VS Code C/C++
环境配置实用手册

Windows 10/11 版

第三版 修订于 2024/11/28

前言

当你翻开本手册时，大抵只有两种情况：一种情况是这是你首次接触 VS Code C/C++环境（尽管你可能不是第一次接触 C/C++），另一种情况时你的 VS Code C/C++环境无法正常使用了。

对于第一种情况的同学而言，也就是对于首次安装 VS Code C/C++环境的同学们而言，请你们不要畏惧本手册的长度。这本手册主要是由各种屏幕截图构成，实际上并没有特别多的步骤，而且其中有不少文本也是废话（比如目前这段）。只要耐心、仔细地跟着手册做一遍，你的环境一定不会有问题！

对于第二种情况的同学而言，也就是对于环境遇到问题需要修复的同学而言，接下来的话十分重要。

解决任何环境问题的基本方法是关闭重开、卸载重装。关闭重开我相信你已经试过了，并且没有效果。这时你需要做的，就是卸载你的环境，然后按照本教程重新安装一遍。

你不需要卸载一切，或者具体来说大部分情况下你不需要卸载你的 VS Code，卸载重装 VS Code 一般也不能解决问题。你只需要卸载你的 MinGW 编译器（如果是用 MSYS 安装的，请卸载 MSYS；如果是用解压的方法安装的，请删除原先解压出的文件夹；如果是用 MinGW 在线安装程序安装的，请找到 MinGW 卸载程序卸载 MinGW），然后删除存放.c 和.cpp 代码的那个文件夹里面的.vscode 文件夹。之后，请跟着本教程重新配置一遍环境。我相信你的问题一定会解决，因为我之前就按照这种方式解决了好几个同学的问题！

总之，你一定要相信你可以独立完成 VS Code C/C++环境的配置。其实你的助教以及热心的学长并没有什么魔法，他们帮你安装或修复环境时遵照的也是本教程的方法。

相信自己，你一定可以做到！

第三版修订者 李甘
Nictheboy Li <nictheboy@outlook.com>
于 2024 年 11 月 28 日

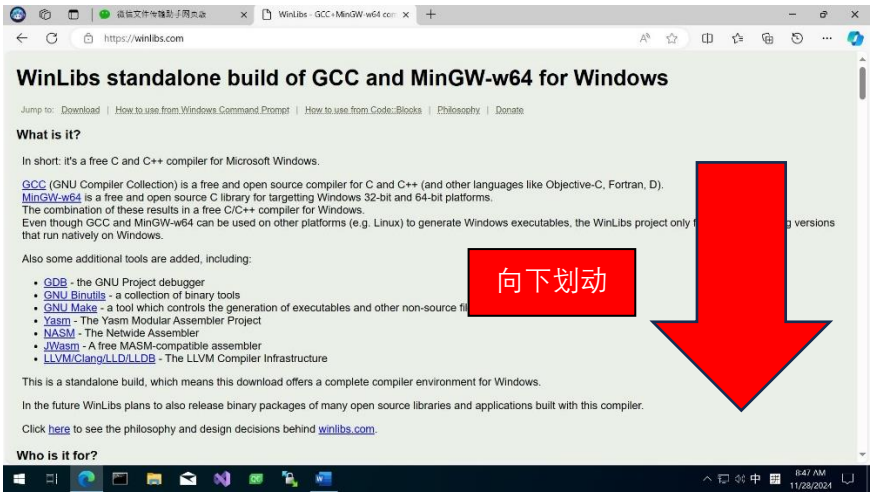
目录

一、下载（或通过助教获取）MinGW-W64

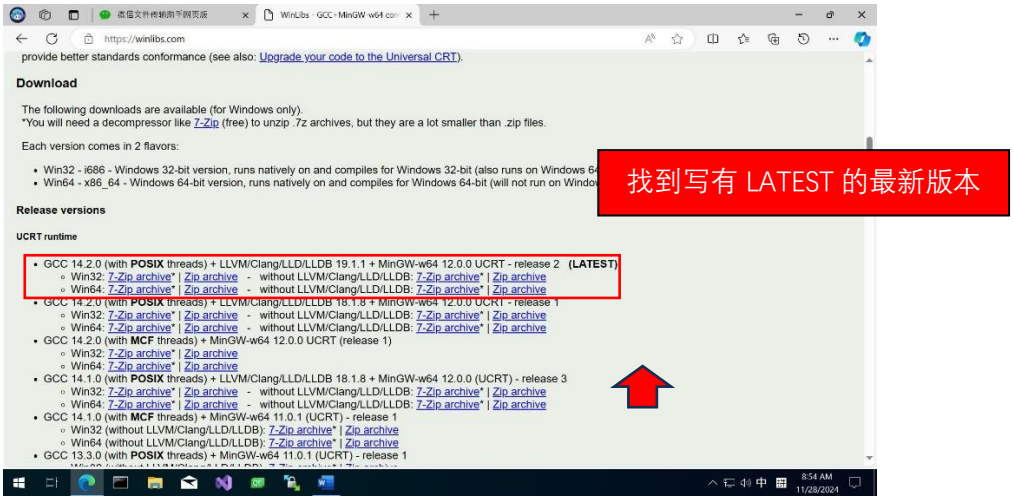
推荐直接通过助教获取此文件。因为在不使用网络工具的前提下，下面的下载链接下载速度很慢（10kb/s）（使用工具可达到 3mb/s）。下面的下载方法主要是给助教提供参考，请助教提前借助有关工具完成下载。

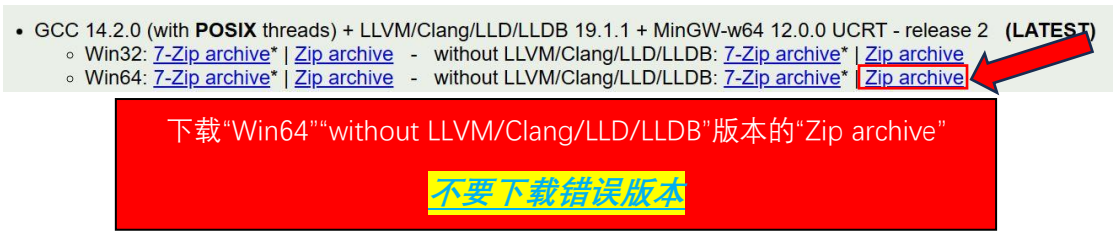
版本问题至关重要！请务必下载“without LLVM/Clang/LLD/LLDB”的版本以避免烦恼！

下载地址：https://winlibs.com/
打开网页后，向下划动



看到此列表后，选择最新版本的“Win64”“without LLVM/Clang/LLD/LLDB”版本的“Zip archive”





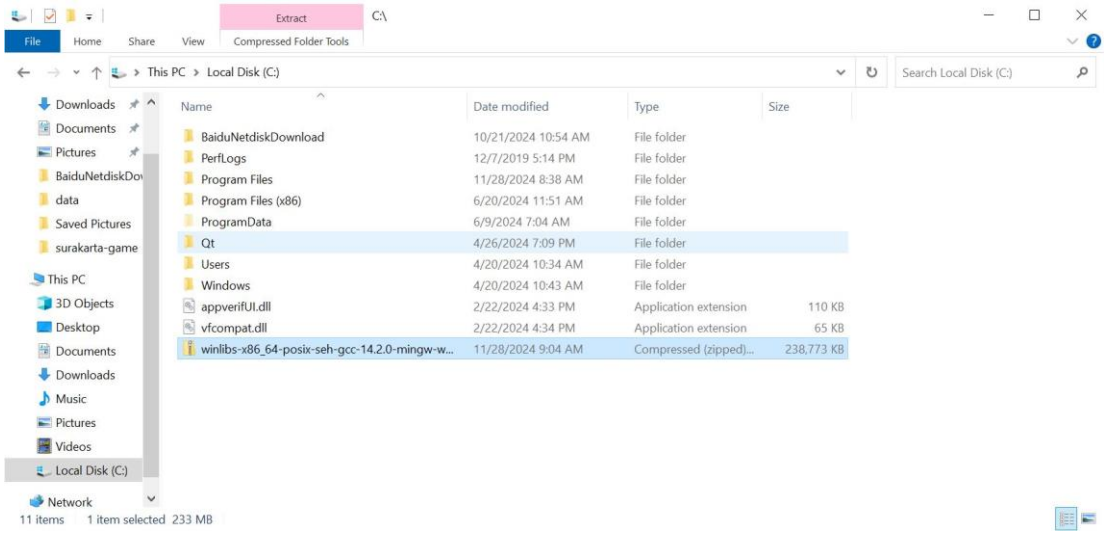
如果下载过程较慢，请联系助教获取此文件，或使用访问国际互联网的有关工具下载。

二、解压 MinGW-W64

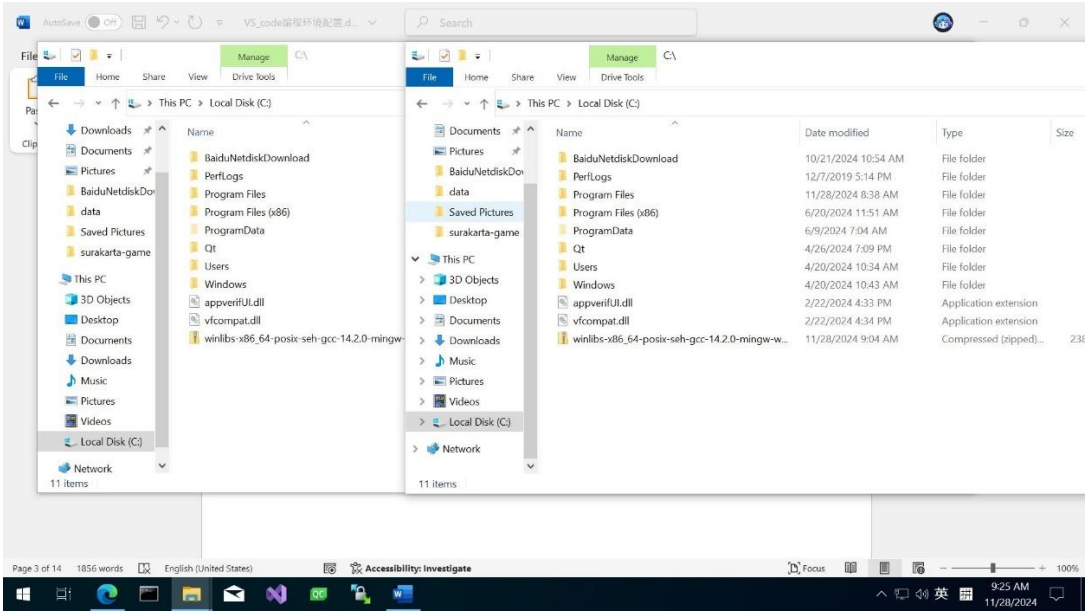
本节内容（解压 MinGW-W64）实际上就是将刚刚下载的压缩包内的“mingw64”文件夹解压至 C:\mingw64，熟悉“压缩文件”和“解压”的同学可不阅读本节内容，自行完成解压。如果不熟悉“压缩文件”和“解压”也不用担心，只要一步一步跟着本节教程完成即可。

根据具体情况，也可以解压至 C:\mingw64 外的其他地方，如 D:\Programs\mingw64 等。如果这么做，请注意两件事情。第一，保证该路径不含中文，如不能为“D:\新建文件夹\mingw64”。第二，在后续“三、配置 MinGW-W64 的环境变量”和“六、编写 VS Code 的文件夹配置文件”的步骤中，需要将所有 C:\mingw64\bin 改为相应的路径，如 D:\Programs\mingw64\bin。

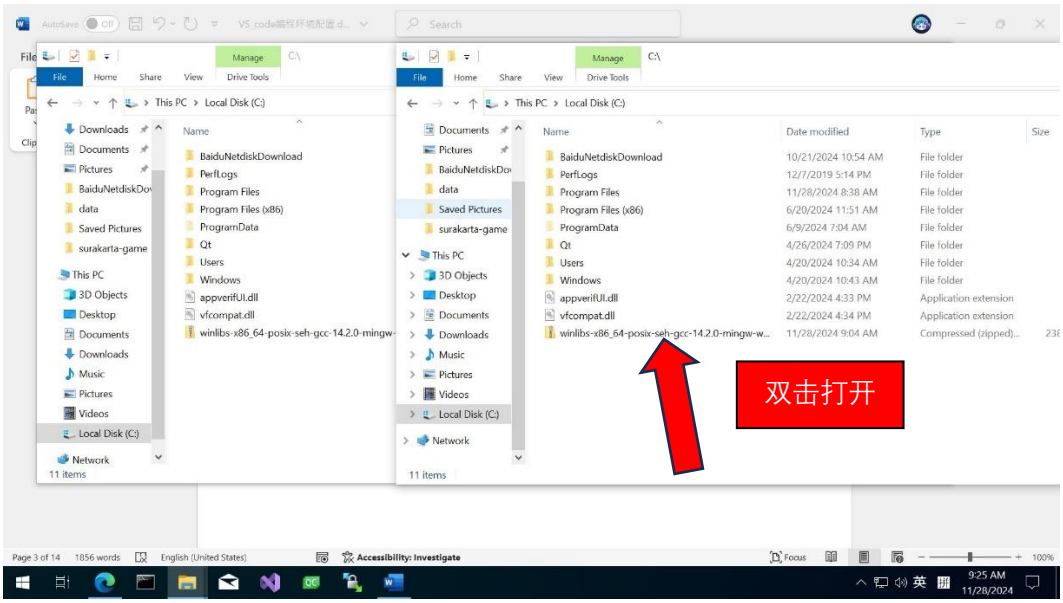
将刚刚获得的文件（文件名类似于 winlibs-x86_64-posix-seh-gcc-14.2.0-mingw-w64ucrt-12.0.0-r2.zip）置于 C 盘根目录

