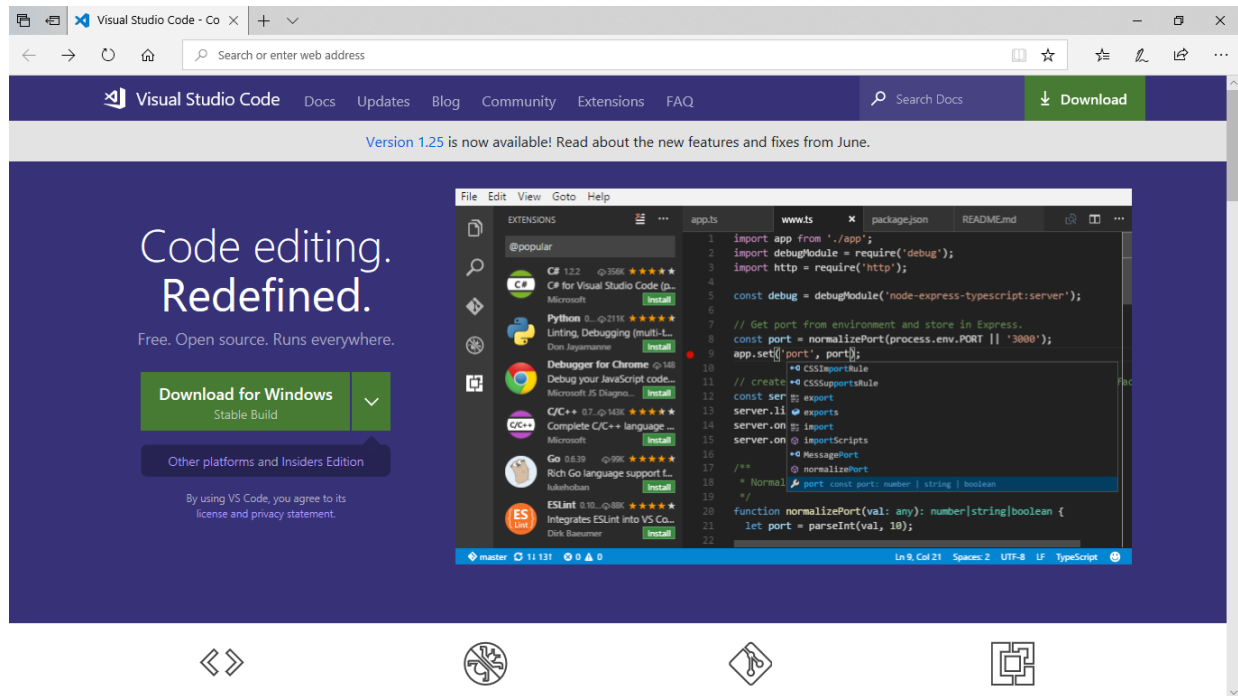


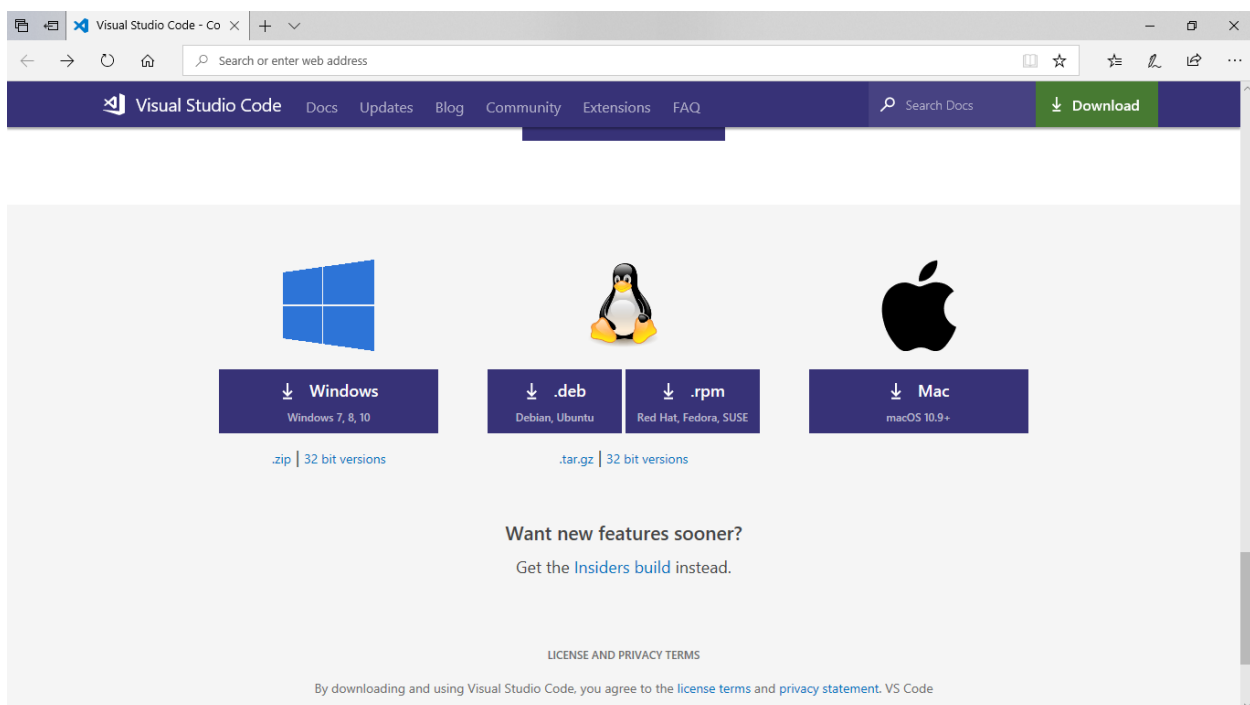
I. Download và cài đặt Visual Studio Code

Bước 1: Download Visual Studio Code

+ Truy cập <https://code.visualstudio.com/>



+ Lựa chọn phiên bản phù hợp với hệ điều hành và tải về

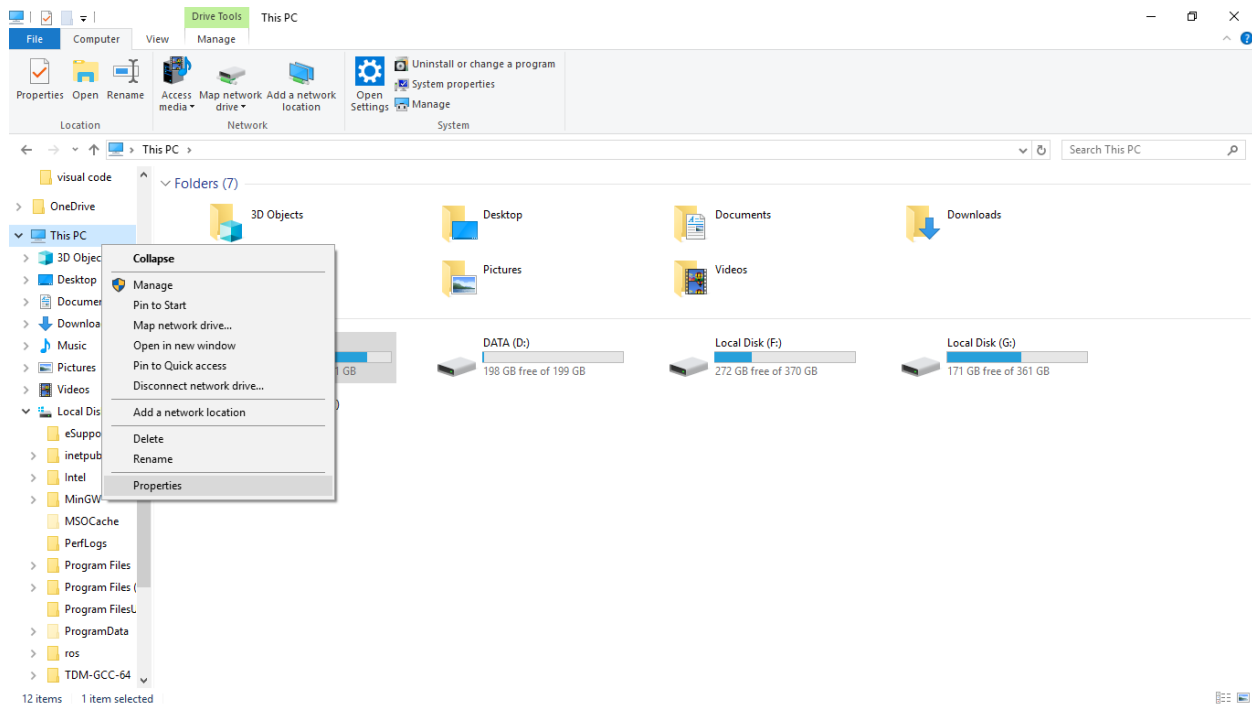


Bước 2. Download và cài đặt MinGW

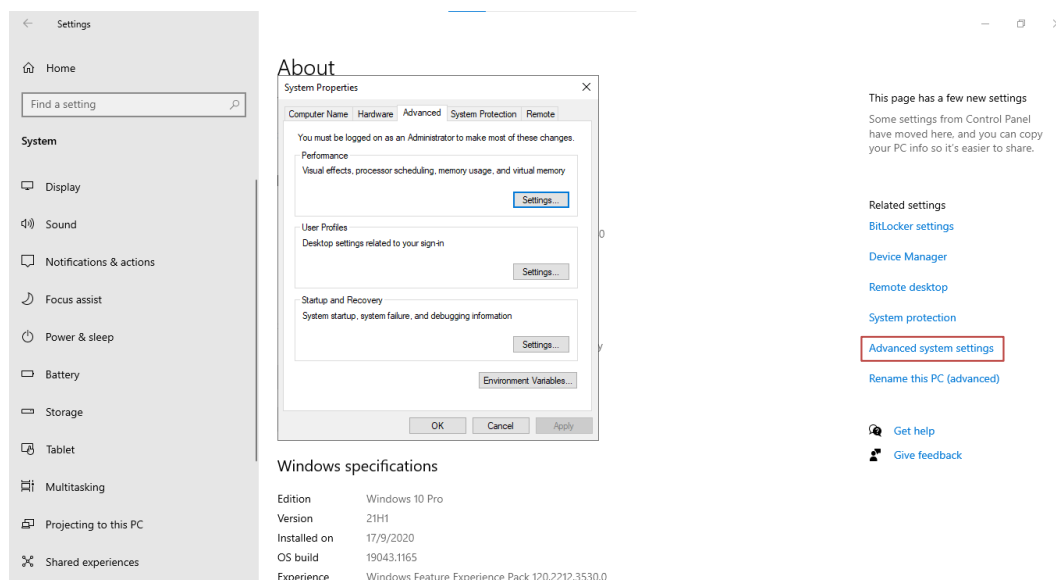
+ Truy cập [đây](#) để download MinGW và xem hướng dẫn [MSYS2](#) để tiến hành cài đặt.

