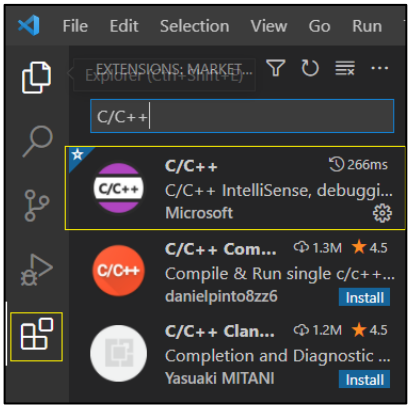
HOW TO INSTALL SOFTWARE DEVELOPMENT TOOLS FOR C/C++ PROGRAMMING

Please follow one of the below installation instructions depending on the operating system (Windows/macOS/Linux/) that you use.

Windows

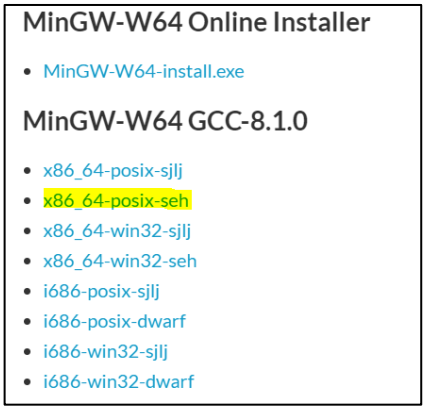
1. Install **VS Code**
    - Download the VS Code installer for Windows
    - Double-click at the downloaded file to run the installer

2. Install the **C/C++ extension for VS Code**
    - Open VS Code
    - Click at the Extensions icon in the Activity Bar on the left side of VS Code
    - Type "C/C++" in the Search box to search for the C/C++ extension
    - Click at the **Install** button of the extension

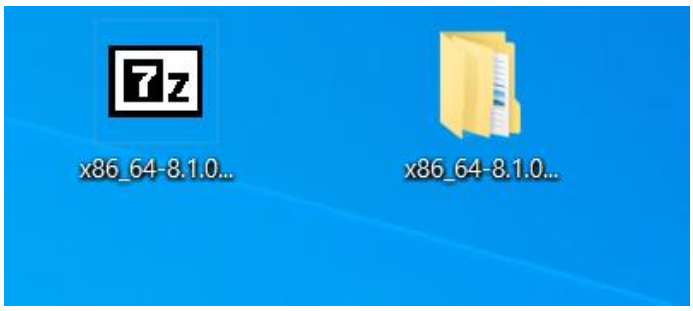


3. Install **gcc, g++ and C/C++ standard libraries**

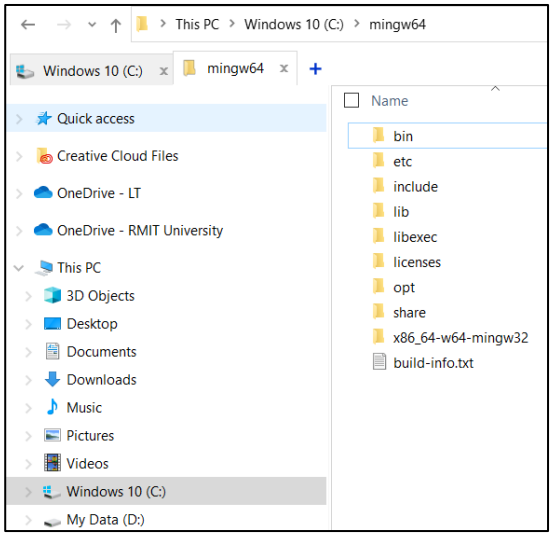
    a. Go to this link: MinGW-w64. Download **x86_64-posix-seh** with latest version.



