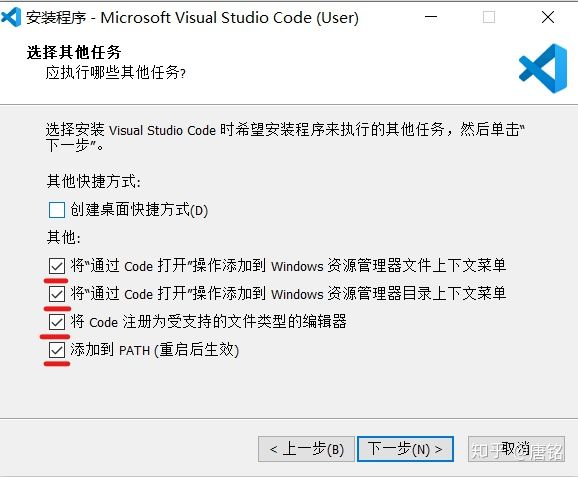
# **windows 10上使用vscode编译运行和调试C/C++**

主要讲如何在vscode上实现**编译**/**运行**/**调试**C以及C++，如果是初学者，就请完全按照文章的步骤进行，**如果觉得图片不够清晰，点击一下图片会显示清晰的大图**

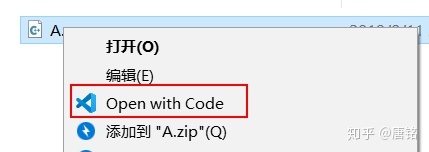
**[知乎连接](https://zhuanlan.zhihu.com/p/77645306)**

## **一、前期准备**

1. 首先在[微软官网](https://code.visualstudio.com/)下载并安装好visual stdio code，建议在安装时将选项全勾选上（注意:超链接打开请Ctrl+鼠标左键）



勾选后在文件和文件夹的右键菜单中会多出 Open with Code这个选项，也就是用vscode打开当前文件/文件夹

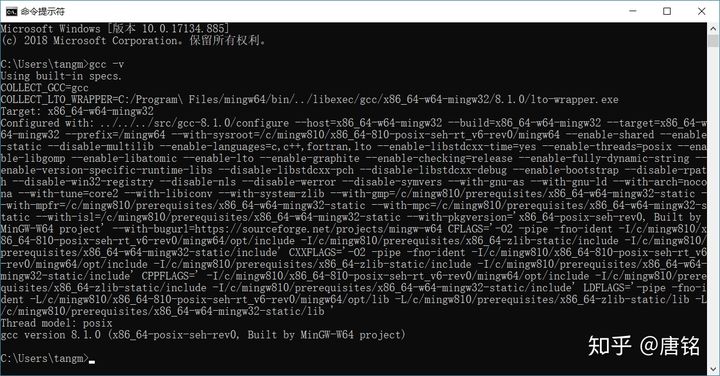


1. 因为vscode只是一个代码编辑器，没有自带有C/C++的编译器，因此首先需要安装一个C/C++编译器并且设置环境变量，这里使用[mingw-w64](https://link.zhihu.com/?target=https://sourceforge.net/projects/mingw-w64/files/mingw-w64/" \t "https://zhuanlan.zhihu.com/p/_blank)。因为网络问题mingw-w64很难下载，建议离线下载然后手动添加环境变量

离线下载的地址：[Mingw-w64](https://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win64/Personal%20Builds/mingw-builds/8.1.0/threads-posix/seh/x86_64-8.1.0-release-posix-seh-rt_v6-rev0.7z/download)

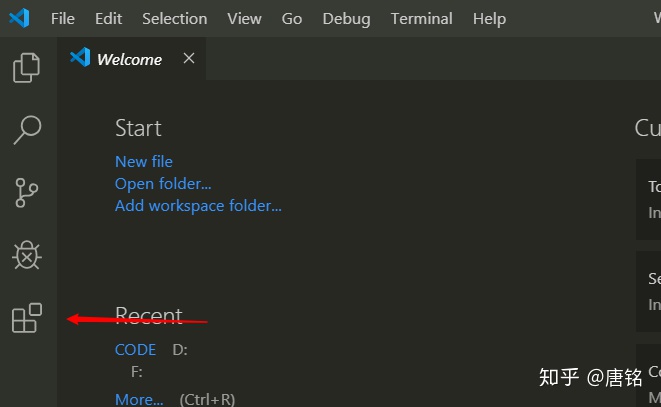
**为节省篇幅，具体的细节参见**：[安装流程](https://www.cnblogs.com/TAMING/p/9945389.html)

