

# Better IDE: Coding with C/C++

Yu-Hsuan Chen

2019 Spring, Introduction to Programming

# Outline

- 為何不用Dev-C++
- VSCode介紹
- 如何安裝GNU C Compiler
- GCC / G++指令與終端機操作
- VSCode上不用終端機的替代方案

# **棄用Dev-C++的幾個理由**

# 過時的版本

- Dev-C++ 最後一次更新已經是2015年，版本號停留在5.1.1
- 內附的GNU Compiler也停留在4.9.2



# 不靈活的 Formatter

## 好的排版賞心悅目 不好的排版讓人翻桌

右邊的Code  
你看得下去嗎？

```
[*] 新文件1
1 #include<conio.h>
2 #include<stdio.h>
3 int main()
4 {
5     float e[100][100],w[100][100],p[100],q[100],t;
6     int i,n,j,k,l,m,r,root[100][100];
7     printf("this is optimal binary search tree\n");
8     printf("enter the value of n\n");
9     scanf("%d",&n);
10    printf("enter the values of the probability\n");
11    for(i=1;i<=n;i++)
12    {
13        scanf("%f",&p[i]);
14    }
15    printf("enter the values of the probability of dummy\n");
16    for(i=0;i<=n;i++)
17    {
18        scanf("%f",&q[i]);
19    }
20    for(i=1;i<=n+1;i++)
21    {
22        e[i][i-1]=q[i-1];
23        w[i][i-1]=q[i-1];
24    }
25    for(l=1;l<=n;l++)
26    {
27        for(i=1;i<=n-l+1;i++)
28        {
29            j=i+l-1;
30            e[i][j]=100.0;
31            w[i][j]=w[i][j-1]+p[j]+q[j];
32            for(r=i;r<=j;r++)
33            {
34                t=e[i][r-1]+e[r+1][j]+w[i][j];
35                if(t<e[i][j])
36                {
37                    e[i][j]=t;
38                    root[i][j]=r;
39                }
40            }
41        }
42    }
43 }
```

# 堪用的 Debugger

Debugger的Trace功能簡  
單、有時候會無法用

A screenshot of a debugger's user interface. At the top, there are tabs for '專案' (Project), '類別' (Category), and '除錯' (Debug). Below the tabs, a code editor window titled 'testset2.cpp' displays the following C++ code:

```
1 #include<stdio.h>
2
3 int main()
4 {
5     int arr[1000];
6     int i;
7     for (i=0;i<1000;i++)
8     {
9         arr[i] = i;
10    }
11 }
```

The line 'arr[i] = i;' is highlighted in blue, indicating it is the current line of execution. The toolbar below the code editor contains icons for '編譯器訊息' (Compiler Messages), '資源檔' (Resource), '編譯紀錄' (Compile Log), '除錯' (Debug) (which is checked), '搜尋結果' (Search Results), and '最小化' (Minimize).



謎樣的  
optimization  
某一些非正規的寫法  
Dev-C++會自己幫你修正

```
#include<iostream>

int main()
{
    int a;
    std::cout << a;
}
```

思考一下上面的a值應該會印出什麼?

```
#include<iostream>

int main()
{
    int a;
    std::cout << a;
}
```

正常來說，應該是個任意的亂數  
但是Dev-C++會自己幫你初始化成0啊

```
1 // Example program
2 #include <iostream>
3 #include <string>
4
5 int main()
6 {
7     int a;
8     std::cout << a;
9 }
10 8:19: warning: 'a' is used uninitialized in this function [-Wuninitialized]
```