+ Thêm đường dẫn đến thư mục bin của Mingw-64 vào Windows PATH environment variable, theo các bước bên dưới:

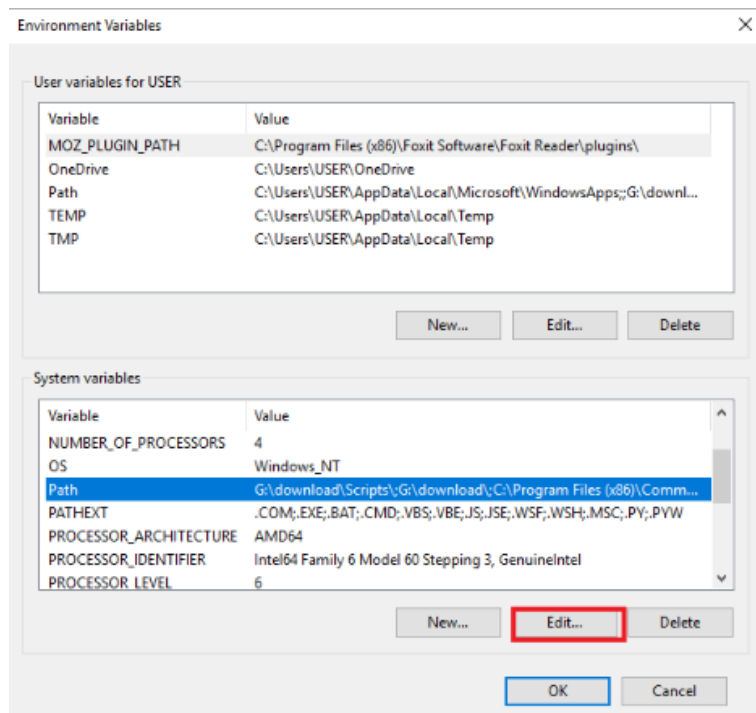
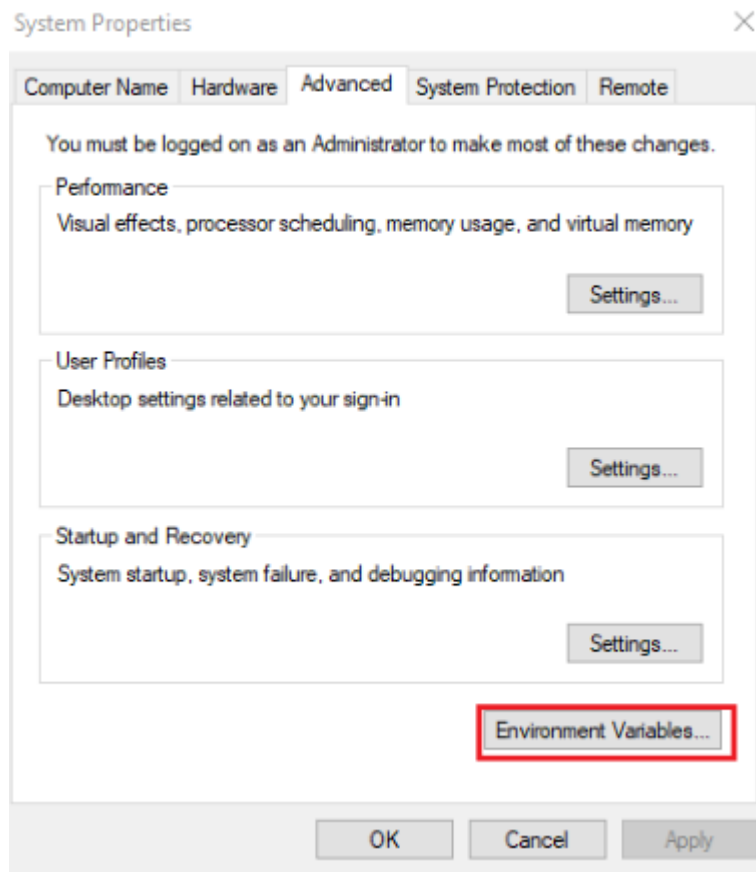
- Click chuột phải vào ThisPC chọn Properties



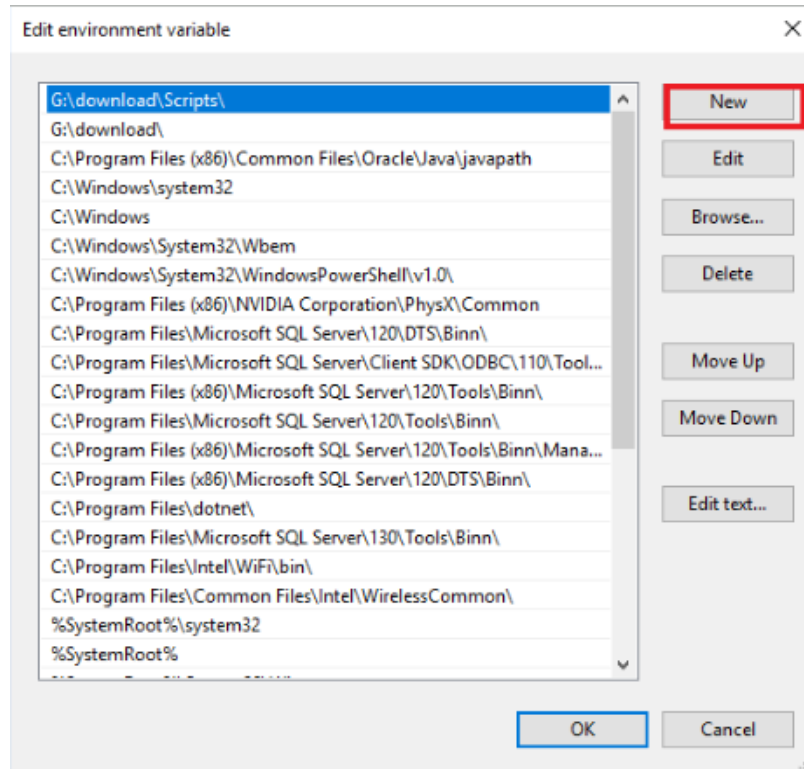
- Chọn Advanced system settings



- Chọn Environment Variables...

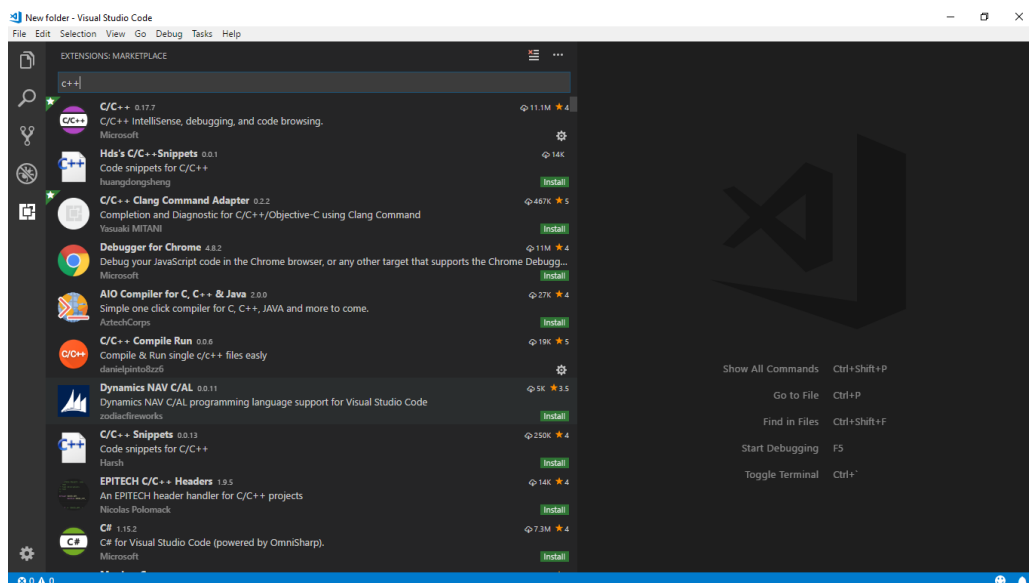


- Chọn New rồi Paste đường dẫn `C:\msys64\mingw64\bin` và OK

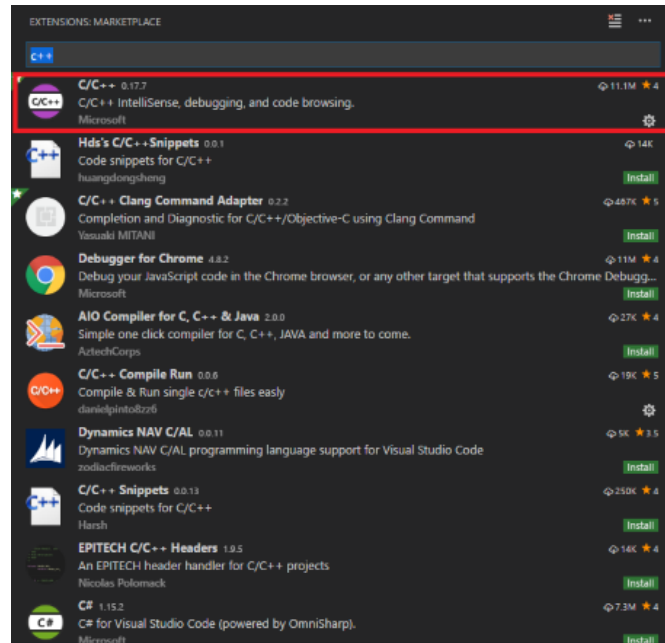


II. Run Visual Studio Code

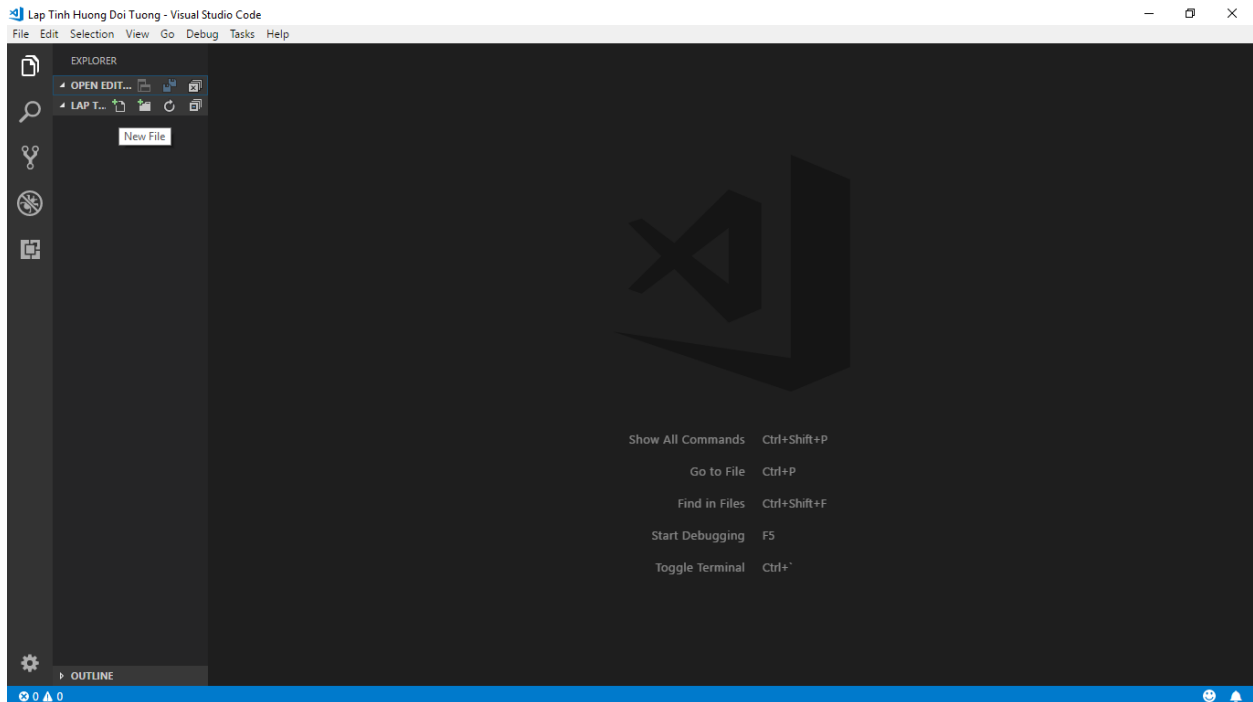
- + Khởi động Visual Studio Code -> File -> OpenFolder (đường dẫn sẽ đặt project)
- + Nhấn tổ hợp Ctrl + Shift + X , Search C++