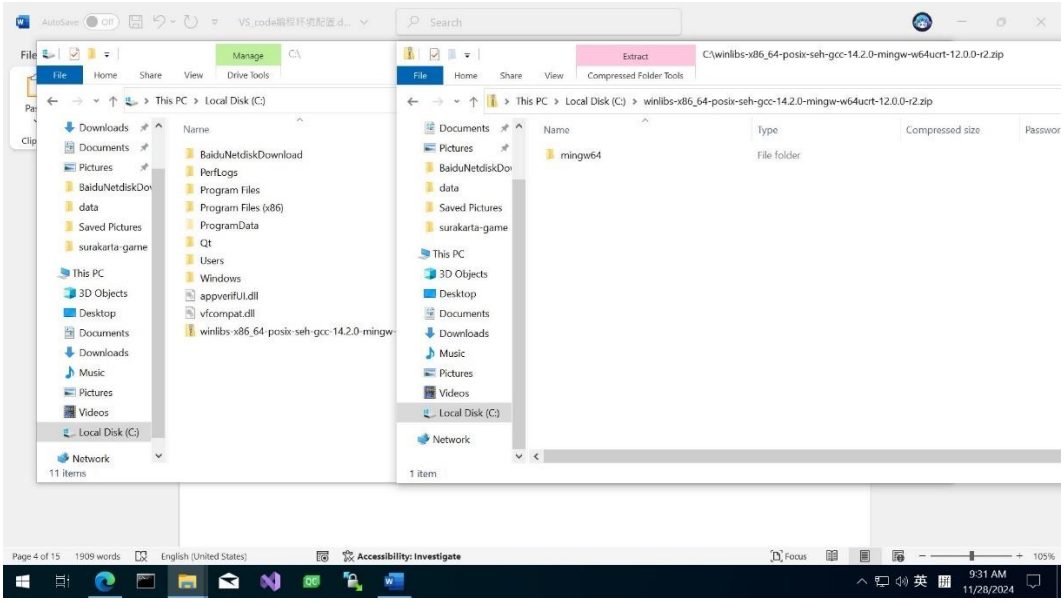


打开两个资源管理器（文件管理器）窗口，都打开 C 盘

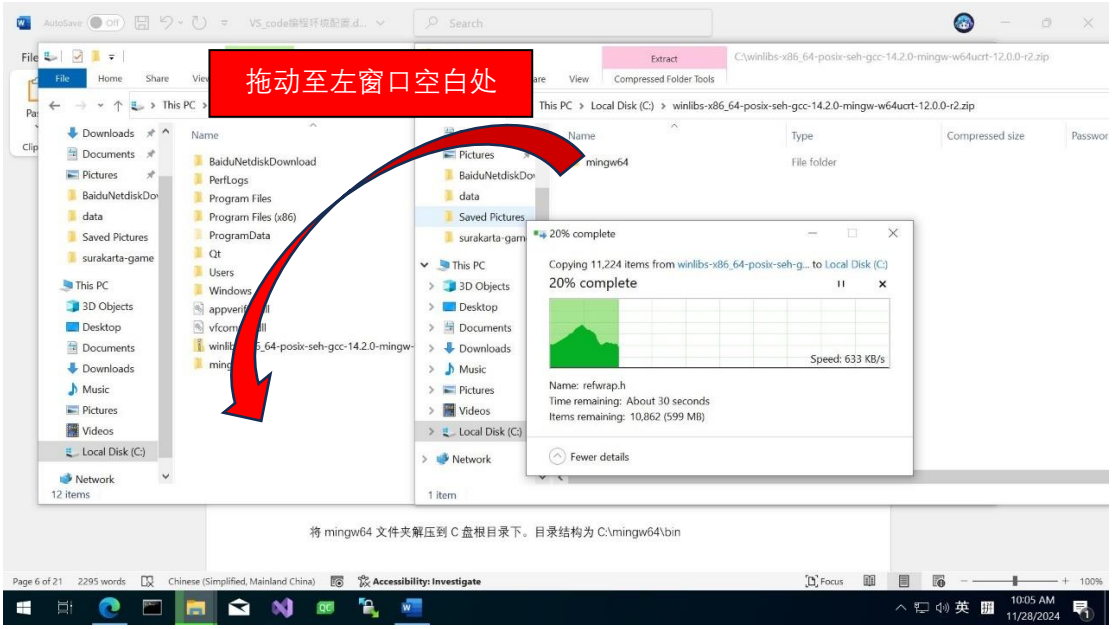


在右侧的窗口中双击刚刚下载的文件，打开这一压缩文件



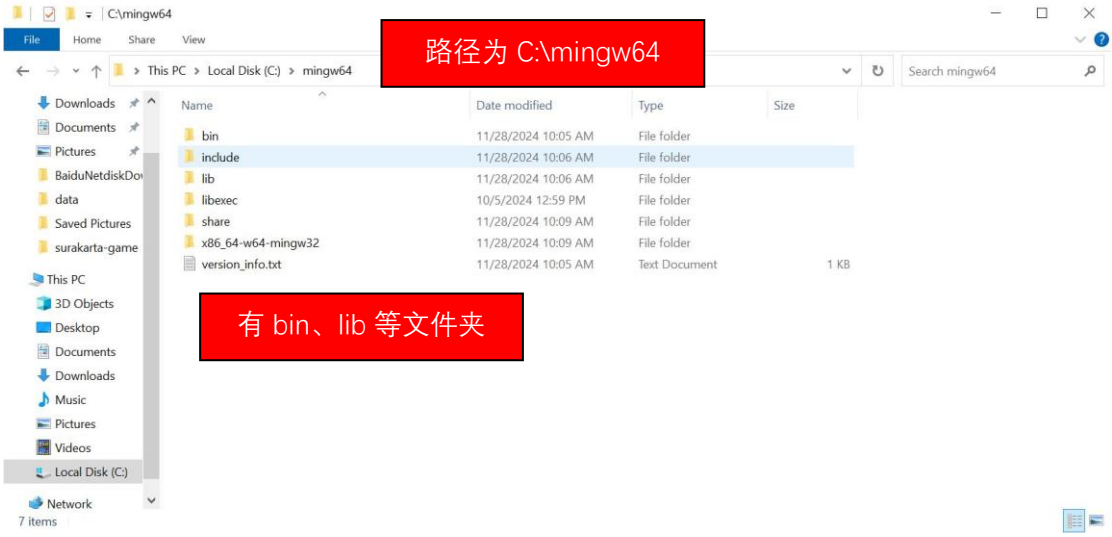


将右侧窗口的 mingw64 文件夹拖拽至左侧窗口。请拖拽至空白处，不要拖拽到“Windows”、“Program Files”等文件夹内。如果提示需要管理员权限，请点击允许。



由于 mingw64 中小文件较多，该解压过程耗时较长，可能需要十到二十分钟，请耐心等待

解压完成后，C 盘根目录下应有 mingw64 文件夹，其中内容类似下图

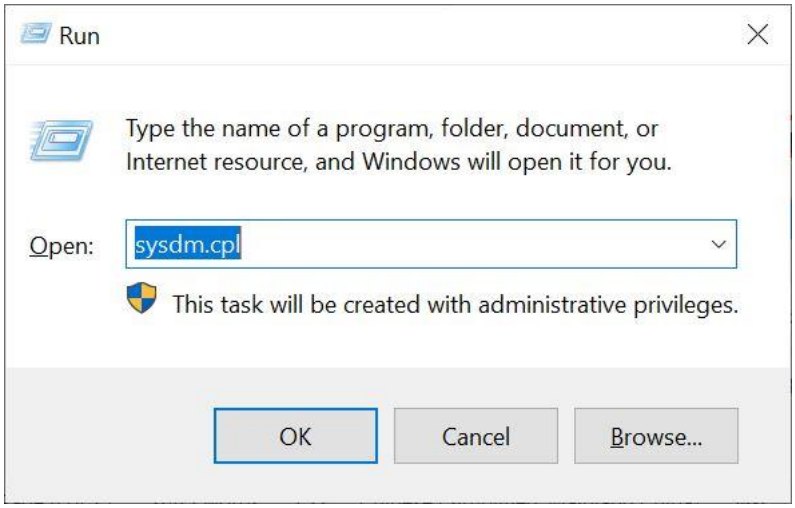


最后，不要忘记删除安装包（文件名类似于 winlibs-x86_64-posix-seh-gcc-14.2.0-mingw-w64ucrt-12.0.0-r2.zip）哦！

三、配置 MinGW-W64 的环境变量

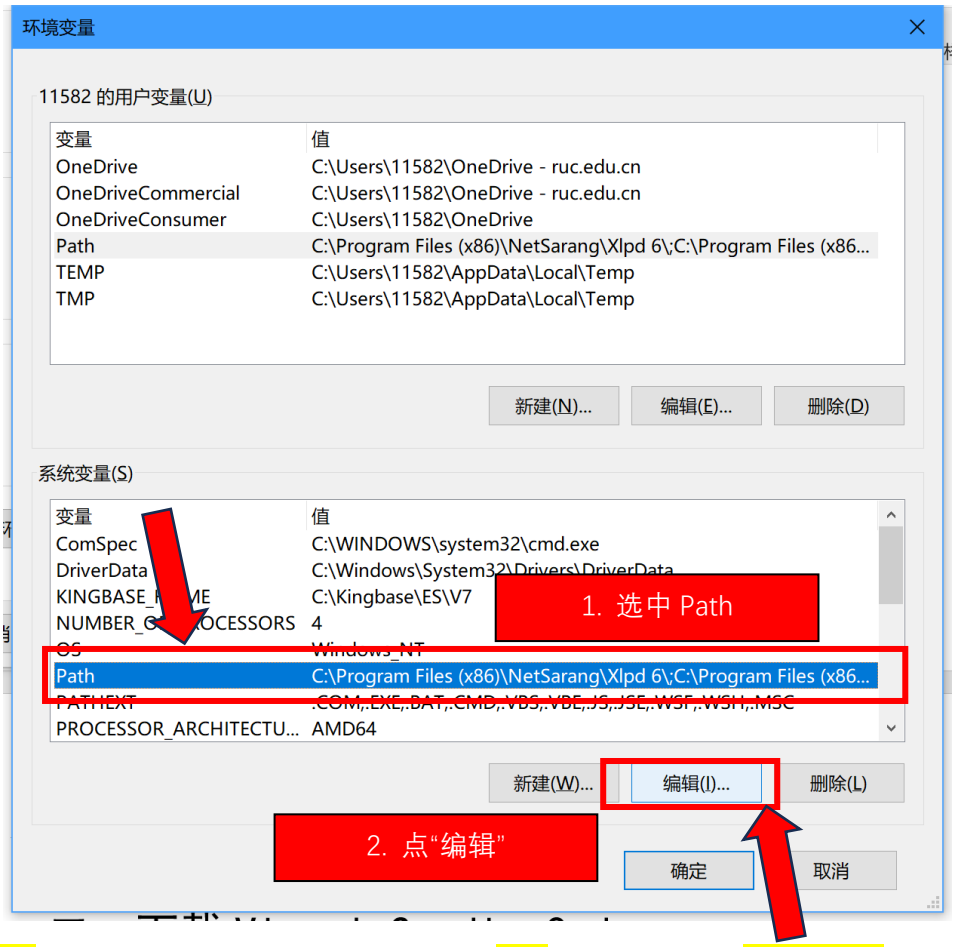
下面将 MinGW-W64 编译器的路径（即 C:\mingw64）添加到系统环境变量：

