如下所示，代码中为什么要有这样的结构呢？

编写头文件test.h时，在头文件开头写上两行，如下：

#ifndef \_\_TEST\_H

#define \_\_TEST\_H

"

...... #内容

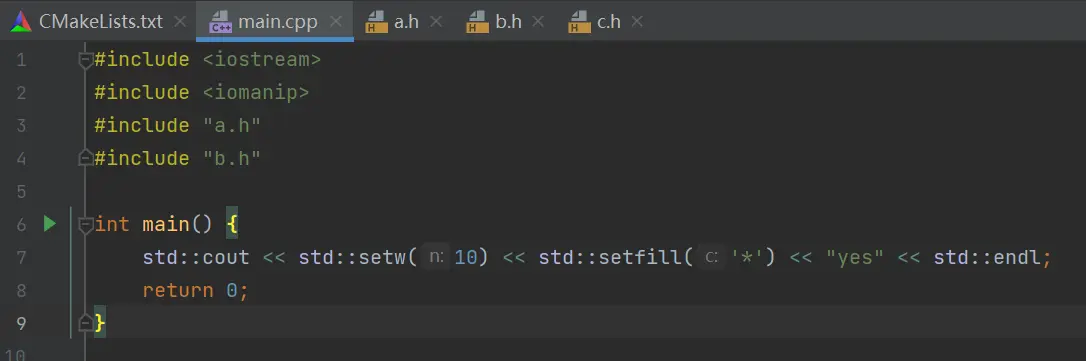
......

"

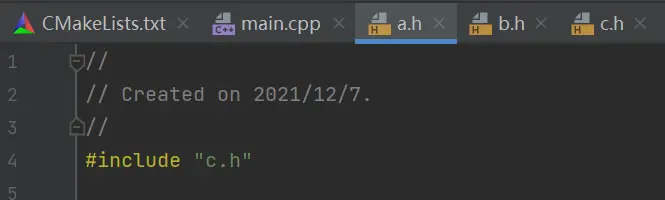
头文件结尾写上一行：

#endif

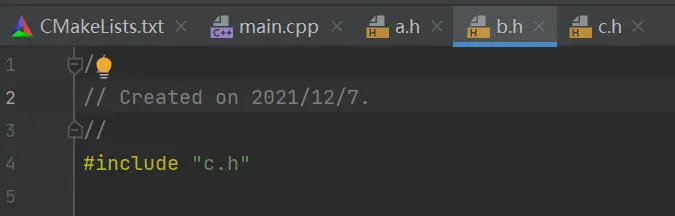
**这是因为：**防止文件中的内容被重复定义  
我们来看下面这样一个例子，定义了main.cpp，a.h，b.h，c.h，其中main.cpp包含了a.h和b.h，a.h包含了c.h，b.h也包含了c.h。



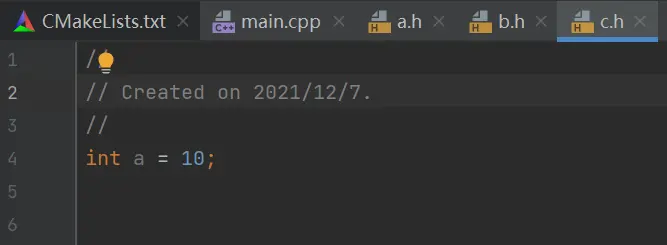
main.cpp



a.h

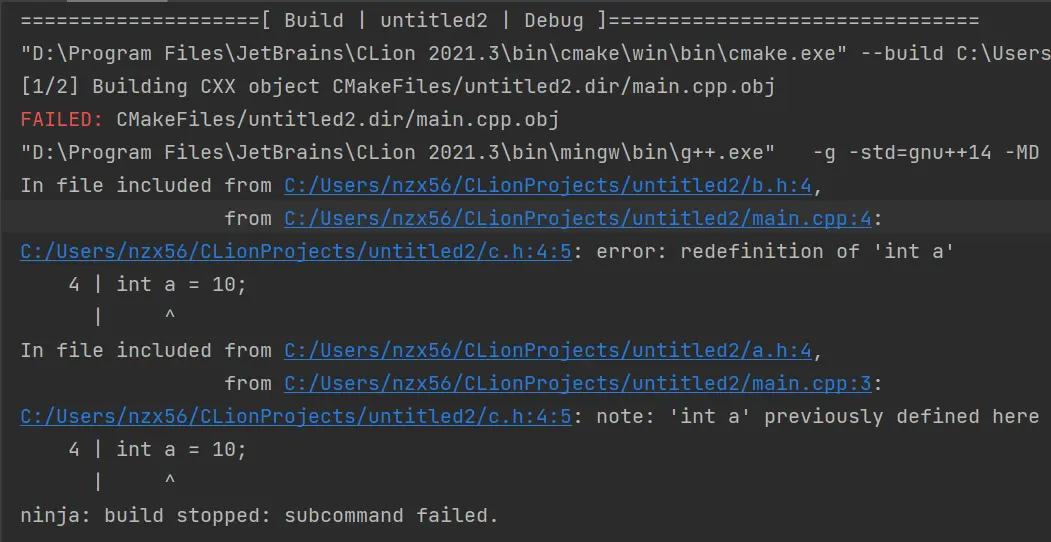


b.h



c.h

这样编译是无法通过的，因为在编译main.cpp时先运行#include "a.h"，此时编译了a.h中的c.h，也即给int a变量做了定义，后运行到#include "b.h"，此时编译了b.h中的c.h，又给int a进行了重定义，所以报了重定义的错误：



发生重定义错误

解决的方法就是，使用#ifndef系列语句块将c.h中的int a = 10;这一语句包装起来，包装结果如下：

#ifndef UNTITLED2\_C\_H

#define UNTITLED2\_C\_H

int a = 10;

#endif //UNTITLED2\_C\_H

这样就可以通过编译，main.cpp文件也可以正常运行了。所以我们在编写头文件时，尽量将所有语句放在#ifndef语句块中，防止重定义报错。至于命名规则，通常#ifndef以及#define后面接的是头文件名称的全部字符大写，如#ifndef \_\_C\_H，当然也可以自行任意命名，为编译器添加一个可识别的标签即可

参考链接：[#ifndef用法\_william\_hehe的博客-CSDN博客](https://links.jianshu.com/go?to=https%3A%2F%2Fblog.csdn.net%2Fwilliam_hehe%2Farticle%2Fdetails%2F108230885)