此时我们要验证一下mingw是否可用，**打开cmd输入gcc -v**，出现版本信息就表明mingw是可用的，如果未出现版本信息则需要再检查一下，如果这里出现问题，后面全部无法进行

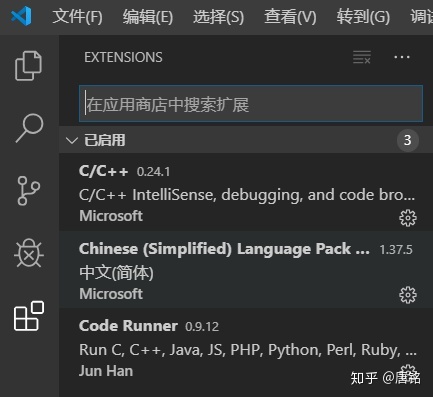


1. 安装必要的插件，打开vscode，点击左面竖排第五个按钮，搜索并安装上如下三个插件

* chinese(simplified)
* C/C++
* Code Runner



安装好以后重启一下vscode，界面切换为中文，为了避免麻烦，暂时不要安装其他插件



## **二、编译运行程序**

编译运行程序介绍两种方法

1. 使用之前安装好的Code Runner插件一键编译运行程序
2. 打开vscode的集成终端使用命令行编译和运行

**点击：文件>新建文件**，出现一个空白文档，将下面的代码复制粘贴到空白文档中

**#include** <stdio.h>**int** **main**(){

**char** name[100];

printf("What is your name?\n");

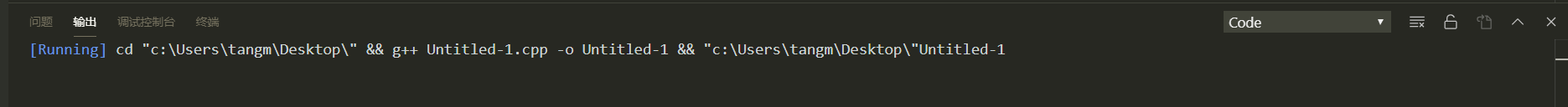
scanf("%s",name);

printf("Hello,%s,nice to meet you!\n",name);}

然后按Ctrl+S**将文件后缀名改为.c或.cpp**后保存到电脑上，然后点右上角的三角形按钮



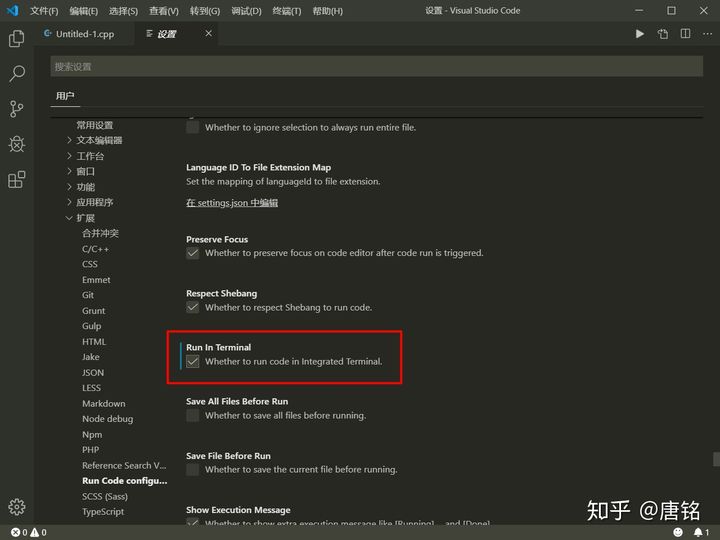
但现在会出现下面的情况



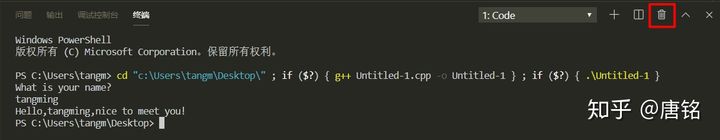
程序段中使用了scanf函数从键盘读取一串字符，这时无法输入数据，也无法结束程序，需要重启一下vscode