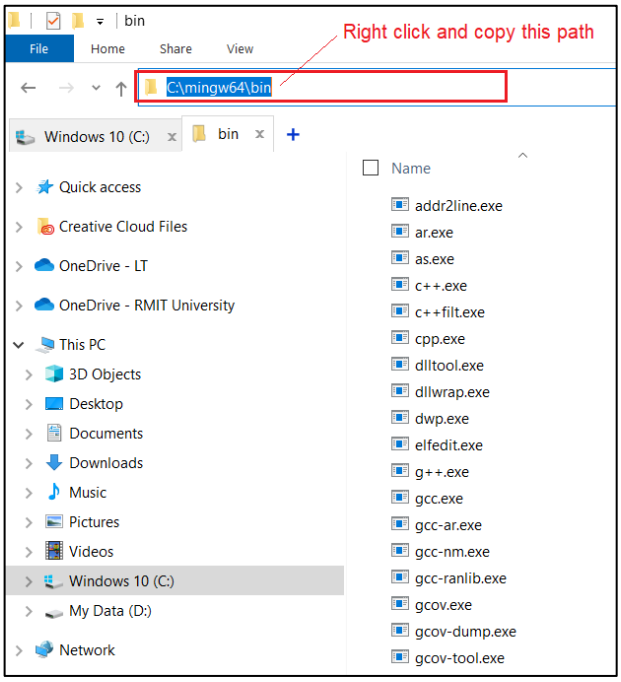
- After you download the file **x86_64-8.1.0-release-posix-seh-rt_v6-rev0.7z**, right click on it and select "**Extract Here**" with Winrar or 7-Zip tool to extract it. You will get the extracted folder as below:



- Now, open the extracted folder, you will see a folder namely **mingw64** inside. Select and copy that folder to your **C drive.** You will see it as below.

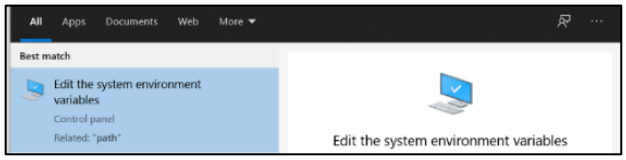


b. Browse to **bin** folder then <u>copy its path</u>. With instructions above, the path will be "***C:\mingw64\bin***" (it is fine if you copy it in another place and have different path but must have no space characters in the path).

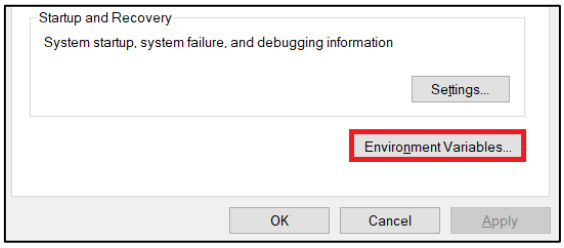


c. Add the above path to the Windows **PATH environment variable** so that gcc can run without the full path in Windows Terminal (PowerShell and Command Prompt):

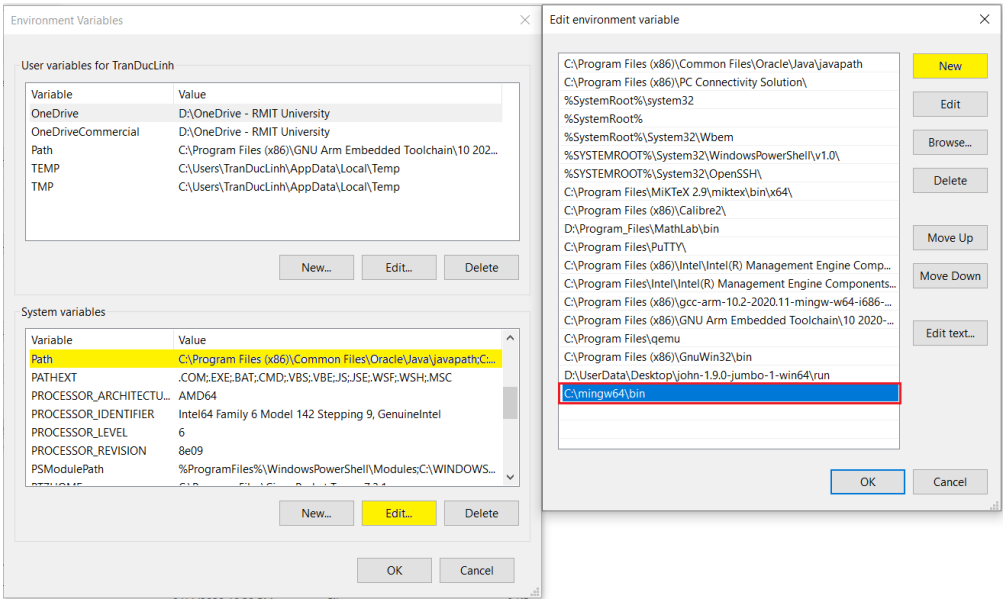
- In search bar, search for "**variables**" and select "**Edit the system environment variables**".



