

C++开发环境搭建手册（MAC版）

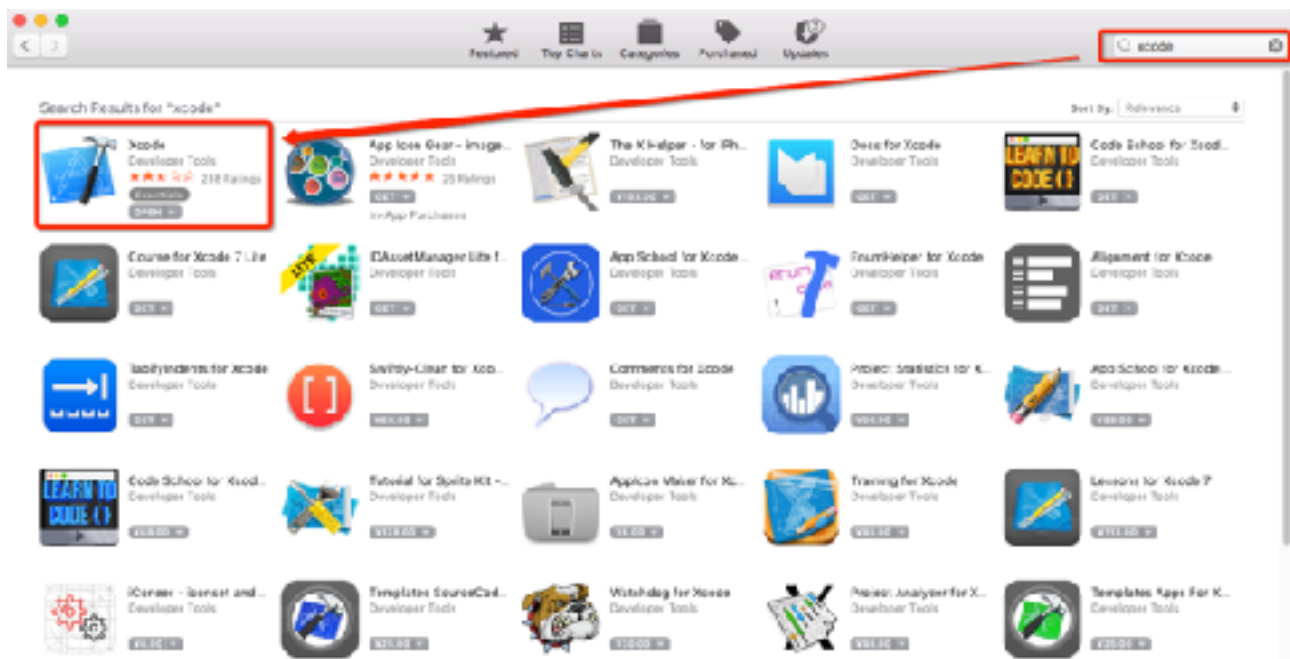
c++编译器（gcc/g++）

- macOS自带c++编译环境，不需要额外安装😊

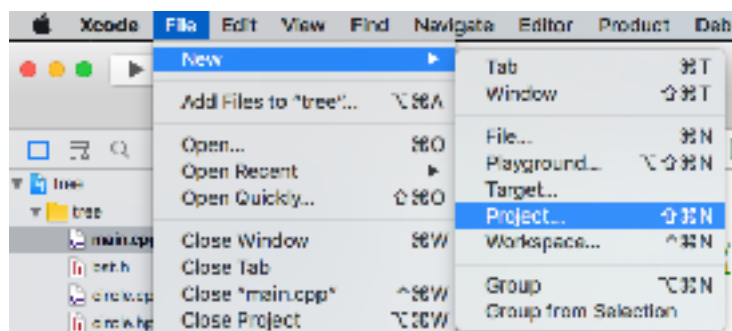
Xcode（轻量级图形化IDE）

- 本课程将使用Xcode作为演示环境
- Xcode可以在mac的AppStore中下载安装（需要appleId账号，没有的话注册一个即可），安装过程是傻瓜式的

