Environment setup Guide On MacOS

Before Start

在开始之前，您可能需要了解以下内容：

- Mac下有gcc，但是默认版本是clang12。现在的xcode command line tool自带clang，强行占用了gcc这个命令名，但是可以通过下载GCC安装包来安装GCC。
- 我们这门课用的编译器是gcc14（gcc11以上其实都可以，但是gcc14最好，不会影响作业的正确性，但是会影响编程调试的体验）

下载brew

brew 是帮助我们在命令行快速安装其他应用的工具，因此首先安装 brew。打开mac的终端，输入安装 brew 的命令：

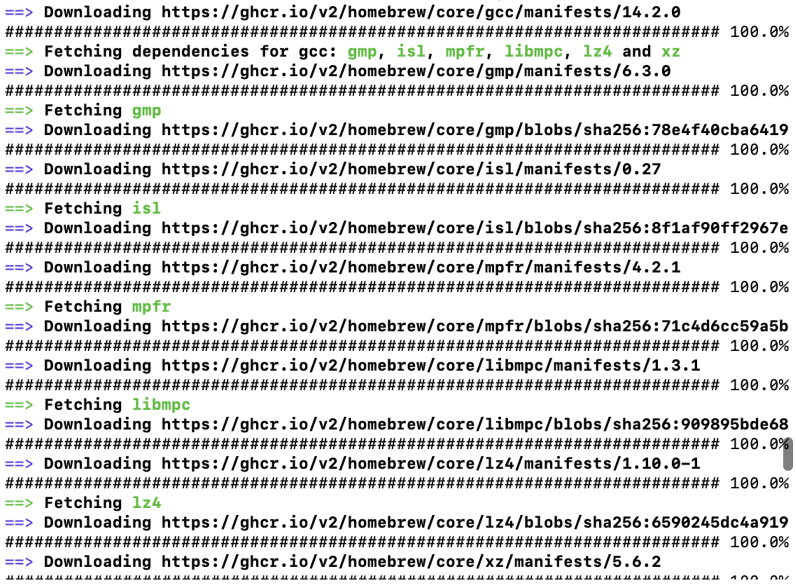
```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

安装gcc14

然后使用 brew 安装 gcc14：

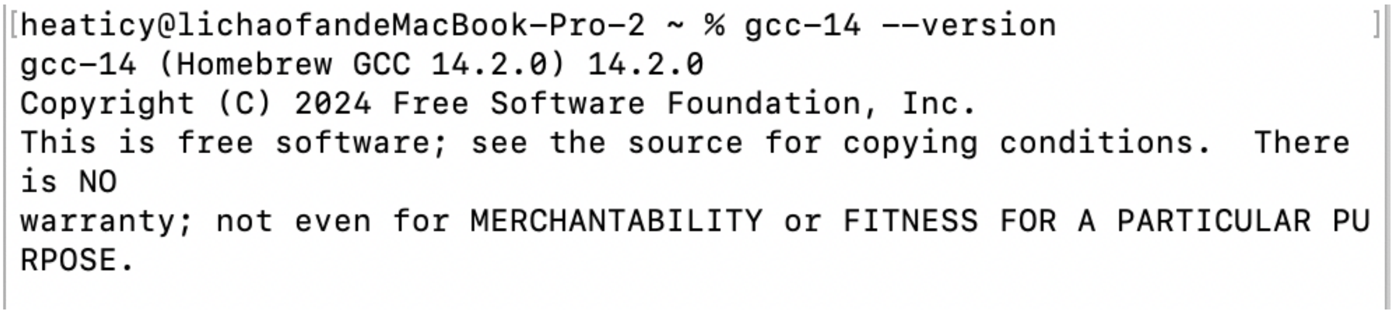
```
brew install gcc@14
```

如果出现类似的结果，说明安装正在进行：