- Select "**Environment Variables**"



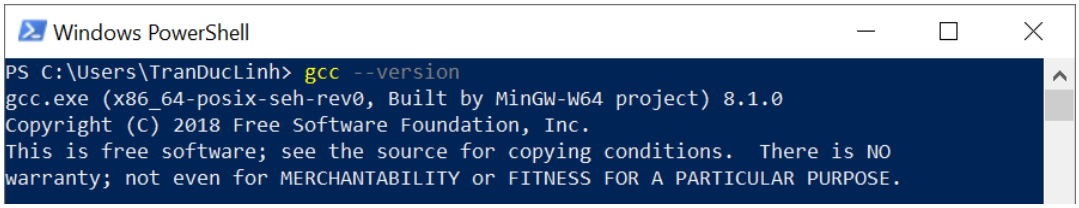
- Under **System variables** area, select **Path** > **Edit**. In the **Edit environment variable** window, press **New** and input path to the bin directory of our installed GCC tool. After that, press **OK** in all windows to complete PATH setting.



d. Test the gcc tool to make sure that it is installed properly:

Type ***powershell*** in Search bar to open PowerShell

Type "**gcc --version**". If it is installed correctly, you should see the result as below:

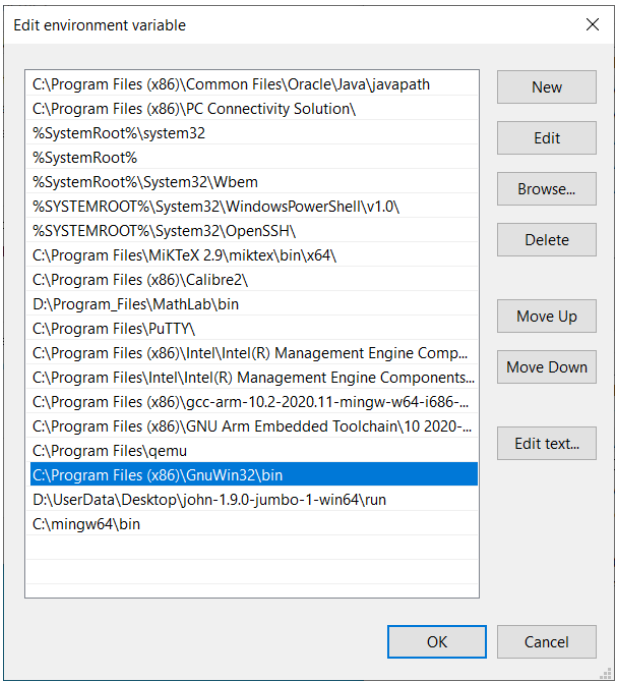


4. Install GNU **Make** tool

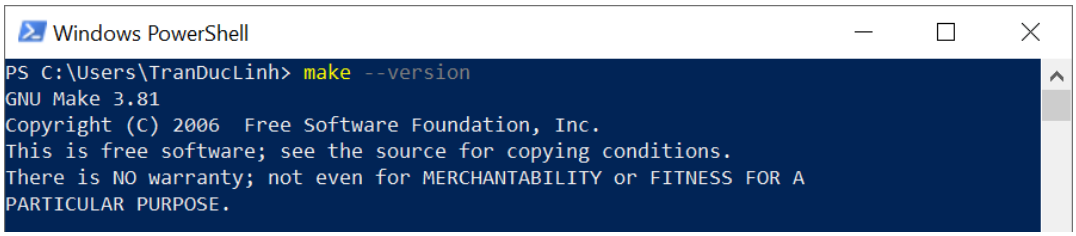
   a. Go to the link http://gnuwin32.sourceforge.net/packages/make.htm

      Select "Complete package, except sources" Setup to download and install it.

   b. Similary, add path of its bin directory (by default is *C:\Program Files (x86)\GnuWin32\bin*) to system PATH environment variable.



