

配置 VS Code C/C++开发环境

By 助教高海涵

这是一篇配置环境的文档，配置环境是学习一个语言必须要做的工作，配置一个合适的 IDE，可以提高开发效率，使写代码变成一件让人享受的事。

PART 1:前言（电脑如何执行 C 语言和其它高级语言代码？）

这一部分只是简单介绍，时间不够可以略过。

我们写的 C 语言代码在电脑中存储的是其实是一个个字符，但是对于机器而言只能执行机器码（二进制文件），这时候我们就需要进行一个翻译的过程，这个翻译的过程叫做编译。（当然，C 语言变成机器码的过程和一般想象中的翻译还是有所不同，这些内容将在之后的课程介绍）

现在常用的编译器随操作系统平台不同有所差异，在 Unix/Linux 操作系统上，常用的编译器叫 GNU，但是它不能应用于 Windows 环境，所以就有了 MinGw，你可以理解 MinGw 是 Windows 下的 GNU。

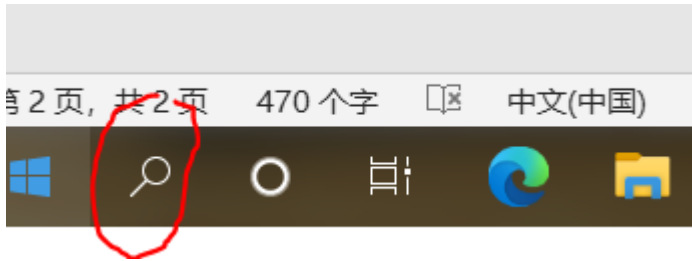
GNU 不仅可以编译 C 语言代码，还有调试作用，这将会在之后介绍。

PART 2:安装 MinGw 并配置环境变量

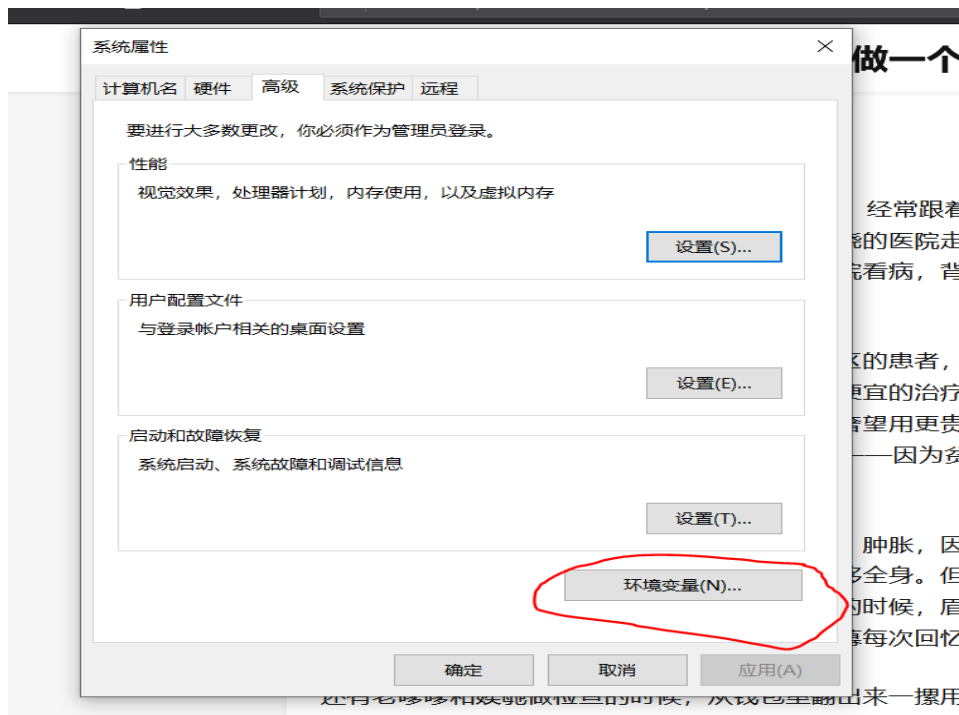
安装 MinGw: MinGw 的压缩包已经在群文件中了, 将 MinGw 的安装包下载下来, 解压至电脑的某个位置 (请记住这个位置, 之后在 **vscode** 的配置文件中要用!)