按 Win+R，输入 sysdm.cpl，回车

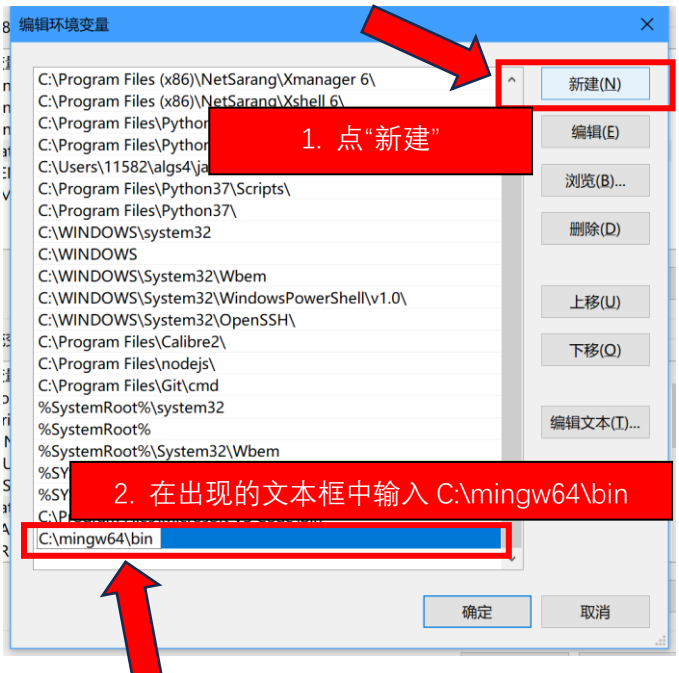


依次点击“高级”“环境变量”：

选中 Path，**点击编辑**：



点击新建，在出现的文本框中输入 C:\mingw64**\bin**，这是编译系统**可执行文件**所在目录。如果“二、解压 MinGW-W64”中解压到了别的路径，请填写 mingw64 目录中 bin 目录的路径



然后依次点击确定，确定，确定关闭上面三个界面。

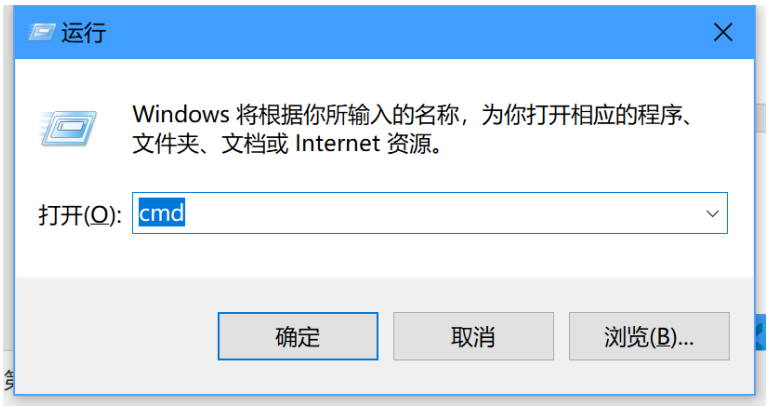
#重要# #重要# #重要#

不要删除现有的设置

此外，如果你的操作系统因为版本不同的原因，不是显示如上图那样的列表下拉框，而是只有一个输入框，那么在此输入框的末尾先输入一个分号(英文输入状态)，再把相应路径添加在它后面。

查看是否成功

按 Win+R 键，出现下面窗口，输入 cmd，按回车键:



在命令行界面输入命令：gcc --version

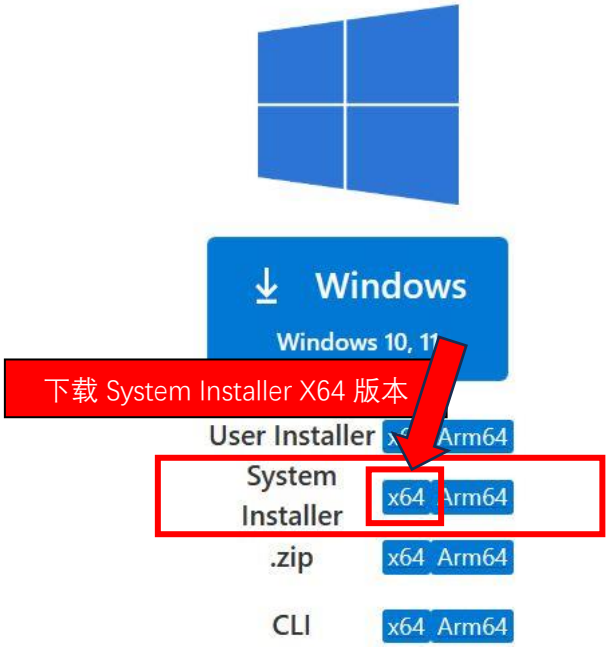
如果输出的内容类似下图，且第一行末尾的版本号（下图中为 14.2.0）与安装包文件名（类似于 winlibs-x86_64-posix-seh-gcc-14.2.0-mingw-w64ucrt-12.0.0-r2.zip）中的版本号一致，则说明环境变量设置成功。

如果显示不是内部或外部命令，也不是可运行程序，证明上面环境变量没有配置好，重新按上述流程再走一遍。

如果显示的版本号不与与安装包文件名（类似于 winlibs-x86_64-posix-seh-gcc-14.2.0-mingw-w64ucrt-12.0.0-r2.zip）中的版本号一致，则说明你的电脑中之前的系统环境变量中有另一个版本的编译器，在配置环境变量界面将 C:\mingw64\bin 上移到第一个即可。

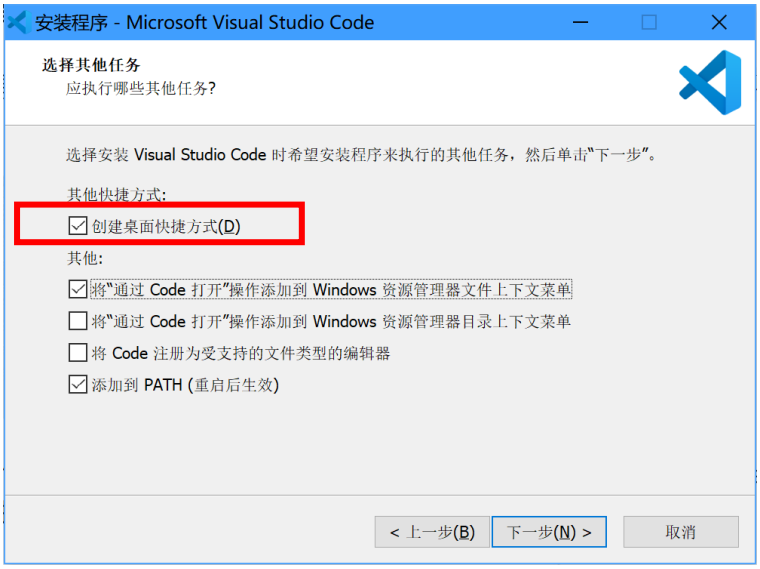
四、下载并安装 VS Code

进入下载界面 https://code.visualstudio.com/Download
下载 Windows 版本的 System installer x64（不要选 User installer）。

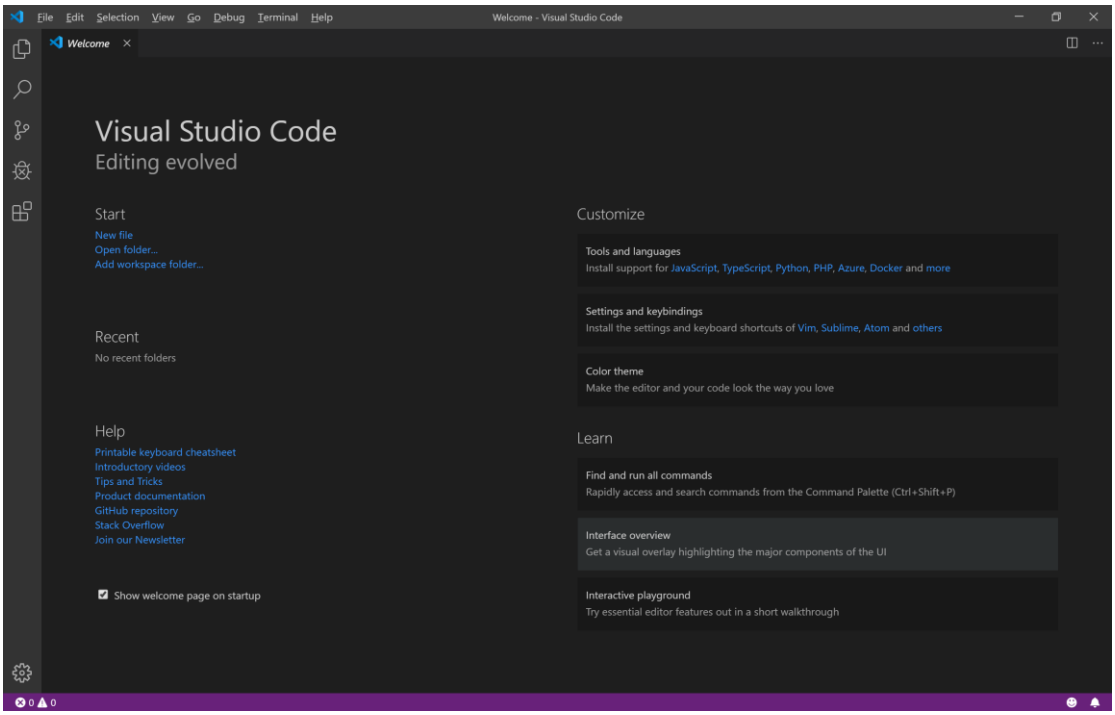


双击安装程序，可勾选“创建桌面快捷方式”。

强烈不推荐勾选“将‘通过 Code 打开’操作添加到 Windows 资源管理器文件上下文菜单”，因为大部分情况下用 VS Code 单独打开一个 C/C++文件无法正常编译、调试，只有打开文件夹时各种配置文件才能生效。这一功能总是带来烦恼，却很少被使用到。



安装完成后打开 VS Code，看到这一界面说明安装成功！欢迎来到 VS Code 的世界



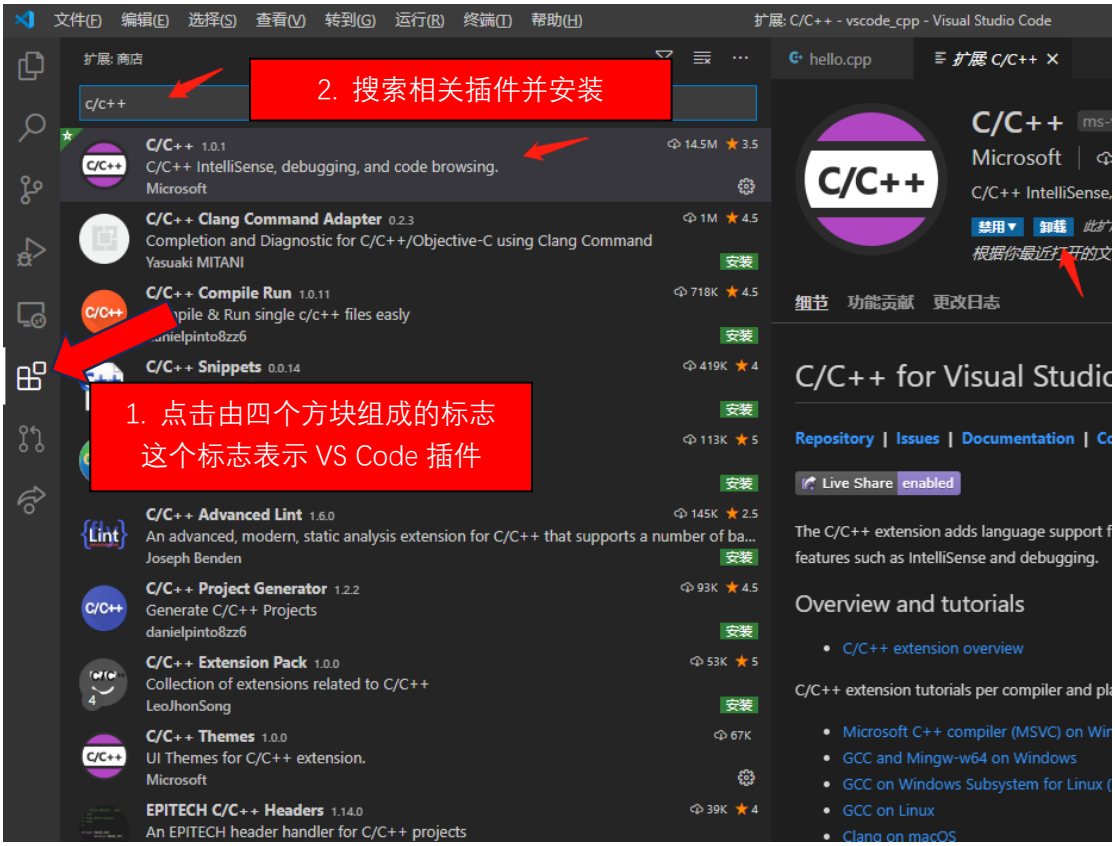
五、安装若干 VS Code 插件并开启自动保存

安装插件

按照下图所示方法安装 (1) C/C++插件 (2) 汉化插件

1) 在搜索框输入 C/C++，点第一个插件，点击 install 安装
2) 在搜索框输入 chinese，找到 Chinese (Simplified) Language Pack for Visual Studio Code 插件，点击 install 安装

安装完**两个插件**后，关闭 VS Code 软件并重新打开。



注：不推荐使用 Code Runner，因为 Fn+F5 或 F5 提供的“调试”功能同样可以被当作运行，而 Code Runner 生成的.exe 文件与.c/.cpp 文件混在一起，非常混乱。