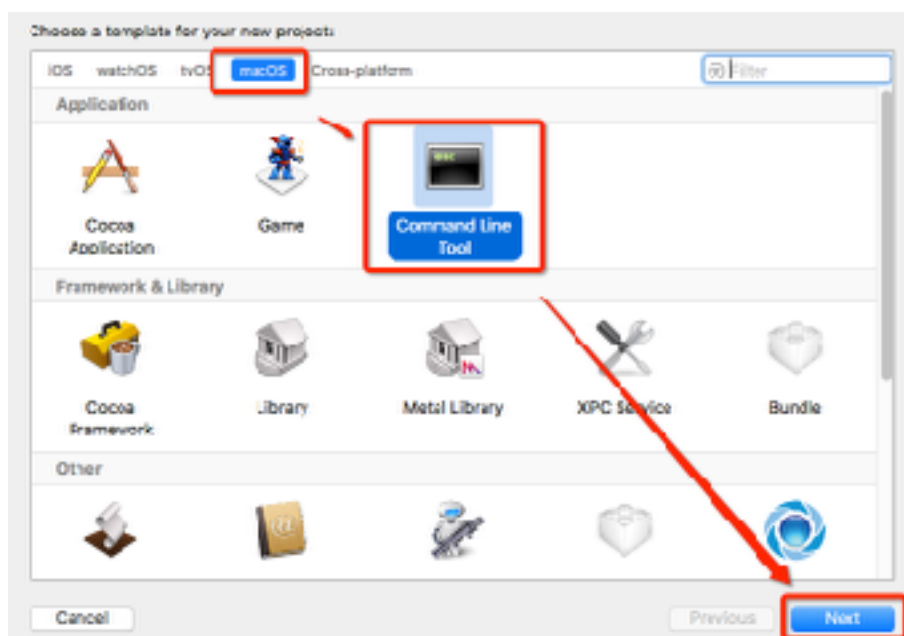




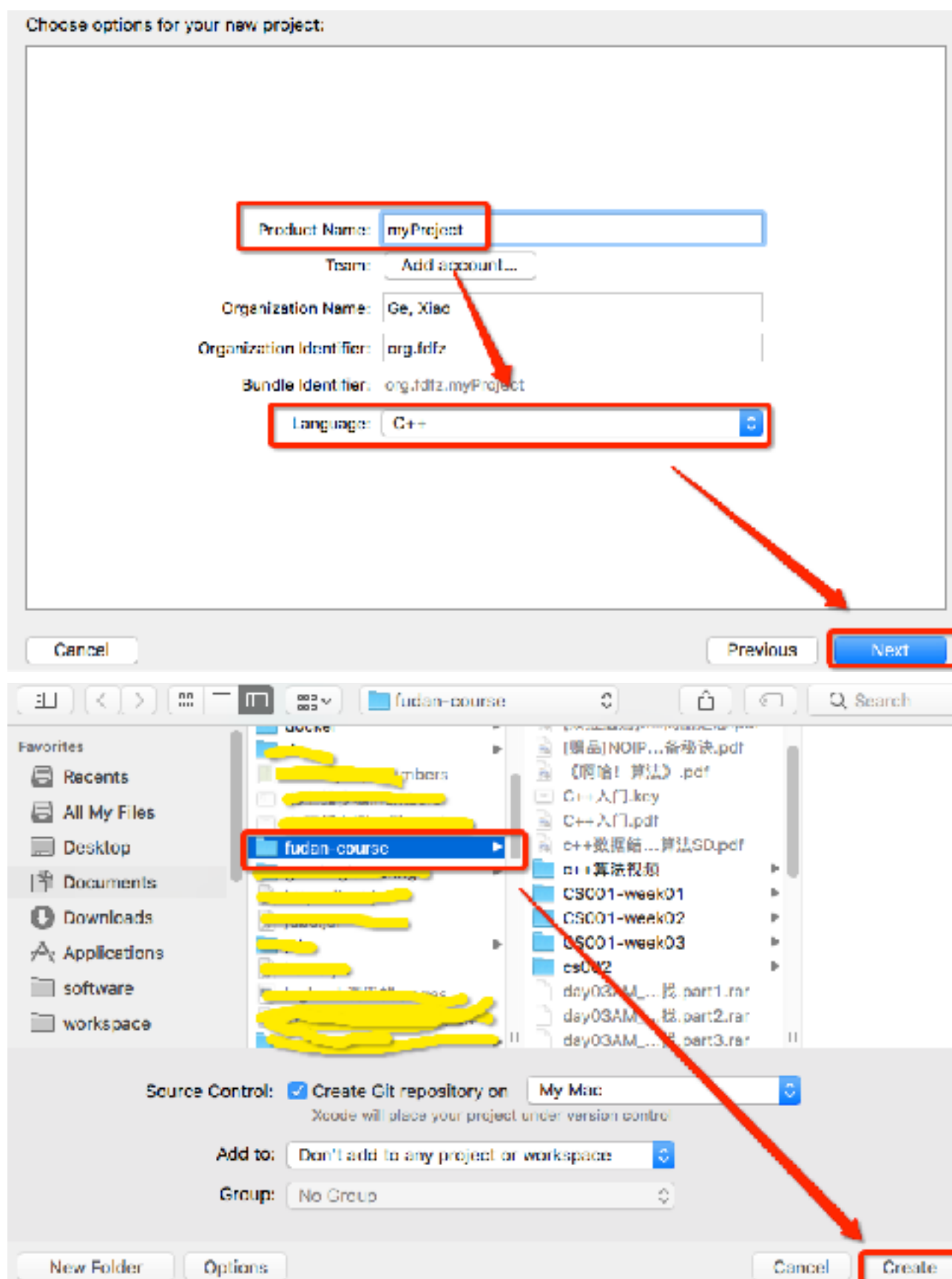