配置环境变量, 这里请各位自己 search 一下环境变量的含义, 本文只是告诉大家如何配置环境变量。

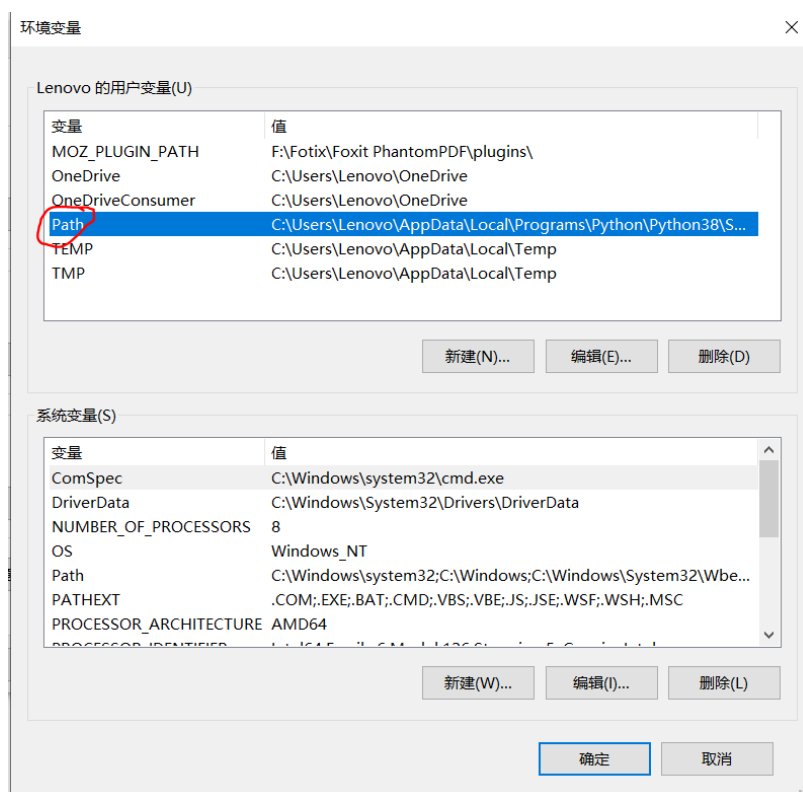
点击 Windows 搜索栏目



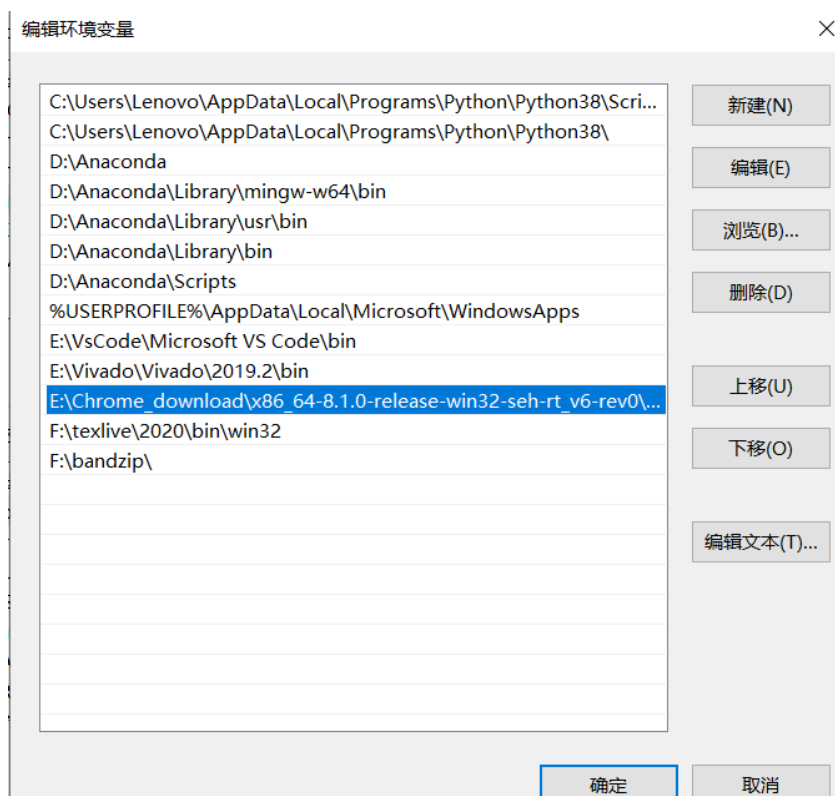
输入: 高级系统设置, 选择 查看高级系统设置, 进入一下窗口, 选择变量变量



在 环境变量 窗口，你可以查看系统中已经存在的环境变量，点击 Path



进入 path，查看自己的配置的环境变量



这里选择新建，将刚刚解压的 MinGw 文件夹下 bin 文件夹的位置复制粘贴到新建的条目中，即完成环境变量的配置

为了检测我们配置环境变量是否配置成功，我们需要打开命令行（在搜索一栏输入 cmd，进入命令行界面），在命令行下输入 gcc -v，查看输出，正确的输出如下图所示

```
C:\Users\Lenovo>gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=E:/Chrome_download/x86_64-8.1.0-release-win32-seh-rt_v6-rev0/mingw64/bin/./libexec/gcc/x86_64-w64-m
ingw32/8.1.0/lto-wrapper.exe
Target: x86_64-w64-mingw32
Configured with: ../../src/gcc-8.1.0/configure --host=x86_64-w64-mingw32 --build=x86_64-w64-mingw32 --target=x86_64-w
64-mingw32 --prefix=/mingw64 --with-sysroot=/c/mingw810/x86_64-810-win32-seh-rt_v6-rev0/mingw64 --enable-shared --enable
-static --disable-multilib --enable-languages=c,c++,fortran,lto --enable-libstdcxx-time=yes --enable-threads=win32 --ena
ble-libgomp --enable-libatomic --enable-lto --enable-graphite --enable-checking=release --enable-fully-dynamic-string --
enable-version-specific-runtime-libs --disable-libstdcxx-pch --disable-libstdcxx-debug --enable-bootstrap --disable-rpat
h --disable-win32-registry --disable-nls --disable-werror --disable-symvers --with-gnu-as --with-gnu-ld --with-arch=noco