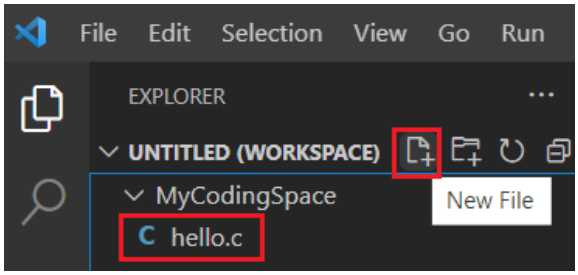
   and check again with command "**make --version".**



5. **Get familiar with VS Code**

   a. Close and re-open **VS Code** if it is opening. Click on **File** > **Add Folder to Workspace.** Browse to/ create a folder where you want to save all of your programs (e.g. D:\MyCodingSpace), and select it.

   b. Create a new file namelly **hello.c** with example program as below



Copy the following code and paste it into your hello.c file:

```c
#include <stdio.h>
int main()
{
    int num;
    printf("Enter an integer: ");
    scanf("%d", &num);
    printf("Got number = %d", num);
    return 0;
}
```

c. Click on **File** > **Auto Save** to turn on Auto Save feature for all files (very helpful feature !).

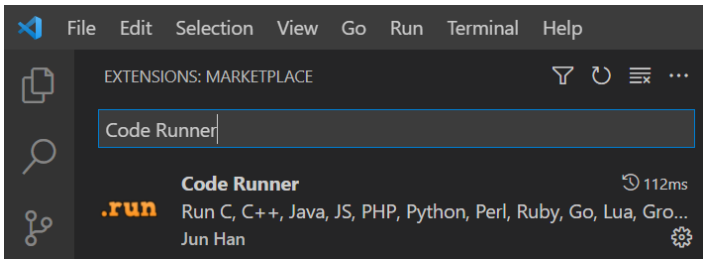
d. Click on **Teminal** > **New Terminal** to open Windows' Powershell terminal.

e. In the terminal window, type "**gcc hello.c**" to compile the hello.c file. By default it will generate a.exe in Windows as a result (can type "**ls**" to see all files). Then, type "**./a.exe**" to run the program. It should work properly as below:

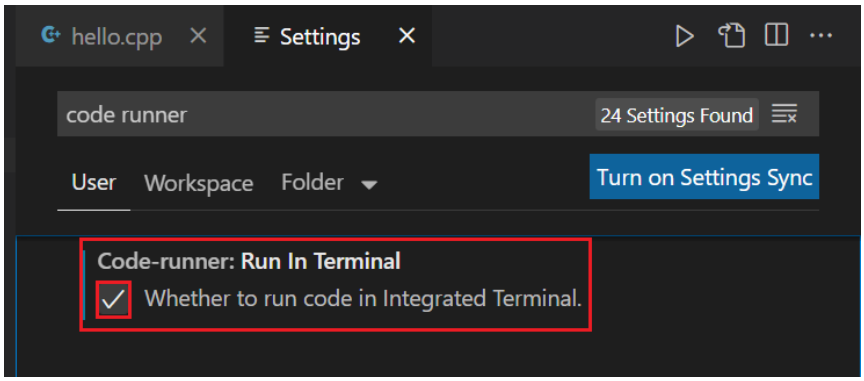
```
PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE

PS C:\Users\TranDucLinh\MyCodingSpace> gcc hello.c
PS C:\Users\TranDucLinh\MyCodingSpace> ls

    Directory: C:\Users\TranDucLinh\MyCodingSpace

Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        6/10/2021   4:17 PM          54198 a.exe
-a----        6/10/2021   3:53 PM            187 hello.c

PS C:\Users\TranDucLinh\MyCodingSpace> ./a.exe
Enter an integer:
```

6. Install and use **Code Runner** extension for "click to compile and run" feature.

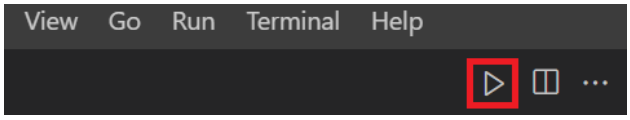
- Click at the Extensions icon in the Activity Bar on the left side of VS Code
  Type "Code Runner" in the Search box to search for it, and click at the **Install** button of the extension.