- 创建工程：打开Xcode，选择File→New→Project



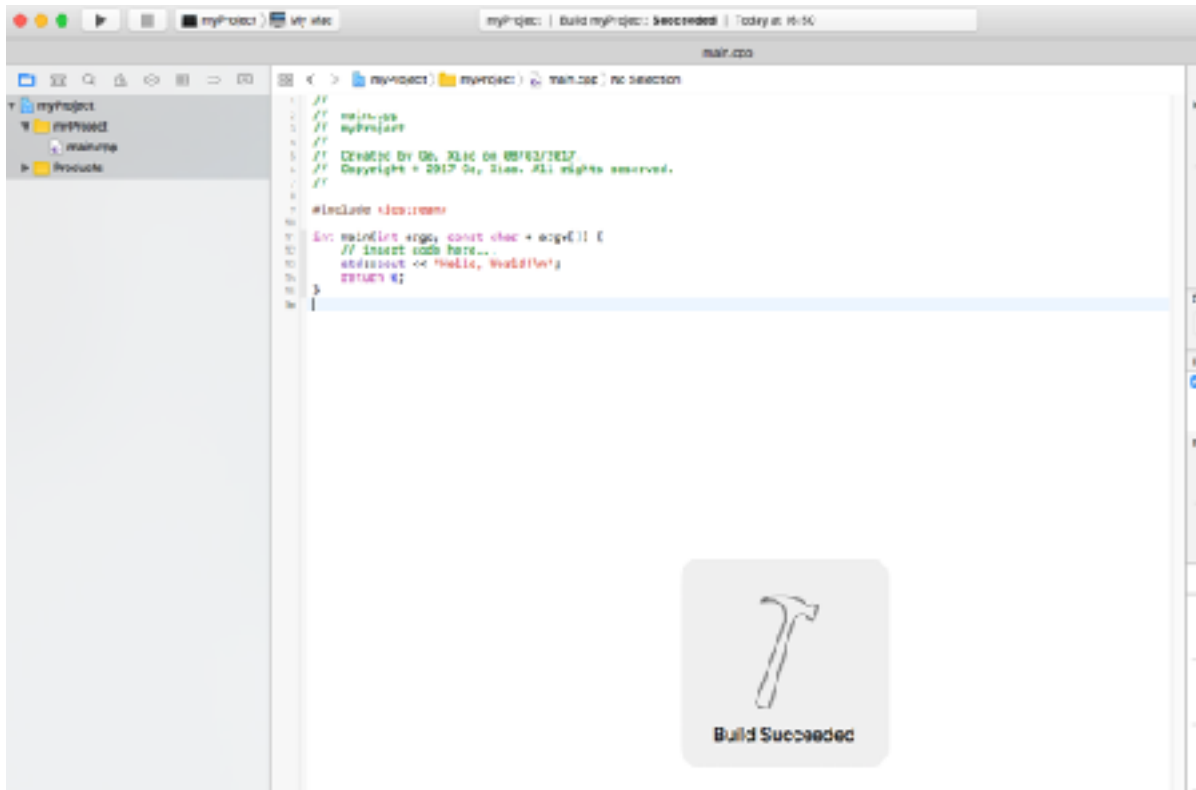
- 选择命令行工具 (Command Line Tool)