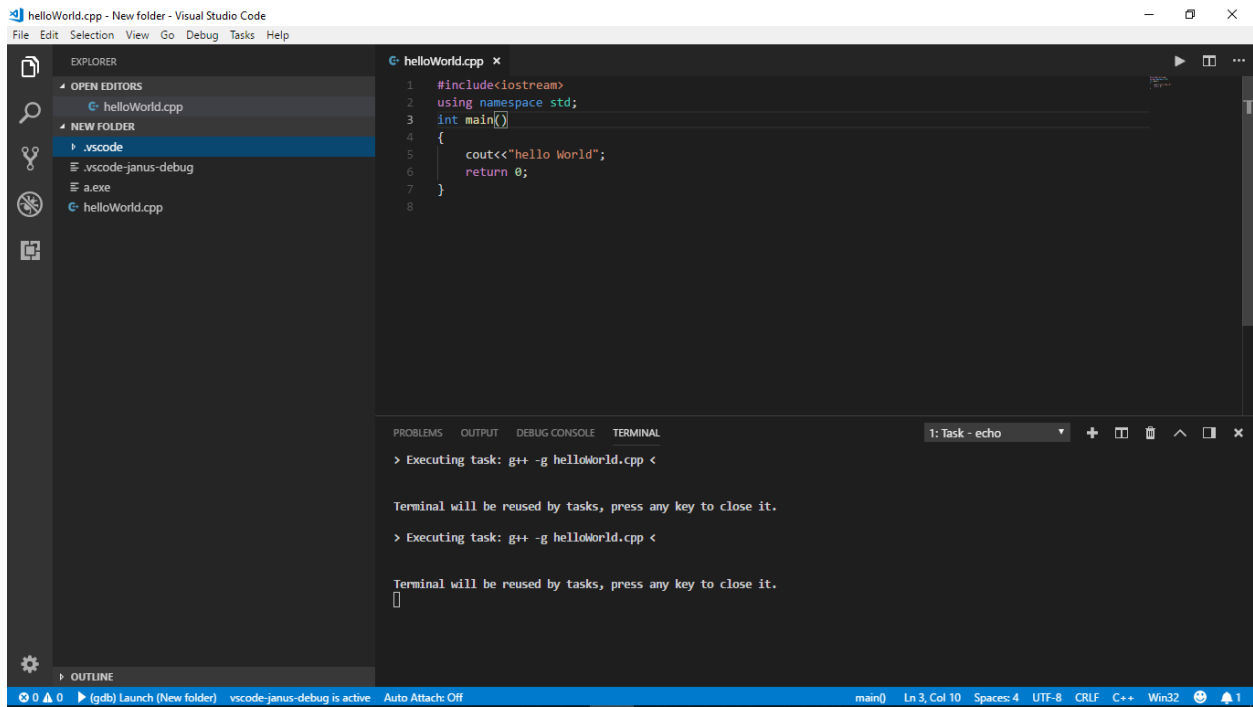


+ Tiến hành Install C/C++




+ Tại thư mục ta vừa chọn, add thêm file helloworld.cpp

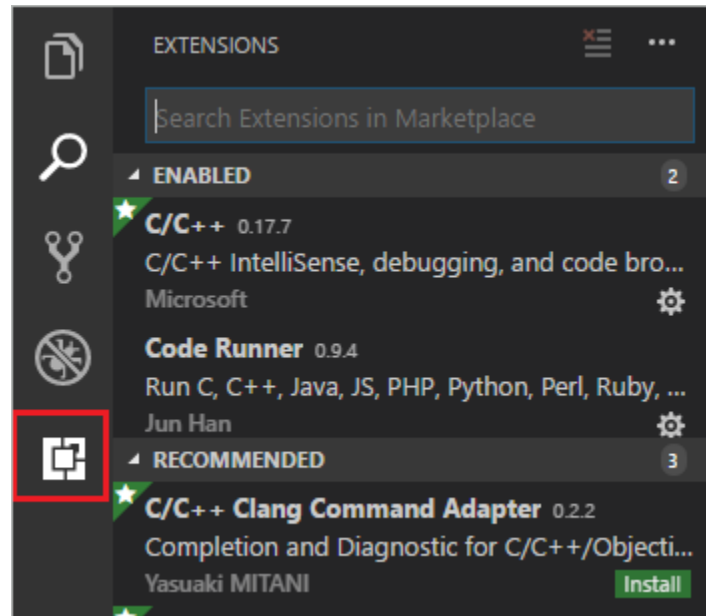




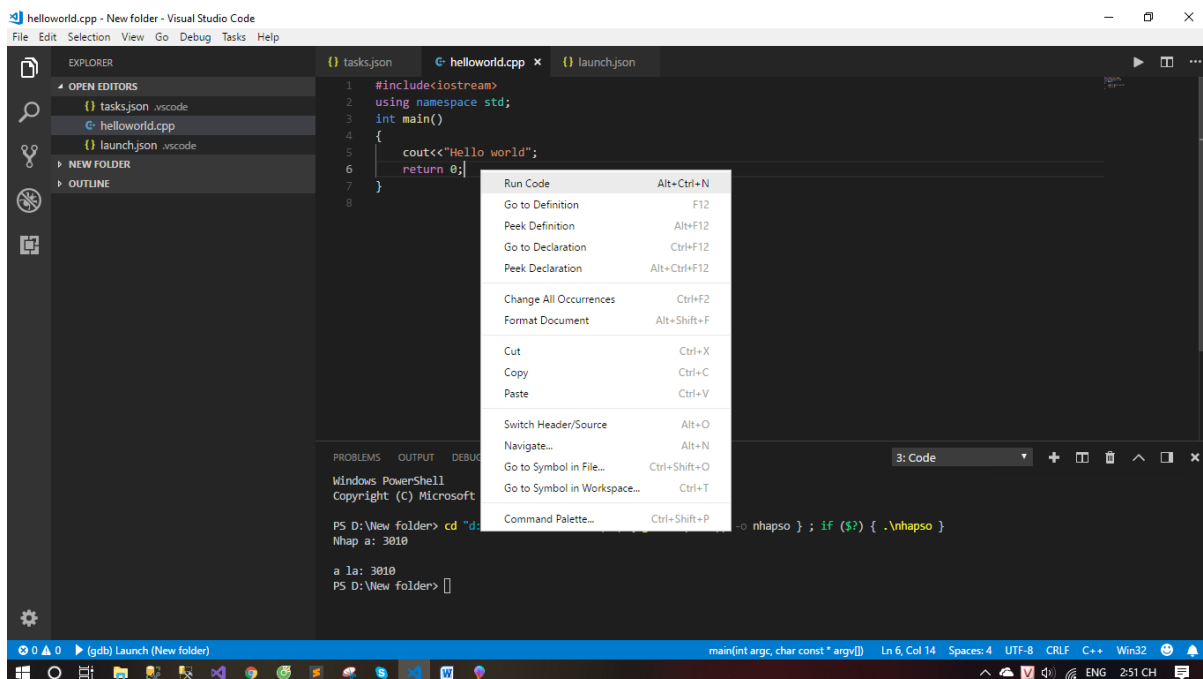
III. Build code

Cách 1: Install Code Runner

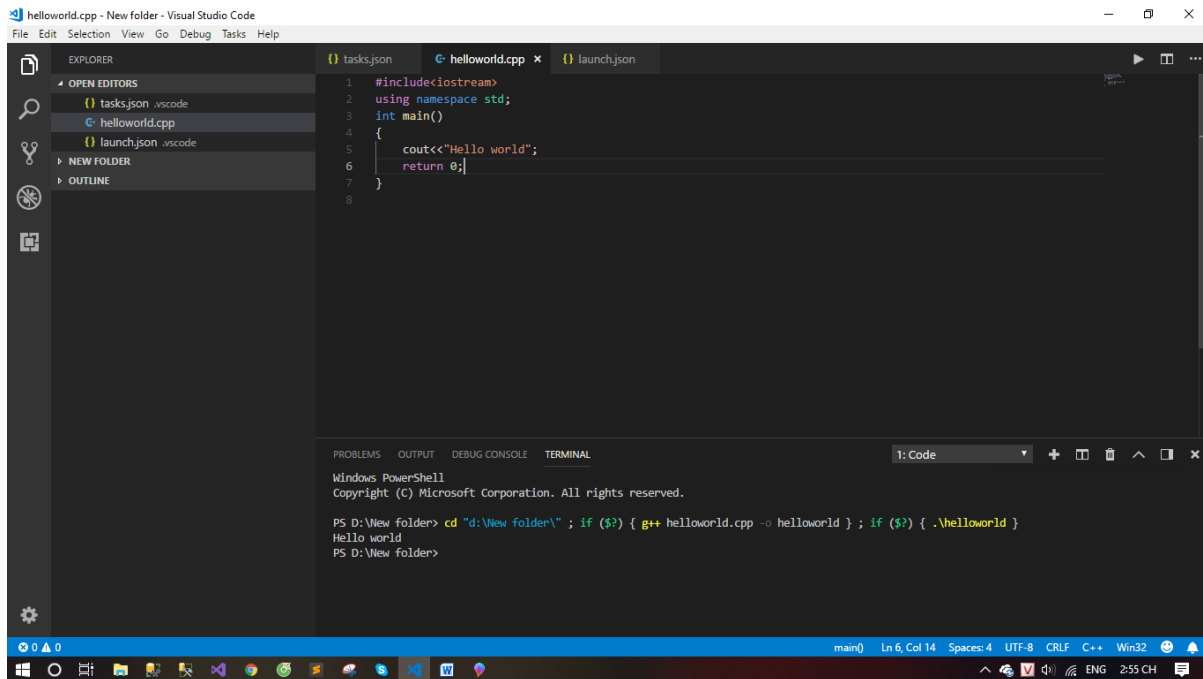
+ Nhấn biểu tượng  trên thanh bên trái màn hình hoặc nhấn ctrl+shift+X -> Search: Code Runner -> Install, sau khi Install xong nhấn Reload



+ Muốn build chương trình: chuột phải chọn Run code (Alt + Ctrl + N)



+ Kết quả thực thi như sau:



The screenshot shows the Visual Studio Code interface with a C++ file named `helloworld.cpp` open. The code is as follows:

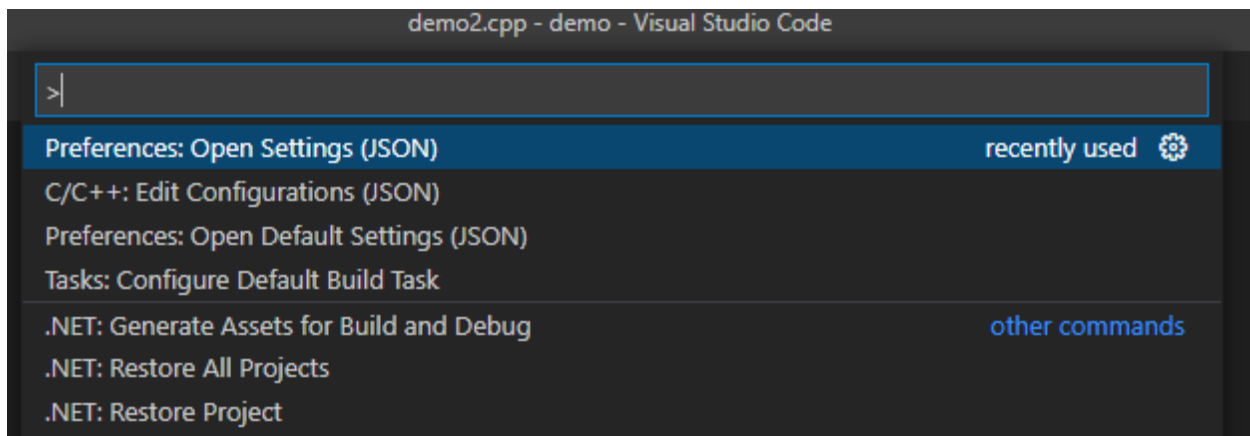
```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     cout<<"Hello world";
6     return 0;
7 }
8
```

The Explorer sidebar on the left shows the file structure with `tasks.json`, `helloworld.cpp`, and `launch.json`. The Terminal at the bottom shows the command `g++ helloworld.cpp -o helloworld` and the output `Hello world`.