然后依次打开：**文件>首选项>设置>用户>拓展>Run Code Configuration**  
找到**Run In Terminal**勾选上



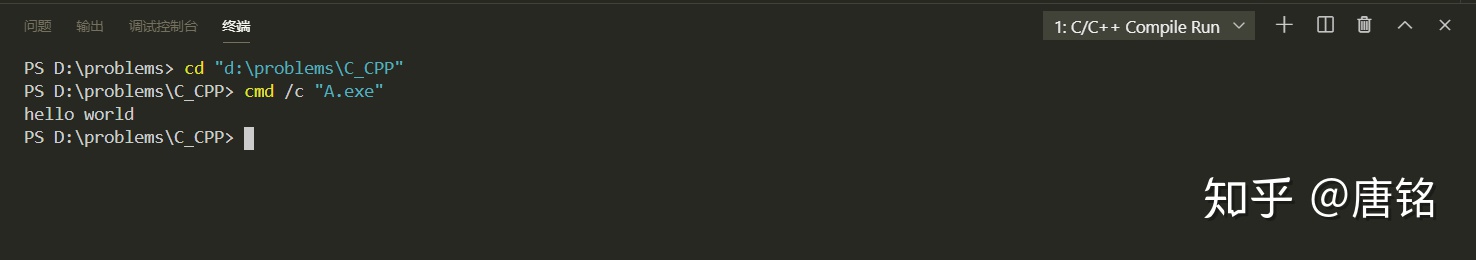
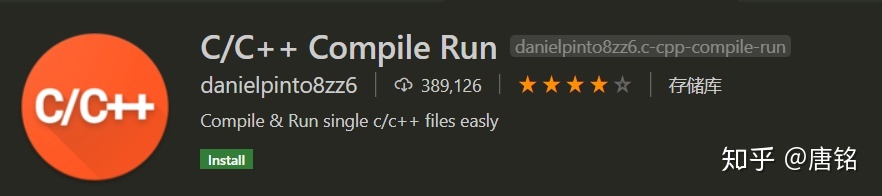