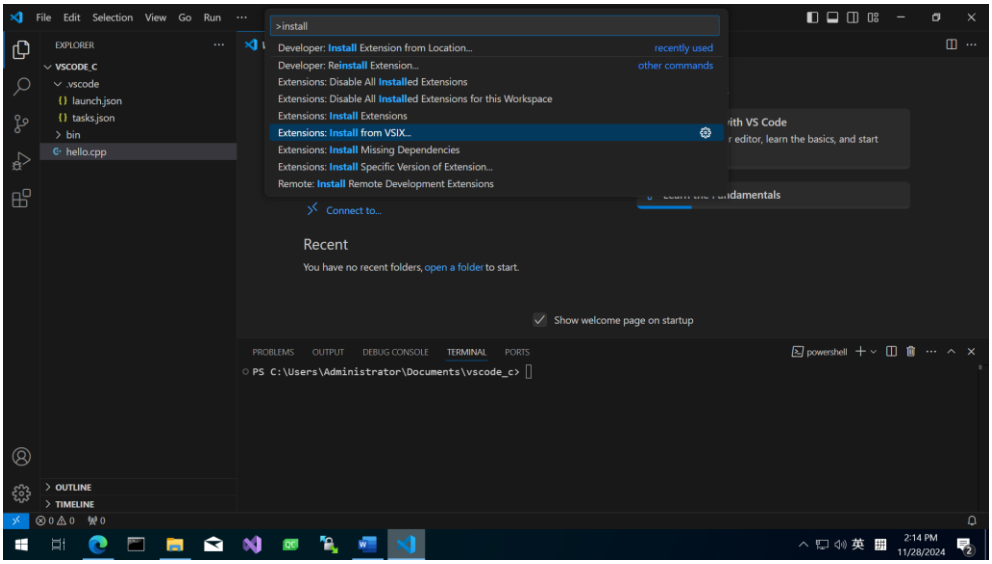
离线安装 VS Code 插件

有时 VS Code 插件的下载会遇到一些问题（这种情况很少见，但并非不存在），这时我们可以离线安装 VS Code 插件。

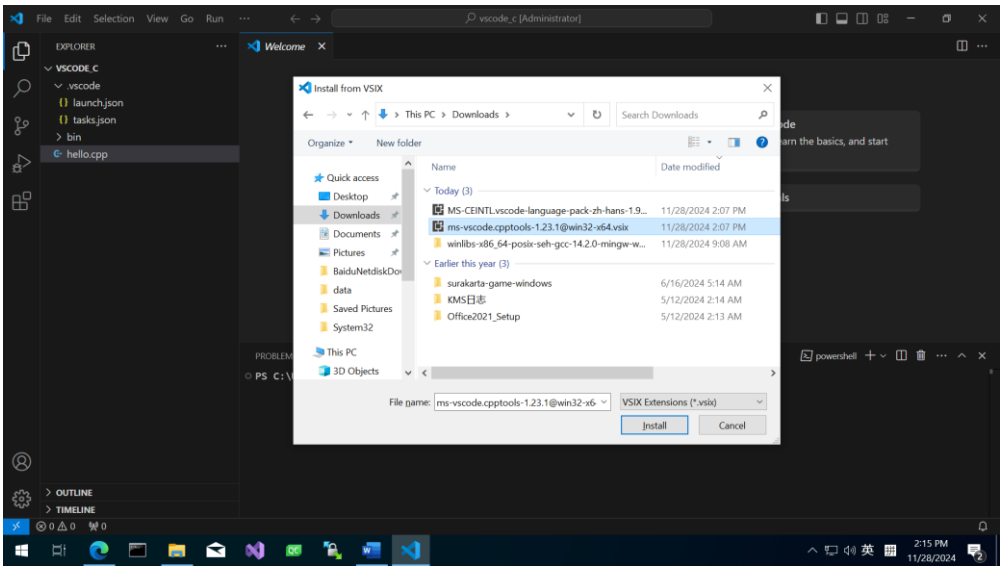
我们需要下载 VS Code 插件的离线安装包。
这些安装包可以从 https://marketplace.visualstudio.com/VSCode 下载得到，如果无法下载，助教可以直接将这些安装包的拷贝发送给大家。

例如，刚刚提及的的 C/C++插件的文件名为 ms-vscode.cpptools-1.23.1@win32-x64.vsix，汉化插件的文件名为 MS-CEINTL.vscode-language-pack-zh-hans-1.96.2024112709.vsix

安装 VS Code 插件离线安装包时，我们按下 Ctrl+Shift+P，输入 install，选中“Extensions: Install from VSIX…”



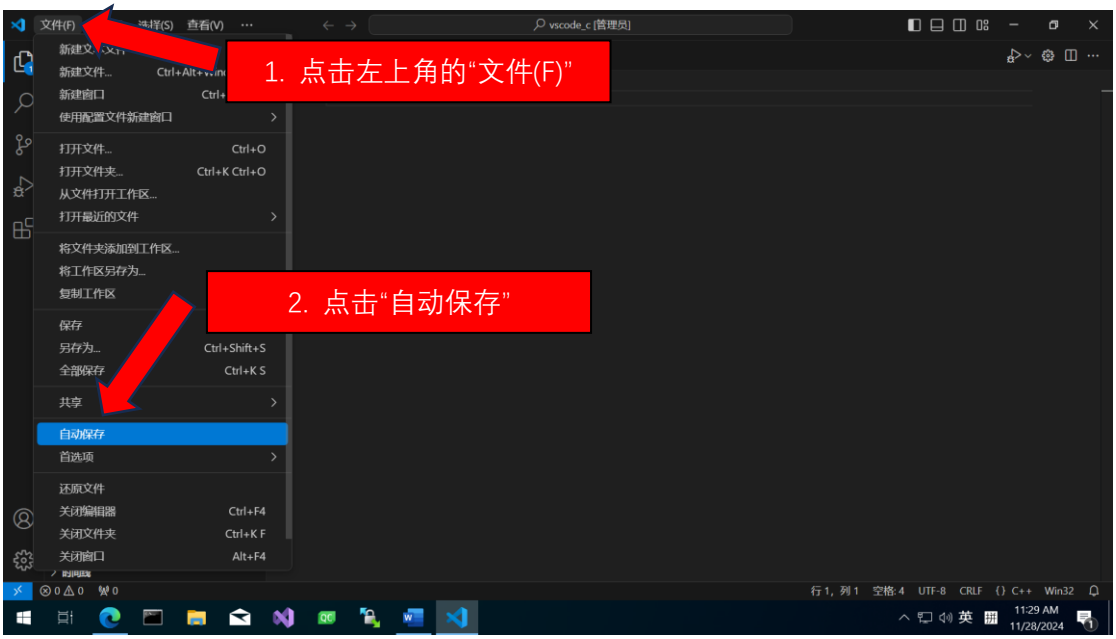
打开需要安装的离线安装包

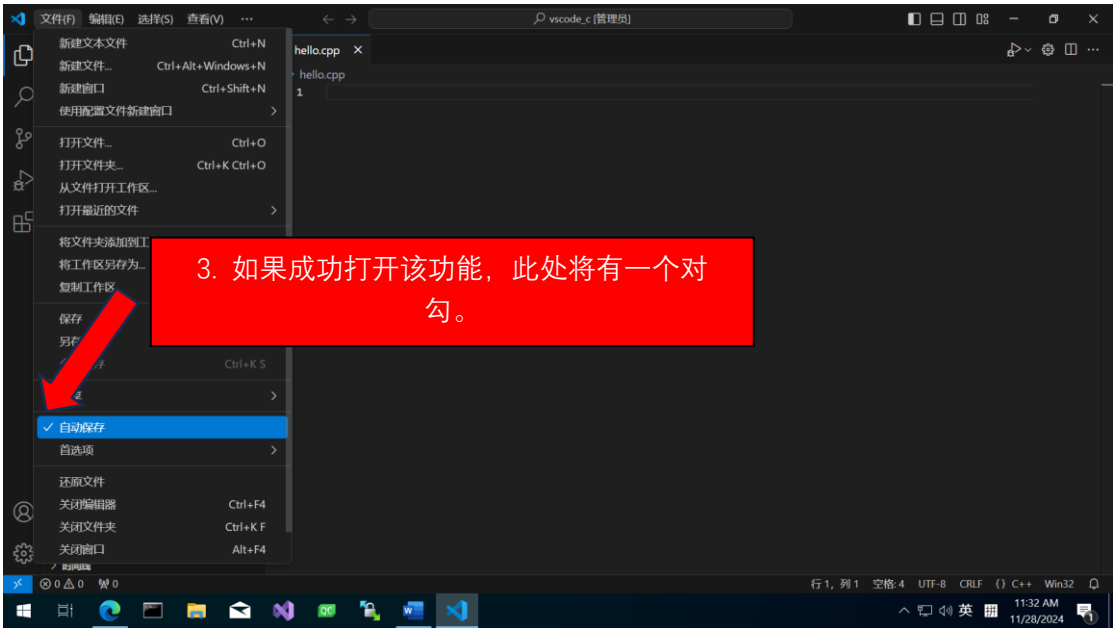


之后耐心等待几十秒，我们就会发现插件安装成功了。

开启自动保存

打开 VS Code 的自动保存功能，这一功能可以免去无穷无尽的烦恼。





六、编写 VS Code 的文件夹配置文件

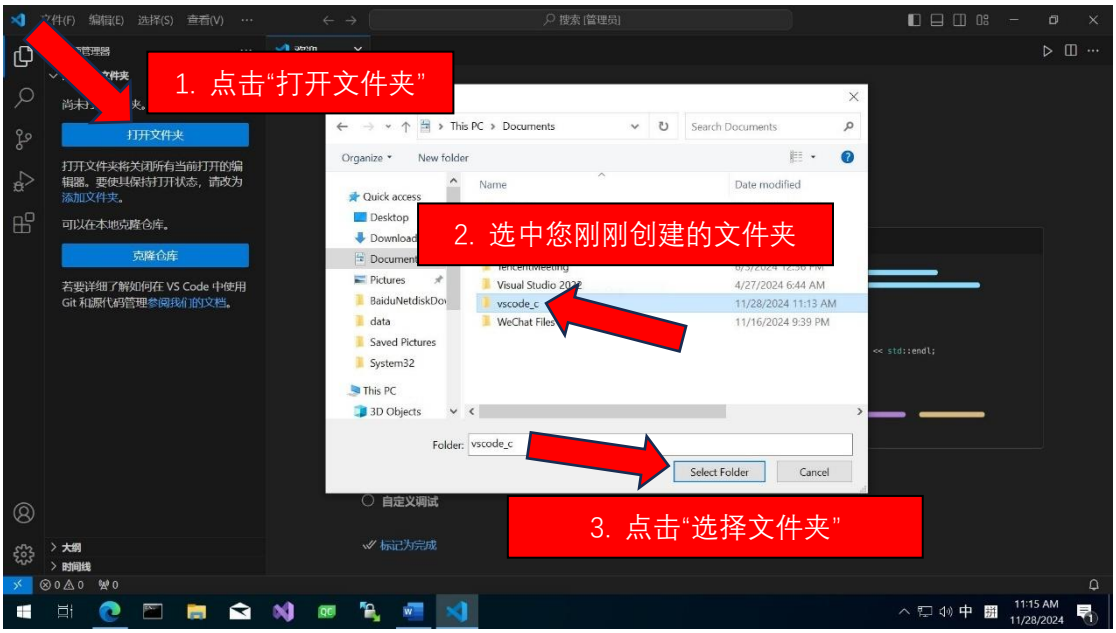
恭喜你！配置 VS Code C/C++环境的任务已经几乎要完成了。

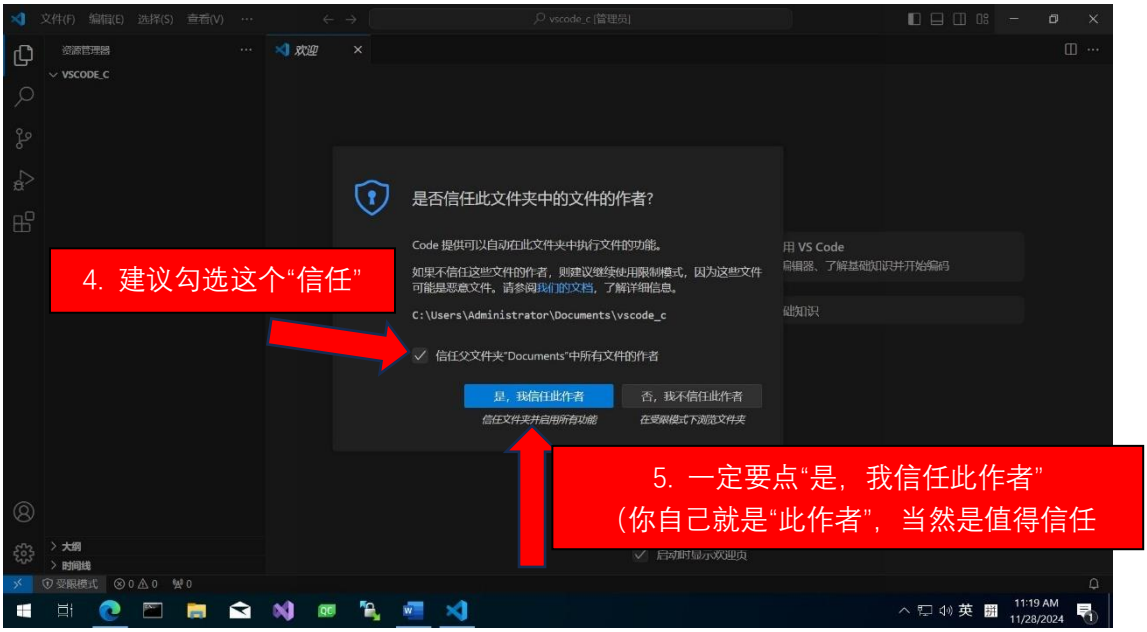
创建 vscode_c 文件夹

请你找一个你喜欢的位置，创建一个文件夹，我们假设这个文件夹的名字是 vscode_c

这个文件夹的路径中不能包含中文（如 C:\Users\某人\Document\vscode_c），也不能包含空格（如 D:\VS Code C）或除“_”“-”外的任何特殊字符（如 D:\vscode.c）