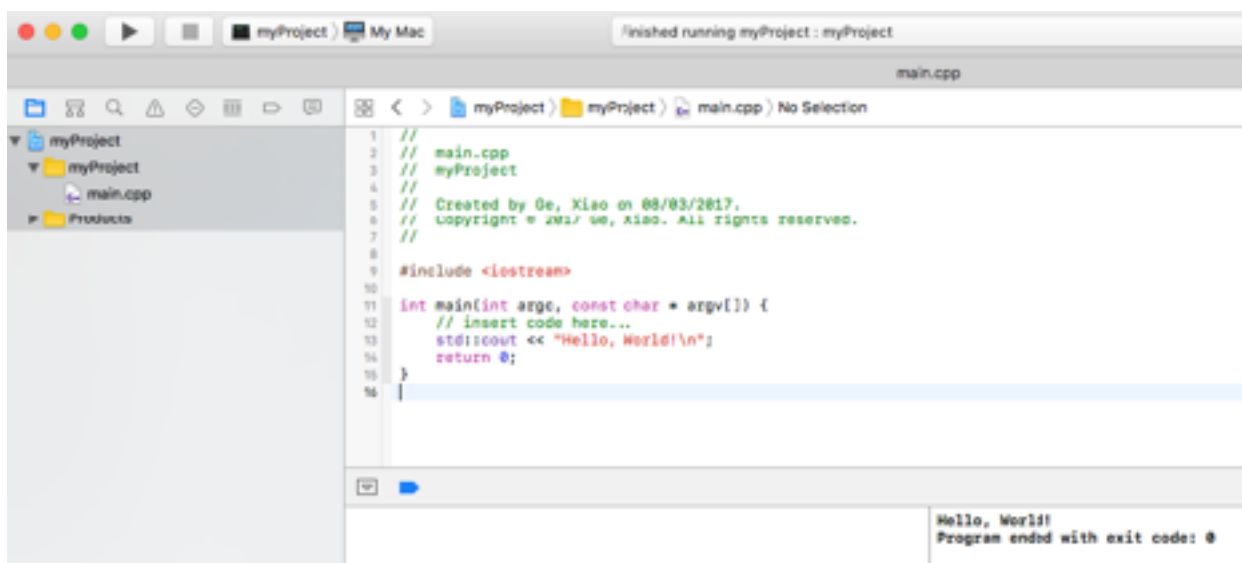
- 输入工程名字，并选择语言、工程路径。点确认创建工程



- 这时Xcode会自动创建一个HelloWorld的主函数main.cpp。你也可以修改这个代码文件，写入自己的代码
- 点Product→Build或使用⌘+B快捷键编译代码，如编译成功会显示Build Succeeded