```
==> Downloading https://ghcr.io/v2/homebrew/core/gcc/manifests/14.2.0
######################################################################### 100.0%
==> Fetching dependencies for gcc: gmp, isl, mpfr, libmpc, lz4 and xz
==> Downloading https://ghcr.io/v2/homebrew/core/gmp/manifests/6.3.0
######################################################################### 100.0%
==> Fetching gmp
==> Downloading https://ghcr.io/v2/homebrew/core/gmp/blobs/sha256:78e4f40cba6419
######################################################################### 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/isl/manifests/0.27
######################################################################### 100.0%
==> Fetching isl
==> Downloading https://ghcr.io/v2/homebrew/core/isl/blobs/sha256:8f1af90ff2967e
######################################################################### 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/mpfr/manifests/4.2.1
######################################################################### 100.0%
==> Fetching mpfr
==> Downloading https://ghcr.io/v2/homebrew/core/mpfr/blobs/sha256:71c4d6cc59a5b
######################################################################### 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/libmpc/manifests/1.3.1
######################################################################### 100.0%
==> Fetching libmpc
==> Downloading https://ghcr.io/v2/homebrew/core/libmpc/blobs/sha256:909895bde68
######################################################################### 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/lz4/manifests/1.10.0-1
######################################################################### 100.0%
==> Fetching lz4
==> Downloading https://ghcr.io/v2/homebrew/core/lz4/blobs/sha256:6590245dc4a919
######################################################################### 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/xz/manifests/5.6.2
######################################################################### 100.0%
```

然后等待安装完成。