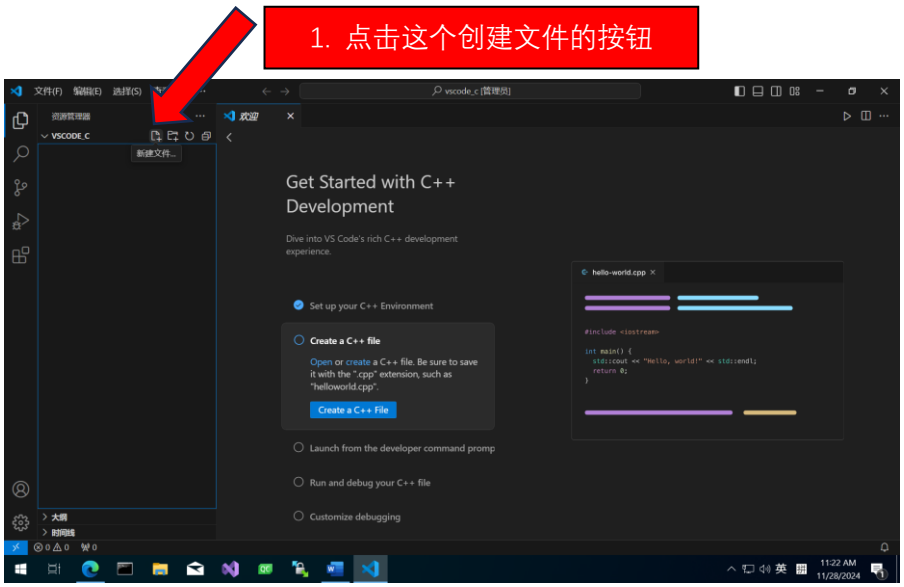
现在，我们用 VS Code 打开这一文件夹。

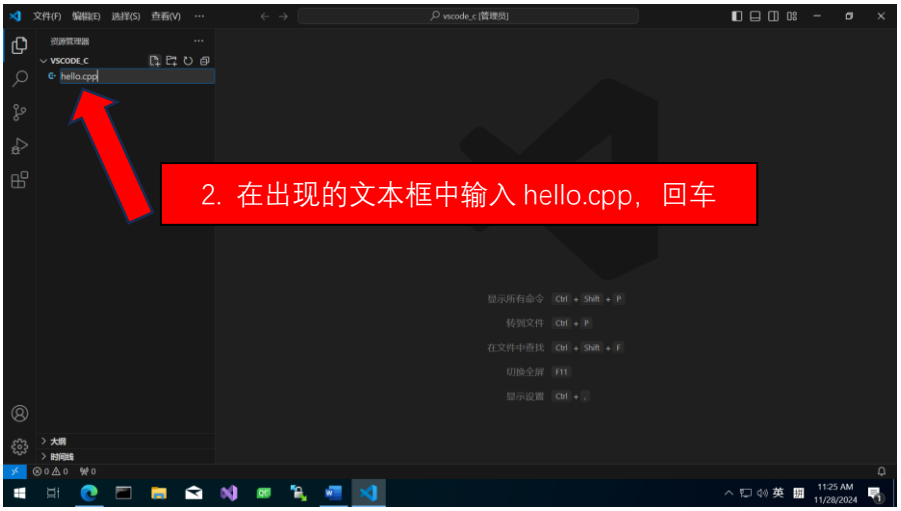




创建 hello.cpp

虽然你可能只需要 C 语言环境，但只要 C++环境能用，C 语言肯定也能用，但 C 能用 C++不一定能用，所以请创建 hello.cpp 而非 hello.c





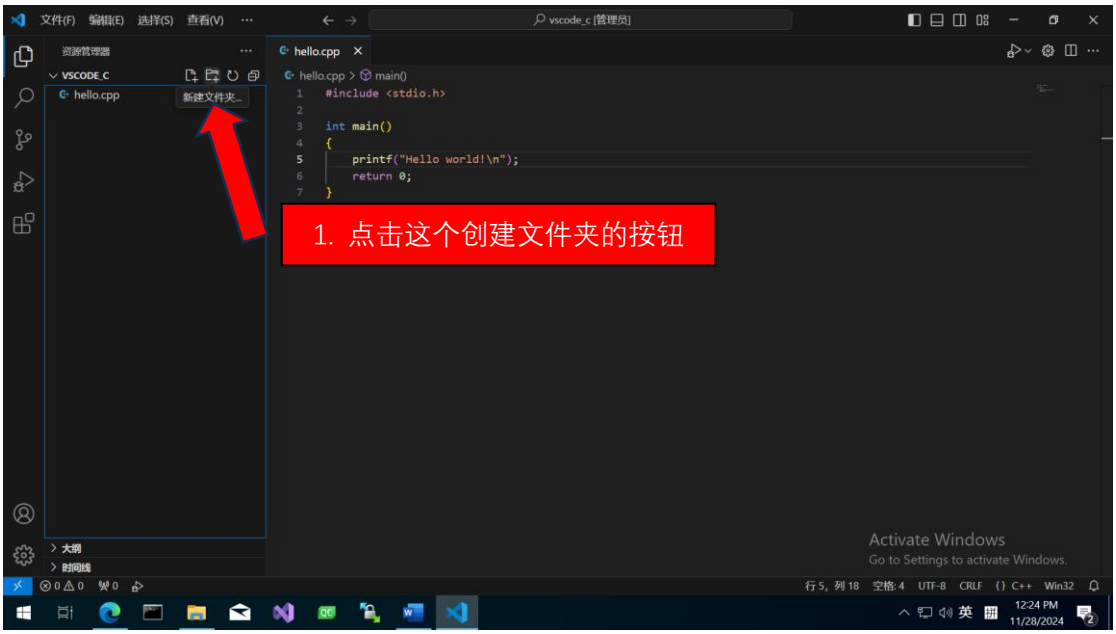
在 hello.cpp 中粘贴如下代码

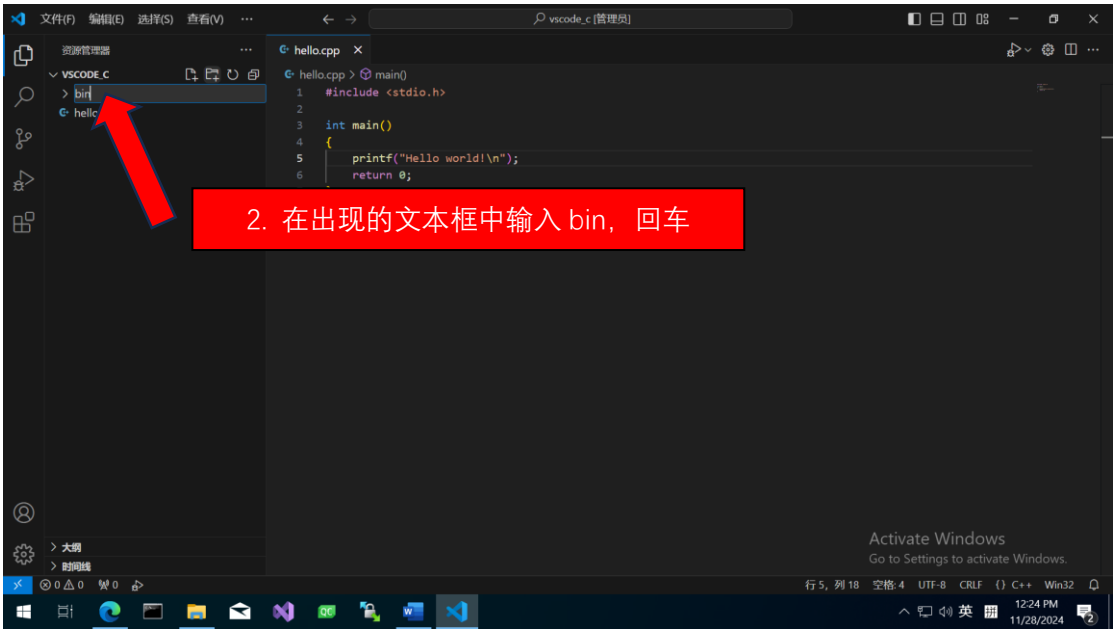
```
#include <stdio.h>

int main()
{
	printf("Hello world!\n");
	return 0;
}
```