Note: Trường hợp chạy bằng Run code thì khi có dữ liệu đầu vào và chạy nhiều file .cpp cùng 1 lúc:

Cấu hình lại file setting:

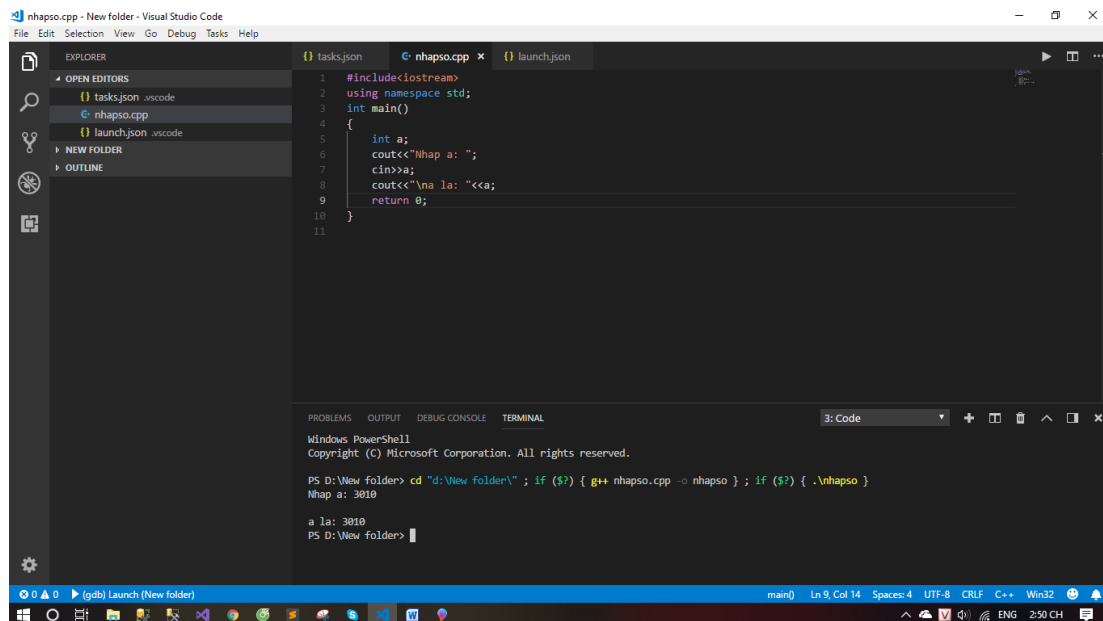
+ Nhấn **Ctrl + Shift + P**, chọn **Preferences: Open Settings (JSON)**



+ Thêm **"code-runner.runInTerminal": true** vào file settings.json vừa mở để có thể chạy file có chứa dữ liệu đầu vào

```
{ settings.json X
C: > Users > Admin > AppData > Roaming > Code > User > {} settings.json > ...
1  {
2      "code-runner.runInTerminal": true,
3      "terminal.integrated.defaultProfile.windows": "Git Bash"
4  }
```

+ Ví dụ mẫu:



+ Thêm **"code-runner.executorMap": {**

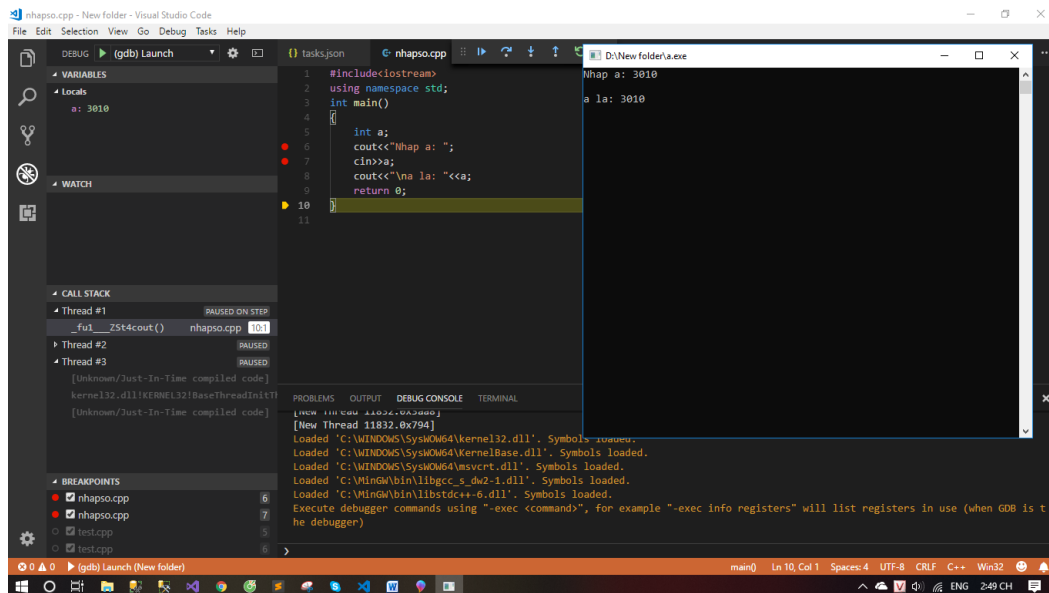
"cpp": "cd \$dir && g++ -std=c++14 *.cpp -o \$fileNameWithoutExt && \$dir\$fileNameWithoutExt",
}

vào file settings.json để có thể chạy nhiều file .cpp cùng 1 lúc:

```
{ settings.json X
C: > Users > Admin > AppData > Roaming > Code > User > {} settings.json > ...
1  {
2      "code-runner.runInTerminal": true,
3      "code-runner.executorMap": {
4          "cpp": "cd $dir && g++ -std=c++14 *.cpp -o $fileNameWithoutExt && $dir$fileNameWithoutExt",
5      },
6      "terminal.integrated.defaultProfile.windows": "Git Bash"
7  }
```

Chạy bằng debug: Xem phần **debug** phía sau để biết rõ hơn

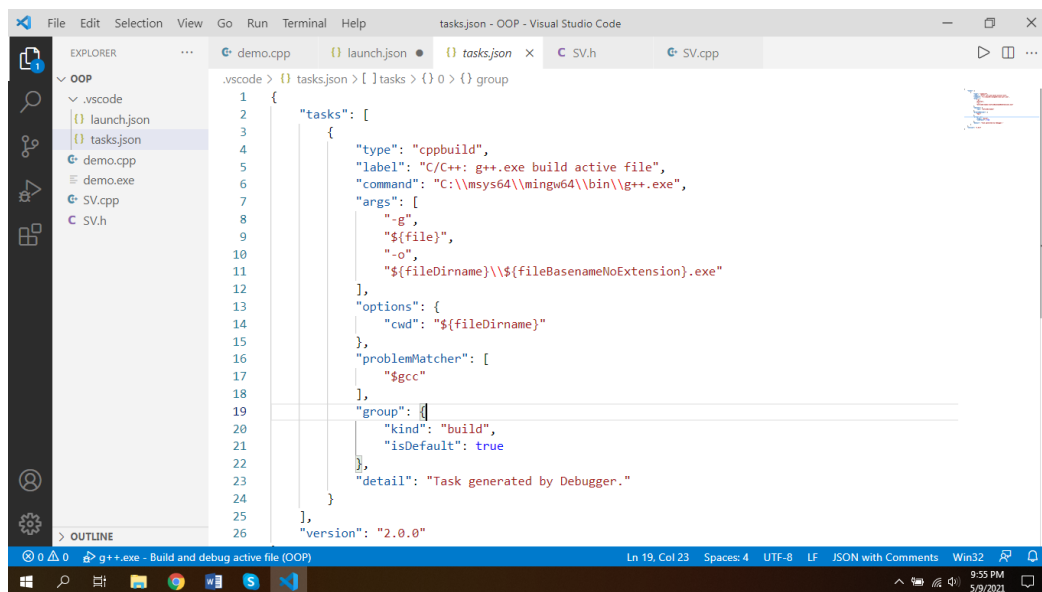
+ Ví dụ mẫu:



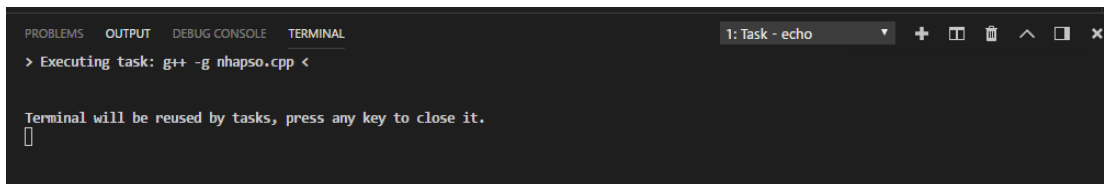
Cách 2: Chạy bằng Terminal

+ Tạo file **tasks.json**:

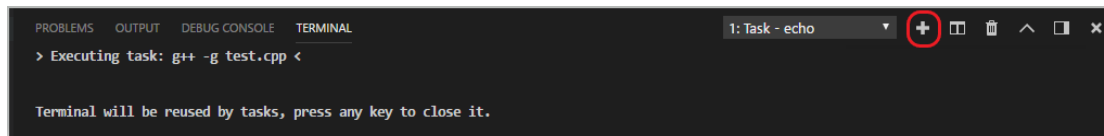
- B1. Mở Command Palette (**Ctrl+Shift+P**);
- B2. Chọn lệnh Tasks: Configure Tasks ..., nhấn Create tasks.json file from templates, sau đó chọn Others;



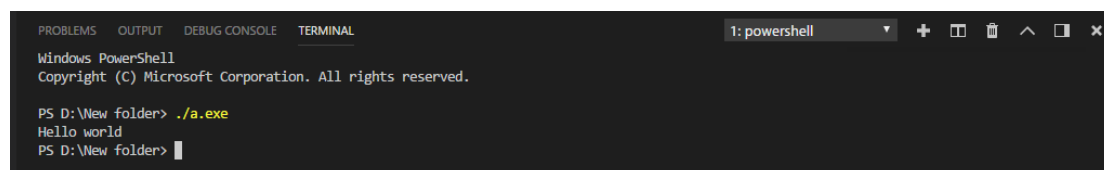
+ Build code: Nhấn **Tasks: Run Build Task** (**Ctrl+Shift+B**) để build



+ Thực thi và xem kết quả: chọn nút + bên phải phía dưới màn hình:



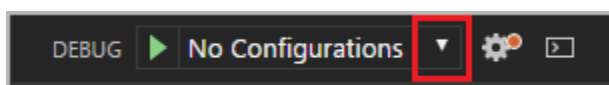
+ Gõ **.(tenfilebuild).exe**. Ví dụ: trong đó: a.exe là file được tạo ra trước đó khi ta nhấn **Tasks: Run Build Task** (**Ctrl+Shift+B**)