- Go to File > Preferences > Setting. Search for Code Runner. Scroll down to find and tick on option "**Code-runner: Run In Terminal**" (will make the program to run in terminal when we hit Run button).



- Now, you can hit the RUN button to compile and run your program.

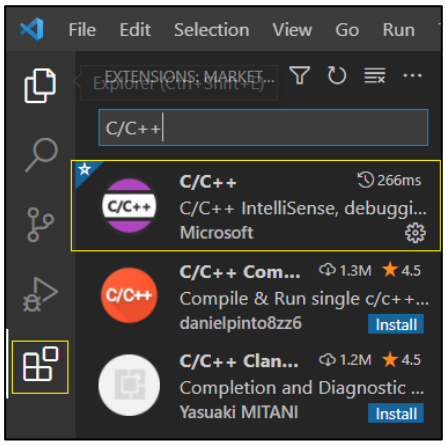


Mac OS

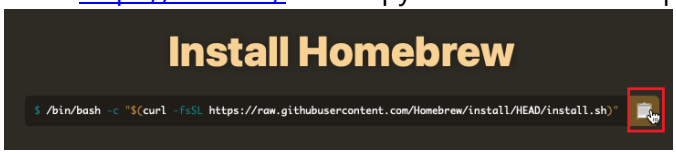
1. Install VS Code
    - Download the VSCode to a directory
    - Open Finder and navigate to that directory
    - Double-click at the zip file to unzip it
    - Drag the **Visual Studio Code.app** to the **Applications** directory to make it available in **Launchpad**

    *Note: If you use macOS Catalina (macOS 10.15), you may see a message "**Visual Studio Code can't be opened because Apple cannot check it for malicious software**". This is because Visual Studio Code is not currently notarized but it will run just fine on macOS Catalina. To work around the notarization check, open **Launchpad** > **System Preferences** > **Security & Privacy** > **General** and choose **Open Anyway**.*

2. Install the **C/C++ extension for VS Code**
    - Open VS Code
    - Click at the Extensions icon in the Activity Bar on the left side of VS Code
    - Type "C/C++" in the Search box to search for the C/C++ extension
    - Click at the **Install** button of the extension



3. Install gcc, g++ and C/C++ standard libraries
    a. Install homebrew
    - Go to https://brew.sh/ and copy the installation script



    - Open the terminal, paste the script and run it

```
linhtd@MacOSs-iMac ~ % /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.
com/Homebrew/install/HEAD/install.sh)"
==> Checking for `sudo` access (which may request your password).
Password:
==> This script will install:
/usr/local/bin/brew
/usr/local/share/doc/homebrew
/usr/local/share/man/man1/brew.1
/usr/local/share/zsh/site-functions/_brew
/usr/local/etc/bash_completion.d/brew
/usr/local/Homebrew

Press RETURN to continue or any other key to abort
==> /usr/bin/sudo /usr/sbin/chown -R linhtd:admin /usr/local/Homebrew
==> Downloading and installing Homebrew...
Updating files: 100% (2721/2721), done.
HEAD is now at 976f9daa1 Merge pull request #12235 from Homebrew/dependabot/bund
ler/Library/Homebrew/sorbet-0.5.9226
Updated 1 tap (homebrew/core).
==> Installation successful!
```

b. Install gcc tool from homebrew:

- Run command "**brew install gcc**"

```
linhtd@MacOSs-iMac ~ %
linhtd@MacOSs-iMac ~ %
linhtd@MacOSs-iMac ~ % brew install gcc
==> Downloading https://ghcr.io/v2/homebrew/core/gcc/manifests/11.2.0
Already downloaded: /Users/linhtd/Library/Caches/Homebrew/downloads/21
0783e77b227b8210d559abfe3514cdb95c915619fa3f785ad212120d6a36f9--gcc-11
.2.0.bottle_manifest.json
==> Downloading https://ghcr.io/v2/homebrew/core/gcc/blobs/sha256:da67
5b722172d8866c8c3eed38a107ebdb7fb8c5e9a9a8589382d55
Already downloaded: /Users/linhtd/Library/Caches/Homebrew/downloads/d2
7cf99015ea79170ceb740a8b1b5554d952d9d1b0cdfd9005a819c5c32953f8--gcc--1
1.2.0.big_sur.bottle.tar.gz
==> Pouring gcc--11.2.0.big_sur.bottle.tar.gz
🍺  /usr/local/Cellar/gcc/11.2.0: 2,163 files, 459.7MB
linhtd@MacOSs-iMac ~ %
```

- Check install success by **brew info gcc**

```
linhtd@MacOSs-iMac ~ % brew info gcc
gcc: stable 11.2.0 (bottled), HEAD
GNU compiler collection
```

<u>If the installation FAIL</u>, type following commands:

```
rm -rf /usr/local/Homebrew/Library/Taps/homebrew/homebrew-core
brew tap homebrew/core
```

- As above, the gcc version 11 has been installed. Check again by command **gcc-11 –version.**

```
linhtd@MacOSs-iMac ~ % gcc-11 --version
gcc-11 (Homebrew GCC 11.2.0) 11.2.0
Copyright (C) 2021 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

linhtd@MacOSs-iMac ~ %
```

Note: *<u>You may get another version</u> depending on the current time of installation. For example, if you got version 12, type gcc-12 –version.*

c. Create a <u>symbolic link</u> (shortcut) file for gcc.

On Mac OS, the default compiler is clang. Thus, to use gcc, we need to create a symbolic link namely gcc which links to our actual gcc-11 file (so that we can just type gcc to use our actual gcc version 11 tool)

- Check where the gcc-11 is installed
  which gcc-11

```
linhtd@MacOSs-iMac ~ % which gcc-11
/usr/local/bin/gcc-11
```

You may get **/usr/local/bin/gcc-11** as the result, then it is in **/usr/local/bin/** (some people may get another location, e.g. /opt/local/bin/gcc-11).

- **cd** to the location that you get above and list out content by flowing commands.
  You should see gcc-11 file there:

  cd /usr/local/bin
  ls

Note: in "**ls**" and "**ln**" commands, l is **L letter in lower-case** (not i letter).

```
linhtd@MacOSs-iMac ~ % cd /usr/local/bin
linhtd@MacOSs-iMac bin % ls
brew
c++-11
cpp-11
g++-11
gcc-11
gcc-ar-11
gcc-nm-11
gcc-ranlib-11
gcov-11
gcov-dump-11
gcov-tool-11
gdc
gdc-11
```

- Create a symbolic link file namely gcc which links to our actual gcc-11 file as below
  ln -s gcc-11 gcc

- Check again with "**ls**" command. You should see both files gcc (symbolic link) and actual gcc-11 file.

  ls

```
linhtd@MacOSs-iMac bin % ls
brew
c++-11
cpp-11
g++-11
gcc
gcc-11
gcc-ar-11
gcc-nm-11
gcc-ranlib-11
gcov-11
gcov-dump-11
gcov-tool-11
gdc
gdc-11
```

d. Now, **close and reopen the terminal**, type "**gcc –version**" to check that whether we have successfully have configured it correctly or not. You should see it as Homebrew GCC (not clang) as below:

```
linhtd@MacOSs-iMac ~ % gcc --version
gcc (Homebrew GCC 11.2.0) 11.2.0
Copyright (C) 2021 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

linhtd@MacOSs-iMac ~ %
```