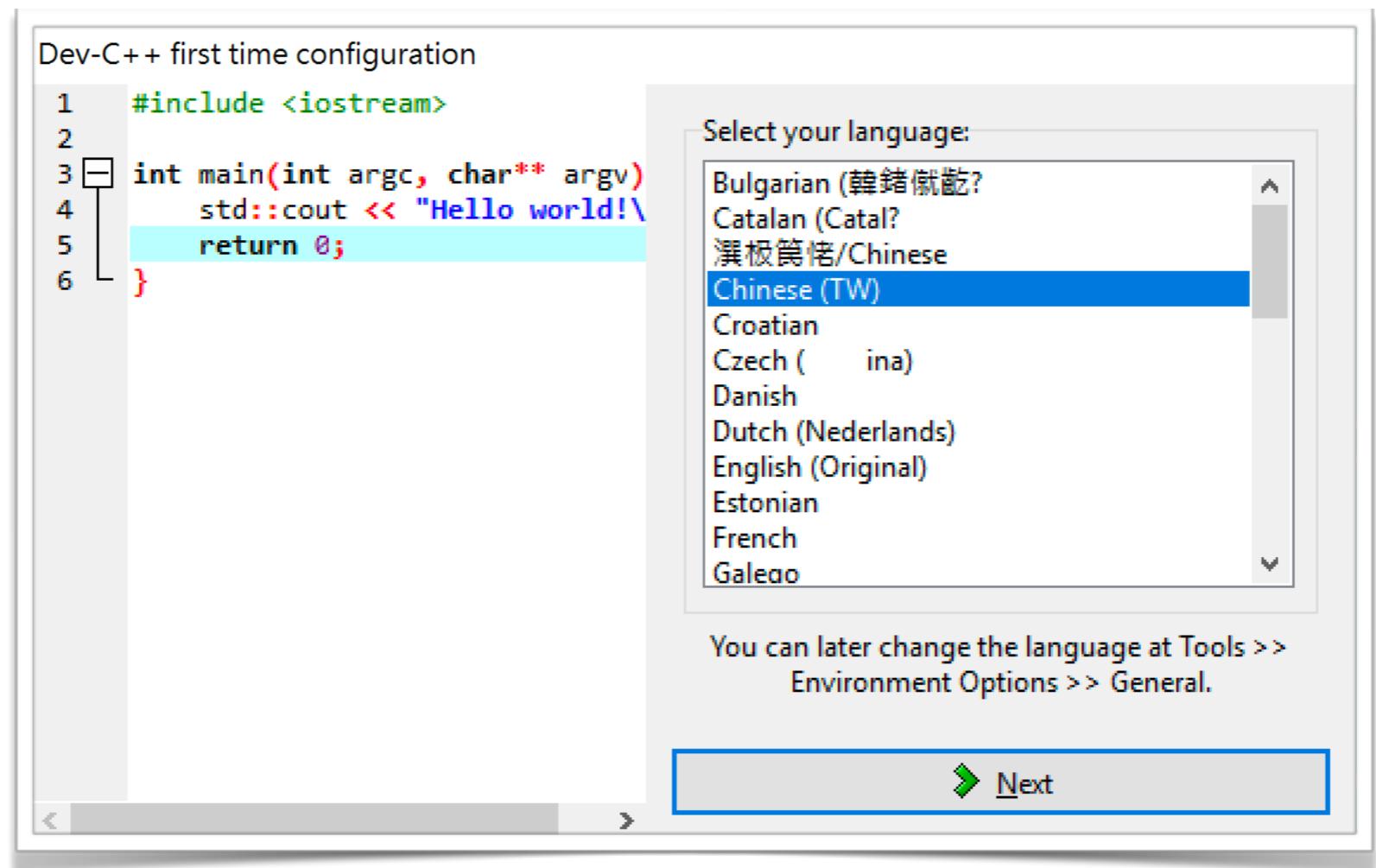
這種用法Compiler通常會  
提出Warning的

因為Dev-C++幫你了一把  
也沒有主動告訴你出問題了  
然後你就覺得答案正確，開開心心上傳E-tutor

通過	執行結果
是	AC: 完全正確
是	AC: 完全正確
是	AC: 完全正確
否	WA: 錯誤的結果
是	AC: 完全正確

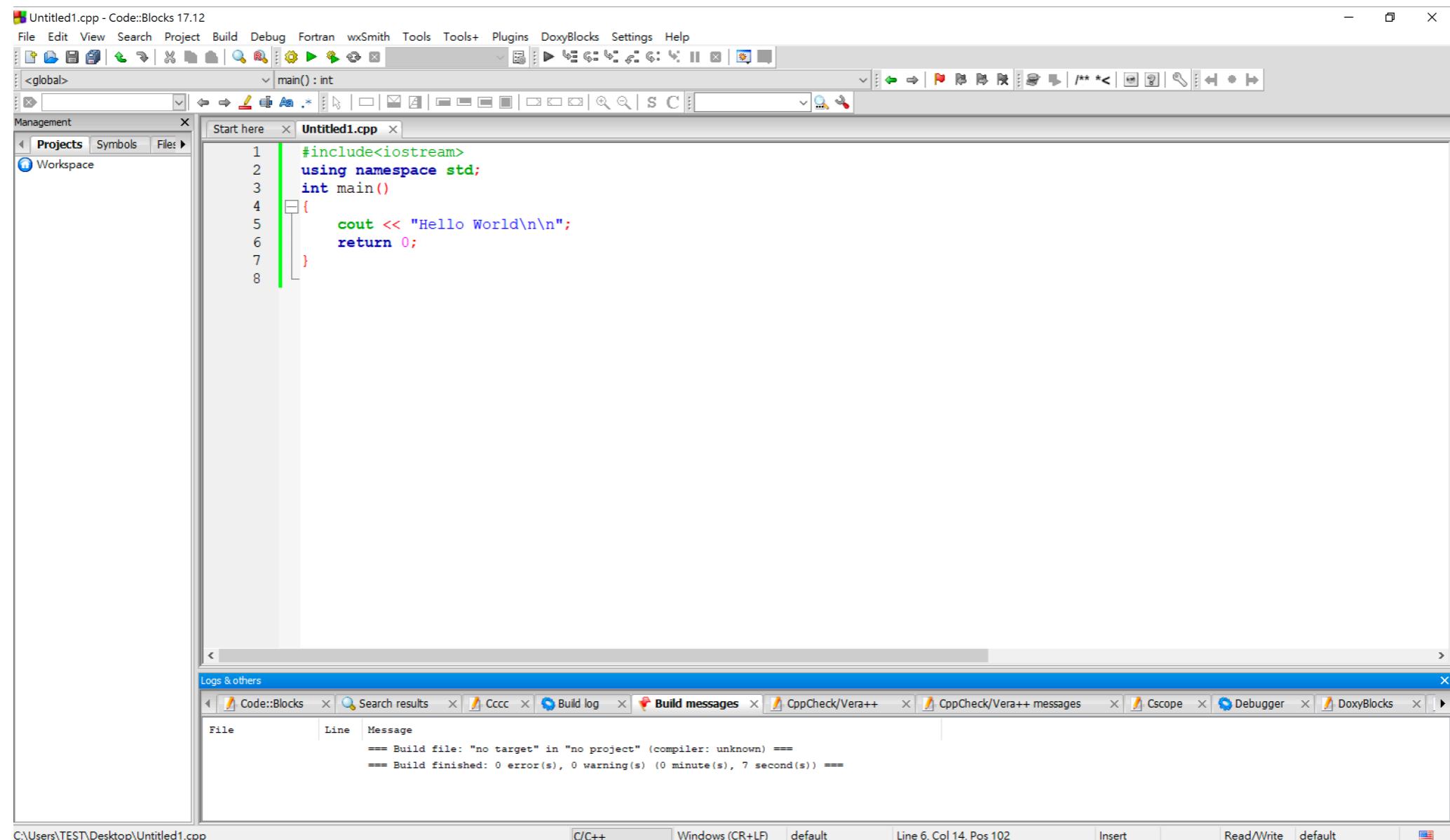
看吧，這下WA了吧，而且是保密測資

**那為什麼教學現場還是這麼多老師喜歡用？**



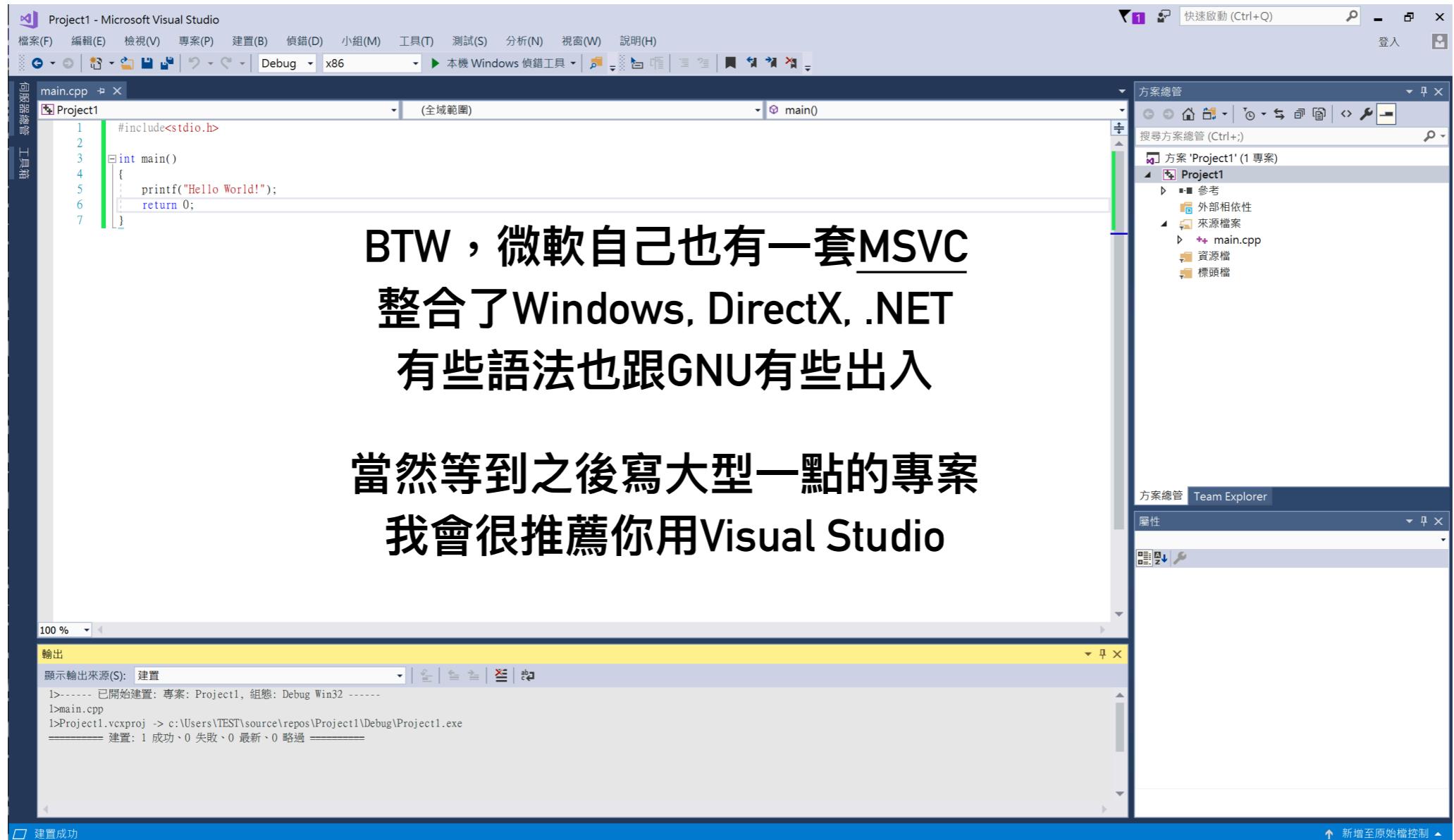