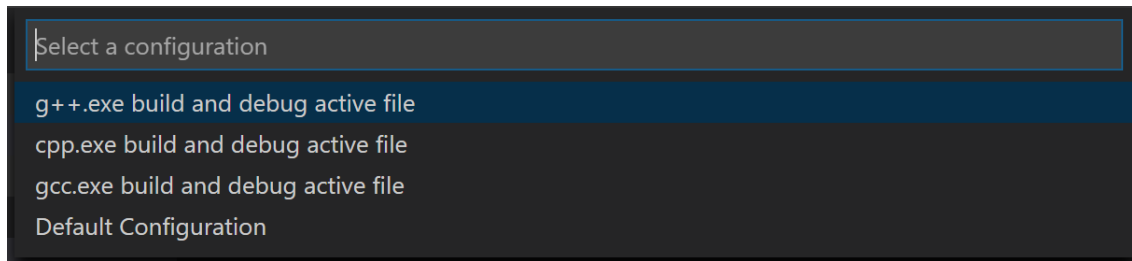
IV. Debug code

+ B1. Kích biểu tượng  trên thanh bên trái màn hình hoặc nhấn ctrl+shift+D

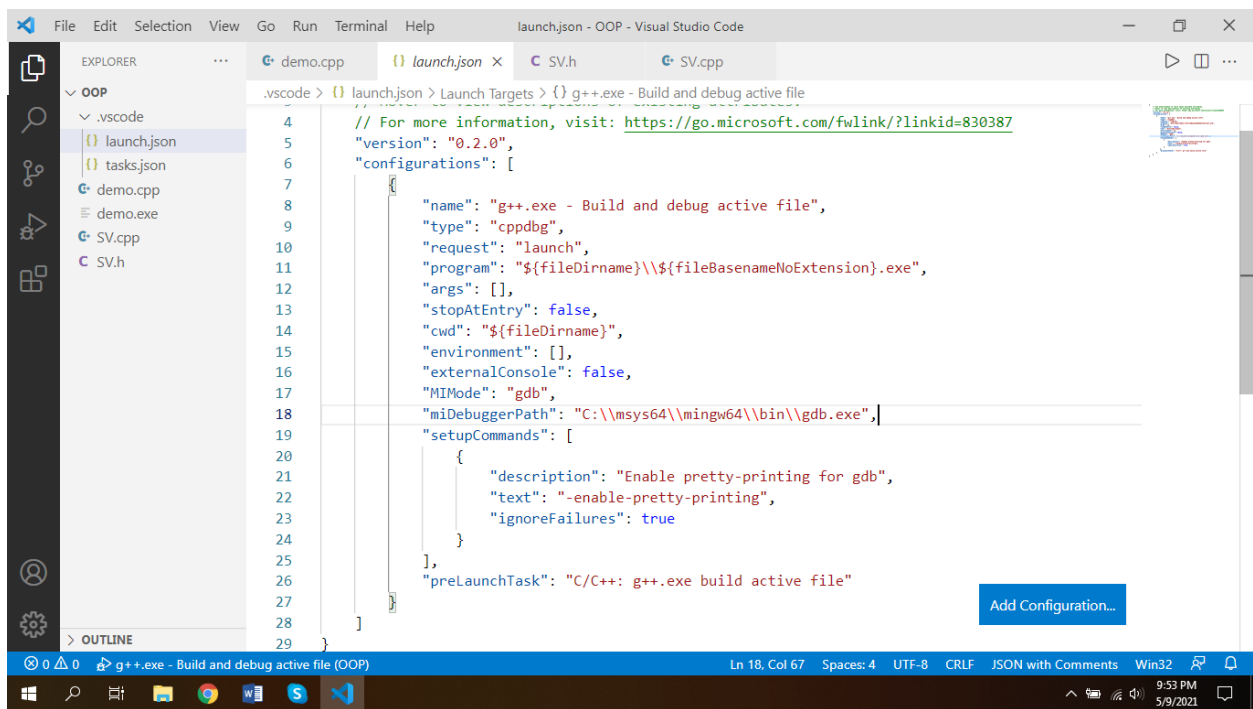
+ B2. Để debug được, tạo file **launch.json**:



- Nhấn vào mũi tên chọn Add Configuration > chọn **C++ (GDB/LLDB)**, sau đó chọn **g++.exe build and debug active file**.

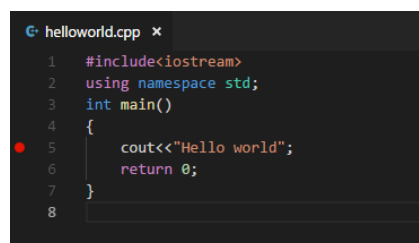


- Tại file launch.json, thêm preLaunchTask (nếu dùng Code Runner thì không cần), ví dụ như sau:

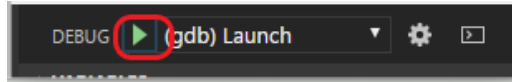


- Trong đó “C/C++: g++.exe build active file” trong preLaunchTask là phân label được lấy trong file Task.json

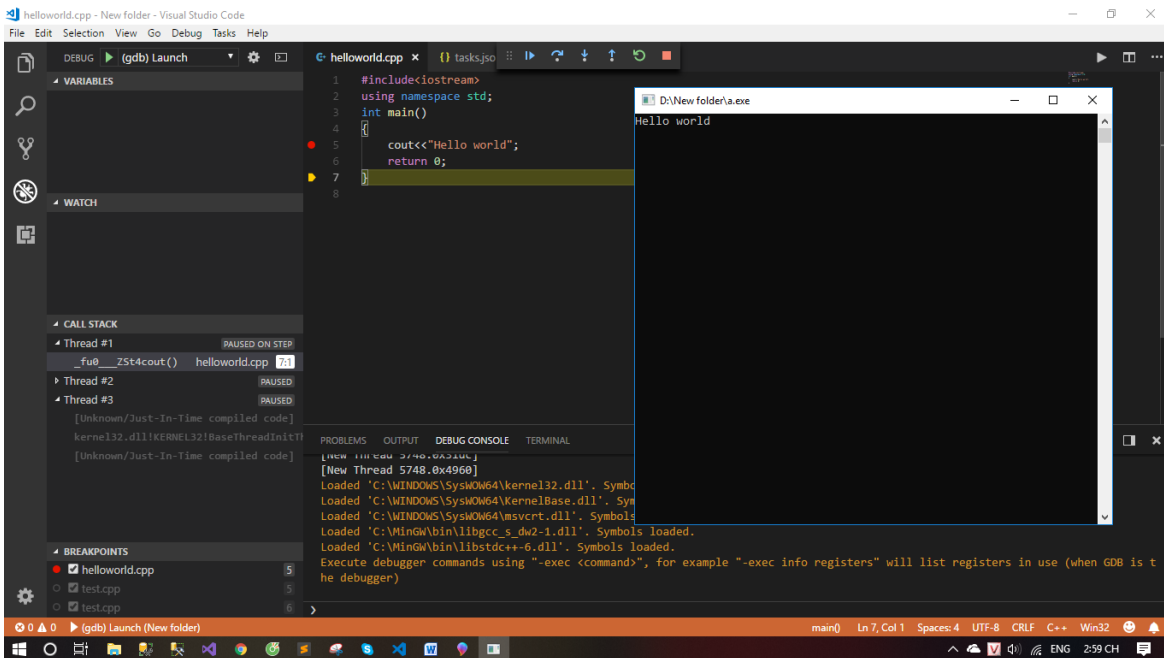
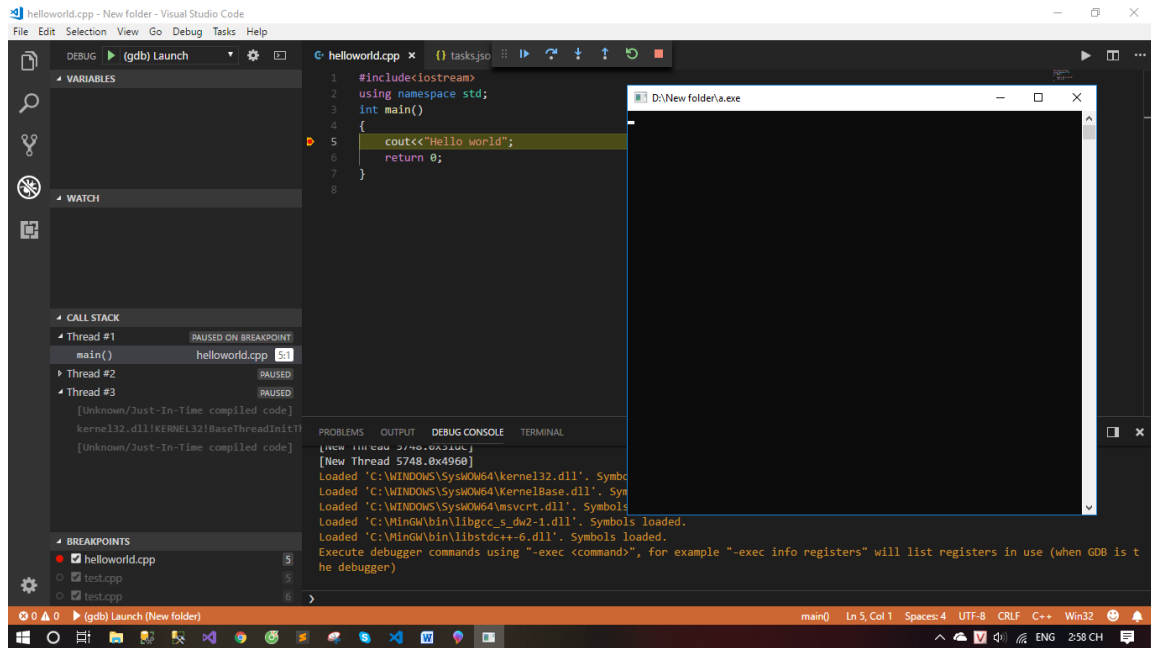
+ B3: Bật chế độ debug:



- Nhấn

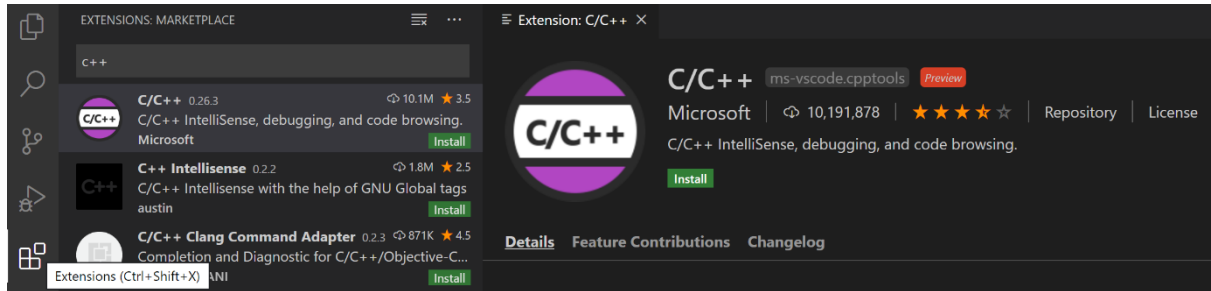


để debug chương trình



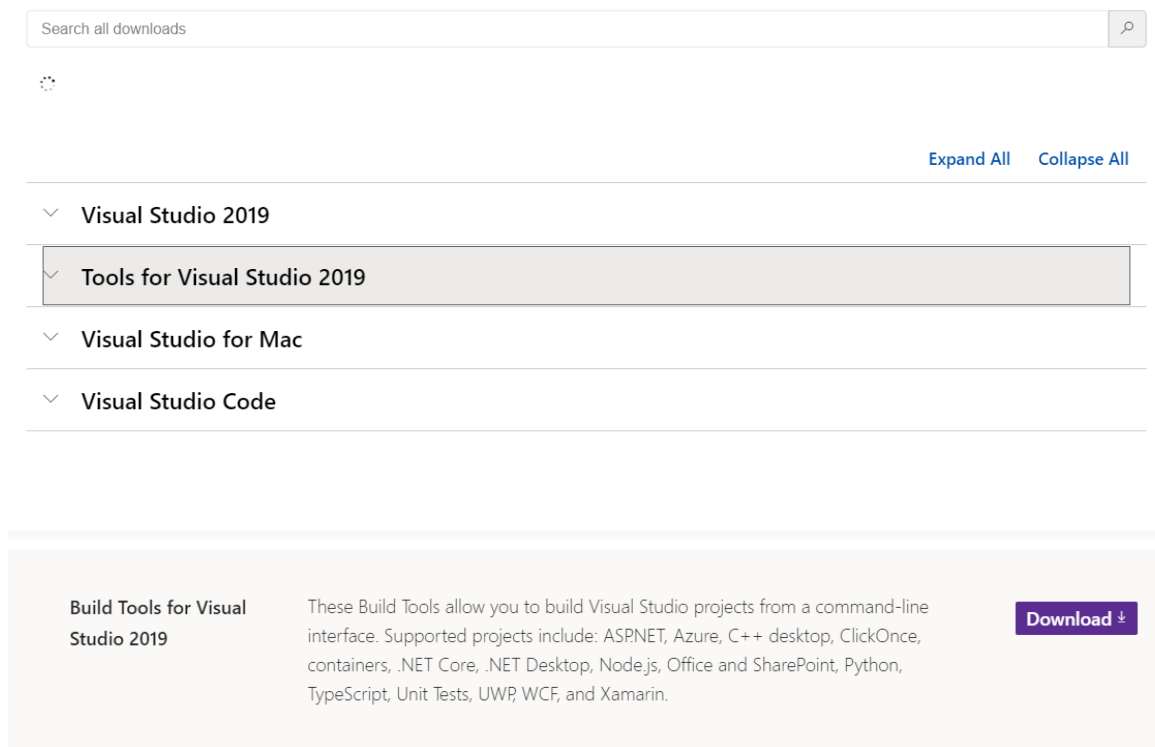
Hướng dẫn chạy C++ bằng VSCode

1. Cài đặt **Visual Studio Code**.
2. Cài đặt **C/C++** extension cho VS Code. Có thể cài đặt **C/C++** extension bằng cách search 'C++' trong Extensions của VS code (Ctrl+Shift+X).