```
heaticy@lichaofandeMacBook-Pro-2 ~ % gcc-14 --version
gcc-14 (Homebrew GCC 14.2.0) 14.2.0
Copyright (C) 2024 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 PU
RPOSE.
```

这样你的gcc14就下好了

按照https://www.bilibili.com/video/BV1id4y1n7Kw/

里面的配置vscode，但是文件换成下面的

settings.json:

注意从pdf复制的时候，换行符的问题

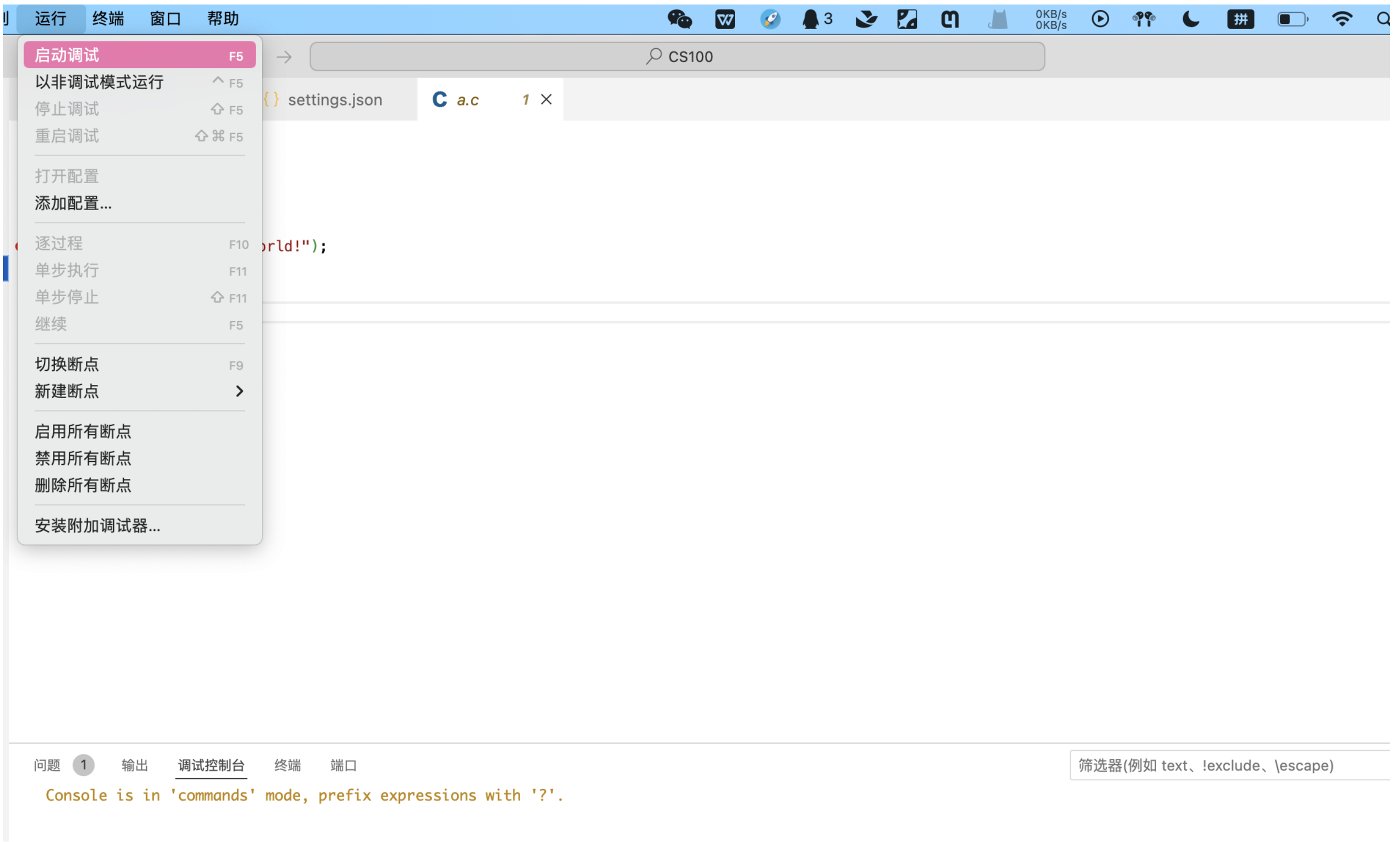
```
{
    "code-runner.executorMap": {
        "cpp": "cd $dir && g++-14 '$fileName' -o '$fileNameWithoutExt' -std=c++17 -Wall -Wpedantic -Wextra && echo 'compilation ends.' && ./'$fileNameWithoutExt'",
