c 開發環境設置 – windows

注意！無論是編譯器(gcc compiler)或是程式開發環境(Visual Studio Code)的安裝途徑，都請不要包含中文，容易導致後續程式編譯或執行時產生問題。

1. Use MinGW to download and install gcc compiler

<https://alexmav04.github.io/computer/windows-with-mingw-w64/>

這個連結寫得相當詳細，同學可以參考這位網友寫的步驟透過MinGW 下載及安裝gcc compiler，接著對環境變數做設置，之後開啟命令提示字元(command) 來確定gcc是否成功安裝。

[注意]

此位網友所撰寫的流程中有提到g++編譯器，gcc 為c語言的編譯器，而g++則為c++語言的編譯器，這邊同學先有個概念，之後開發c++語言要轉換編譯器。

[提醒]

下載及安裝完gcc之後，要記得按照這位網友寫的，設定環境變數(environmental variables)。特別注意這位網友有提到在環境變數設置時，要同時加入兩個路徑 (path)，一個為你gcc compiler的位置，一個為你MinGW資料夾的位置。

為什麼要在環境變數設定path，由於我們是使用命令提示字元做C檔案的編譯及執行，有別於傳統使用visual c++ 已經整合好一個使用者介面給使用者操作，我們是在命令提示字元直接跟機器打交道，是shell script，我們必須告訴shell編譯檔案時要用哪一個編譯器以及他的位置。



由於shell指令有太多可以說，我們目前是使用最廣泛的BASH，在此補充幾個連結給大家參考:

Shell references:

1. <https://www.cyut.edu.tw/~ywfan/1109linux/201109chapter11shell%20script.htm>
2. <https://www.ibm.com/docs/zh-tw/aix/7.1?topic=shell-environment-variables-in-c>

環境變數也設置好後，我們開啟命令提示字元，確定gcc 是否安裝成功，輸入 gcc -v

若是成功會出現以下畫面:

Text

Description automatically generated

上面紅框的內容是在告訴你，系統找到這個gcc在哪(target)，同時也告訴你gcc的版本是多少。到這邊我們已經有compiler了，接著我們還需要一個寫程式的環境(editor)

1. 安裝visual studio code

請由此網址下載visual studio code: <https://code.visualstudio.com/>

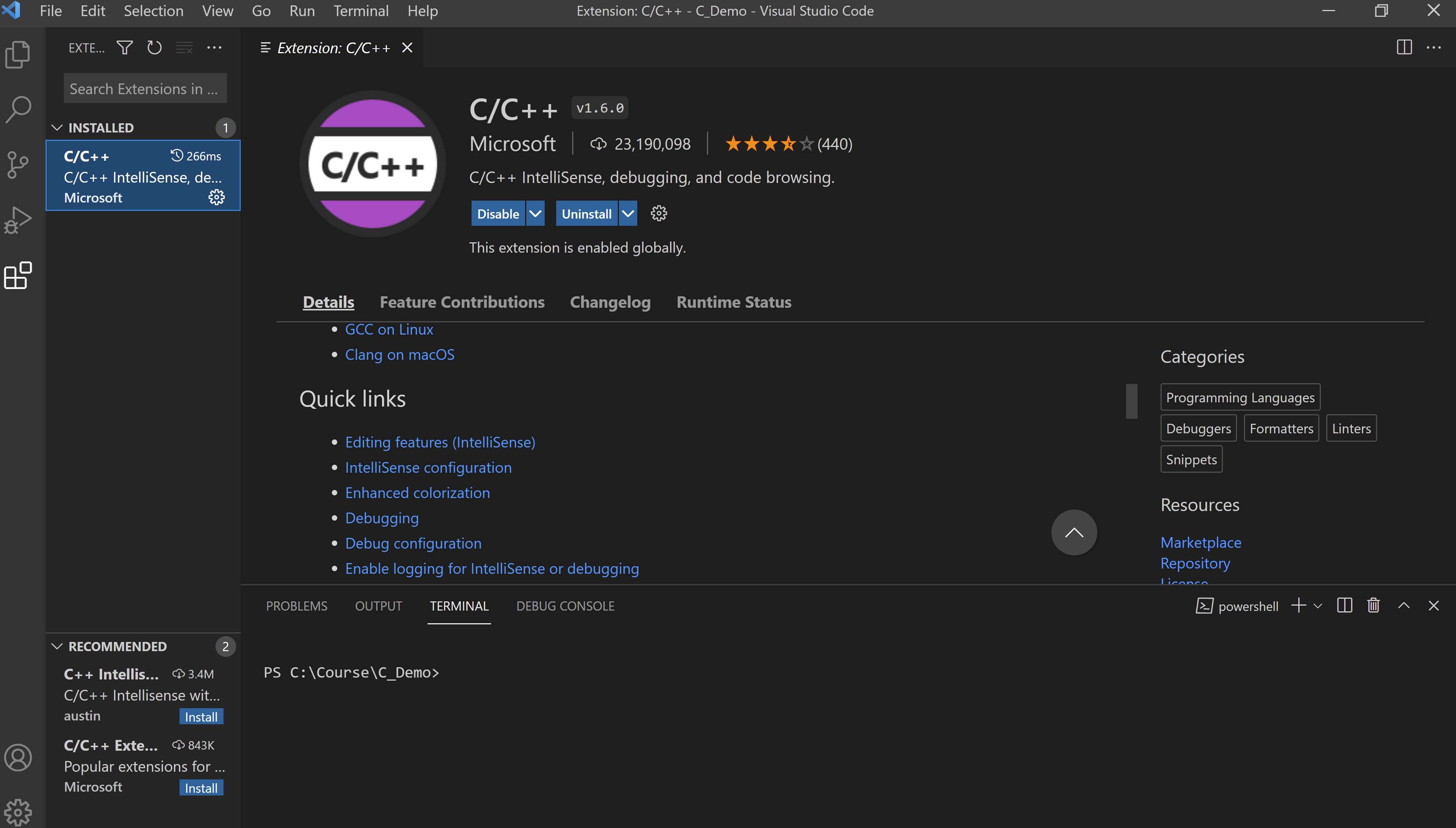
由於此軟體下載版本為win-64 bits，請確認你的windows系統作業版本是否為64位元。