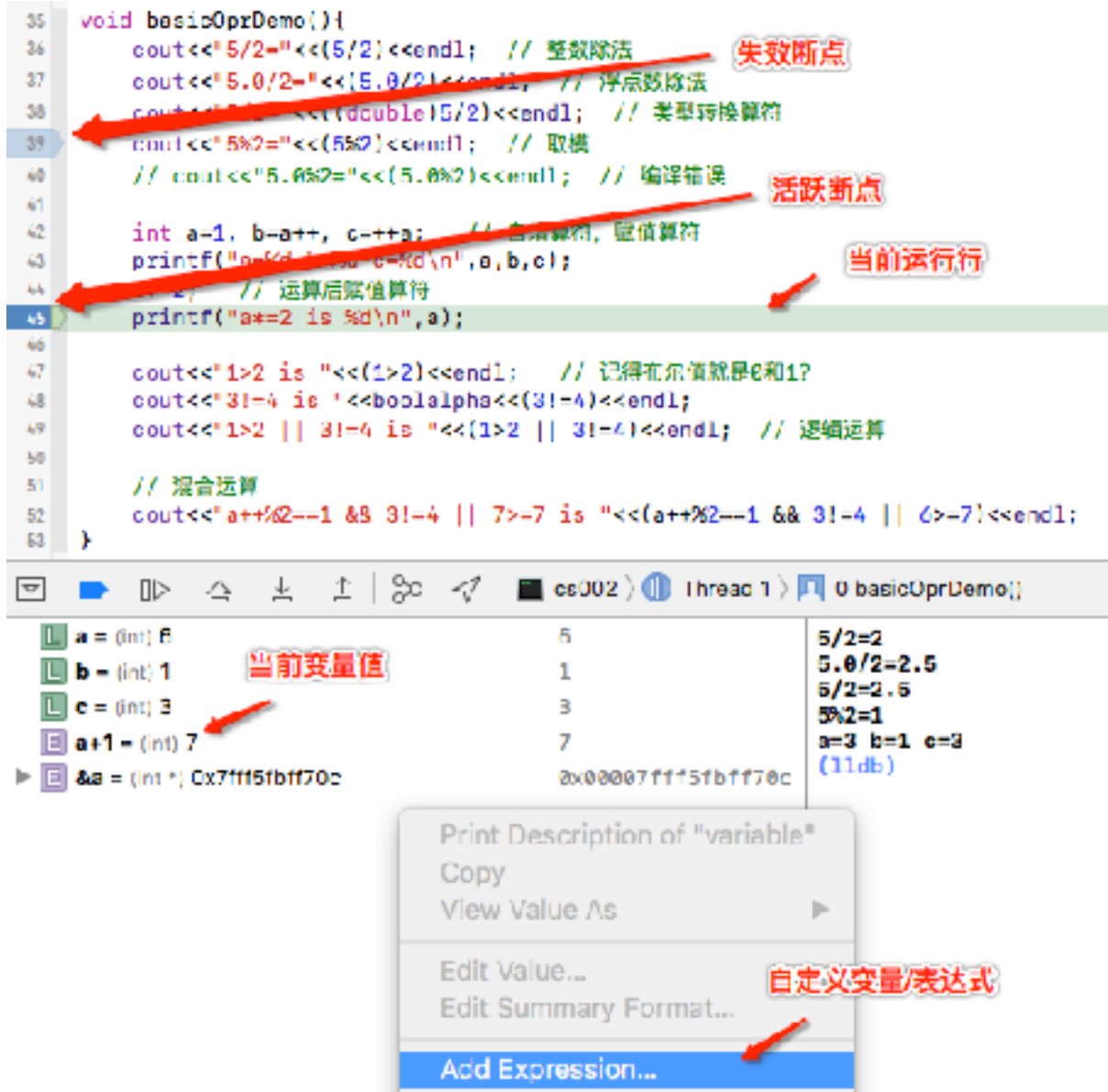


- 点Product→Run或使用⌘+R快捷键运行代码，会在下面的控制台区域看到输出。如果代码需要输入，也在同一区域输入



- 看到输出Hello World字样。恭喜你！你已经会用Xcode进行C++编程了

- 调试步骤：设置断点→运行程序→添加监控变量→单步执行



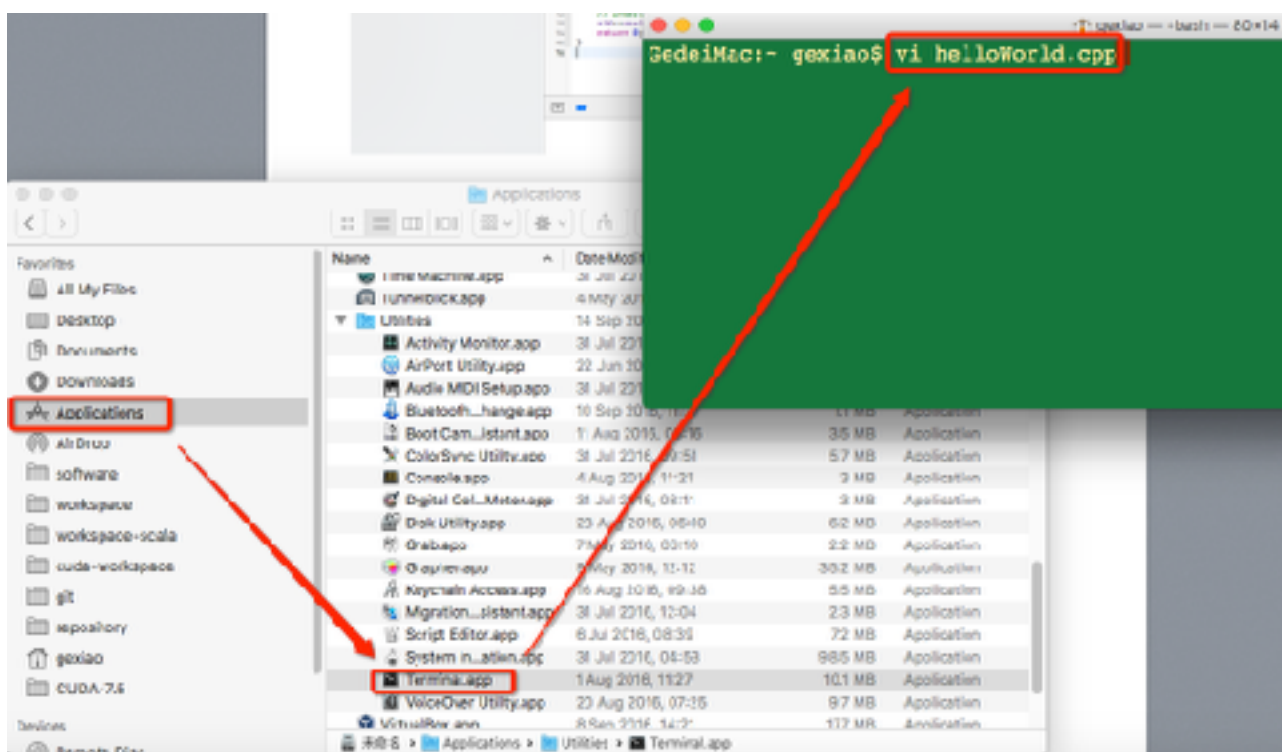
- 快捷键

- Continue: 继续执行直到遇到断点
- Continue to Current Line: 继续执行直到运行到光标所在行
- Step Over: 单步执行 (不进入函数内部)
- Step Into: 单步执行 (进入函数内部)
- Step Out: 继续执行直到当前函数结束

Continue	^⌘Y
Continue To Current Line	^⌘C