na --with-tune=core2 --with-libiconv --with-system-zlib --with-gmp=/c/mingw810/prerequisites/x86_64-w64-mingw32-static --
with-mpfr=/c/mingw810/prerequisites/x86_64-w64-mingw32-static --with-mpc=/c/mingw810/prerequisites/x86_64-w64-mingw32-s
tatic --with-isl=/c/mingw810/prerequisites/x86_64-w64-mingw32-static --with-pkgversion= x86_64-win32-seh-rev0, Built by
MinGW-W64 project --with-bugurl=https://sourceforge.net/projects/mingw-w64 CFLAGS=-O2 -pipe -fno-ident -I/c/mingw810/x
86_64-810-win32-seh-rt_v6-rev0/mingw64/opt/include -I/c/mingw810/prerequisites/x86_64-zlib-static/include -I/c/mingw810/
prerequisites/x86_64-w64-mingw32-static/include CXXFLAGS=-O2 -pipe -fno-ident -I/c/mingw810/x86_64-810-win32-seh-rt_v6
-rev0/mingw64/opt/include -I/c/mingw810/prerequisites/x86_64-zlib-static/include -I/c/mingw810/prerequisites/x86_64-w64
-mingw32-static/include CPPFLAGS= -I/c/mingw810/x86_64-810-win32-seh-rt_v6-rev0/mingw64/opt/include -I/c/mingw810/pre
quisites/x86_64-zlib-static/include -I/c/mingw810/prerequisites/x86_64-w64-mingw32-static/include LDFLAGS=-pipe -fno-i
dent -L/c/mingw810/x86_64-810-win32-seh-rt_v6-rev0/mingw64/opt/lib -L/c/mingw810/prerequisites/x86_64-zlib-static/lib -L
/c/mingw810/prerequisites/x86_64-w64-mingw32-static/lib
Thread model: win32
gcc version 8.1.0 (x86_64-win32-seh-rev0, Built by MinGW-W64 project)
```