这既是一段标准的 C 语言代码，也是一段标准的 C++代码。

创建 bin 文件夹

接下来你将创建一个名为“bin”的文件夹，这个文件夹未来将存放所有编译产生的.exe 文件

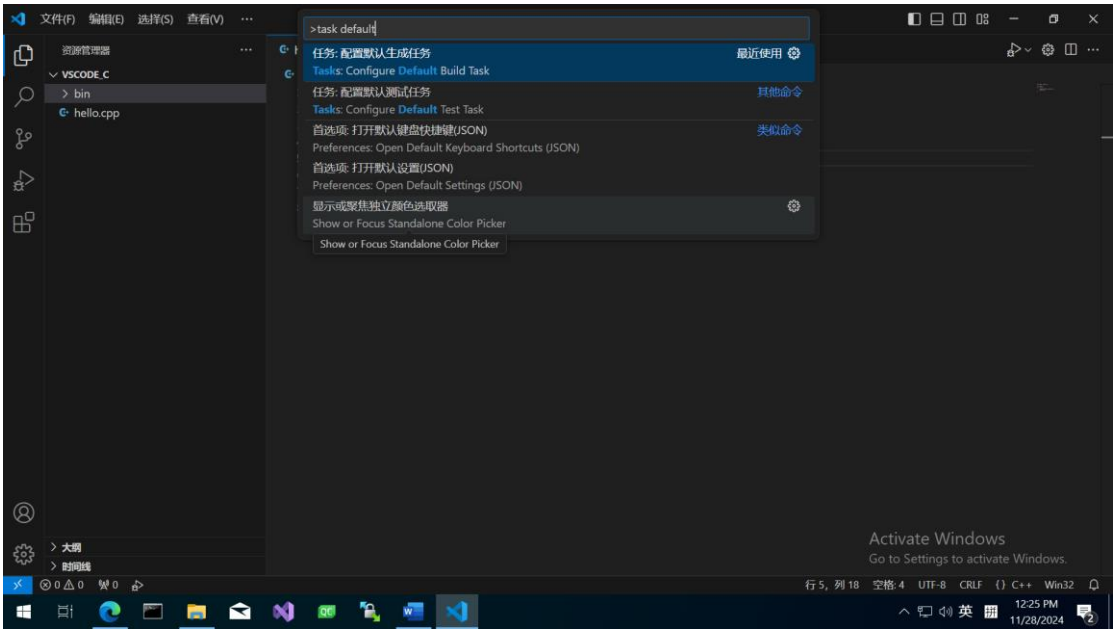




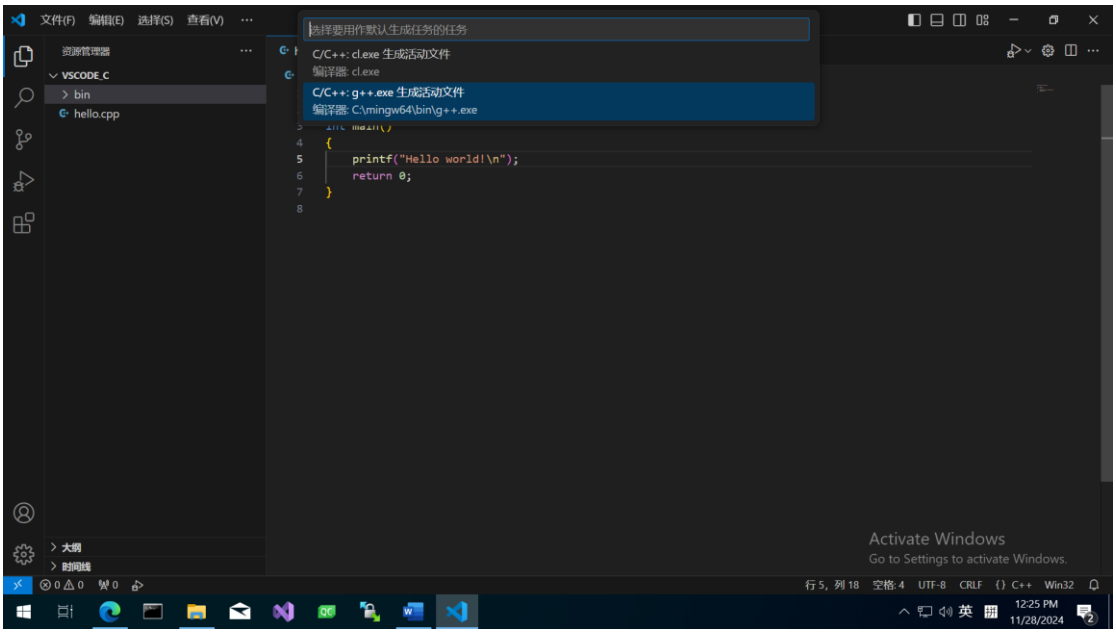
生成并修改 tasks.json

这一步非常重要，因为这一步中的 tasks.json 的内容直接决定了你的代码能否被正常编译。

按下 Ctrl+Shift+P，输入 task default，找到“任务：配置默认生成任务”，回车
注意，不要错选为“任务：配置任务”，这两个是不一样的



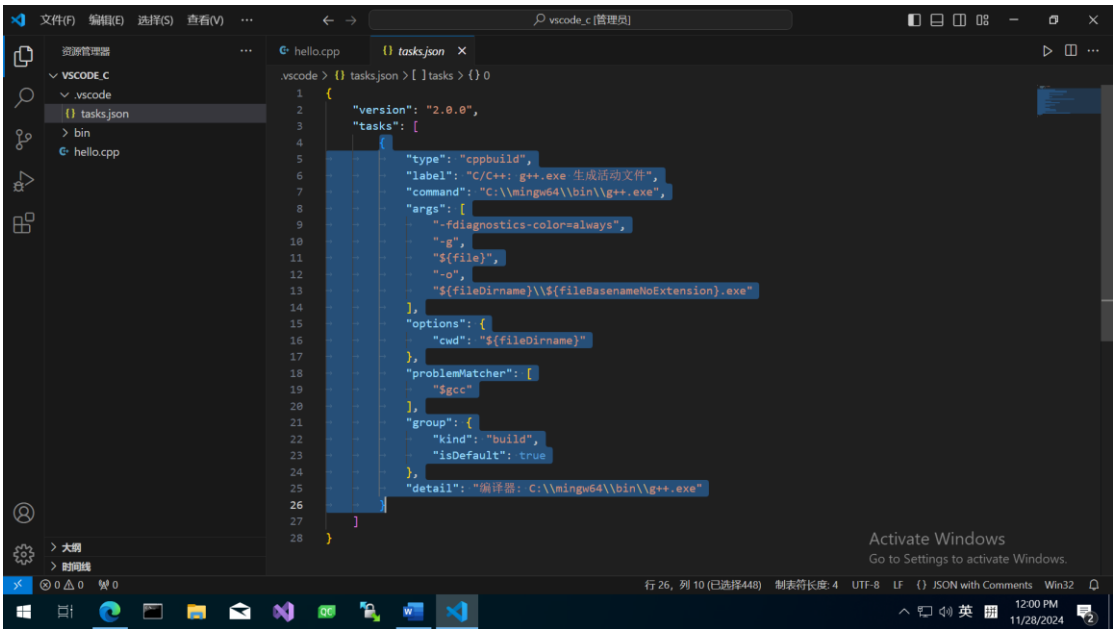
在显示的若干选项中，找到路径为 C:\mingw64\bin\g++.exe 的选项。



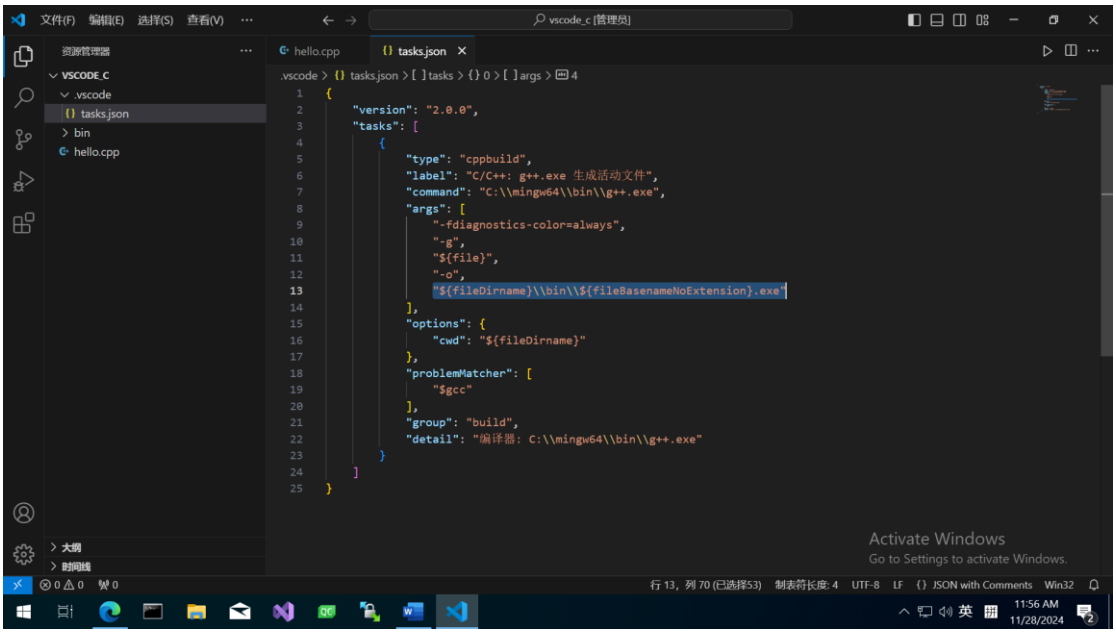
如果“二、解压 MinGW-W64”中解压到了别的路径，请根据实际情况选择正确的 g++.exe

如果这里不显示您刚刚安装的 MinGW-W64（也就是 g++.exe），则说明环境变量配置有问题，请重新将“三、配置 MinGW-W64 的环境变量”完整地再做一遍

现在，我们得到了下图所示的 tasks.json 文件

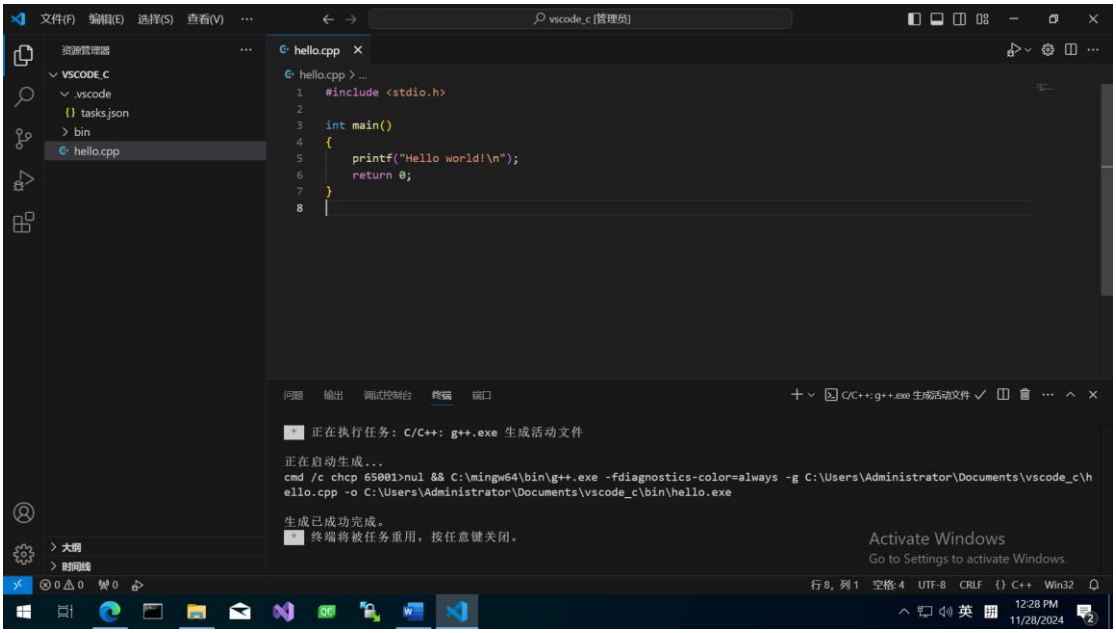


将 "${fileDirname}\\${fileBasenameNoExtension}.exe"
修改为 "${fileDirname}\\bin\\${fileBasenameNoExtension}.exe"
以使得生成的.exe 文件位于 bin 文件夹中