因為他是少數簡單好上手、介面有中文的IDE

**讓我們看看其他IDE的長相...**



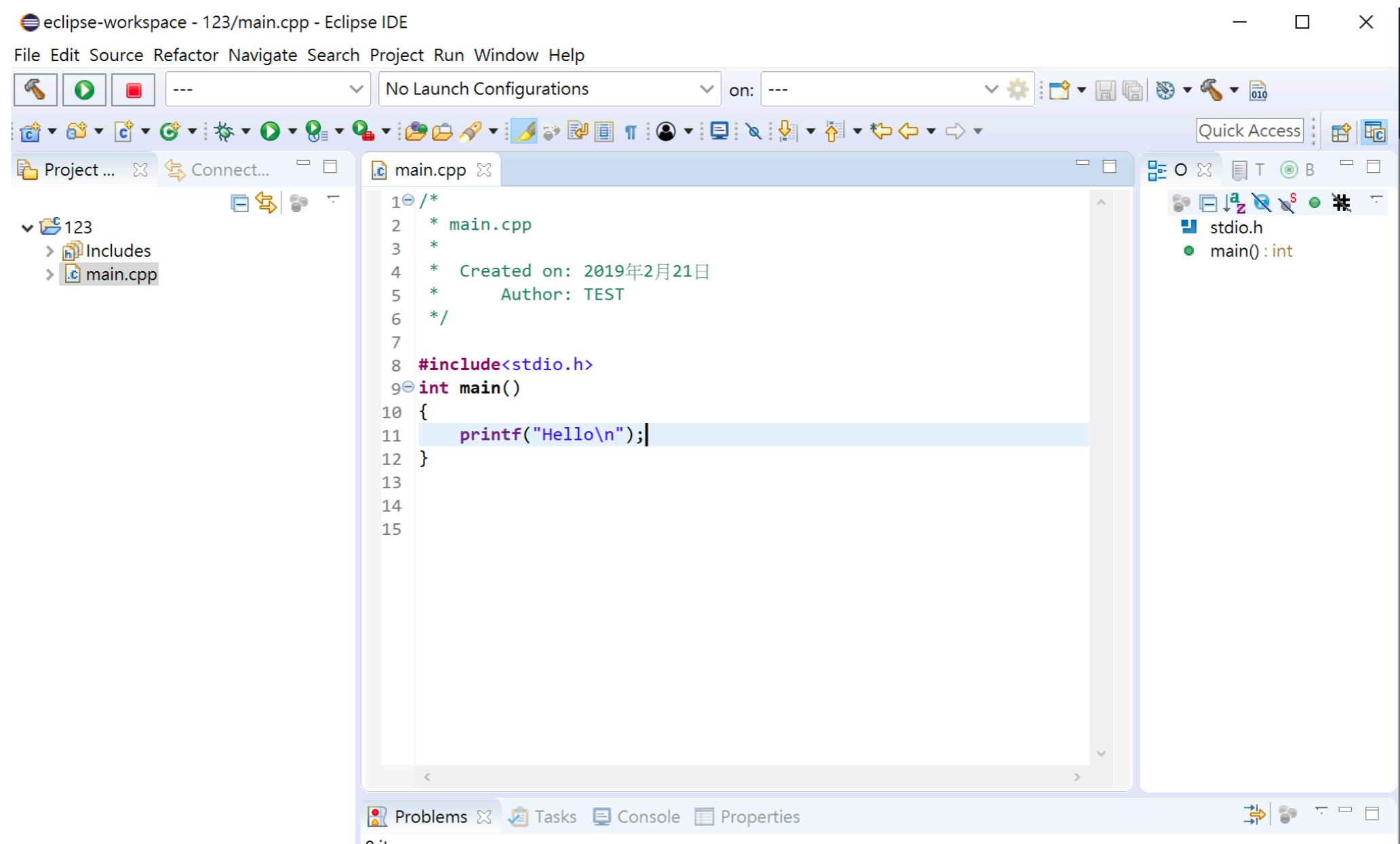
# Code::Blocks

- 最新版本17.12
- 沒有中文介面
- 具自動完成功能



# Visual Studio

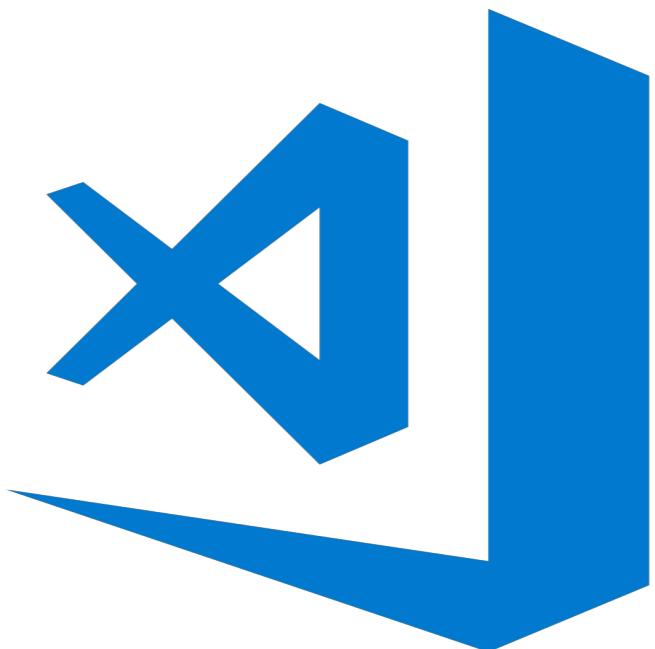
- 最新版本2017
- 有中文
- 功能最強大、介面較複雜
- 占空間(VC環境約需6GB)



# Eclipse

- 最新版本2018-12
- 部分中文化(有語言包)
- 不好說(裝Eclipse之前還需要裝Java)

