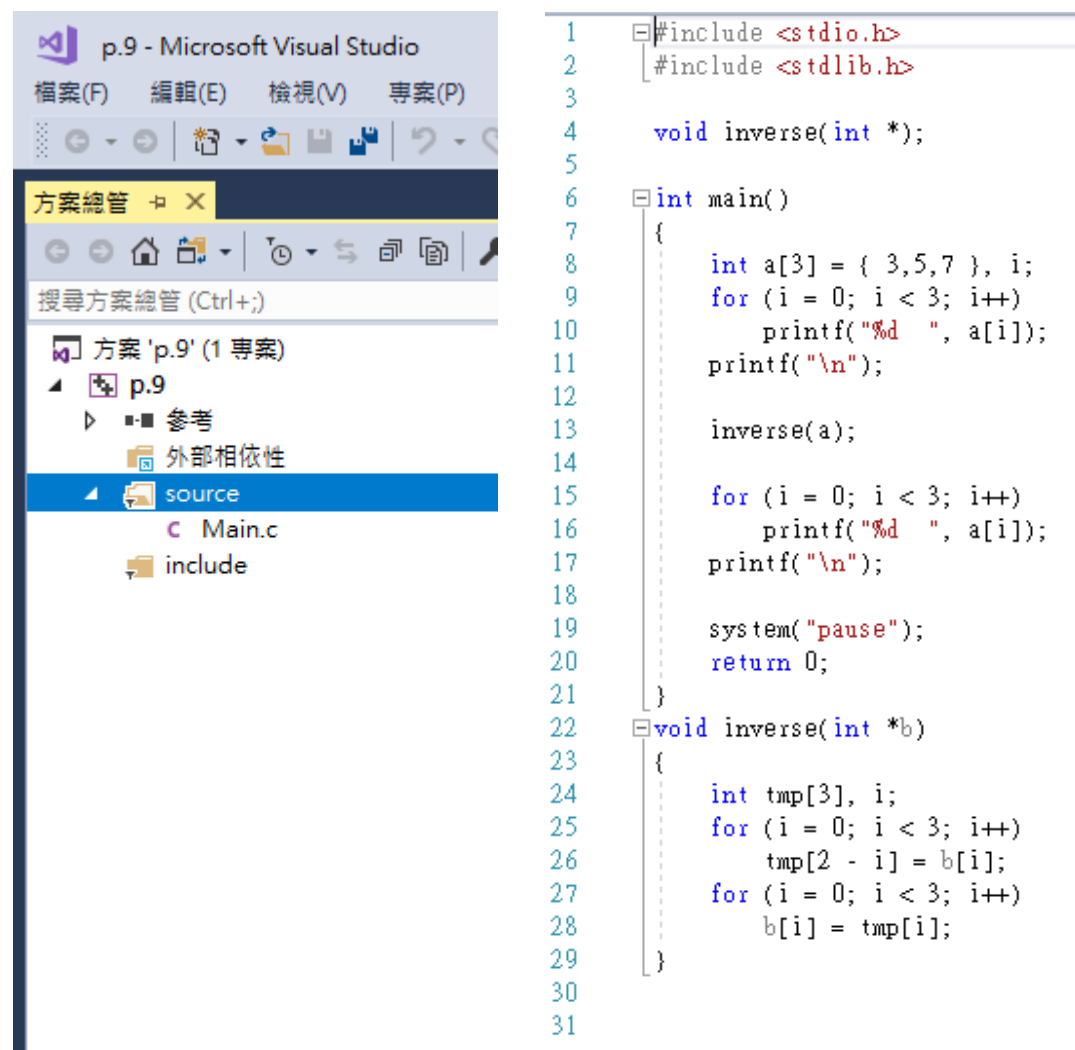
The values of the original are:
0 1 2 3 4
The values of the modified array are:
0 2 4 6 8

Effects of passing array element by value:

The value of a[3] is 6
Value in modifyElement is 12
The value of a[3] is 6
請按任意鍵繼續 . . .
```

- Call by Address

1. 開始撰寫 C 語言程式



2. 執行測試結果



• 檢查