关闭 tasks.json，重新使得当前文件为 hello.cpp。

按下 Ctrl+Shift+B，如果看到下方“**终端**“栏中有“**生成已成功完成**”字样，则说明 tasks.json 配置无误。

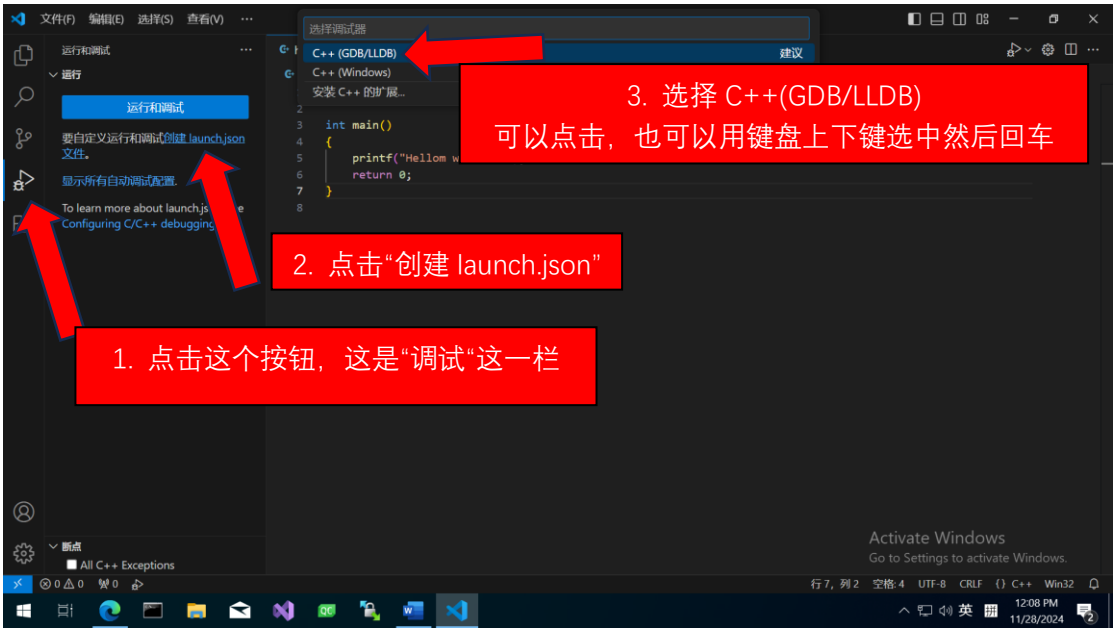


如果发生了错误，看到“**生成已完成，但出现错误**”等字样，请删除 tasks.json，重新把本小节“生成并修改 tasks.json”再操作一遍

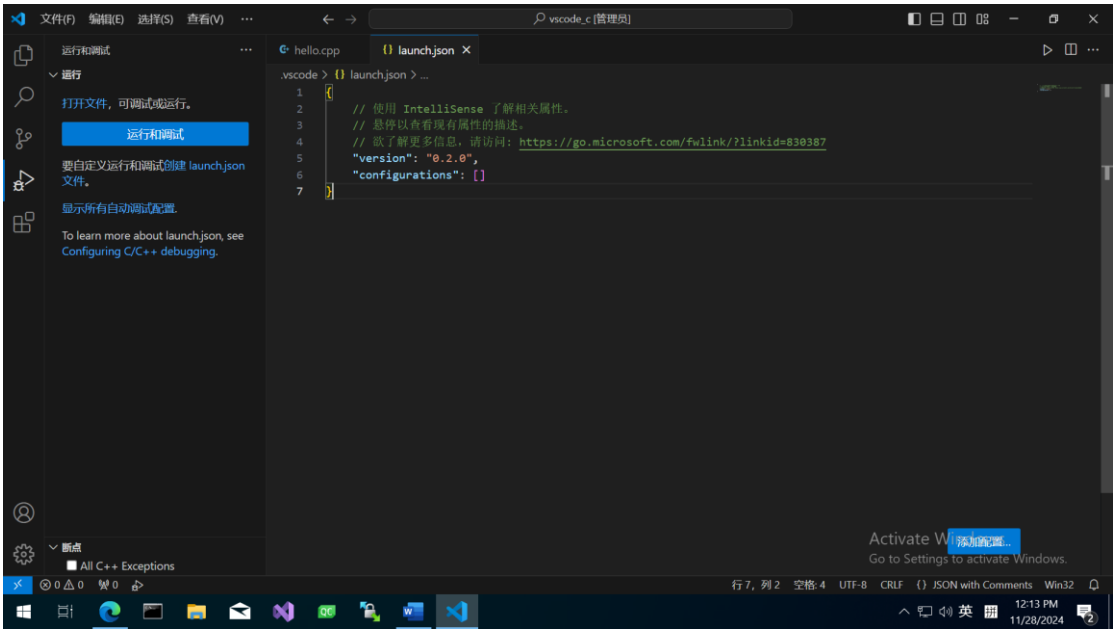
生成并修改 launch.json

这一步同样非常重要，因为 launch.json 的内容直接决定了你是否可以正常调试代码。

点击左侧的调试那一栏，然后点击“创建 launch.json 文件”，在顶部弹出的列表中点击“C++(GDB/LLDB)”这一项

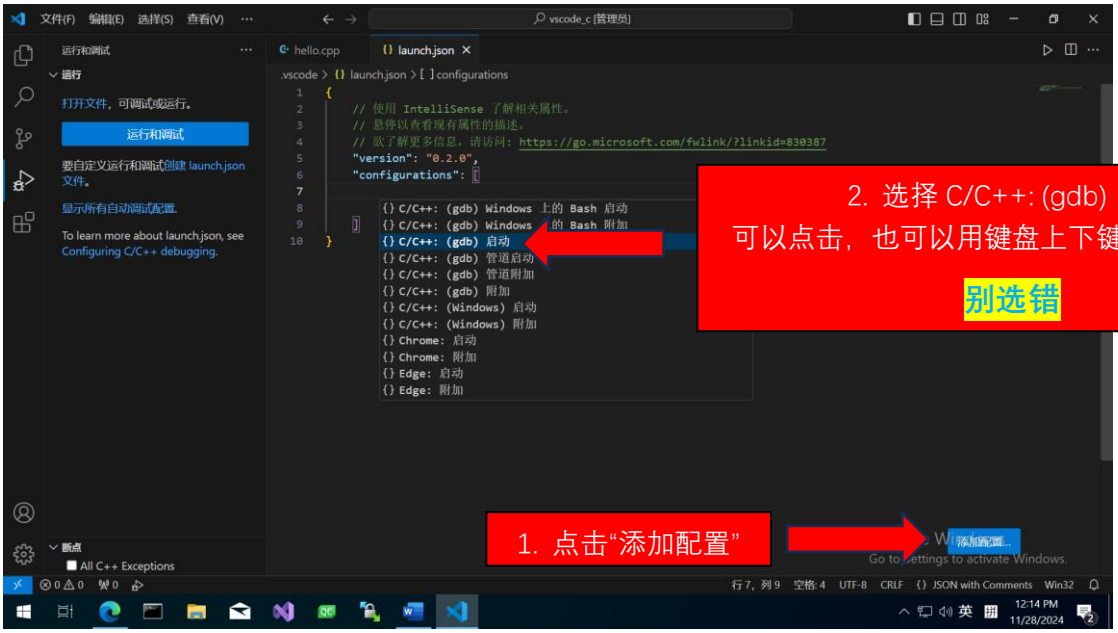


之后你会得到这样的一个 launch.json



点击“添加配置”，然后选择“C/C++: (gdb) 启动”

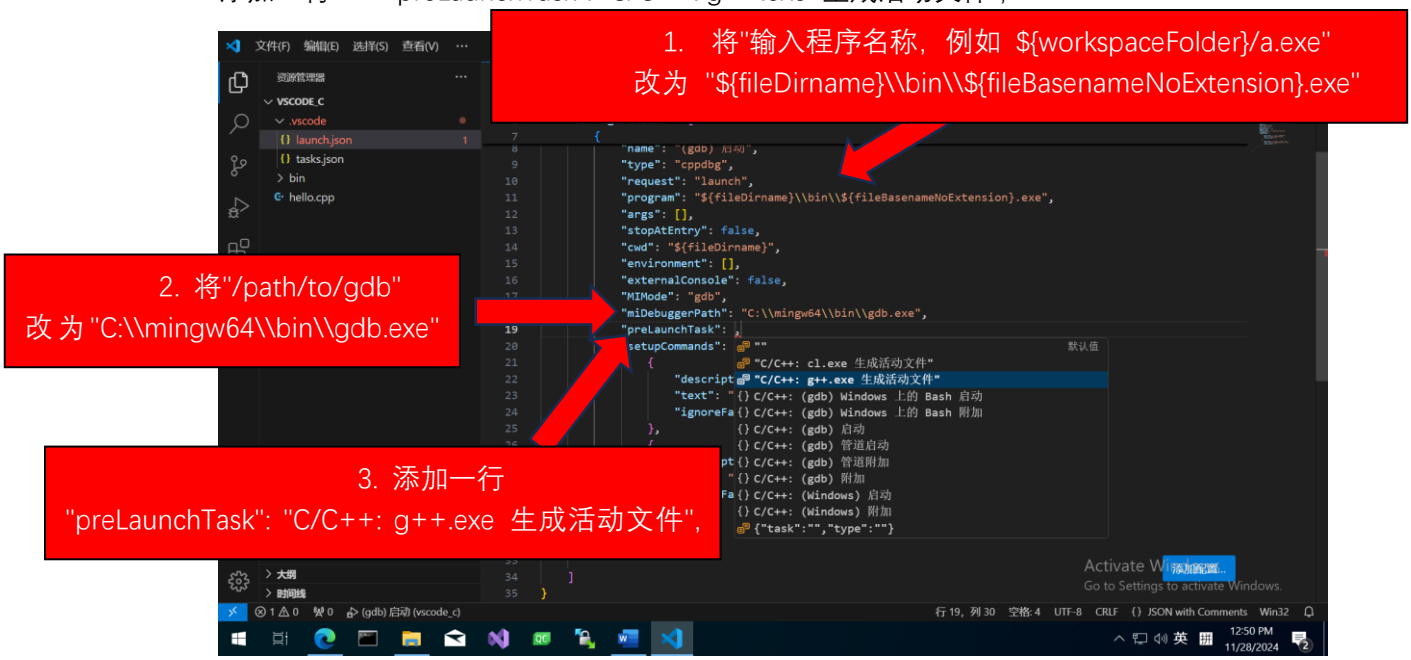
注意，千万不要选择“C/C++: (gdb) Windows 上的 Bash 启动”，尽管它在第一项。



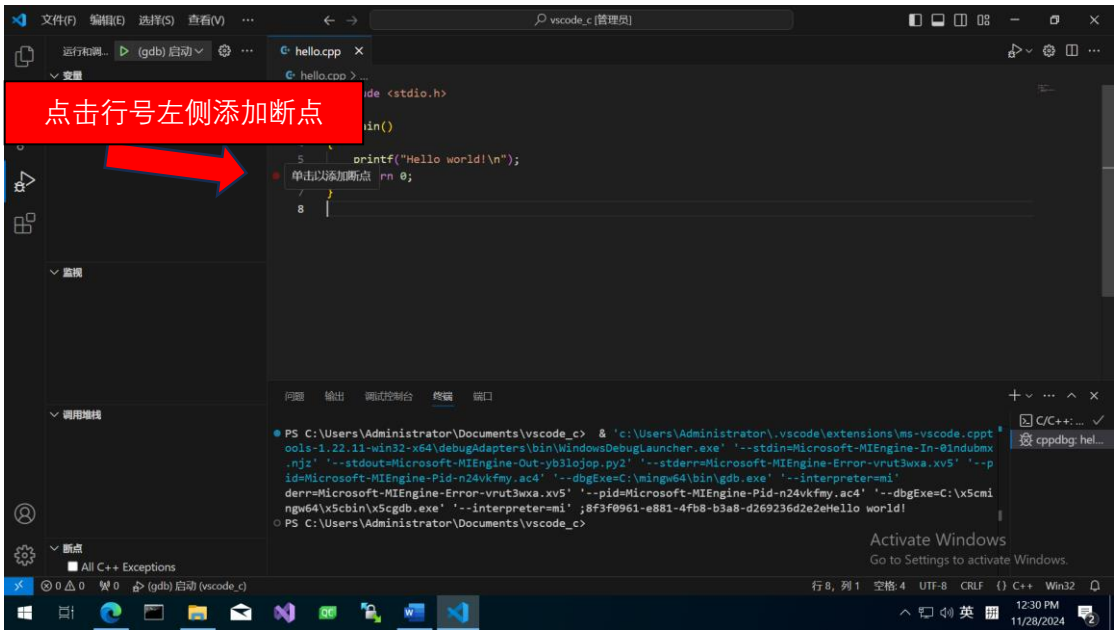
将　　"输入程序名称，例如 ${workspaceFolder}/a.exe"
改为　　"${fileDirname}\\bin\\${fileBasenameNoExtension}.exe"　　（和 tasks.json 中一致）

将　　"/path/to/gdb"
改为　　"C:\\mingw64\\bin\\gdb.exe"
（如果“二、解压 MinGW-W64”中解压到了别的路径，请根据实际情况填写 gdb.exe 路径）

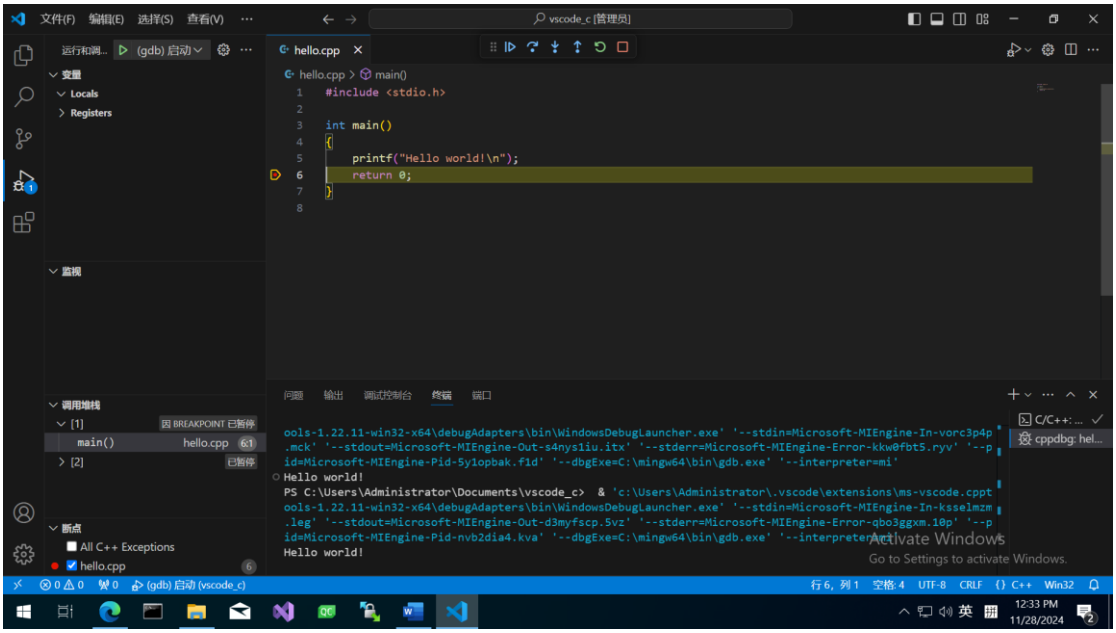
添加一行　　"preLaunchTask": "C/C++: g++.exe 生成活动文件",



在 hello.cpp 的 return 0; 处添加断点



按下 Fn+F5 或 F5（如果一个不行就试一试另一个），如果看到下图所示的暂停状态，则说明调试功能已经正常！恭喜你，你成功完成了 VS Code C/C++环境的配置！

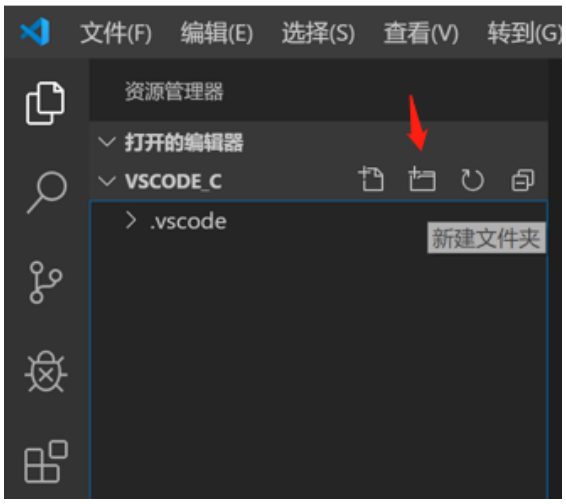


如果出现了报错或者卡死什么事情都不发生，请删除 launch.json，重新把本小节（生成并修改 launch.json）的内容再做一遍。

七、使用 VS Code

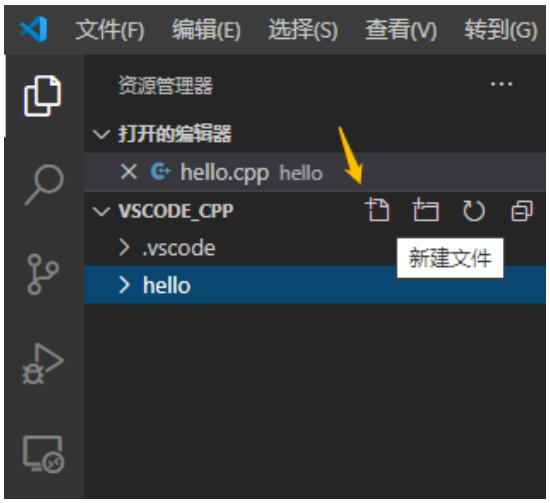
创建文件夹

参见“六、编写 VS Code 的文件夹配置文件 —— 创建 bin 文件夹”