**如果沒有強大的IDE  
何不自己組一個？**



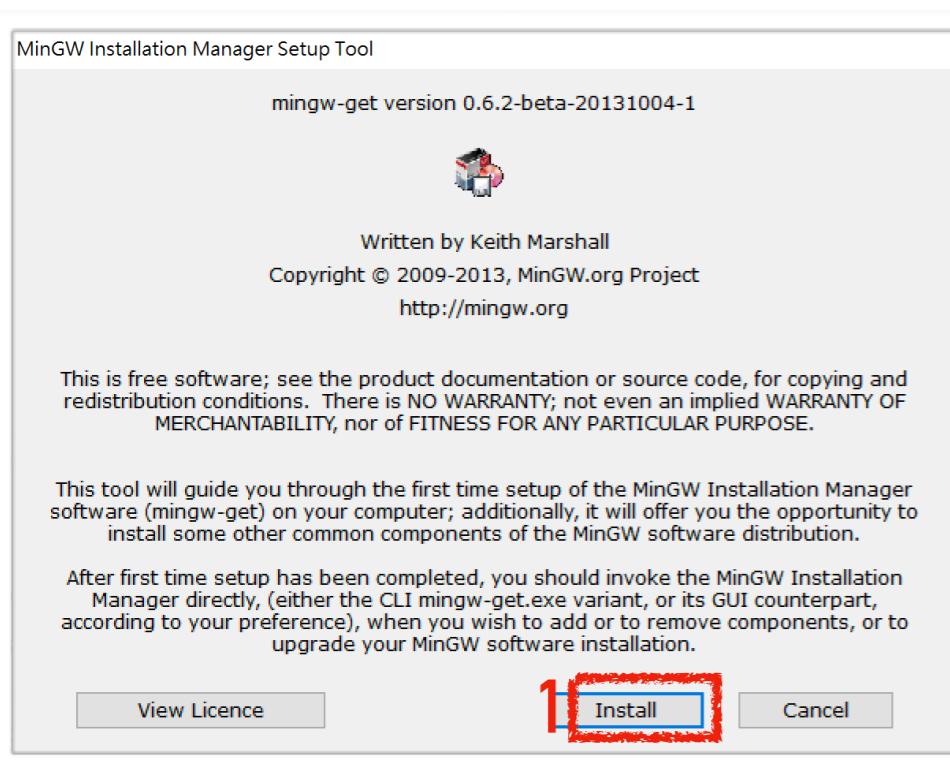
今日兩大主角

Visual Studio Code + GCC Compiler

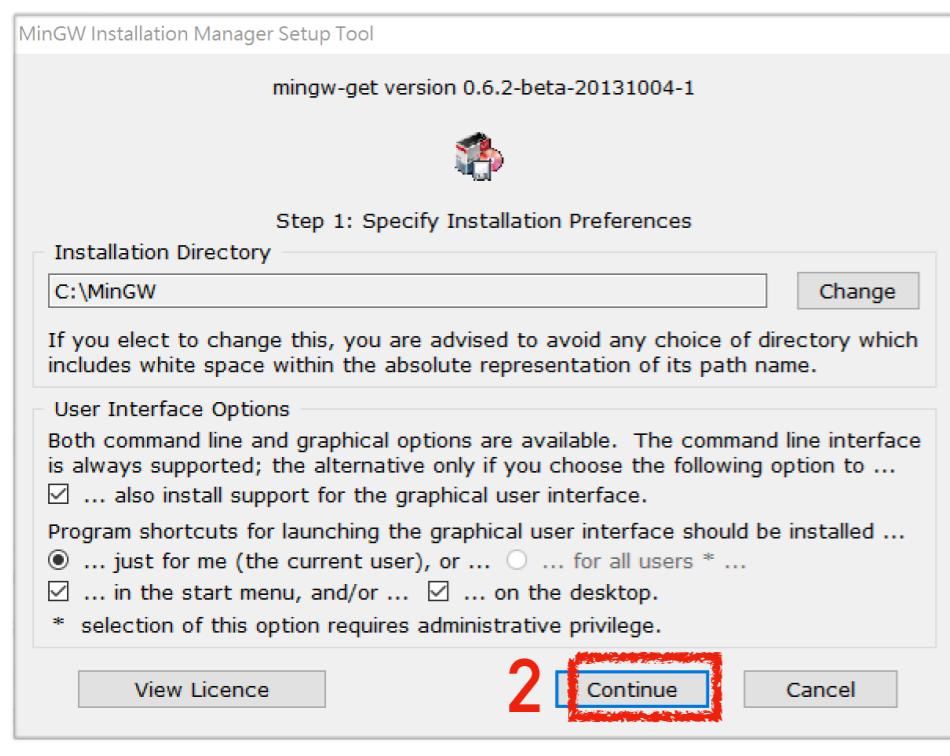
# 安裝MinGW<sub>1</sub>

- MinGW是將GCC編譯器和GNU Binutils移植到Windows底下的產物。我們要透過MinGW下載GNU C的編譯器
- 下載MinGW Installer：  
<https://sourceforge.net/projects/mingw/>

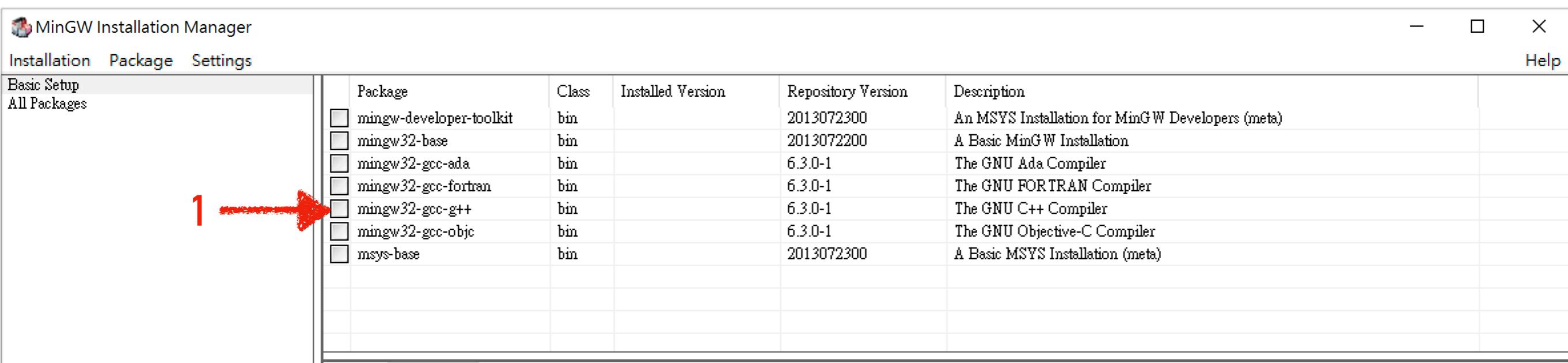
# 安裝MinGW 2



This screenshot shows the second step of the setup process. It displays the software's logo and version information. The title 'Step 2: Download and Set Up MinGW Installation Manager' is visible. A message indicates that the catalogue update has completed. Below this, a progress bar shows 'Processed 113 of 113 items : 100 %'. A large green bar indicates the completion of the download. A 'Details' section shows the command-line output of the download process, which includes unpacking the setup file, updating the installation database, and registering the component. At the bottom, there are three buttons: 'View Licence', 'Continue' (which is highlighted with a red box), and 'Quit'. The number '3' is written next to the 'Continue' button.



# 安裝MinGW 3

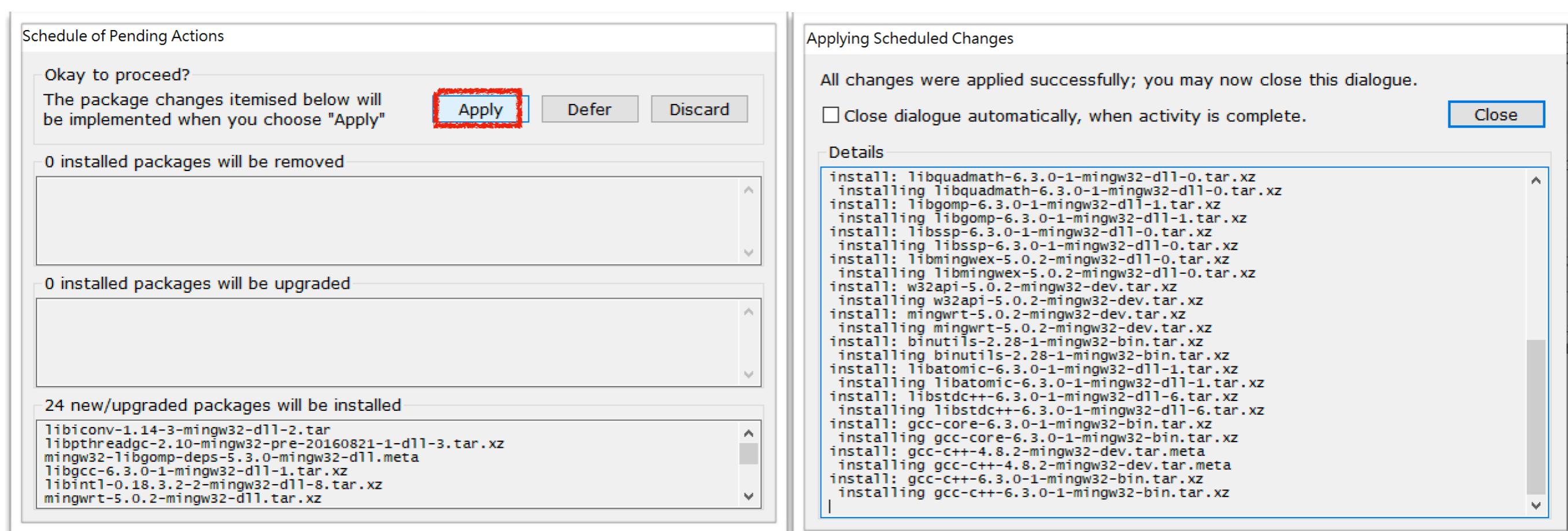


The image contains three screenshots of the MinGW Installation Manager:

- Screenshot 1:** Shows the "All Packages" list. A red arrow labeled "1" points to the "All Packages" link.
- Screenshot 2:** Shows the "General" tab of the "mingw32-gcc-g++" package details. A context menu is open with options: "Unmark" (disabled), "Mark for Installation" (selected and highlighted in blue), "Mark for Reinstallation", "Mark for Upgrade", and "Mark for Removal". A red number "2" is placed over the "Mark for Installation" option.
- Screenshot 3:** Shows the application menu with "Apply Changes" selected and highlighted in blue. A red number "3" is placed over the "Apply Changes" option.