这样就代表 OS 可以找到 gcc 这个环境变量

PART 3: 配置 vscode json 文件

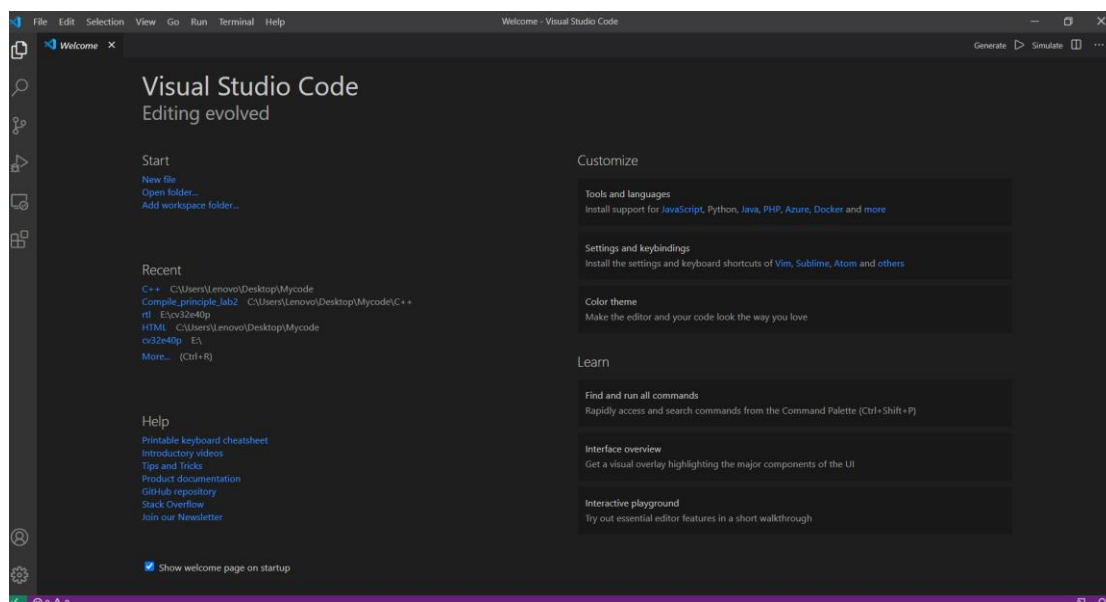
Vscode 只是一个编辑器，为了让它变成一个 IDE,我们需要告诉它一些信息，比如编译器类型和位置等，这就需要配置 json 文件。