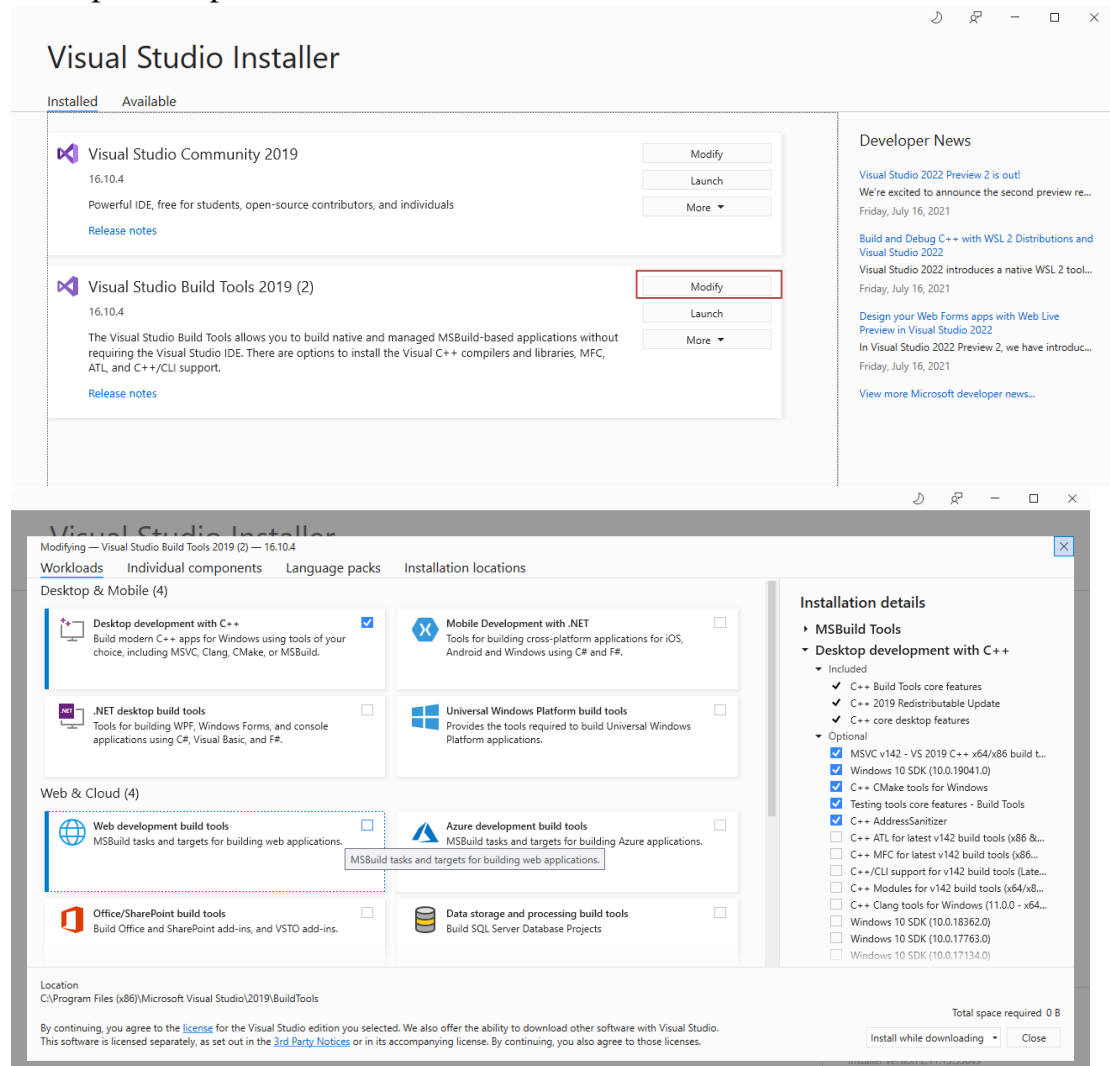
3. Cài đặt bộ công cụ biên dịch **Microsoft Visual C ++ (MSVC)**.
 - Nếu như chưa cài Visual Studio Build Tools ở Visual Studio Installer. Từ trang **Downloads** của **Visual Studio**, tại **Tools for Visual Studio** chọn download cho **Build Tools for Visual Studio**, sau đó cài đặt.

All Downloads

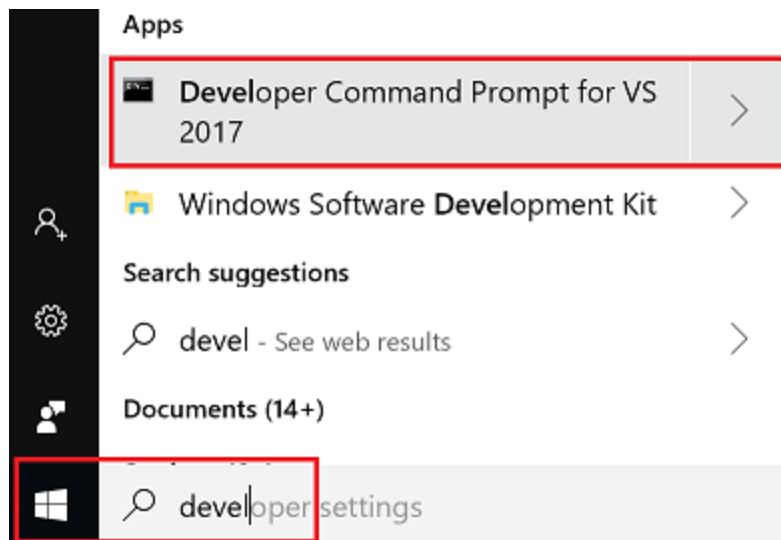


- Nếu đã cài Visual Studio Build Tools, mở Visual Studio Installer từ Windows Start menu, nhấn Modify của Visual Studio Build Tools và kiểm tra rằng C++

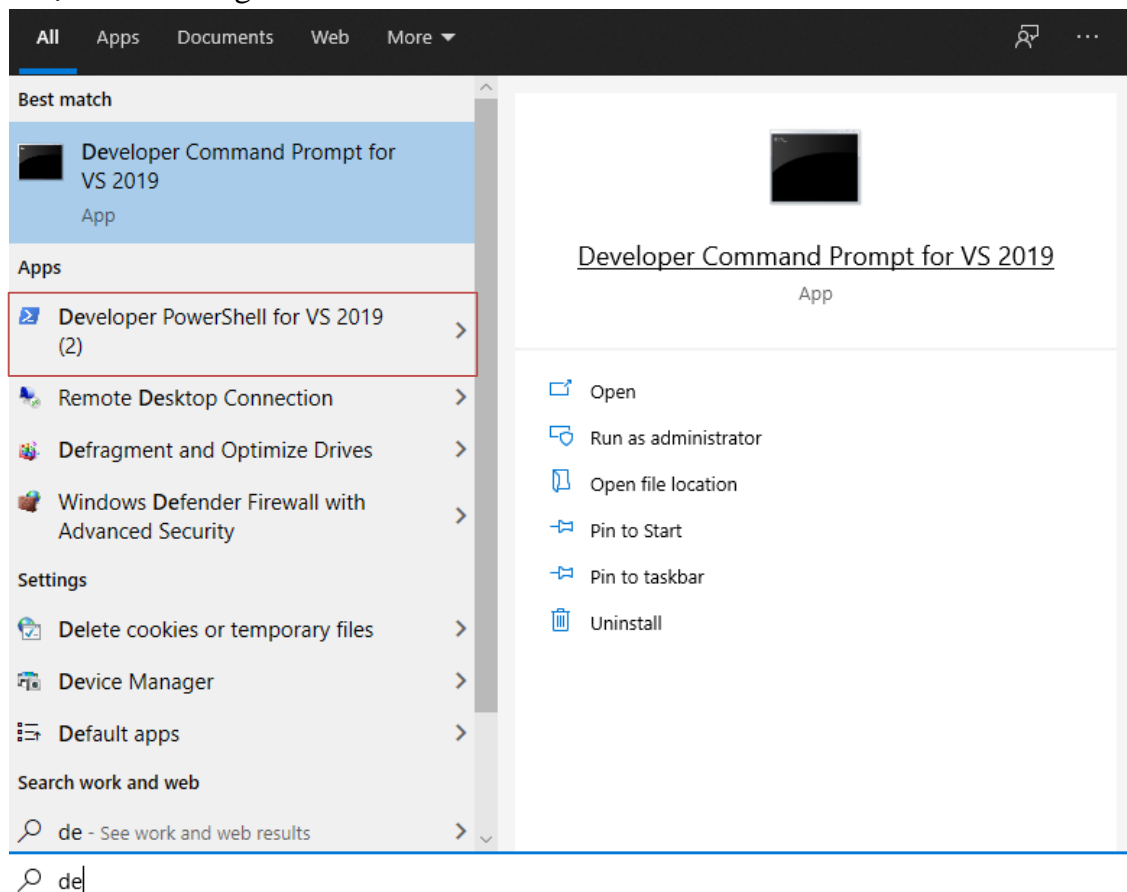
workload đã được check hay chưa. Nếu chưa được cài đặt thì click checkbox Desktop development with c++ và nhấn install để cài đặt.



4. Kiểm tra **Microsoft Visual C++(MSVC)** đã được cài đặt
 - Để sử dụng MSVC từ command line hoặc VS Code **phải** chạy project từ **Developer Command Prompt for Visual Studio**.
 - Gõ **developer** ở Windows Start menu, sẽ thấy tên được hiển thị sau đó click vào để mở.