# 安裝MinGW\_4

- 點擊Apply開始下載必要元件



# 設定GCC PATH 1

- 控制台 > 系統與安全性 > 系統 > 進階系統設定



This screenshot shows the 'System' settings in the Control Panel. It displays basic computer information and system details. The '檢視電腦的基本資訊' (View basic computer info) section includes:

- Windows 版本: Windows 10 企業版
- © 2018 Microsoft Corporation. 著作權所有，並保有所有權利。

The '系統' (System) section provides hardware and software specifications:

處理器:	Intel(R) Core(TM) i7-7
已安裝記憶體 (RAM):	2.00 GB
系統類型:	64 位元作業系統 · x64
手寫筆與觸控:	手寫筆支援

The '電腦名稱、網域及工作群組設定' (Computer name, domain, and workgroup settings) section lists:

電腦名稱:	DESKTOP-S62VA3C
完整電腦名稱:	DESKTOP-S62VA3C

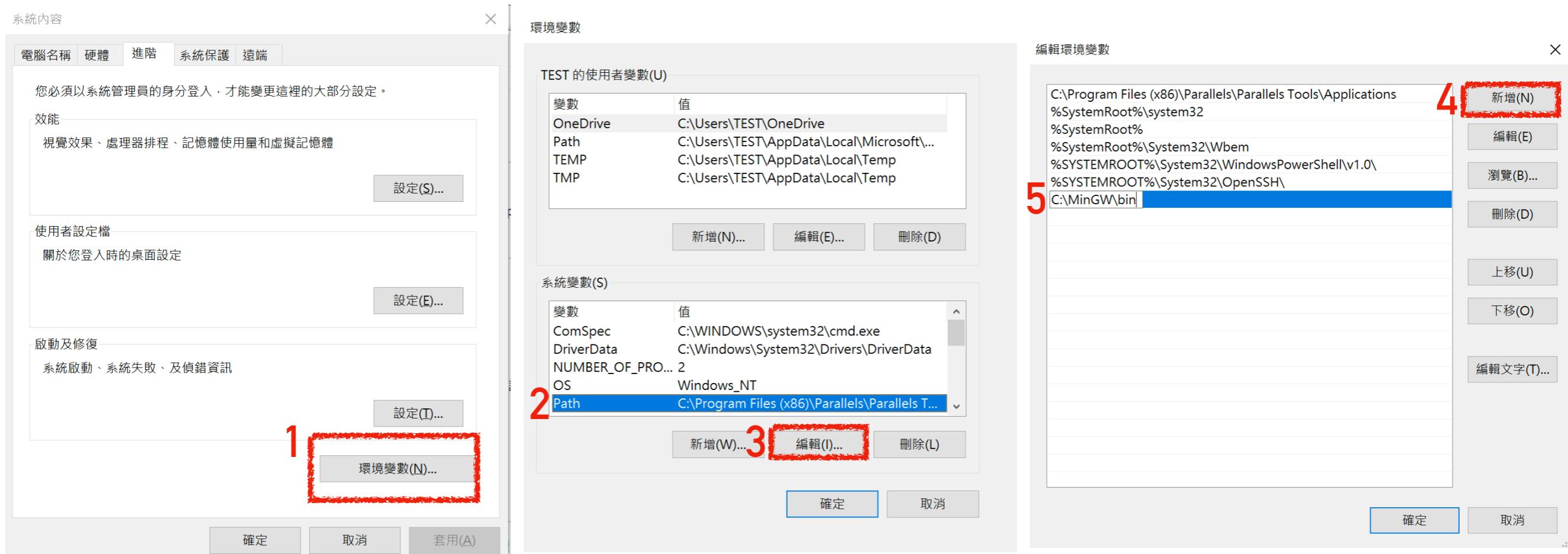
The 'Windows 啟用' (Windows Activation) section includes:

- Windows 已啟用 (Windows is activated)
- 閱讀 Microsoft 軟體授權條款 (Read the Microsoft Software License Terms)

At the bottom, the product identification code is listed: 產品識別碼: 00329-00000-00003-AA748.

# 設定GCC PATH 2

- 在系統變數中加入 **C:\MinGW\bin**  
加入完成點擊確定關閉所有對話框



# 設定GCC PATH 3

- 打開命令提示字元，輸入gcc -v 跟 g++ -v  
如果可以正確看到GCC版本就表示成功



```
命令提示字元
Microsoft Windows [版本 10.0.17763.1]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。

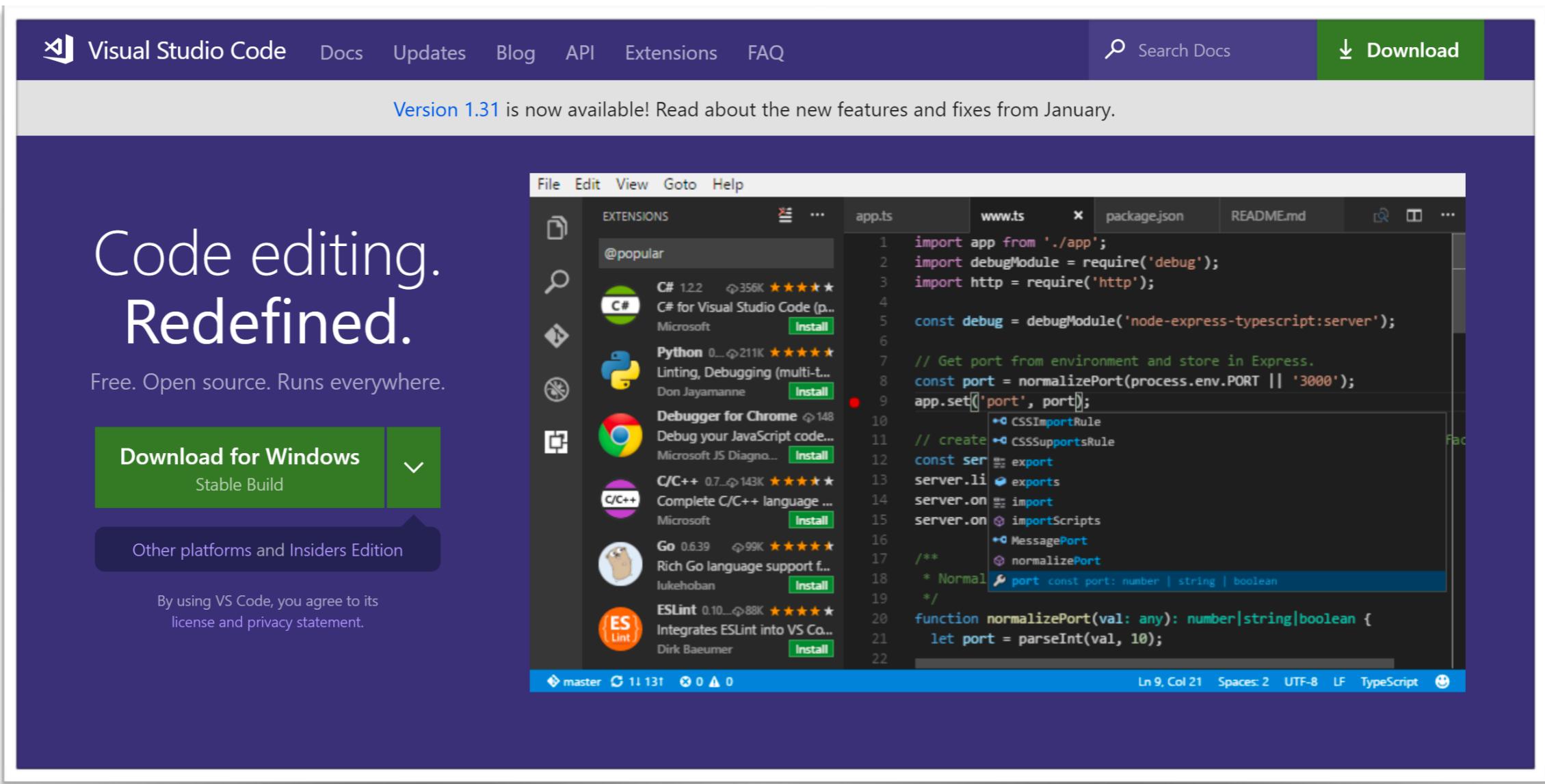
C:\Users\TEST>gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=c:/mingw/bin/../libexec/gcc/mingw32/6.3.0/lto-wrapper.exe
Target: mingw32
Configured with: ./src/gcc-6.3.0/configure --build=x86_64-pc-linux-gnu --host=mingw32 --target=mingw32 --with-gmp=/mingw --with-mpfr --with-mpc=/mingw --with-isl=/mingw --prefix=/mingw --disable-win32-registry --with-arch=i586 --with-tune=generic --enable-languages=c,c++,objc,obj-c++,fortran,ada --with-pkgversion='MinGW.org GCC-6.3.0-1' --enable-static --enable-shared --enable-threads --with-dwarf2 --disable-sjlj-exceptions --enable-version-specific-runtime-libs --with-libiconv-prefix=/mingw --with-libintl-prefix=/mingw --enable-stdcxx-debug --enable-libgomp --disable-libvtv --enable-nls
Thread model: win32
gcc version 6.3.0 (MinGW.org GCC-6.3.0-1)

C:\Users\TEST>g++ -v
Using built-in specs.
COLLECT_GCC=g++
COLLECT_LTO_WRAPPER=c:/mingw/bin/../libexec/gcc/mingw32/6.3.0/lto-wrapper.exe
Target: mingw32
Configured with: ./src/gcc-6.3.0/configure --build=x86_64-pc-linux-gnu --host=mingw32 --with-gmp=/mingw --with-mpfr=/mingw --with-mpc=/mingw --with-isl=/mingw --prefix=/mingw --disable-win32-registry --target=mingw32 --with-arch=i586 --enable-languages=c,c++,objc,obj-c++,fortran,ada --with-pkgversion='MinGW.org GCC-6.3.0-1' --enable-static --enable-shared --enable-threads --with-dwarf2 --disable-sjlj-exceptions --enable-version-specific-runtime-libs --with-libiconv-prefix=/mingw --with-libintl-prefix=/mingw --enable-stdcxx-debug --with-tune=generic --enable-libgomp --disable-libvtv --enable-nls
Thread model: win32
gcc version 6.3.0 (MinGW.org GCC-6.3.0-1)

C:\Users\TEST>
```