现在程序已经可以正常运行了，此时程序是运行在vscode的集成终端上，并不会额外弹出一个外部窗口

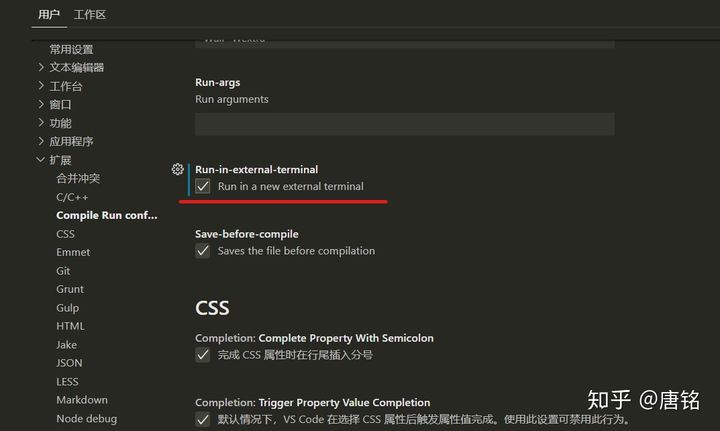


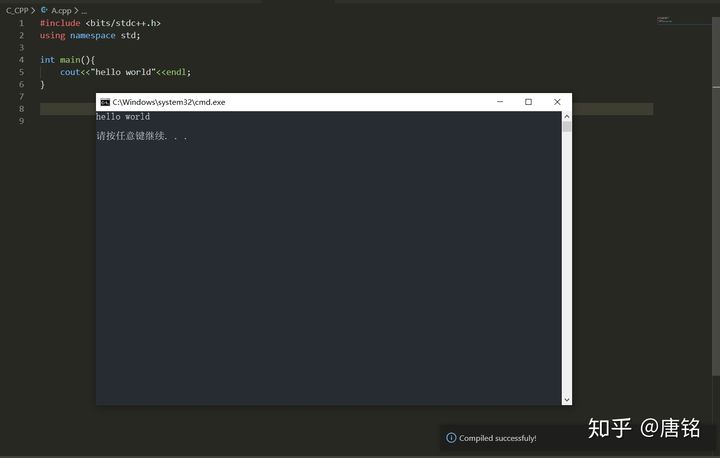
点击右上角的**垃圾桶**图标可以直接结束程序运行并关闭集成终端

**替补插件：**编译运行C/C++也可以选择C/C++ Compile Run插件，按f6一键运行



相较于code runner，C/C++ Compile Run插件在勾选上:文件>首选项>设置>扩展>Compile Run Configuration的Run-in-external-terminal选项后可以出现和dev / codeblocks / vc一样的外部窗口





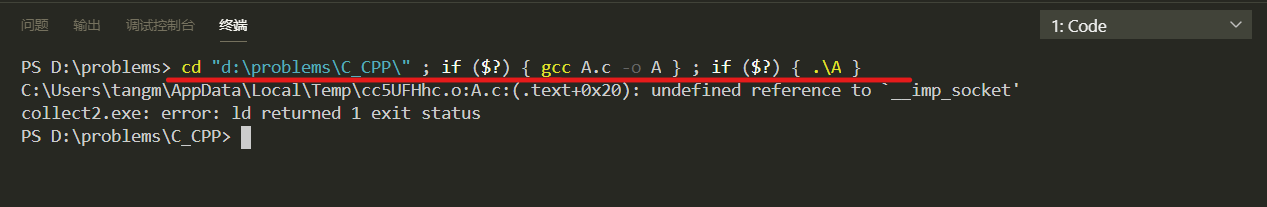
用插件运行程序是一种偷懒的方法，它本质是替用户输入了命令，它存在局限，**看一个例子**

假设说我们在代码中使用了 winsock2.h这个头文件，我们用刚刚的Code Runner插件的方式编译会无法通过

**#include** <winsock2.h>**#include** <stdio.h>**int** **main**(){

SOCKET client\_socket**=**socket(AF\_INET,SOCK\_STREAM,IPPROTO\_TCP);

printf("Compile successfully!");}



造成问题是原因是：Code Runner是按照预先设定好的规则，自动在终端执行编译运行的指令，也就是图片上红色下划线标注出的

g++ Untitled-1.cpp -o Untitled-1 ;

如果是使用了winsock2.h，同时又是使用gcc/g++编译，在编译时我们应该在编译指令中额外添加-lwsock32指令，而Code Runner默认下并不会添加这条指令