Step Over	F6
Step Into	F7
Step Out	F8

vi (文本编辑器)

- 高手推荐
- macOS自带vi编辑器，不需要安装😊
- 打开mac的命令行终端Terminal（如果没有快捷方式，在Finder里找~/Applications/Utilities/Terminal.app）
- 输入vi helloWorld.cpp创建代码文件并进入编辑器界面（如果文件已经存在则相当于修改文件）



- 按一下 i 键进入写入状态，输入helloWorld代码



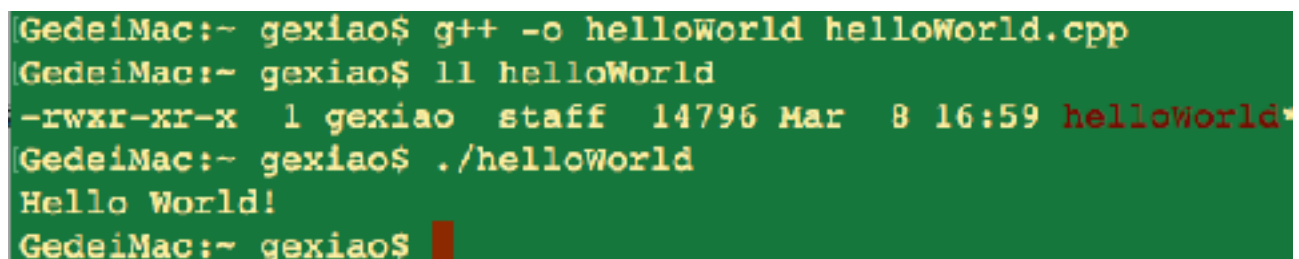
```
1 #include<iostream>
2
3 int main(){
4     std::cout<<"Hello World!\\n";
5     return 0;
6 }
7
-- INSERT --
```