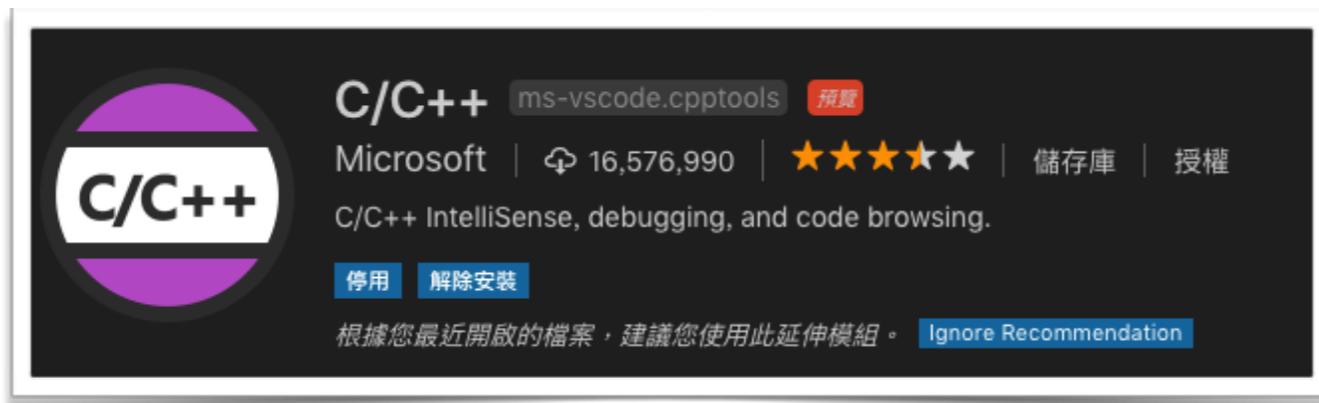
# 安裝Visual Studio Code

- 下載網址：<https://code.visualstudio.com/>



# 下載Code的額外套件

- Visual Studio Code本身是開源在Github的，有海量的第三方套件可以使用，而且VSCode本體更新的速度很勤快（大約一個月一次版本更新，有BUG幾乎不會拖太久）
- 請下載以下套件
  - Chinese (Traditional) Language Pack
  - C/C++ (ms-vscode.cpptools)



# 開始寫第一份Code

A screenshot of a code editor window titled "Untitled-1". On the left, there is a vertical toolbar with icons for file operations, search, and other functions. The main area shows a file named "test.cpp" with the following code:

```
1 #include<stdio.h>
2
3 int main()
4 {
5     printf("Hello World\n!!");
6     return 0;
7 }
8
```

A red arrow points from the text "較可惜的地方是這些元件還無法及時檢查出語法的錯誤 (例如忘記打分號)" to the code editor area.

The code editor has a dark theme. A tooltip labeled "擴充功能 C++ Intellisense" is displayed over the code. The word "int" is being typed, and a dropdown menu shows suggestions:

- int
- int16\_t
- int32\_t
- int64\_t
- int8\_t

The tooltip also contains the following text:

保存檔案之後  
自動完成的功能就會開始輔助你  
當然隨時Ctrl + S是好習慣

C++ keyword

較可惜的地方是這些元件還無法及時  
檢查出語法的錯誤 (例如忘記打分號)

這裡可以切換這份檔案的語言  
(快速鍵 : Ctrl + K , M)

切換至C/C++之後對應的附加元件就會啟動



# 編譯寫好的程式碼 1

The screenshot shows the Visual Studio Code interface. On the left, there's a sidebar with icons for file operations like Open, Save, Find, Cut/Copy/Paste, and Undo/Redo. The main editor area contains a C++ file named 'test.cpp' with the following code:

```
1 #include<stdio.h>
2
3 int main()
4 {
5     printf("Hello World!!\n");
6 }
```

On the right, there's a large text overlay in white that reads: "Code裡有整合系統的終端機" and "打程式、測試都可以在整個Code完成".

Below the editor, there's a terminal window labeled '2'. The terminal tab is highlighted with a red box. The terminal output shows the Windows command prompt and some directory navigation:

```
Microsoft Windows [版本 10.0.17763.1]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\TEST>cd desktop
C:\Users\TEST\Desktop>
```

At the bottom of the terminal window, there's a question: "該怎麼用終端機Compile?".

In the bottom-left corner of the interface, there's a gear icon with the number '1' above it, and a status bar at the very bottom with various icons and text.

# 編譯寫好的程式碼 2

- 因為Code本身沒有編譯按鈕，所以需要以終端機輔助
- 基本指令：  
  gcc [檔案名稱] -o [輸出檔案名稱]  
  g++ [檔案名稱] -o [輸出檔案名稱]

gcc表示用C編譯，g++表示用C++編譯

**gcc main.c**

**gcc main.c -o main.exe**

**g++ main.cpp**

**g++ main.cpp -o main.exe**

如果沒有附加 -o 參數  
預設產生的執行檔名字為 a.exe

# 執行程式 1

- 依照使用的終端機差異，執行程式的方法也有點不一樣。

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.17763.1]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權
C:\Users\TEST\Desktop>g++ test.cpp -o test.exe
C:\Users\TEST\Desktop>test
Hello World!!
C:\Users\TEST\Desktop>test.exe
Hello World!!
C:\Users\TEST\Desktop>
```

使用cmd

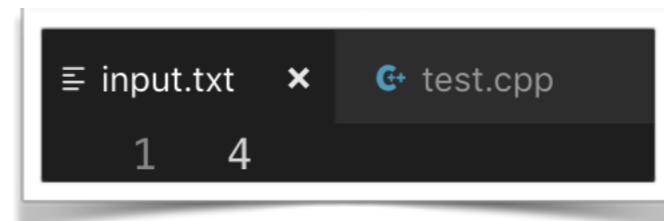
**在PowerShell要執行檔案  
前端需要加上「./」**

```
Windows PowerShell
Windows PowerShell
著作權 (C) Microsoft Corporation. 著作權所有，並保留一切權利。
PS C:\Users\TEST\Desktop> g++ test.cpp -o test.exe
PS C:\Users\TEST\Desktop> ./test.exe
Hello World!!
PS C:\Users\TEST\Desktop> ./test
Hello World!!
PS C:\Users\TEST\Desktop> test
test : 無法辨識 'test' 詞彙是否為 Cmdlet、函數、指令檔或可執行程式
確認路徑是否正確，然後再試一次。
位於 線路:1 字元:1
```

使用PowerShell

# 執行程式 2 檔案輸入/輸出