Hoặc có thể dùng PowerShell



- Để kiểm tra xem đã cài đặt đúng trình biên dịch c++ (**cl.exe**) chưa, gõ 'cl' sẽ thấy dòng thông báo bản quyền, version và mô tả sử dụng cơ bản.


```
Developer Command Prompt for VS 2019

*****
** Visual Studio 2019 Developer Command Prompt v16.4.5
** Copyright (c) 2019 Microsoft Corporation
*****

C:\Program Files (x86)\Microsoft Visual Studio\2019\BuildTools>cl
Microsoft (R) C/C++ Optimizing Compiler Version 19.24.28316 for x86
Copyright (C) Microsoft Corporation. All rights reserved.

usage: cl [ option... ] filename... [ /link linkoption... ]

C:\Program Files (x86)\Microsoft Visual Studio\2019\BuildTools>
```

5. Tạo project

Tại **Developer Command Prompt for Visual Studio**, dùng lệnh *cd* <đường dẫn folder của project muốn chạy>

```
Developer Command Prompt for VS 2019 (2)

*****
** Visual Studio 2019 Developer Command Prompt v16.10.4
** Copyright (c) 2021 Microsoft Corporation
*****

C:\Program Files (x86)\Microsoft Visual Studio\2019\BuildTools>cl
Microsoft (R) C/C++ Optimizing Compiler Version 19.29.30040 for x86
Copyright (C) Microsoft Corporation. All rights reserved.

usage: cl [ option... ] filename... [ /link linkoption... ]

C:\Program Files (x86)\Microsoft Visual Studio\2019\BuildTools>cd C:\Users\Admin\Downloads\demo
C:\Users\Admin\Downloads\demo>
```

Sau đó dùng lệnh ***code*** . để mở thư mục

```
Developer Command Prompt for VS 2019 (2)

*****
** Visual Studio 2019 Developer Command Prompt v16.10.4
** Copyright (c) 2021 Microsoft Corporation
*****

C:\Program Files (x86)\Microsoft Visual Studio\2019\BuildTools>cl
Microsoft (R) C/C++ Optimizing Compiler Version 19.29.30040 for x86
Copyright (C) Microsoft Corporation. All rights reserved.

usage: cl [ option... ] filename... [ /link linkoption... ]

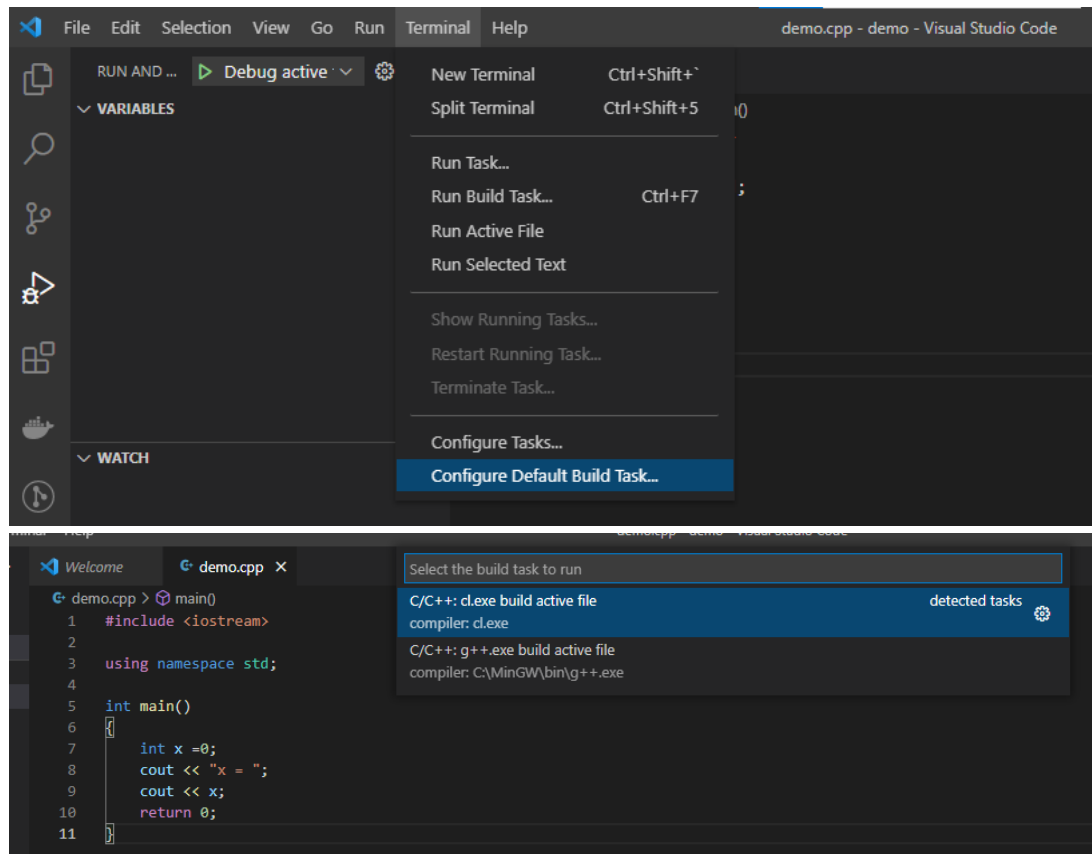
C:\Program Files (x86)\Microsoft Visual Studio\2019\BuildTools>cd C:\Users\Admin\Downloads\demo
C:\Users\Admin\Downloads\demo>code .
C:\Users\Admin\Downloads\demo>
```

Tạo file

```
demo.cpp x settings.json
demo.cpp > main()
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int x = 0;
8      cout << "x = ";
9      cin >> x;
10     cout << x;
11     return 0;
12 }
```

6. Build file

- Tạo file **tasks.json** để VS Code build (compile) chương trình. Task sẽ gọi Microsoft C++ compiler để tạo file .exe dựa vào source code.
- Từ thanh menu chọn **Terminal > Configure Default Build Task**. Chọn **cl.exe build active file**



- Sẽ tạo ra file **tasks.json** trong thư mục **.vscode**, mở file đó ra sẽ thấy dữ liệu như hình bên dưới :

```
.vscode > {} tasks.json > ...
1  {
2      "version": "2.0.0",
3      "tasks": [
4          {
5              "type": "cppbuild",
6              "label": "C/C++: cl.exe build active file",
7              "command": "cl.exe",
8              "args": [
9                  "/Zi",
10                 "/EHsc",
11                 "/nologo",
12                 "/Fe:",
13                 "${fileDirname}\\${fileBasenameNoExtension}.exe",
14                 "${file}"
15             ],
16             "options": {
17                 "cwd": "${fileDirname}"
18             },
19             "problemMatcher": [
20                 "$msCompile"
21             ],
22             "group": {
23                 "kind": "build",
24                 "isDefault": true
25             },
26             "detail": "compiler: cl.exe"
27         }
28     ]
29 }
```

- *args*: chỉ định các đối số dòng lệnh sẽ được chuyển đến cl.exe. Các đối số này phải được chỉ định theo thứ tự mà compiler mong đợi. Tác vụ này yêu cầu compiler C++ lấy tệp đang hoạt động (muốn compile) (`${file}`), biên dịch nó và tạo tệp thực thi trong thư mục hiện tại (cùng với file muốn compile) (`${fileDirname}`) có cùng tên với tệp đó với đuôi là .exe (`${fileBasenameNoExtension}.exe`), ví dụ: muốn compile file demo.cpp thì sau khi thực hiện task.json sẽ tạo ra file demo.exe
- *label*: tên task, có thể đặt tên tùy ý
- *cwd*: Thư mục làm việc hiện tại của task runner lúc khởi động
- *problemMatcher*: trình phân tích cú pháp đầu ra để sử dụng tìm lỗi và warnings output trong compiler (với cl.exe thì có kết quả tốt nhất khi dùng \$msCompile)

- *"isDefault"*: nếu true cho phép run task khi nhấn Ctrl+Shift+B tại file muốn build, nếu false thì chỉ có thể run task khi nhấn vào Terminal trên menu, chọn Tasks: Run Build Task.
- Muốn build demo.cpp, tại file đó khi task.json đã được cài đặt, nhấn Ctrl+Shift+B hoặc trên menu Terminal > Tasks: Run Build Task
- Nếu thành công thì sẽ được hình như bên dưới

```

TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

> Executing task: C/C++: cl.exe build active file <

Starting build...
cl.exe /Zi /EHsc /nologo /Fe: C:\Users\Admin\Downloads\demo\demo.exe C:\Users\Admin\Downloads\demo\demo.cpp
demo.cpp

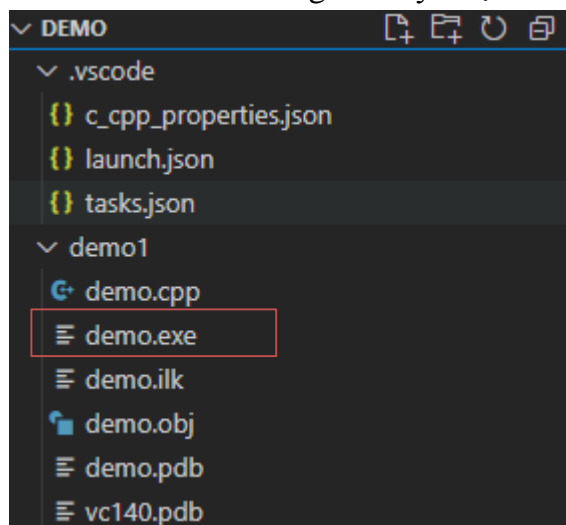
Build finished successfully.

Terminal will be reused by tasks, press any key to close it.

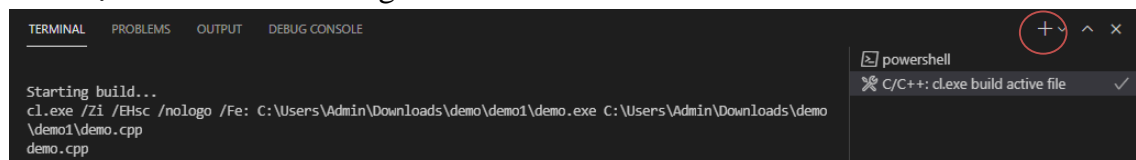
```

7. Run file

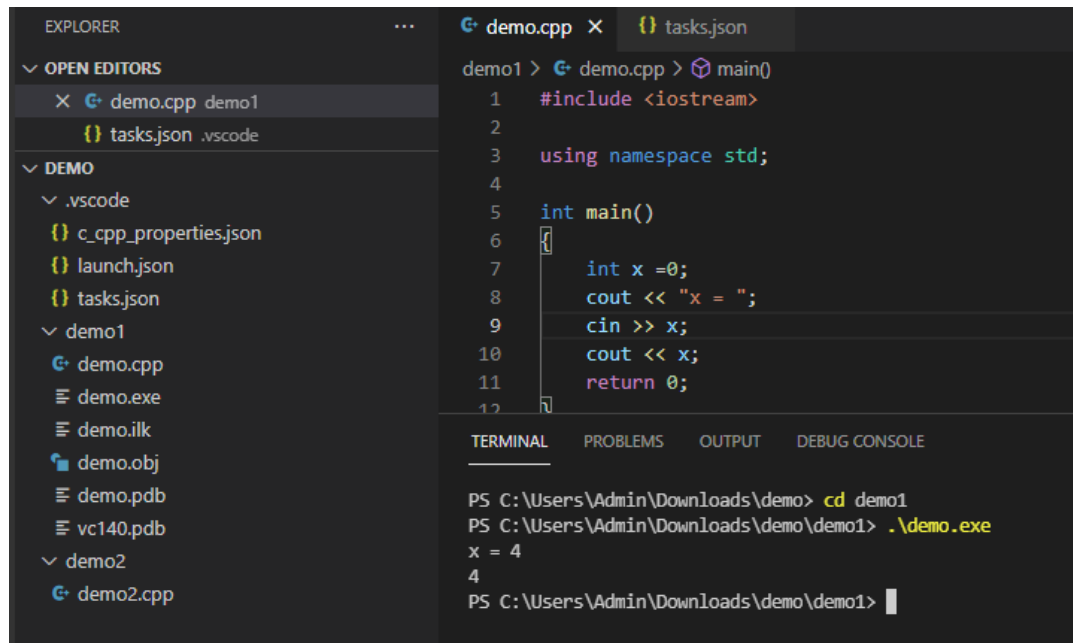
- Sau khi build file xong sẽ thấy được file demo.exe để thực thi code