首先从 Vscode 官网下载 Vscode 安装包安装，这里比较简单，略过，给出 vscode 官网地址

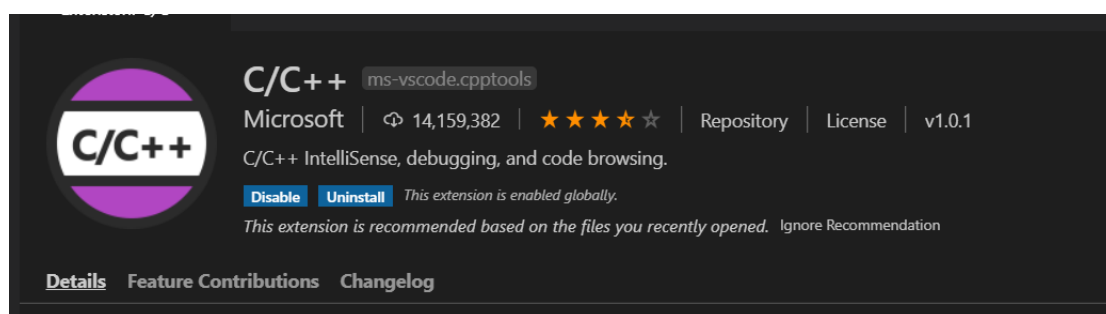
<https://code.visualstudio.com/>

我们需要建立一个文件夹，我们所有的代码都需要在

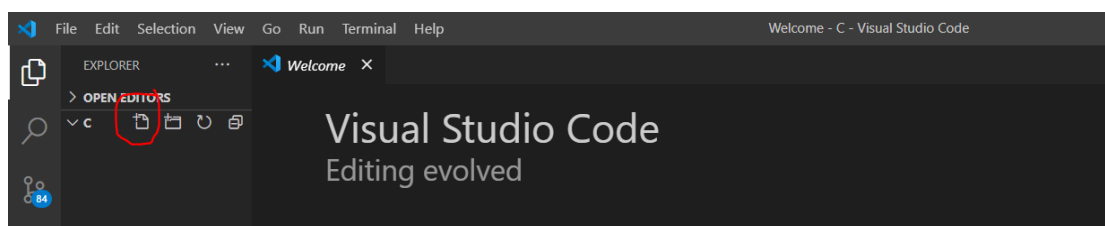
这个文件夹中。助教的文件夹名称叫 C++，当然你也可以自己定义你喜欢的名字，建议放在桌面。
打开 Vscode，进入欢迎界面



选择最左一栏最下面的按钮，点击，进入插件安装界面，在插件搜索中，输入 C，选择你看到的第一个插件，安装。助教这里已经安装，所以显示 uninstall



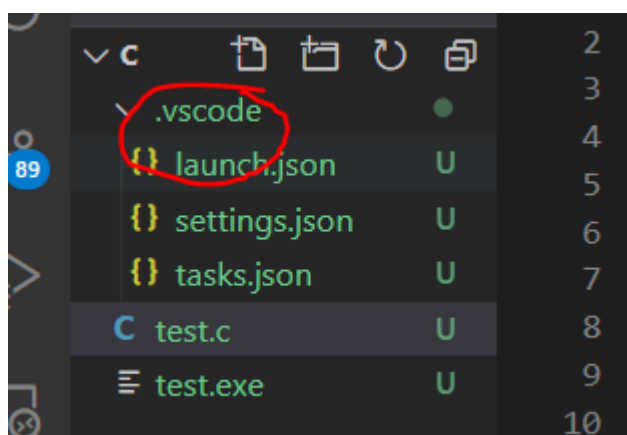
安装完成之后，重启 Vscode，选择左上角的 file，选择 open folder，打开你刚刚创建的存放代码的文件夹，建立第一个文件（单击下图用红圈圈出的部分并输入文件名）注意文件拓展名应该以.c 结尾



这是我的测试程序

```
#include<stdio.h>
int main(){
    printf("Hello world\n");
    int n;
    scanf("%d",&n);
    int j=0;
    for(int i=0;i<n;i++){
        j+=i;
    }
    printf("%d\n",j);
}
```

选择 run-run without debugging, 弹出的 environment 选择 C++ (GDB/LLDB), 随后 vscode 会在你选择的文件夹下创建一个.vscode 文件夹, 里面存放的是配置用的 json 文件。



将群里助教发的 launch.json tasks.json settings.json 复制粘贴进.vscode 文件夹, 有同名文件直接覆盖。
注意, 要将 launch.json 第 18 行改成自己的 gdb.exe

路径!