```
1 #include<stdio.h>
2 int main()
3 {
4     int input;
5     scanf("%d",&input);
6     printf("%d\n",input);
7 }
```



這是一段具有鍵盤輸入、螢幕輸出的程式碼  
假設這個編譯完成的檔案是test.exe

input.txt是這個程式的輸入

**test.exe < input.txt**

← test.exe接受來自input.txt之輸入  
結果輸出在畫面上

**test.exe < input.txt > output.txt**

← test.exe接受來自input.txt之輸入  
結果輸出到output.txt中

**test.exe > output.txt**

← test.exe接受來自鍵盤的輸入  
結果輸出到output.txt中

# 執行程式 3

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.17763.1]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\TEST\Desktop>test.exe <input.txt
Your Input: 100

C:\Users\TEST\Desktop>test.exe <input.txt> output.txt

C:\Users\TEST\Desktop>test.exe > output.txt
100

C:\Users\TEST\Desktop>
```

不管是要單獨開cmd或  
在VSCode整合的終端機操作都可以

```
問題 輸出 偵錯主控台 終端機 1: cmd

Microsoft Windows [版本 10.0.17763.316]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\TEST>cd Desktop

C:\Users\TEST\Desktop>test.exe <input.txt
Your Input: 100

C:\Users\TEST\Desktop>test.exe <input.txt> output.txt

C:\Users\TEST\Desktop>test.exe > output.txt
100

C:\Users\TEST\Desktop>
```

The screenshot shows the VS Code interface with three tabs open:

- test.cpp**: The code is as follows:

```
1 #include<stdio.h>
2
3 int main()
4 {
5     int input;
6     scanf("%d",&input);
7     printf("Your Input: %d\n",input);
8 }
```
- input.txt**: Contains the text "1 100".
- output.txt**: Contains the text "1 Your Input: 100".

**執行結果**

# 執行程式 4 關於EOF(End of file)

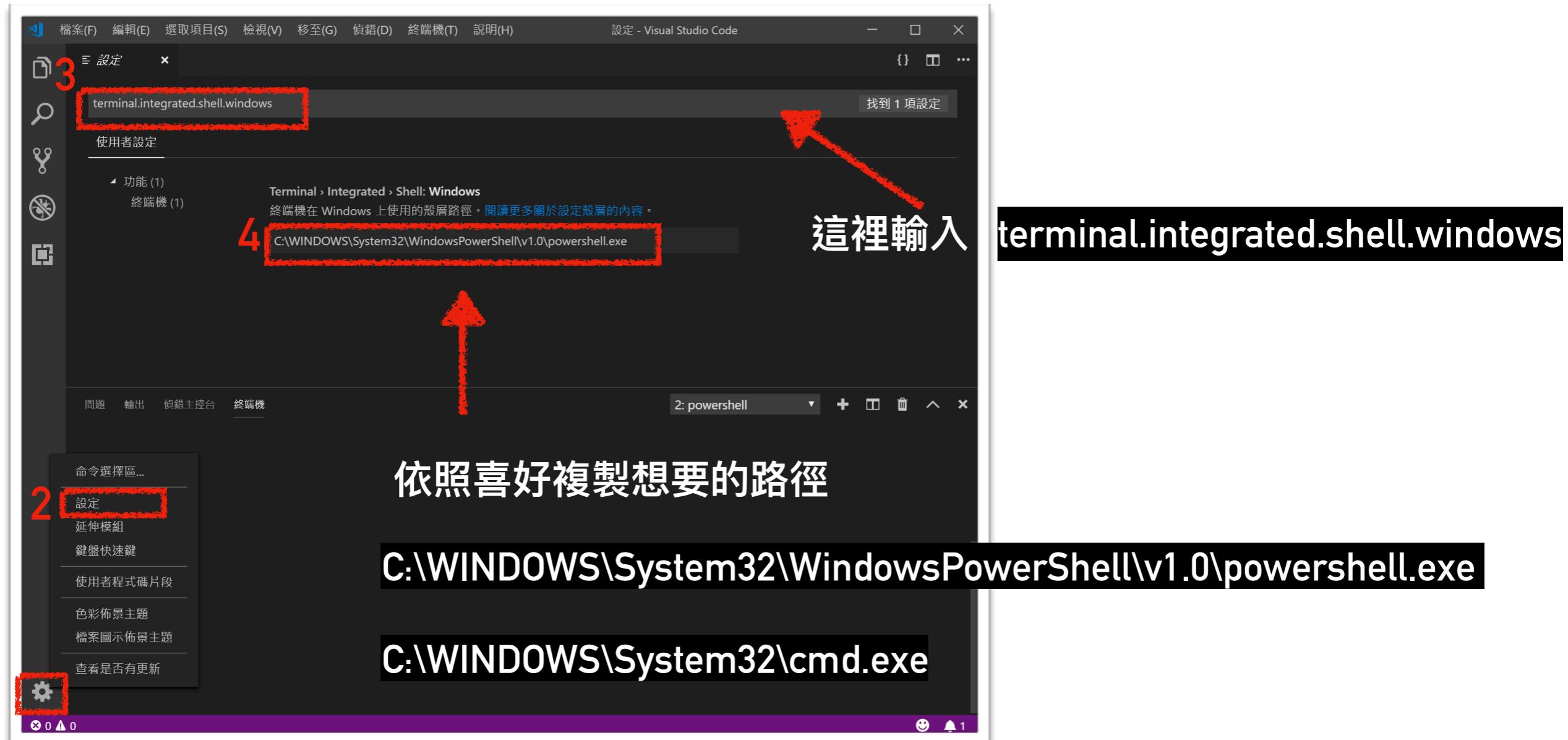
- 在OnlineJudge平台中，大多是用輸入檔案來做測試，所以自然會在讀完檔案之後收到EOF。  
Like this : out.exe <input1.txt> output1.txt  
之後再把output1.txt拿來比對是否一致，對了就Accept(AC)
- 如果你是用鍵盤輸入，要傳送EOF的話，輸入Ctrl + Z，再按Enter就可以了（畫面上會出現^Z）



```
Last login: Wed Mar  6 13:32:38 on ttys001
[Yu-Hsuans-iMac:2-4 yu-hsuanchen$ ./2-4.out
6-320-13394-7
6-320-13394-5
^Z
[1]+  Stopped                  ./2-4.out
Yu-Hsuans-iMac:2-4 yu-hsuanchen$ ]
```