正确的编译指令应该是：

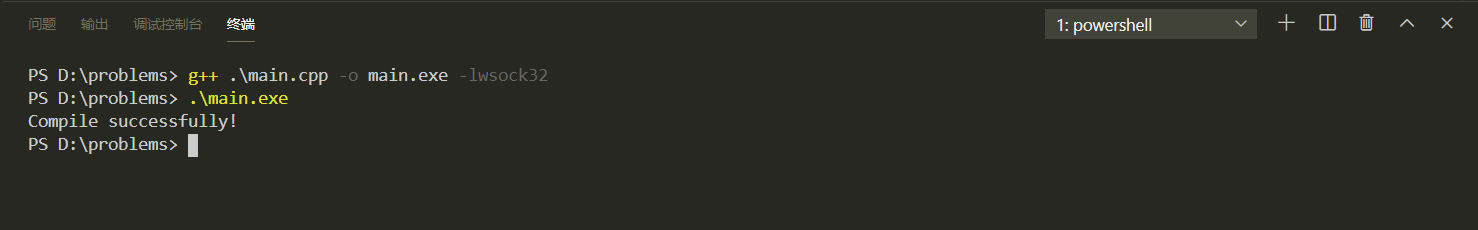
g++ xxx.cpp -o xxx.exe -lwsock32

为了解决这个问题，可以使用比较原始的方法，**直接在终端上用命令行编译和运行，**vscode有一个集成终端，可以连通安装在操作系统上的各种终端

**按 Ctrl+~ 打开集成终端（其实是"`"，打"~"是方便大家找到），输入**

g++ .\文件名.cpp -o 文件名.exe -lwsock32

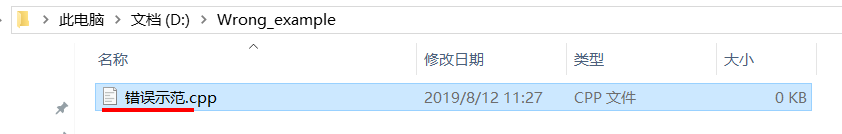
./文件名.exe



**进入正题**

首先在电脑上的某个你习惯的位置新建一个文件夹（工作区），用于存放将要编写的代码，接下来配置好的调试环境，**仅会对存放在这一个文件夹以及文件夹的子目录里的程序生效**

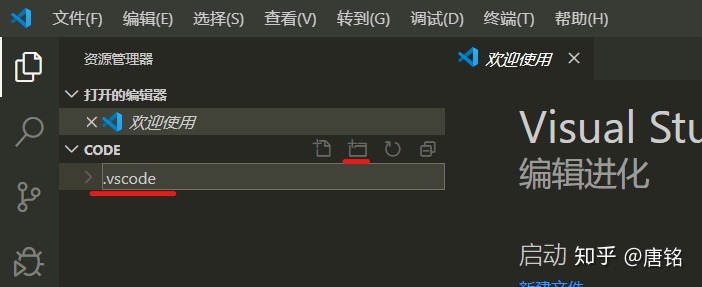
**务必注意！**调试的**文件名**和**文件路径**中不能出现**中文字符！！**否则将无法启动调试！  
这是由于调试用到的mingw中的gdb不支持中文路径！和vscode无关，下面是两个**错误示范**



新建好文件夹后，在vscode界面，点击：**文件>打开文件夹**，用vscode打开刚刚新建的文件夹（安装时勾选了右键打开可以直接右键选通过code打开）



在vscode资源管理器点击新建文件夹图标新建一个.vscode文件夹（注意前面的"."号）



然后在.vscode文件夹下新建两个json文件分别叫做

* launch.json
* tasks.json



**然后将下面的代码复制到对应的json文件中去并保存，要仔细阅读后面两段说明性文字**

**launch.json**

{

"version": "0.2.0",

"configurations": [

{

"name": "C/C++",

"type": "cppdbg",

"request": "launch",

"program": "${fileDirname}/${fileBasenameNoExtension}.exe",

"args": [],

"stopAtEntry": false,

"cwd": "${workspaceFolder}",

"environment": [],

"externalConsole": false,

"MIMode": "gdb",

"miDebuggerPath": "gdb.exe",

"preLaunchTask": "compile",

"setupCommands": [

{

"description": "Enable pretty-printing for gdb",

"text": "-enable-pretty-printing",

"ignoreFailures": true

}

],

},

]

}

**tasks.json**

**如果需要是c语言也就是gcc将下面的command项由g++改为gcc**  
**如果是多文件编译（即函数声明和函数定义分开，不懂的别乱改），需要将args列表中的"${file}"项修改为"${workspaceFolder}/\*.cpp" ，多文件推荐用cmake**

{

"version": "2.0.0",

"tasks": [

{

"type": "shell",

"label": "compile",

"command": "g++",

"args": [

"-g",

"${file}",

"-o",

"${fileDirname}\\${fileBasenameNoExtension}.exe"

],

"problemMatcher": [

"$gcc"

],

"group": {

"kind": "build",

"isDefault": true

}

}

]

}

事实上现在已经可以正常调试了，并且现在可以按 **ctrl+shift+b**直接调用配置好的task编译而不运行程序，类似于一些IDE的编译选项

测试一下，新建一个.cpp文件将下面的代码粘贴进去，并在适当的地方添加上断点

**#include** <bits/stdc++.h>**using** **namespace** std;

**void** **quicksort**(**int** A[],**int** l,**int** r){

**int** m**=**l;

**for**(**int** i**=**l;i**<**r;i**++**){

**if**(A[i]**<**A[r]){

swap(A[i],A[m]);

**++**m;

}

}

swap(A[m],A[r]);