将 settings.json 第三行改成自己的 g++.exe 路径

将 tasks.json 第七行改成自己的 g++.exe 路径

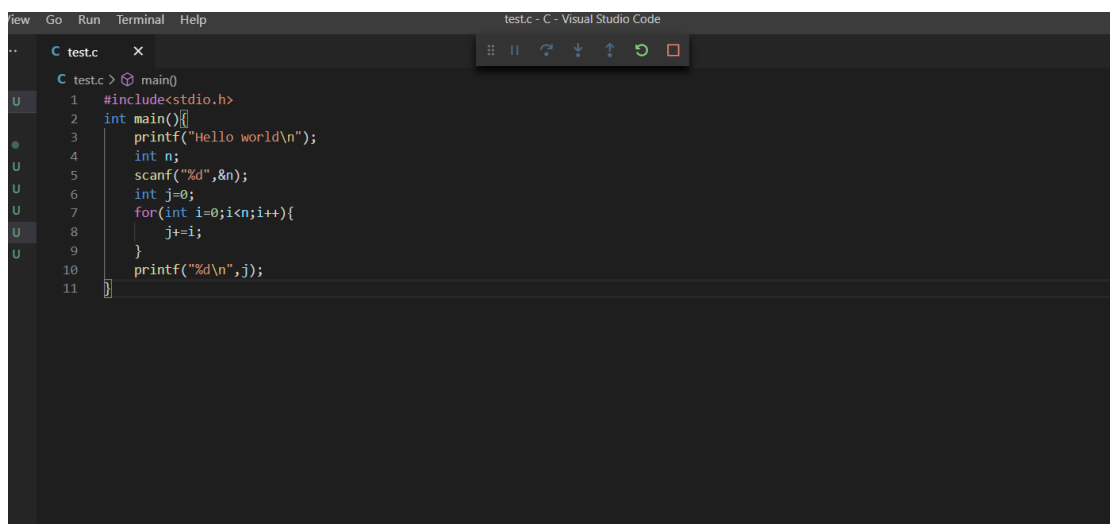
```
15     "environment": {},
16     "externalConsole": false,
17     "MIMode": "gdb",
18     "miDebuggerPath": "E:\\Chrome_download\\x86_64-8.1.0-release-win32-seh-rt_v6-rev0\\mingw64\\bin\\gdb.exe", //改成自己的gdb.exe路径
19     "setupCommands": [
20     {
```

```
2     "C_cpp.default.intellisenseMode": "gcc-x64",
3     "C_cpp.default.compilerPath": "E:\\Chrome_download\\x86_64-8.1.0-release-win32-seh-rt_v6-rev0\\mingw64\\bin\\g++.exe",
4     "C_cpp.default.cppStandard": "c++17",
5     "debug.onTaskErrors": "abort",
6     "files.associations": {
```

```
4     {
5         "type": "shell",
6         "label": "C/C++: g++.exe build active file",
7         "command": "E:\\Chrome_download\\x86_64-8.1.0-release-win32-seh-rt_v6-rev0\\mingw64\\bin\\g++.exe",
8         "args": ["-g", "${file}", "-o", "${fileDirname}\\${fileBasenameNoExtension}.exe"],
9         "options": {
10             "cwd": "${workspaceFolder}"
```

每次修改完毕，需要按下 ctrl 和 C 按键保存修改后的 json 文件，注意要在相应文件的打开界面修改!