下載安裝結束後，我們就可以打開visual studio code 做一些後續套件安裝及編譯器設定。

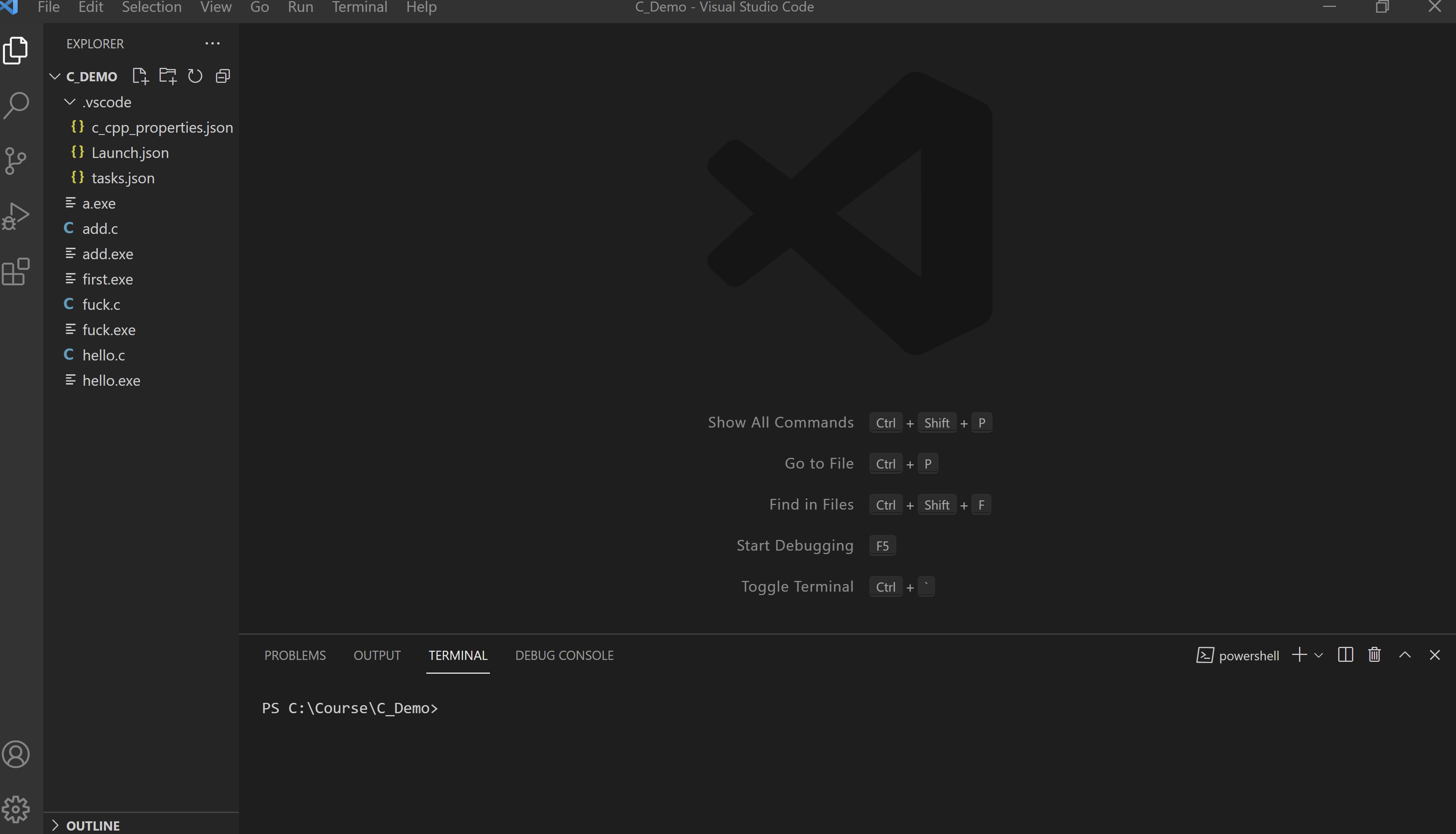
建議同學自行瀏覽官網的說明手冊(英文)，學會查看官方的mannual是學習程式設計很重要的環節!