**if**(m**>**l**+**1)quicksort(A,l,m**-**1);

**if**(m**<**r**-**1)quicksort(A,m**+**1,r);}

**int** **main**(){

**int** A[10];

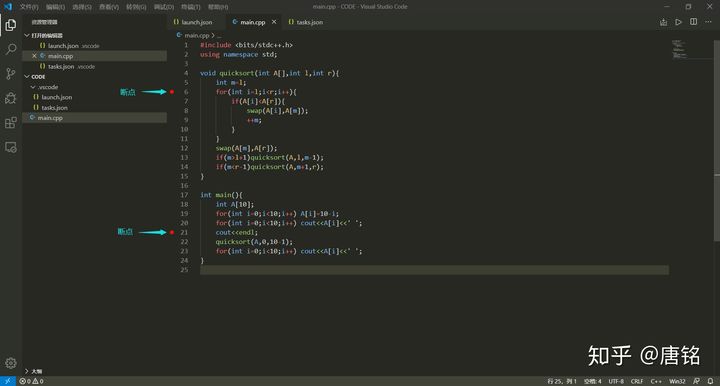
**for**(**int** i**=**0;i**<**10;i**++**) A[i]**=**10**-**i;

**for**(**int** i**=**0;i**<**10;i**++**) cout**<<**A[i]**<<**' ';

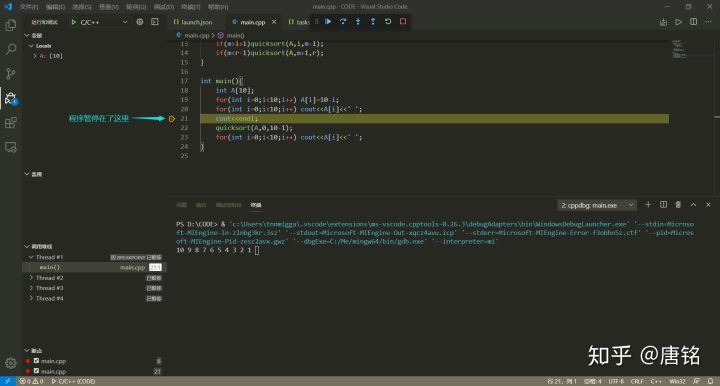
cout**<<**endl;

quicksort(A,0,10**-**1);

**for**(**int** i**=**0;i**<**10;i**++**) cout**<<**A[i]**<<**' ';}



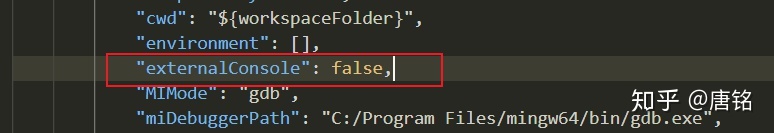
**按f5或者启动调试的按钮启动调试，程序执行到断点处会停下**



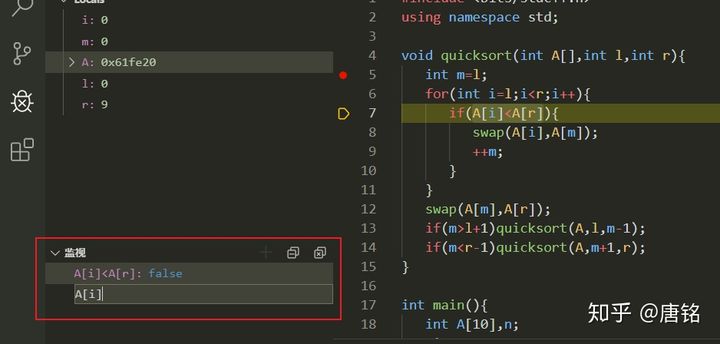
**顶部六个按钮分别代表**

1. 继续执行到下一个断点处
2. 执行下一条语句，遇到函数直接执行完不会跳转进函数
3. 执行下一条语句，遇到函数会跳转进函数继续单步执行
4. 跳出当前所在的函数，如果是主函数会结束程序
5. 重新启动调试
6. 结束调试