接下来我们尝试运行一下刚刚的测试程序 run-run without debugging，进入如下界面

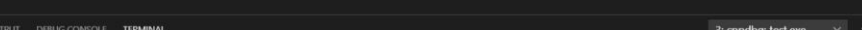


```
view Go Run Terminal Help
test.c - C - Visual Studio Code

C test.c x
C test.c > main()
1 #include<stdio.h>
2 int main(){
3     printf("Hello world\n");
4     int n;
5     scanf("%d",&n);
6     int j=0;
7     for(int i=0;i<n;i++){
8         j+=i;
9     }
10    printf("%d\n",j);
11 }
```

上方多出一栏，这是用于调试用的

注意，vscode 的输入输出不是跳出一个黑色的窗口，而是在下方的 terminal 中输入和输出



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 3: cppdbg: test.exe

尝试新的跨平台 PowerShell <https://aka.ms/pscore6>

```
PS C:\Users\lenovo\Desktop\Mycode> & 'c:\Users\lenovo\.vscode\extensions\ms-vscode.cpptools-1.6.1\debugAdapters\bin\windowsDebugLauncher.exe' '--stdin-Microsoft-MIEngine-In-w
epos8rs.yax' '--stdout-Microsoft-MIEngine-Out-4dful0a.mha' '--stderr-Microsoft-MIEngine-Error-drawrja3.gwu' '--pid-Microsoft-MIEngine-Pid-ib5yrp9h.cm2' '--dbgExe=E:\Chrome_down
load\vx86_64-8.1.0-release-win32-seh-rt_v6-rev0\mingw64\bin\gdb.exe' '--interpreter=mi'
Hello world
10
45
PS C:\Users\lenovo\Desktop\Mycode> & 'c:\Users\lenovo\.vscode\extensions\ms-vscode.cpptools-1.6.1\debugAdapters\bin\windowsDebugLauncher.exe' '--stdin-Microsoft-MIEngine-In-b
hyh8cck.3am' '--stdout-Microsoft-MIEngine-Out-duuyujlg.4ab' '--stderr-Microsoft-MIEngine-Error-8zzuywtv.t0l' '--pid-Microsoft-MIEngine-Pid-0m3i44cl.32c' '--dbgExe=E:\Chrome_down
load\vx86_64-8.1.0-release-win32-seh-rt_v6-rev0\mingw64\bin\gdb.exe' '--interpreter=mi'
Hello world
1
```

第一次启动可能会出现一些报错，这个不用管，过了一会点击 terminal 就能进入终端（如上图所示）上图终端在等待 scanf 输入，我们在终端中输入一个数，程序正常执行打印出 $1+2+3+\dots+n-1$ 的值，如下图所示（这里助教已经运行过一次，所以出现两次输出）

```

3: cppdbg: test.exe
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\Lenovo\Desktop\MyCode> g++ 8 & (Users\Lenovo\vscode\extensions\ms-vscode.cpptools-1.8.1\debugAdapters\bin\windowsDebugLauncher.exe' --stdin-Microsoft-MIEngine-in-w
eposers.yax' --stdout-Microsoft-MIEngine-out-adtuleko.aha' --stderr-Microsoft-MIEngine-error-dmarjs3.gmd' --pid-Microsoft-MIEngine-Pid-1b5yrp0h.em2' --dbgExe=E:\Chrome_down
load\MS64-8.1.0-release-win32-seh-rt_v6-rev0\mingw64\bin\gdb.exe' --interpreter=ml
Hello world
10
45
50
PS C:\Users\Lenovo\Desktop\MyCode> g++ 8 & (Users\Lenovo\vscode\extensions\ms-vscode.cpptools-1.8.1\debugAdapters\bin\windowsDebugLauncher.exe' --stdin-Microsoft-MIEngine-in-b
hyekok.iam' --stdout-Microsoft-MIEngine-out-duniojle.4ab' --stderr-Microsoft-MIEngine-err-or-0zywvtr.tbl' --pid-Microsoft-MIEngine-Pid-0d344cl.32c' --dbgExe=E:\Chrome_down
load\MS64-8.1.0-release-win32-seh-rt_v6-rev0\mingw64\bin\gdb.exe' --interpreter=ml
Hello world
10
45
50
PS C:\Users\Lenovo\Desktop\MyCode>

```

Vscode 配置大功告成！之后的课程我们会教大家如何利用 vscode 调试代码。注意，要使用 vscode 写 C 代码，必须进入我们在 PART3 创建的文件夹