创建文件

参见“六、编写 VS Code 的文件夹配置文件 —— **创建 hello.cpp**”

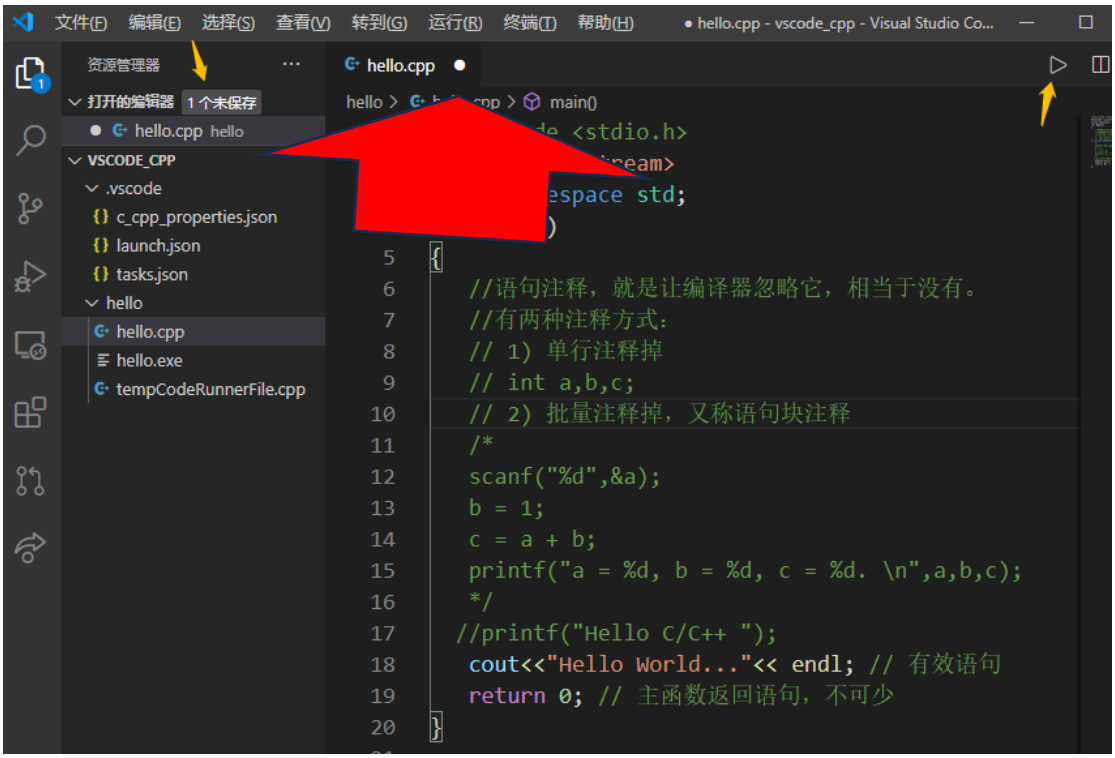


注意：扩展名必须正确，.c 表示 C 语言；.cpp 表示 C++语言。如果你输入代码时，字符全是白的，没有带颜色的文字，说明扩展名错了。 要选中文件夹名 hello 再创建源程序文件。

保存文件与自动保存

如果看到类似下图所示的文件名旁边的原点，说明你没有打开自动保存功能

请参考“五、安装若干 VS Code 插件并开启自动保存 —— 开启自动保存”打开自动保存功能

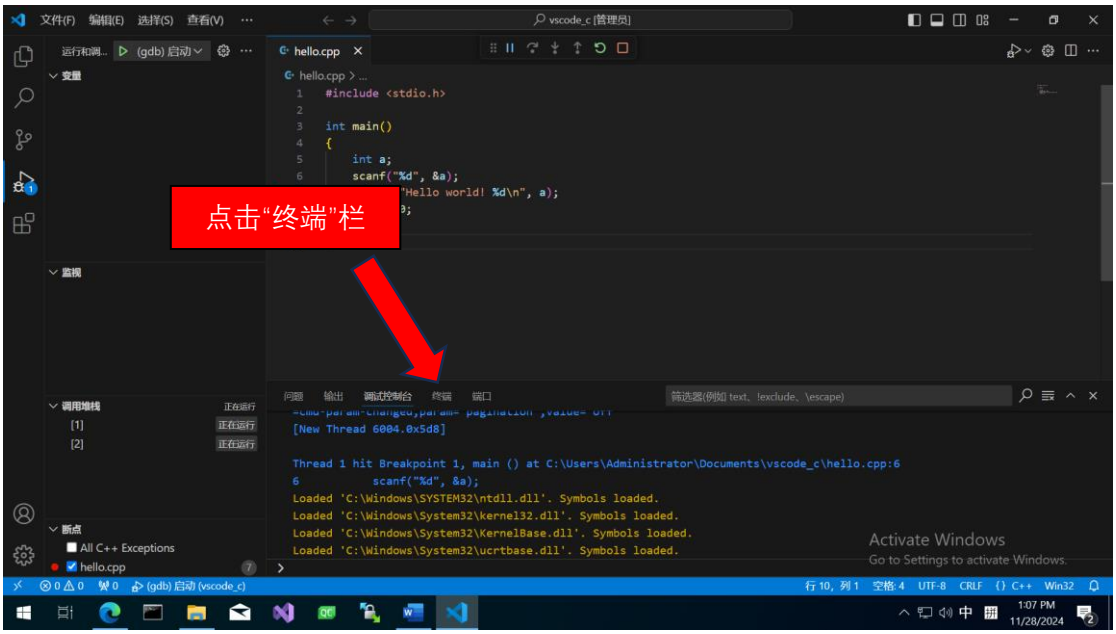


注释掉代码

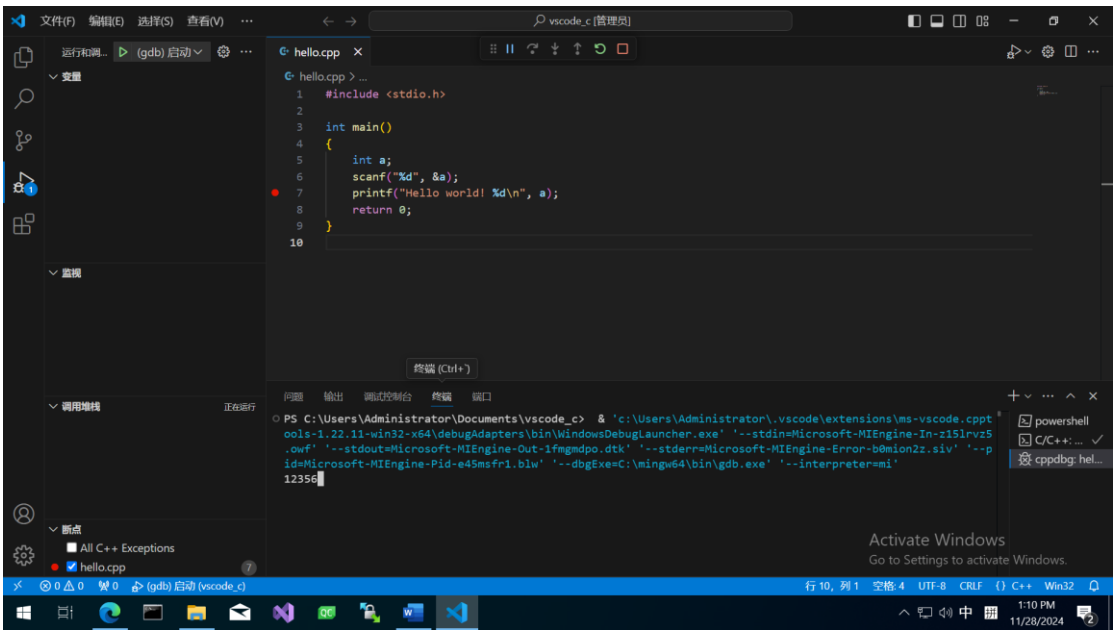
暂时不需要的代码，可以按 Ctrl+/注释掉。 上图绿色代码行就是被注释掉的代码

运行并向终端输入内容

按下 Fn+F5 或 F5 调试代码，注意，程序刚启动时并没有自动显示终端。请点击“终端”栏。



之后便可以向终端输入文本了。



注意：如果看不到小白方块（光标），说明光标仍在编辑区，需要点击终端界面一下，然后才能在终端中输入数字：

看到小白方块，表示终端获得了输入焦点，在键盘上打字才能输入到终端里。

写给助教的话

此处我们假设“助教”为程序设计 I 这门课程的助教，不过考虑到不只有程序设计 I 需要使用 VS Code C/C++环境，也许不只有程序设计 I 的助教会向同学们发放本手册。无论是哪种情况，下面的事情十分重要：

建议在发放本手册时将.docx 格式的文档和.pdf 格式的文档一同发放。这是因为在手机和平板上，.pdf 文件不会有任何格式问题，而.docx 文档中的图示经常会遇到错位等问题。

助教请准备好如下四个文件并通过微信群等手段向同学们发放这四个文件，以免去网络问题带来的无穷烦恼：

1. MinGW-W64 x64 without LLVM/Clang/LLD/LLDB Zip archive
2. VS Code System Installer
3. VS Code C/C++插件的 VSIX 离线安装包
4. VS Code 汉化插件的 VSIX 离线安装包

笔者修订本手册时，四个文件的文件名分别是

1. winlibs-x86_64-posix-seh-gcc-14.2.0-mingw-w64ucrt-12.0.0-r2.zip
2. VSCodeSetup-x64-1.95.3.exe
3. ms-vscode.cpptools-1.23.1@win32-x64.vsix
4. MS-CEINTL.vscode-language-pack-zh-hans-1.96.2024112709.vsix

这四个文件的下载方式分别可以在本手册的以下小节找到：

1. 一、下载（或通过助教获取）MinGW-W64
2. 四、下载并安装 VS Code
3. 离线安装 VS Code 插件
4. 离线安装 VS Code 插件

在笔者修订本手册时，除了第一个文件国内下载非常慢外，其余三个文件的下载均是正常的。但考虑到曾经发生过 VS Code 官网被 DNS 污染的情况，其余三个文件的下载不一定总是正常的。助教可以视当时的网络连通性情况决定发放哪些文件。

除此之外，助教应该尽量鼓励同学们自己阅读手册，跟随手册完成环境的配置。如果同学们总是在手册的某个步骤处遇到问题或者不知道该如何操作，可以考虑适当修订手册。

总之理想情况下，同学们只需要手册就可以完全解决 VS Code C/C++环境问题。这当然是一个宏大的愿望，但手册最大的特点时可以反复的分发和使用。假如手册真的可以完备到那样的程度，那未来助教在配环境方面的任务将减少很多，同学们在配置 VS Code C/C++环境这件事情上浪费的时间也会少很多。

后记

如果你看到了这个后记，你要么是一个很有耐心的人，要么是一个充满好奇的人，要么说明你的 VS Code C/C++环境遇到了一些问题，而你翻遍整本手册却没有找到答案。

如果很不幸是后者，请不必感到烦恼和忧愁，因为你的身边还有无数与你同行的人。你可以联系课程的助教，寻求助教的帮助。也可以找到班里对配环境比较熟悉的同学，看看有没有遇到过一样问题的同学可以传授若干经验于你。也可以寻求计算机协会的帮助。

~~掺杂一点私货，计算机协会超厉害的哦！计算机协会是你校一个历史悠久的学生组织，至今已有超过 30 年的历史，所以我相信不管你是在什么时候阅读本手册，计协一定还坚强地存在着。~~计协里热心的学长会帮助你解答环境配置相关的问题，也可以提供线下的计算机义诊（在修订本手册时，该活动于每周三下午 2：00-5：30 举办），帮助你解决环境问题。请向助教以及身边的同学询问计协相关的信息，从而获得学长们的帮助。

VS Code 环境配置并不是一个很有技术含量的任务，但也并非是一个简单的任务。本手册尽可能的包含了更多的细节，但每个人的电脑环境不一样，有些问题是独特的，也或许有些很常见的问题没有被本手册考虑到。而且 Windows、VS Code 以及 MinGW 本身也处于不断的升级当中，也许以后一些步骤的具体做法将发生变化。所以本手册未来依旧是需要修订的，而这修订，就需要靠屏幕前的你了。

欢迎各位同学对本手册进行进一步的修订！请联系课程助教或者计算机协会如果你想向本手册加入或修改任何内容！你的工作将使许多人受益。

第三版修订者 李甘
Nictheboy Li <nictheboy@outlook.com>
于 2024 年 11 月 28 日