        "c": "cd $dir && gcc-14 '$fileName' -o '$fileNameWithoutExt' -std=c17 -Wall -Wpedantic -Wextra && echo 'compilation ends.' && ./'$fileNameWithoutExt'"
    }
}
```

c_cpp_properties.json: （用于配置"C/C++"插件）

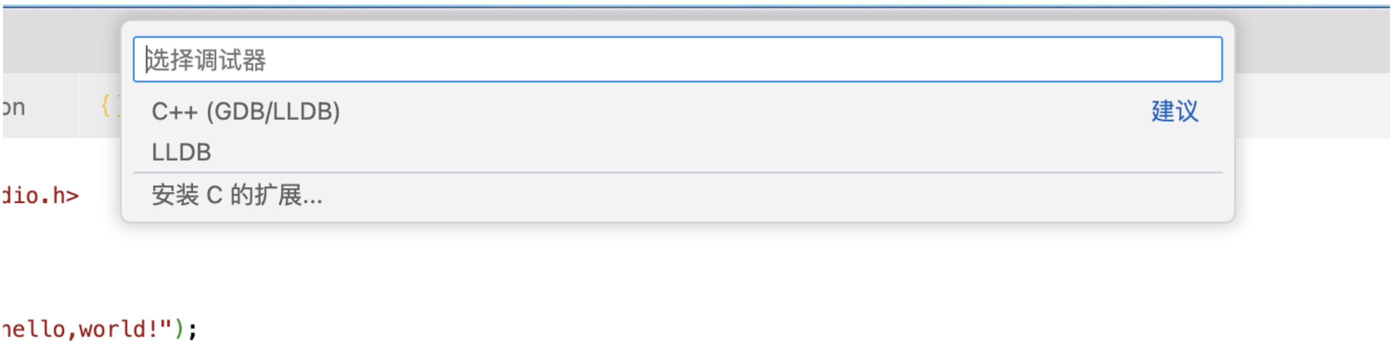
```
{
    "configurations": [
        {
            "name": "Mac",
            "compilerPath": "/opt/homebrew/bin/gcc-14",
            "cStandard": "c17",
            "cppStandard": "c++17",
            "compilerArgs": [
                "-Wall",
                "-Wpedantic",
                "-Wextra"
            ]
        }
    ],
    "version": 4
}
```

GDB调试配置

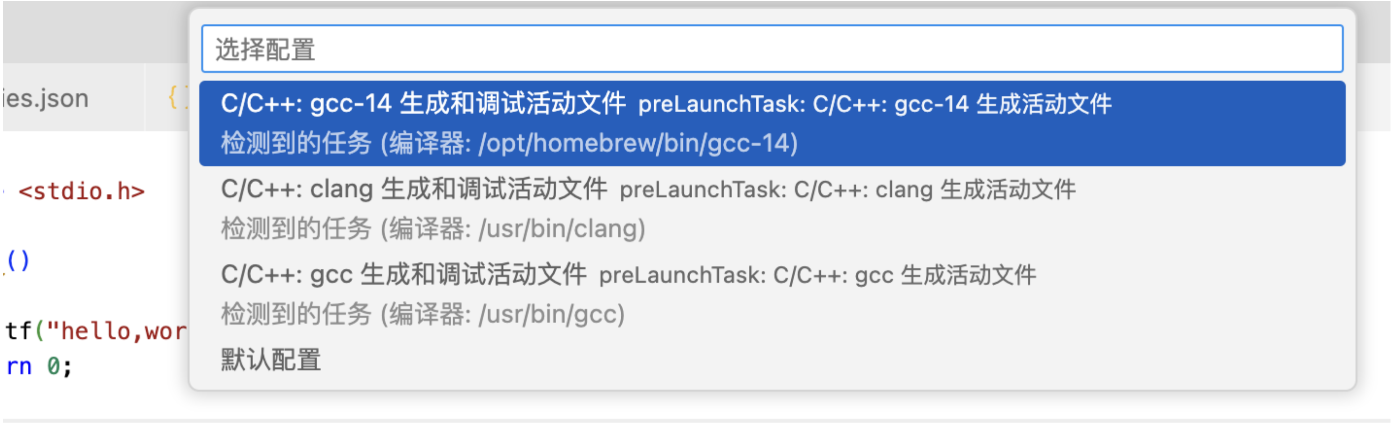
点“运行”—“启动调试”