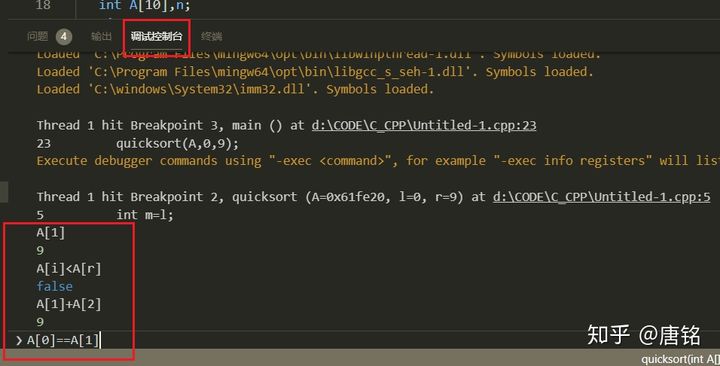
集成终端在底部终端栏输入数据，如果是希望像vs2019一样，启动调试后**显示外部窗口而不是在vscode集成终端进行调试**  
**只需将launch.json中的"externalConsole"后面的值由false改为true**



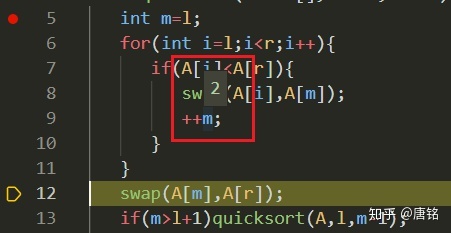
在单步调试的过程中我们可以添加监视来实时监视变量或表达式值的变化



也可以在调试控制台中输入想获取结果的变量名或者表达式获得当前的值



或者直接将鼠标光标移动到变量位置上，会自动显示当前变量的值



如果数组开得过大或者是一些结构较复杂的类或结构体，查询或显示值可能会导致调试程序崩溃

**强调**：调试是属于工作区设置，当前配置的调试环境只会对当前.vscode文件夹所在路径下的文件生效，如果要换用别的文件夹，可以直接把.vscode这个文件夹拷贝过去

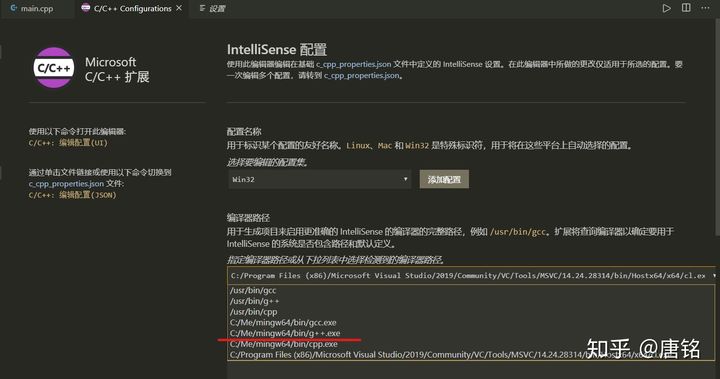
## **补充**

**类似于code runner的问题**

如果是需要有额外的编译指令如-lwsock32,需要调试前事先在tasks.json的args处添加上对应的指令,或者用 // 注释掉launch.json中的 preLaunchTask:"compile"(启动调试前执行编译任务) 这一项,然后自己编译好后再执行调试，也可以直接使用cmake

**找不到头文件问题**

对于安装了vs2019却使用mingw的情况，这时vscode代码检测会默认使用vs2019的msvc编译器的头文件，如果你使用mingw独有的特殊头文件会报找不到头文件的错误。  
鼠标移动到报错头文件会显示一个灯泡，点击，选编辑"includePath"设置，然后在指定编译器下拉列表中选择gcc或者g++



**vscode集成终端问题**

windows 10 1903/1909版本，不能使用cmd（包括cmder），git bash作为默认终端，这两个终端输入clear / cls指令后再关闭终端会造成vscode无响应  
PowerShell作为默认终端时，虽然不存在clear的问题，但如果往终端输入了任何数据或指令又没有按回车就直接关闭了，同样会造成vscode无响应  
WSL没有发现类似的bug

**强迫症治疗**

1903及以上版本windows 10，把PowersShell作为默认终端打开时，会显示一条pscore的广告--"**尝试新的跨平台 PowerShell**[https://aka.ms/pscore6](https://link.zhihu.com/?target=https://aka.ms/pscore6" \t "https://zhuanlan.zhihu.com/p/_blank)"  
1803/1809版本的则会因为bug显示一堆空白