- Mở một Terminal mới bằng cách nhấn nút +

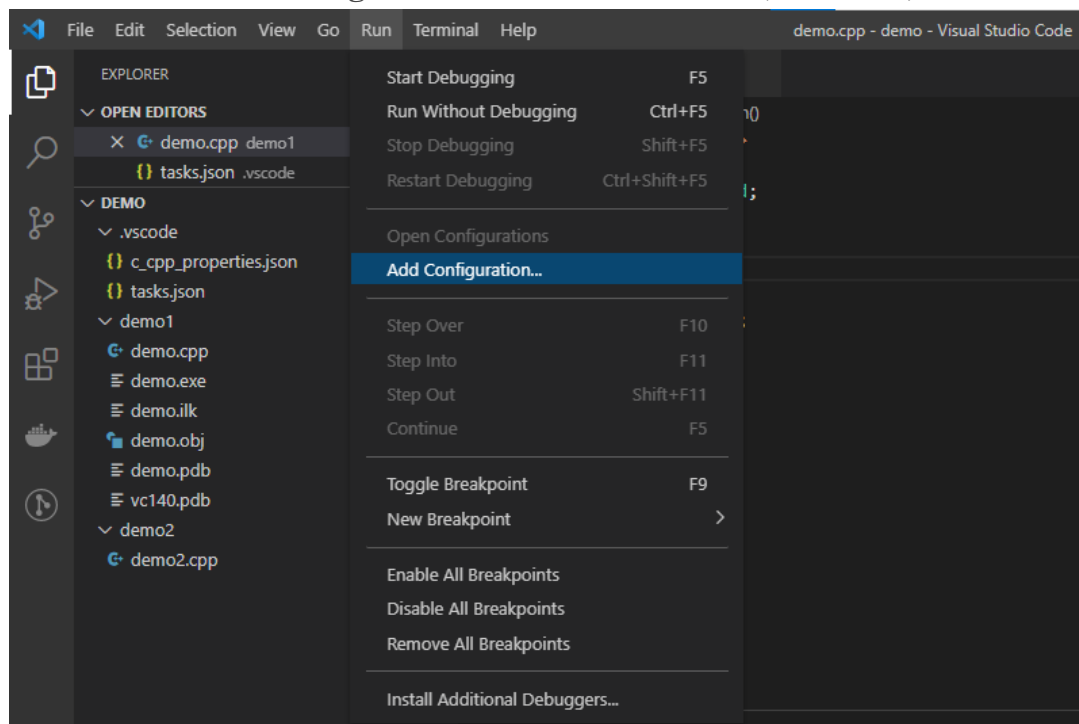


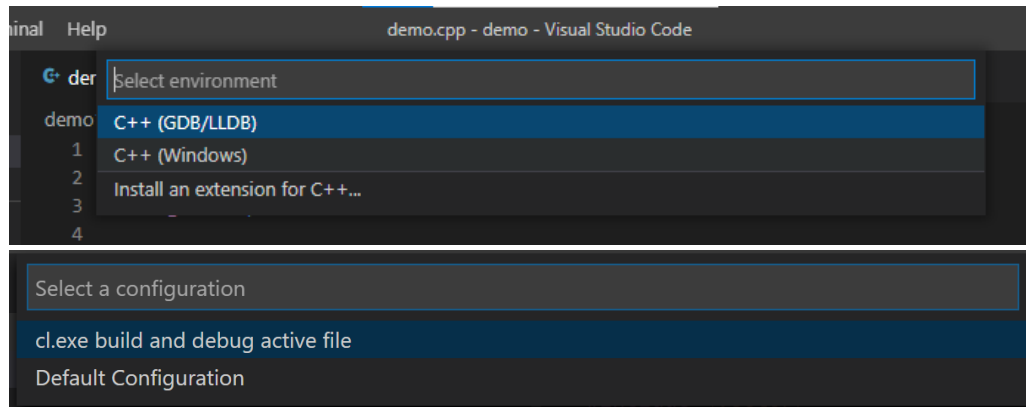
- Vào đúng đường dẫn thư mục chứa file source code, gõ `.\demo.exe` để chạy



8. Debug

- Tạo file **launch.json**
- Chọn **Run > Add Configuration...** sau đó chọn **C++ (Windows)**.





- Sẽ tạo ra file **launch.json** như bên dưới:

```

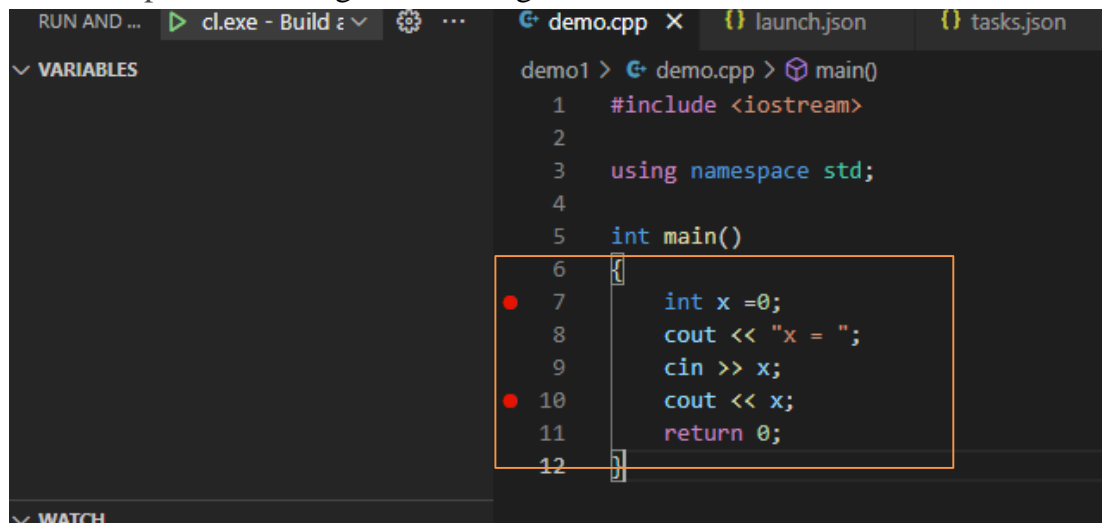
demo.cpp  launch.json  tasks.json
.vscode > {} launch.json > JSON Language Features > [ ] configurations > {} 0
1  {
2      // Use IntelliSense to learn about possible attributes.
3      // Hover to view descriptions of existing attributes.
4      // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
5      "version": "0.2.0",
6      "configurations": [
7          {
8              "name": "cl.exe - Build and debug active file",
9              "type": "cppvsdbg",
10             "request": "launch",
11             "program": "${fileDirname}\\${fileBasenameNoExtension}.exe",
12             "args": [],
13             "stopAtEntry": false,
14             "cwd": "${fileDirname}",
15             "environment": [],
16             "console": "externalTerminal",
17             "preLaunchTask": "C/C++: cl.exe build active file"
18         }
19     ]
20 }

```

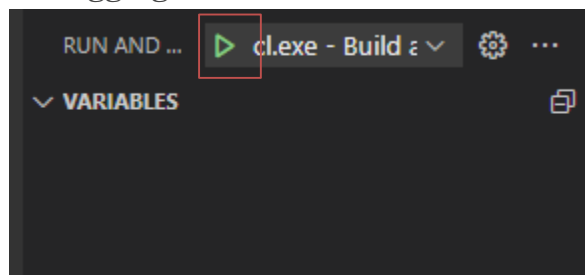
- *program*: chỉ rõ chương trình muốn debug, `${fileDirname}` sẽ là thư mục chứa tệp muốn debug, `${fileBasenameNoExtension}` là tên của tệp. Ví dụ: muốn chạy file `demo.cpp` thì lúc chạy nó sẽ thực thi file `demo.exe`
- *stopAtEntry*: Mặc định sẽ là `false`, lúc chạy thì sẽ không có breakpoints thì sẽ k dừng lại, nếu nó bằng `true` thì debugger sẽ dừng lại sau phương thức `main()`
- *cwd*: Thư mục làm việc hiện tại của ứng dụng mà debugger chạy
- *console*: `externalTerminal` : xuất hiện external console bên ngoài

```
c:\Users\Admin\Downloads\demo\demo1\demo.exe
x = 9
Press any key to continue . . .
```

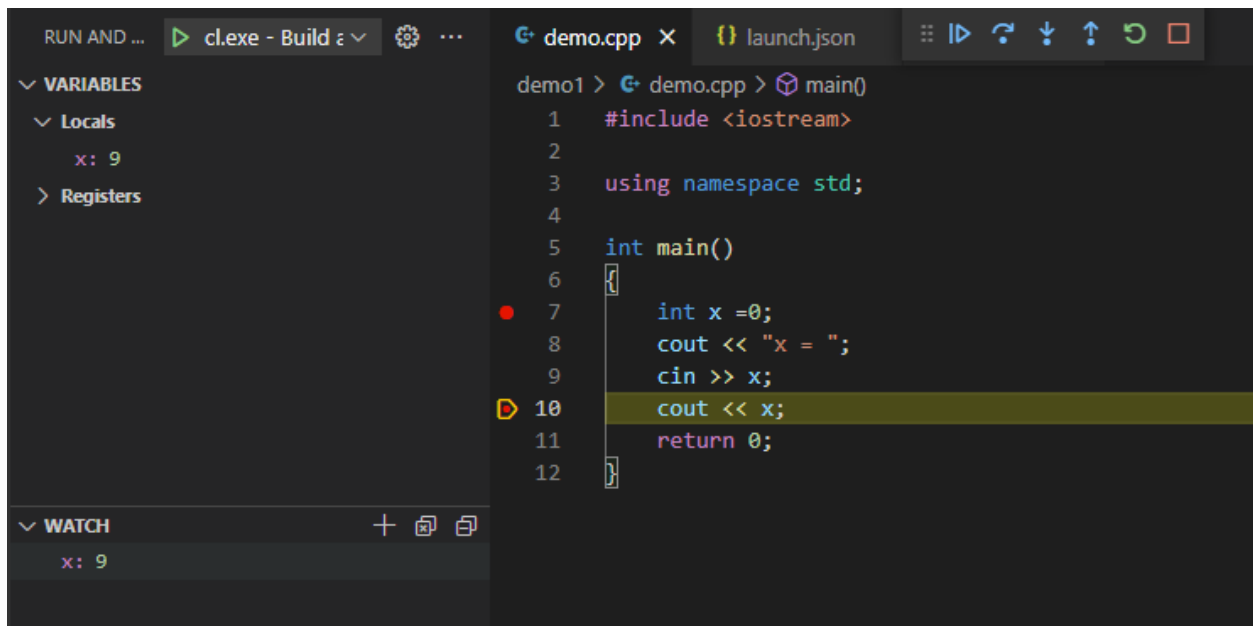
- Đặt breakpoint vào dòng muốn debug



- Bắt đầu debug, tại file muốn debug nhấn F5 hoặc chọn **Run > Start Debugging** hoặc

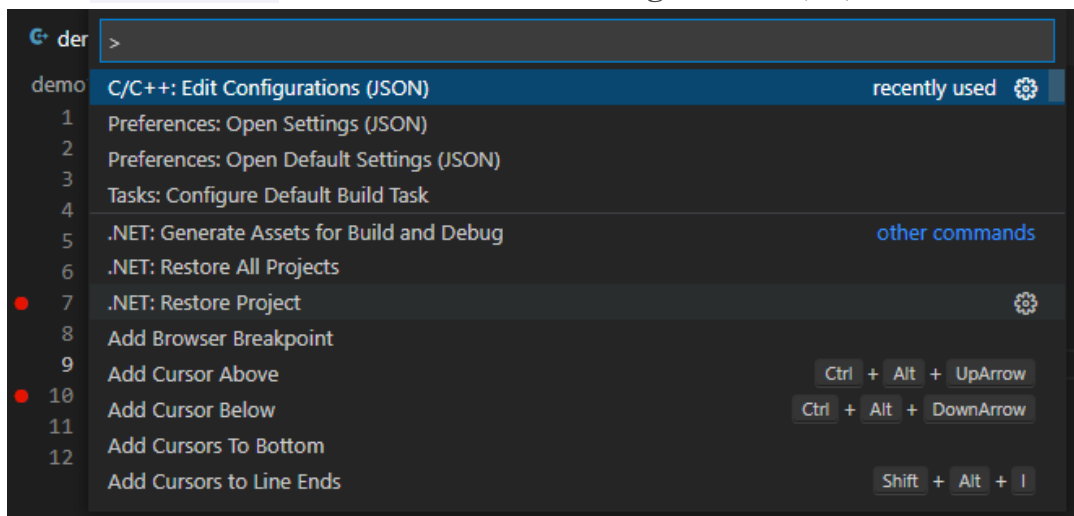


- Xuất hiện như hình bên dưới



9. Tạo file `c_cpp_properties.json`

- File này cho phép thay đổi setting như compiler path, include path, cStandard,...
- Nhấn **Ctrl+Shift+P** chọn **C/C++: Edit Configurations (UI)**



- Sẽ thấy file `c_cpp_properties.json` trong thư mục `.vscode`.