- 按Esc键退出写入状态，输入:wq（冒号也要输入）保存文件并退出vi



```
1 #include<iostream>
2
3 int main(){
4     std::cout<<"Hello World!\\n";
5     return 0;
6 }
7
:wq
```

- 编译代码：在Terminal输入g++ -o helloWorld helloWorld.cpp，生成可执行文件helloWorld
- 运行可执行文件：在Terminal输入 ./helloWorld



```
GedeiMac:~ gexiao$ g++ -o helloWorld helloWorld.cpp
GedeiMac:~ gexiao$ ll helloWorld
-rw-r--r-x 1 gexiao staff 14796 Mar 8 16:59 helloWorld
GedeiMac:~ gexiao$ ./helloWorld
Hello World!
GedeiMac:~ gexiao$
```

- 建议你自己去查一下关于vi文本编辑器的其他命令
- 修改vi配色方案，增加显示行号，修改Tab宽度：

编辑vimrc文件（与键入代码一样）：vi ~/.vimrc

输入下面几行配置，保存退出即可

```
syntax on
set number
set tabstop=4
```

- 调试：vi不能单步调试，你可以在希望调试的断点处输出日志。这是一种工程技能（竞赛里不要用），利用日志调试程序是一种更高端的技能

```
1 #include<iostream>
2
3 using namespace std;
4
5 int main(){
6     int n=10;
7     int a[n];
8     for(int i=0;i<n;i++){
9         scanf("%d",&a[i]);
10    }
11    int max=0;
12    for (int left=0;left<n;left++){
13        for (int right=left;right<n;right++){
14            int sum=0;
15            for (int i=left;i<=right;i++){
16                sum+=a[i];
17                if (sum>max){
18                    cout<<"Bigger Sum "<<sum<<" from "<<left<<" to "<<right<<endl;
19                    max=sum;
20                }
21            }
22        }
23    }
24    cout<<"Max="<<max<<endl;
25    return 0;
26 }
-- INSERT --
```


其他IDE

- 你也许听说过Eclipse和Visual Studio这两款IDE，这都是重量级的开发环境，适合大型项目开发。如果你以后要从事IT/互联网行业，也许你会用到这些工具。不过目前没有必要去了解

Linux指令

- Mac的Terminal类似Windows的命令行工具（cmd），因为MacOS底层是Linux内核，而Terminal就是操作Linux的命令行，因此原则上Terminal可以实现所有Mac系统操作
- 以下是一些常用Linux命令
 - 查看当前文件夹（对应windows的dir命令）：ls -l
 - 进入目录：cd \${目录名}，返回上层目录：cd ..（注意..之前加空格）
 - 创建目录：mkdir \${目录名}
 - 现实文本文件内容：cat \${文件名}（很大的文件不要用，避免刷屏）
 - 复制文件：cp \${源文件} \${目标文件}
 - 移动/重命名文件：mv \${源文件} \${目标目录/目标文件}
 - 删除文件：rm \${文件名}
- 建议你自己去查一下其他命令，或者买一本Linux工具书备查。如果你有志从事互联网行业开发，你早晚要学会使用Linux；即使没有这方面打算，会用Linux也会让你显得很酷😎
- 以下为一些建议你了解的命令和概念（有些比较复杂，不要求一定掌握）：
echo, grep, awk, head, tail, chmod, 重定向, 管道

优劣对比

	优势	劣势
Xcode	1.支持调试 2.有代码智能提示/自动格式化等支持功能，容易上手	1.代码组织受一定限制，如必须组织到工程中（对于竞赛而言并不需要工程）
vi	1.用纯文本编辑器写代码，狂拽炫酷（高手的选择） 2.无调试环境下把代码写对，更具挑战性 3.轻便	1.基本只能用键盘操作（其实熟练了你会觉得鼠标并没有特别大半帮助）

- 最后强调一下，IDE只是协助开发的工具，是帮助我们提高开发效率的。不要对IDE产生过大的依赖，你要学习和领悟的更关键的内容是算法的思想和数学建模的思路