# 切換Code裡綁定的終端機

- 因為Windows 10已經將Powershell做為預設，如果想改成舊版的命令提示字元(CMD)請這樣做

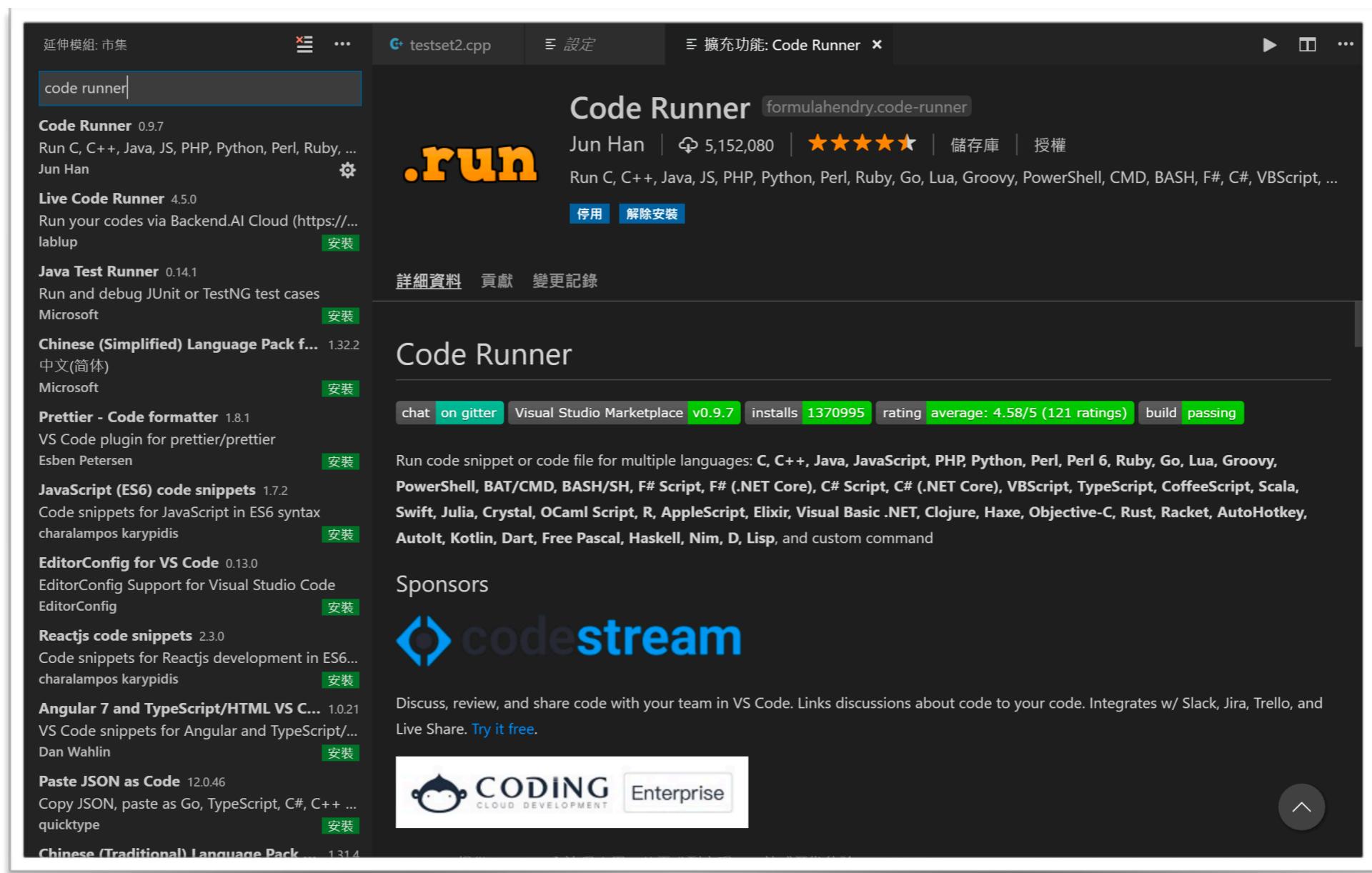


# 終端機常用指令

- cd [資料夾名稱] : 表示前往該資料夾
- cd .. : 表示回到上一層目錄
- cd \ : 表示回到根目錄
- dir : 顯示該目錄底下的資料夾與檔案內容
- 在輸入資料夾或檔案名稱時可以利用Tab鍵呼叫自動完成

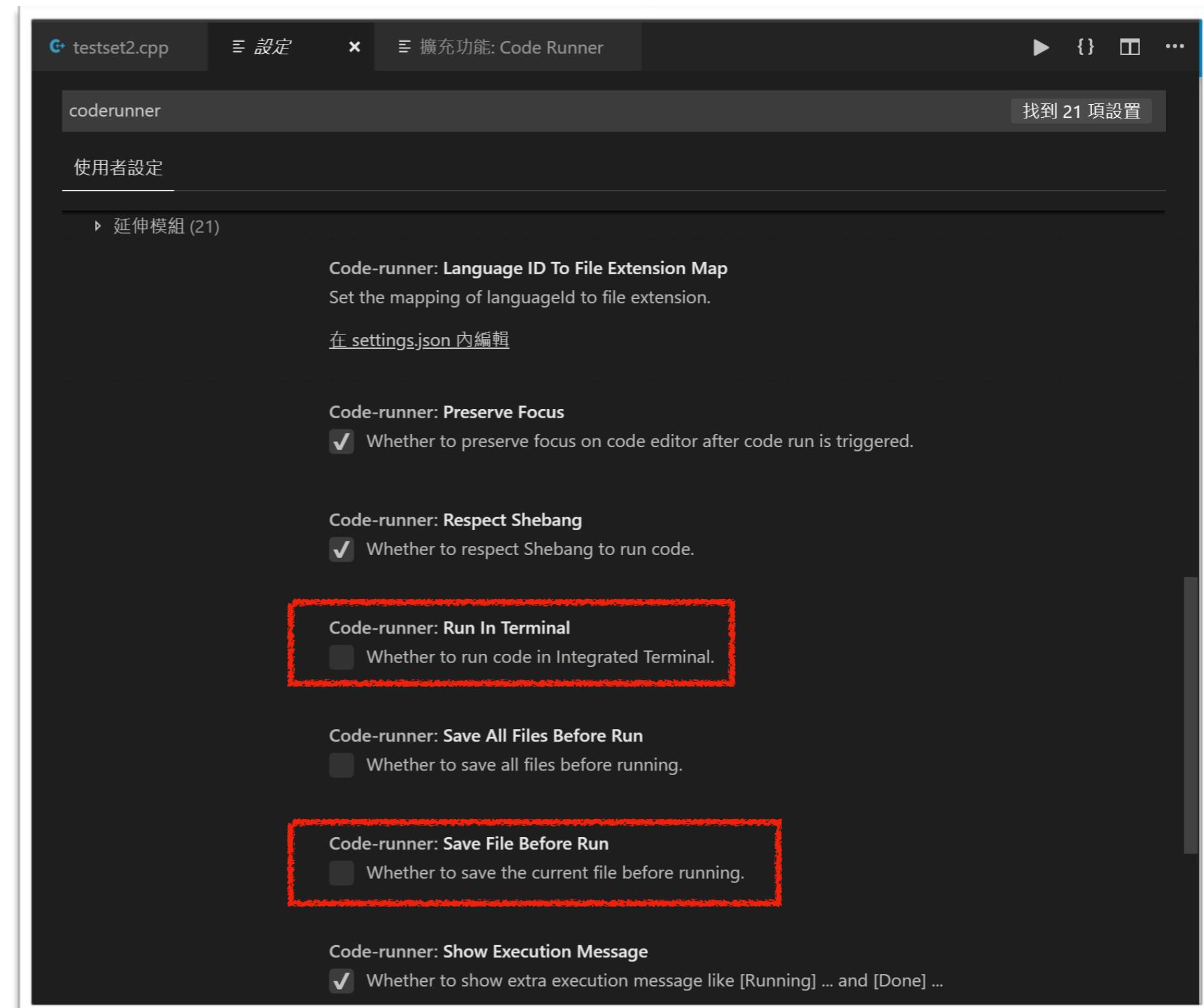
# 不想使用終端機的替代方案

## ● Code Runner (formulahendry.code-runner)



# Code Runner 相關設定

將框起來兩個選項打勾



# 使用Code Runner

The screenshot shows a dark-themed Code Runner interface. In the top-left, there's a code editor window titled 'test.cpp' containing the following C++ code:

```
C++ test.cpp  x
1 #include<stdio.h>
2
3 int main()
4 {
5     int input;
6     scanf("%d",&input);
7     printf("Your Input: %d\n",input);
8 }
```

In the top-right, there's a toolbar with a play button icon (highlighted with a red circle), a save icon, and a more options icon. Below the toolbar, the text '編譯並執行' (Compile and Run) is displayed, followed by the keyboard shortcut '快速鍵：Ctrl + Alt + N'.

At the bottom of the interface, there are tabs for '問題' (Issues), '輸出' (Output), '偵錯主控台' (Debugger Console), and '終端機' (Terminal). The '終端機' tab is active, showing the following terminal output:

```
問題    輸出    偵錯主控台    終端機
1: Code    ▾ + □ ✖ ⌄ ×
Microsoft Windows [版本 10.0.17763.1]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\TEST>cd "c:\Users\TEST\Desktop\" && g++ test.cpp -o test && "c:\Users\TEST\Desktop\"test
4
Your Input: 4
c:\Users\TEST\Desktop>
```

Below the terminal output, the text '停止正在執行的程式：對著終端機輸入Ctrl + C' is displayed.

# 其他編輯器的選擇

Notepad++



Sublime Text



Notepad



Brackets



Komodo Edit

# 鍵盤的浪漫



給我一把鍵盤，我給你整個世界。