```
Console is in 'commands' mode, prefix expressions with '?'.
```



选C++



选择配置

C/C++: gcc-14 生成和调试活动文件 preLaunchTask: C/C++: gcc-14 生成活动文件
检测到的任务 (编译器: /opt/homebrew/bin/gcc-14)

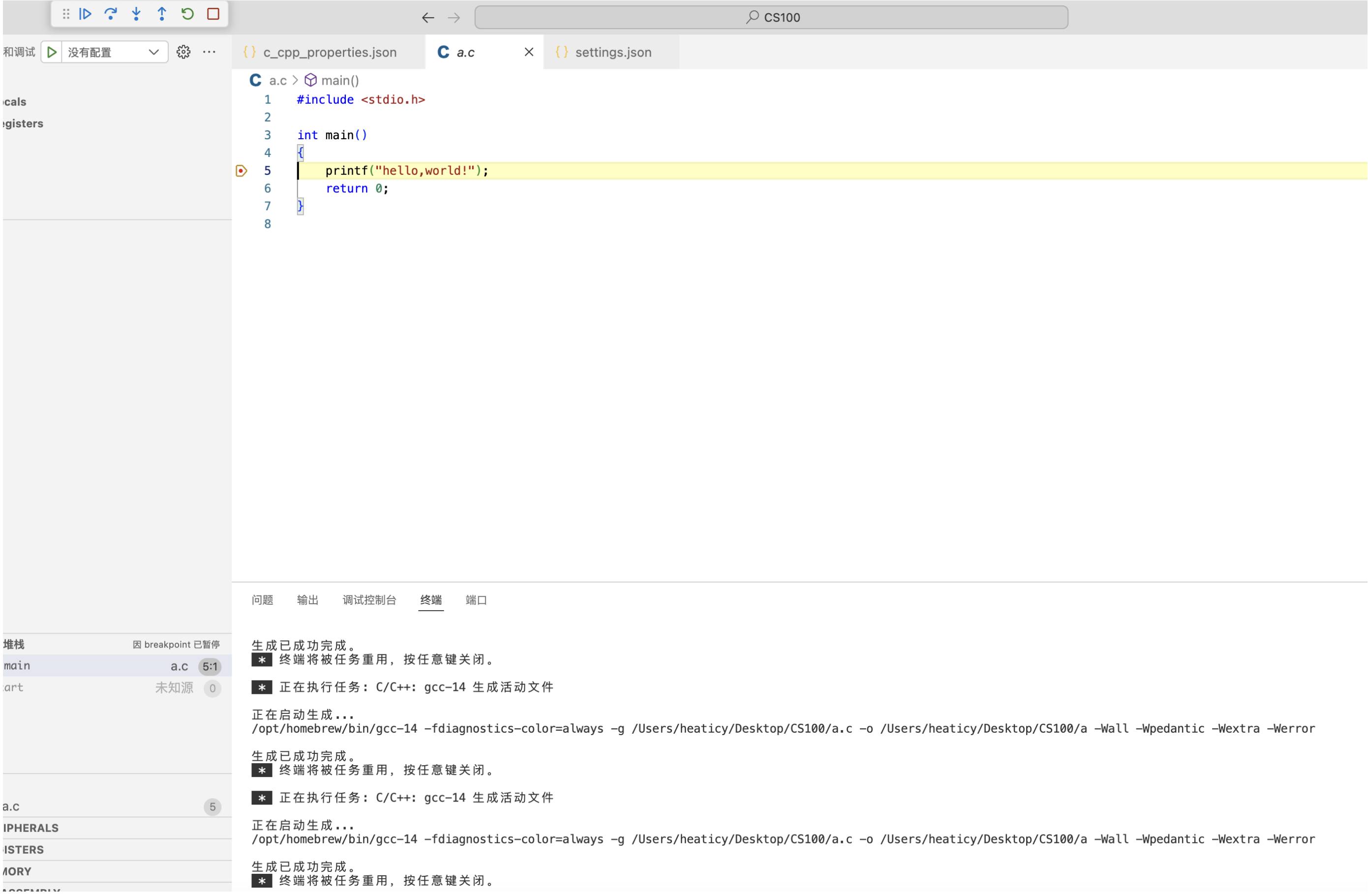
C/C++: clang 生成和调试活动文件 preLaunchTask: C/C++: clang 生成活动文件
检测到的任务 (编译器: /usr/bin/clang)

C/C++: gcc 生成和调试活动文件 preLaunchTask: C/C++: gcc 生成活动文件
检测到的任务 (编译器: /usr/bin/gcc)

默认配置

选gcc-14（其他两个都是clang）

就可以完成断点调试了



```c
#include <stdio.h>

int main()
{
    printf("hello,world!");
    return 0;
}

```

```
生成已成功完成。
 *  终端将被任务重用，按任意键关闭。

 *  正在执行任务: C/C++: gcc-14 生成活动文件

正在启动生成...
/opt/homebrew/bin/gcc-14 -fdiagnostics-color=always -g /Users/heaticy/Desktop/CS100/a.c -o /Users/heaticy/Desktop/CS100/a -Wall -Wpedantic -Wextra -Werror

生成已成功完成。
 *  终端将被任务重用，按任意键关闭。

 *  正在执行任务: C/C++: gcc-14 生成活动文件

正在启动生成...
/opt/homebrew/bin/gcc-14 -fdiagnostics-color=always -g /Users/heaticy/Desktop/CS100/a.c -o /Users/heaticy/Desktop/CS100/a -Wall -Wpedantic -Wextra -Werror

生成已成功完成。
 *  终端将被任务重用，按任意键关闭。
```