4. Install GNU Make tool

a. Check that you already have the make tool or not

make --version

```
linhtd@MacOSs-iMac ~ % make --version
GNU Make 3.81
Copyright (C) 2006  Free Software Foundation, Inc.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.

This program built for i386-apple-darwin11.3.0
```

b. If you do not have it, install it through command

brew install make

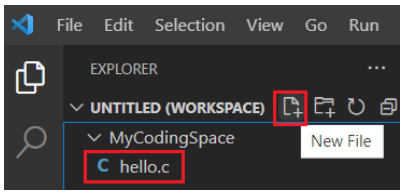
```
linhtd@MacOSs-iMac ~ % brew install make
==> Downloading https://ghcr.io/v2/homebrew/core/make/manifests/4.3-1
######################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/make/blobs/sha256:2019ba646e447
==> Downloading from https://pkg-containers.githubusercontent.com/ghcr1/blobs/sh
######################################################################## 100.0%
==> Pouring make--4.3.big_sur.bottle.1.tar.gz
==> Caveats
```

5. Check and Get familiar with VS Code

a. Create a new folder to store your codes, e.g. MyCodingSpace on your Desktop.

b. Open VS Code. Click on File > Add Folder to Workspace. Browse to that folder and select it.

c. Create a new file namelly hello.c with example program as below



Copy the following code and paste it into your hello.c file:

```c
#include <stdio.h>
int main()
{
    int num;
    printf("Enter an integer: ");
    scanf("%d", &num);
    printf("Got number = %d", num);
    return 0;
}
```

d. Click on File > Auto Save to turn on Auto Save feature for all files (very helpful feature !).

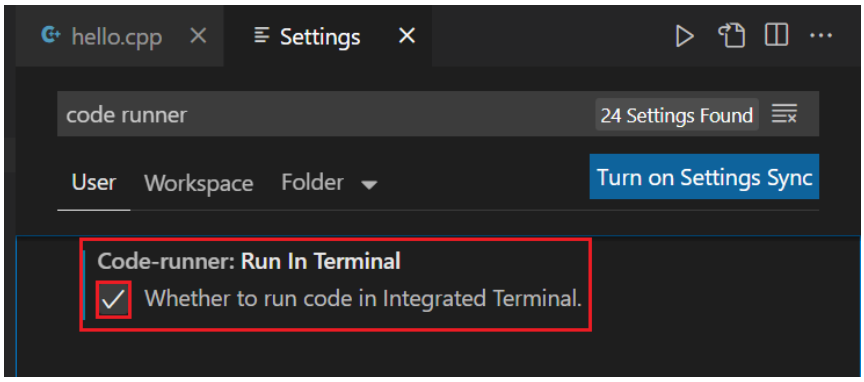
e. Click on Teminal > New Terminal to open Terminal.

f. In the terminal window, type “gcc hello.c” to compile the hello.c file. By default it will generate a.out file as a result (can type “ls” to see all files). Then, type “./a.out” to run the program.

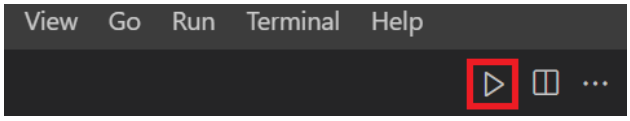
6. **Install and use Code Runner extension for "click to compile and run" feature.**

   a. Click at the Extensions icon in the Activity Bar on the left side of VS Code
   Type "**Code Runner**" in the Search box to search for it, and click at the **Install** button of the extension.

   b. Go to **Code > Preferences > Setting**. Search for Code Runner. Scroll down to find and tick on option "**Code-runner: Run In Terminal**" (will make the program to run in terminal when we hit Run button).



   c. Now, you can hit the **RUN** button to compile and run your program.



Linux (Ubuntu/Debian/Mint)

Similarly for Linux:

1. Install **VS Code**
   - Download the deb file to a directory
   - Open Terminal then navigate to that directory
   - **$ sudo apt install ./<file>.deb**

   *Note: **<file>.deb** must be replaced by the exact name of the downloaded deb file.*

2. Install the **C/C++ extension for VS Code**
   - Open VS Code
   - Click at the Extensions icon in the Activity Bar on the left side of VS Code
   - Type "C/C++" in the Search box to search for the C/C++ extension
   - Click at the **Install** button of the extension

3. Install **gcc, g++ and C/C++ standard libraries** using Terminal
   - **$ sudo apt install gcc g++**
   - **$ sudo apt install build-essential**