https://code.visualstudio.com/docs/cpp/config-mingw

2-1 安裝最基本的套件: C/C++ IntelliSense, debugging, and code browsing



1. 嘗試使用visual studio code 及命令提示字元來執行一個hello world程式。



3-1 新增專案及檔案

打開Visual studio code請新開一個資料夾，之後新增一個file並命名為hello.c，並將以下程式碼貼入:

#include <stdio.h>

int main()

{

  printf("Hello, My Friends!\n");

  return 0;

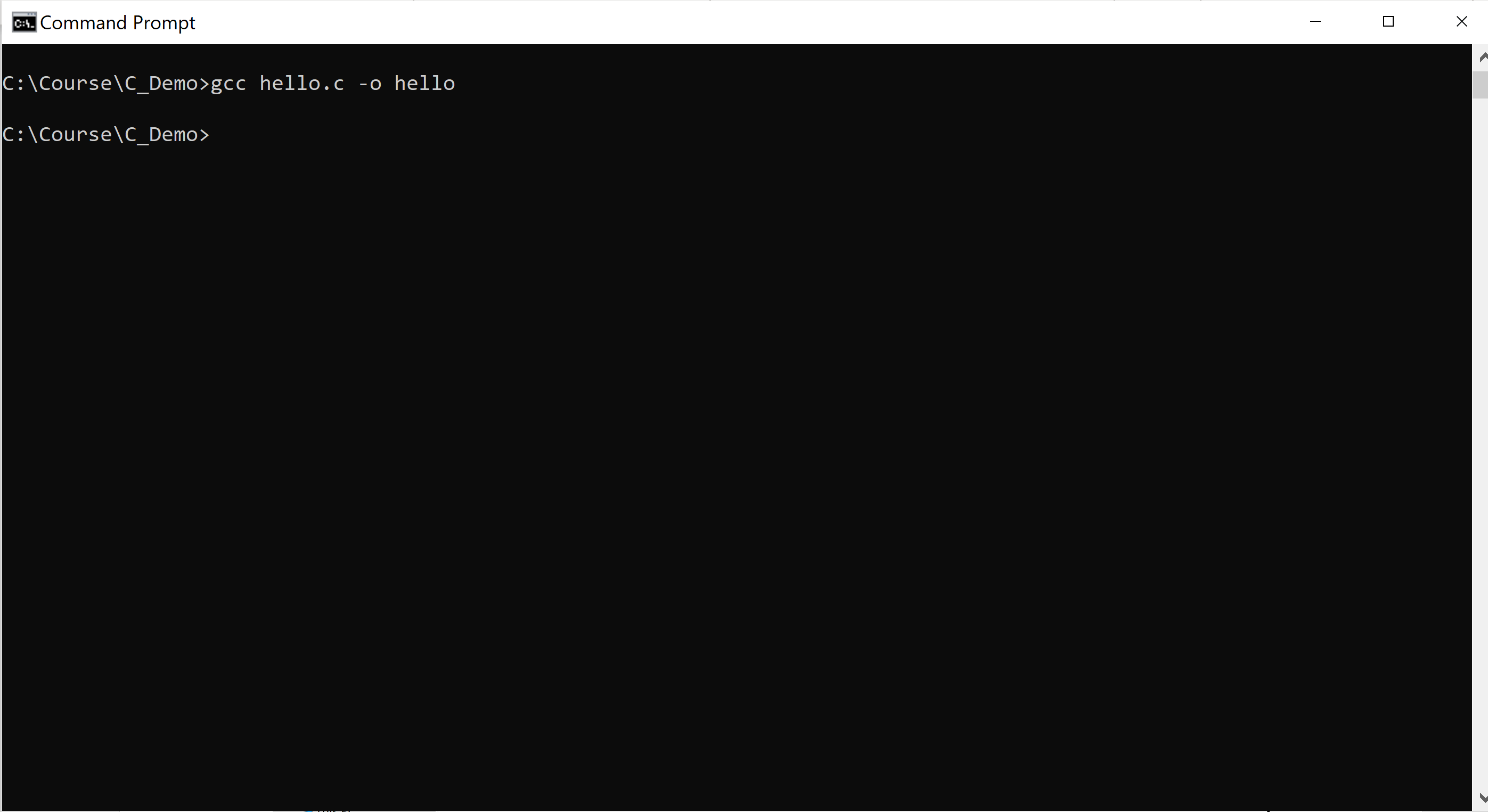
}

之後我們打開命令提示字元，並將目前的目錄更改到你剛剛在Visual studio code新開資料夾的位置，例如老師的資料夾位置是: C:\Course\C\_Demo，命令提示字元操作指令老師在這邊簡單提出幾個:

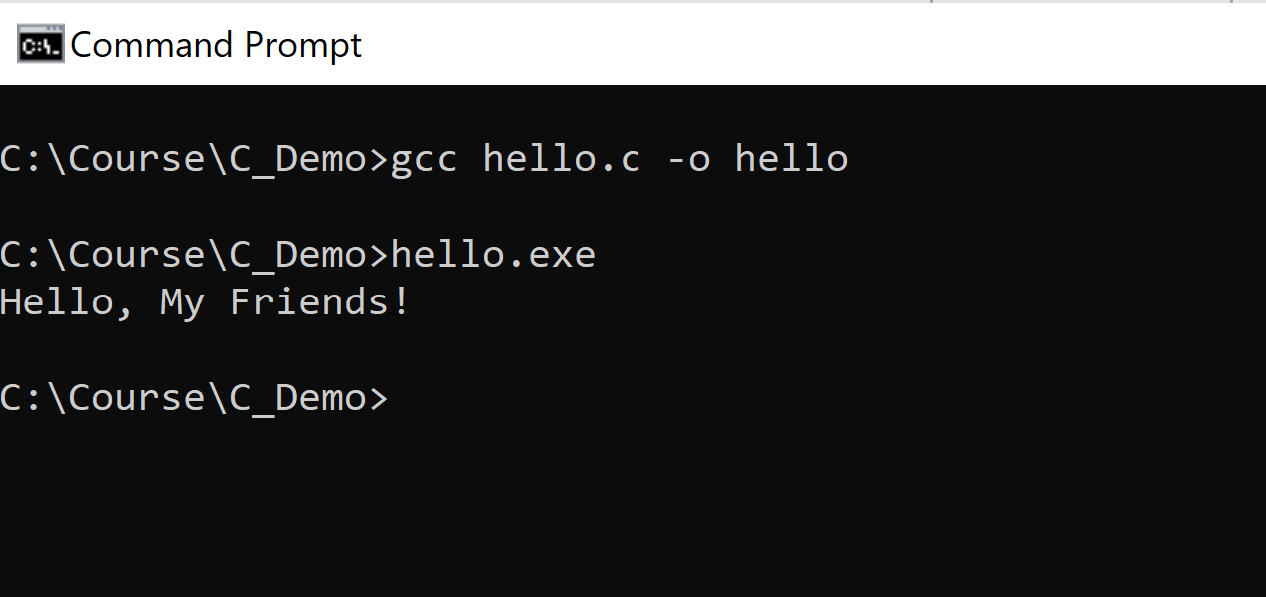
|  |  |
| --- | --- |
| 操作指令 | 用途 |
| cd | 秀出目前資料夾位置 |
| cd.. | 回到上一層 |
| Cd C:\xxx\xxx | 直接打開特定資料夾 |
| cls | 清除屏幕 |
| gcc xxx.c -o xxx | 將xxx C檔案編譯成xxx執行檔 |
| 指令 /? | 詢問此指令的用途 |

此網址提供命令提示字元的教學，請同學多加參考。

<https://lnpcd.blogspot.com/2012/09/00.html>



若是編譯成功，你的命令提示字元會空一行並開始等你輸入新的指令，這是同學可以回到你存放hello.c的資料夾尋找剛剛編譯成功的hello.exe執行檔，接著我們可以執行此檔案，在命令提示字元輸入hello.exe，就可以看到 Hello, My Friends!

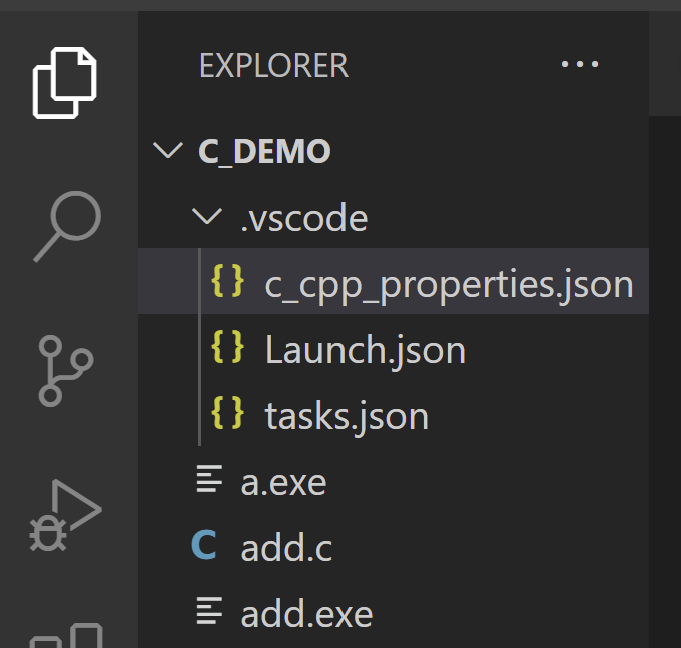


1. 如何在visual studio code 設定你剛剛千辛萬苦裝的gcc為指定compiler

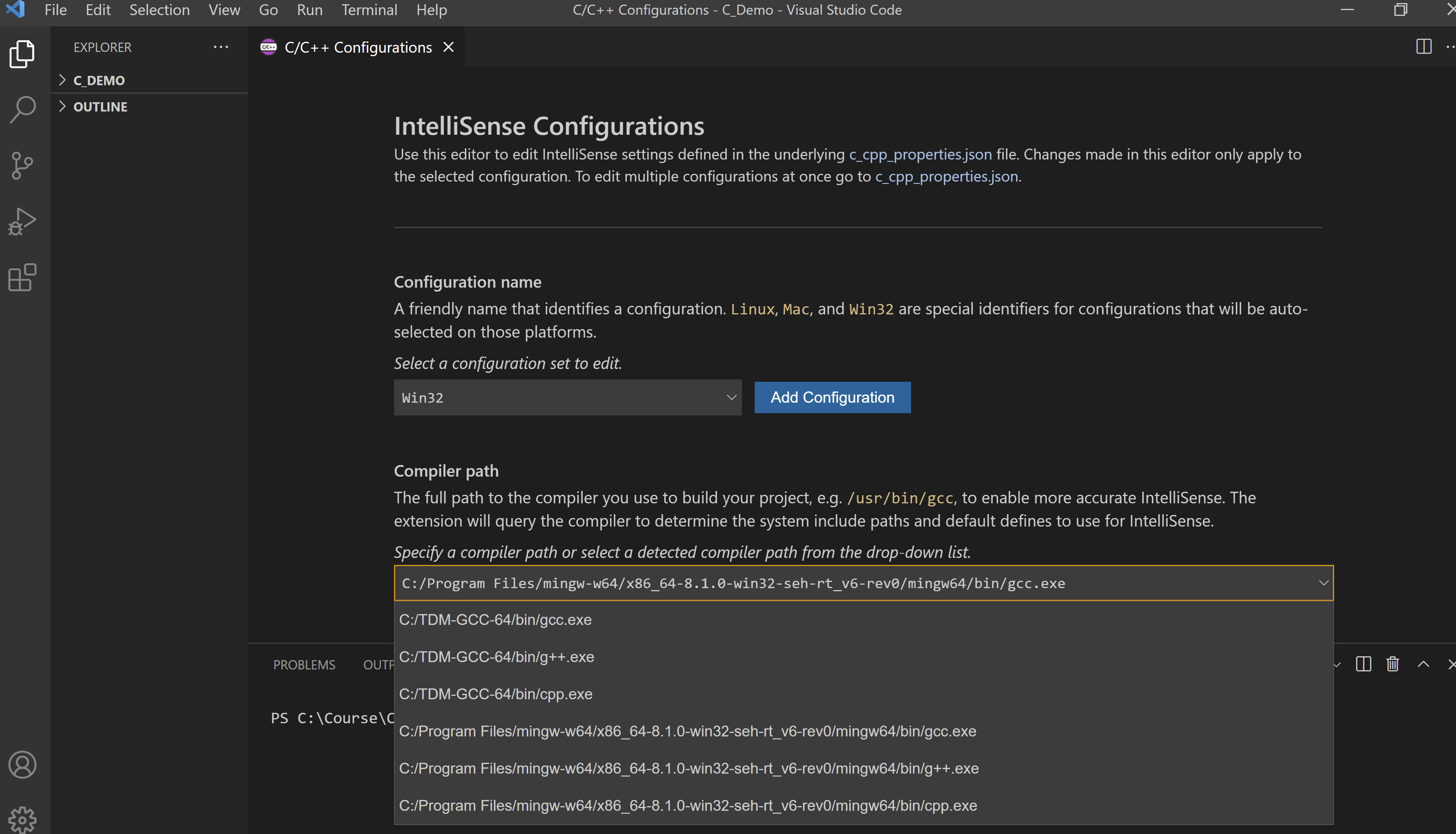
剛剛我們步驟三的編譯以及執行都是必須額外呼叫命令提示字元來完成，如果你覺得這樣麻煩，也可以在visual studio code裡面直接編譯及執行，但是需要完成以下設定首先要在你寫code的同一個資料夾下加入.vscode資料，並加入三個檔案，分別為c\_cpp\_properties.json, Lauch.json, Tasks.json首先c\_cpp\_properties.json，目的是為了設定 Compiler 路徑以及 IntelliSense。

<https://ithelp.ithome.com.tw/articles/10190235>

這位網友列出Visual studio code各項設定步驟以及後續甚麼外加套件建議安裝，總共有五篇文章，請同學參考，注意這位網友提供的程式碼為c++，故在.vscode 資料夾設置的地方是使用g++這個編譯器，同學在參考時要小心。



1. 首先我們先設定 c\_cpp\_properties.json:　按下 F1，輸入：C/C++: Edit Configurations (UI)



設定Configuration name 以及你剛剛裝的gcc compiler 的path

設定完成後，c\_cpp\_properties.json 會存在 .vscode 資料夾內，設定的內容會是(老師的電腦為範例)：

{

    "configurations": [

        {

            "name": "Win32",

            "includePath": [

                "${workspaceFolder}/\*\*"

            ],

            "defines": [

                "\_DEBUG",

                "UNICODE",

                "\_UNICODE"

            ],

            "compilerPath": "C:/Program Files/mingw-w64/x86\_64-8.1.0-win32-seh-rt\_v6-rev0/mingw64/bin/gcc.exe",

            "cStandard": "gnu17",

            "cppStandard": "gnu++14",

            "intelliSenseMode": "windows-gcc-x64"

        }

    ],

    "version": 4

}

1. 接著是tasks.json，目的是編譯（Build） 相關設定，這個task會驅動你的gcc編譯器來編譯你的檔案(source code)，設定步驟如下
   * 在Visual Studio Code主畫面左上的選單中選擇terminal(終端機) > Configure Default Build Task.

A screenshot of a computer

Description automatically generated with medium confidence

* + 接著在中間上方欄位，會出現 C/C++ gcc.exe build active file，點選後即會自動產生task.json在.vscode資料夾

A screenshot of a computer

Description automatically generated with medium confidence

* + 點開task.json，應當長的如下圖所示:

Text

Description automatically generated

* + 最後是Launch.json，目的是偵錯(debugger)相關設定。
  + 在Visual Studio Code主畫面左上的選單中選擇run(執行) > Add configuration

Graphical user interface

Description automatically generated

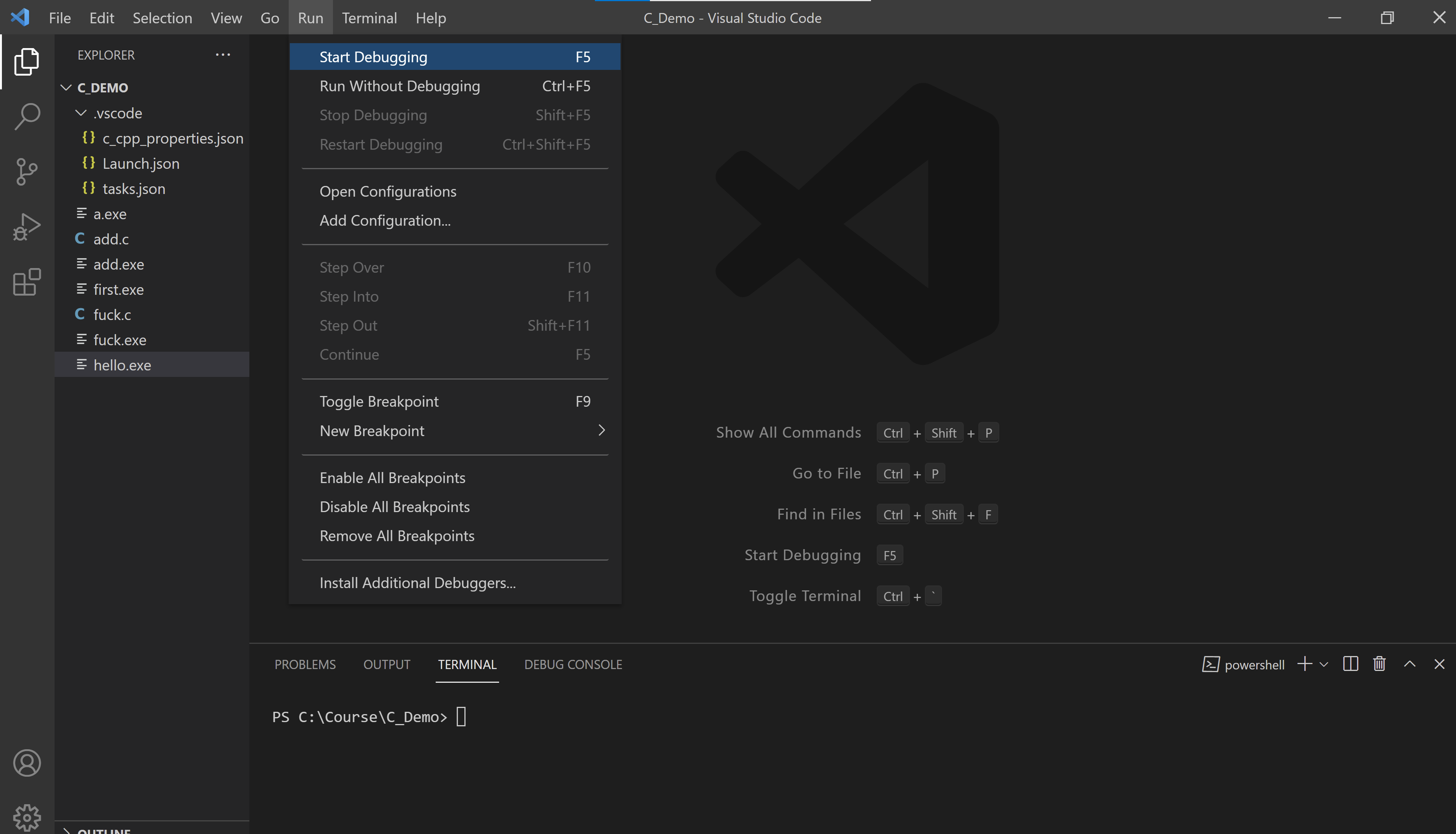
* + 接著在中間上方欄位，會出現 gcc.exe - Build and debug active file.，點選後即會自動產生launch.json在.vscode資料夾
  + 點開launch.json，應當長的如下圖所示:

Text

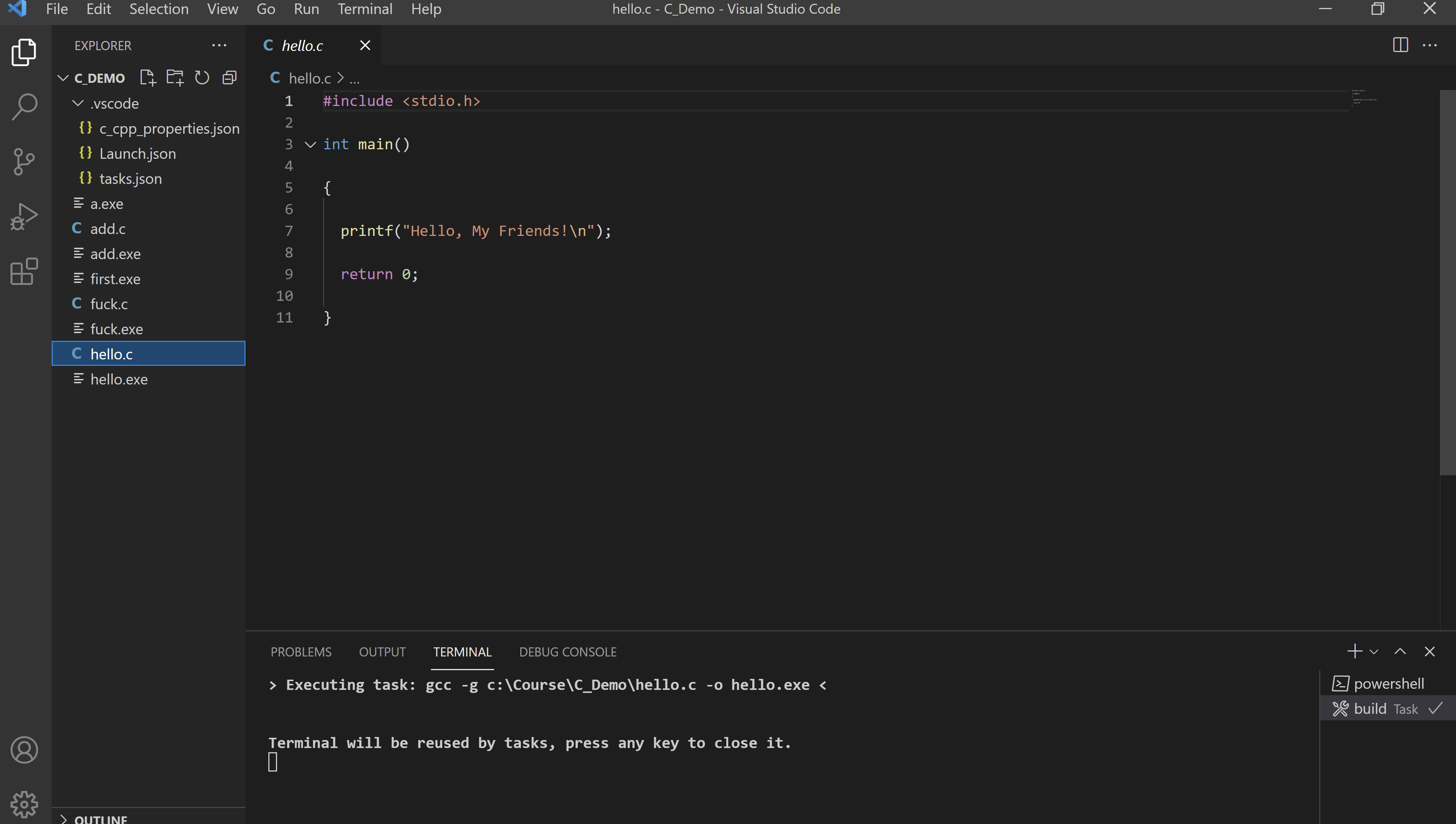
Description automatically generated

1. 確認方才的設定都完成且可以正常運作

使用之前建立的hello.c進行debugging



可以在下方的debug console 以及terminal 觀看程式編譯情況



編譯成功後，我們可以在下方terminal輸入指令./hello.exe，就可以看到Hello, My Friends了

