从 source file到 final executable需要两大步

1.source file-> object file

Translation units: cpp file 每一个cpp file都会生成一个Object file

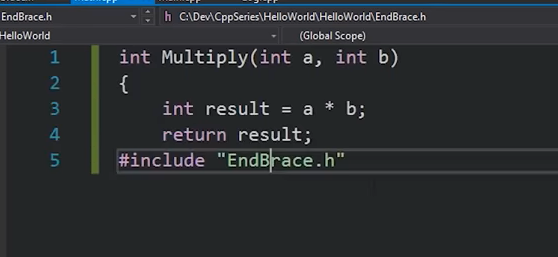
如果你用一个巨大的cpp file include了许多小cpp file,那就会产生一个大object file

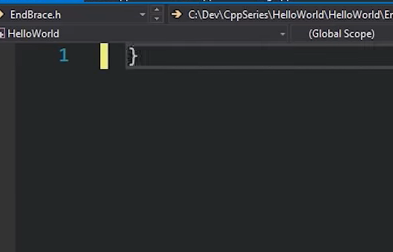
如果没有include，那么每个cpp file都会产生一个小 object file //没有include的话不能使用，但你只要F7成功，那就会产生对应.obj

Preprocessing：就是把 include里的.h文件展开，（代码一一复制），利用ifdef。。。等判定词，生成我们的.i文件 //preprocessing file

这一步是隐藏的，你一般不会看到.i文件 按f7会生成.obj文件

Header file怎么用的



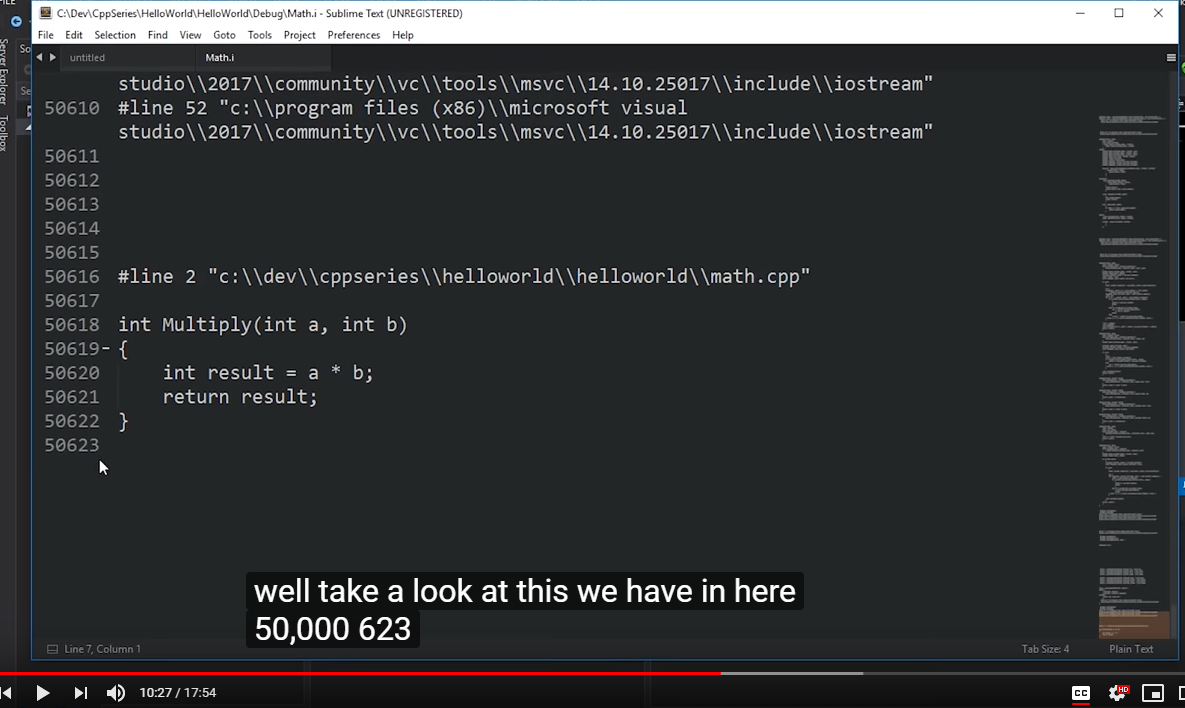


如果没有include，那么这个cpp file不能运行，因为没有最后一个} 符号，而我们这里include了这个.h文件

这个.h文件超级简单，就是一个}

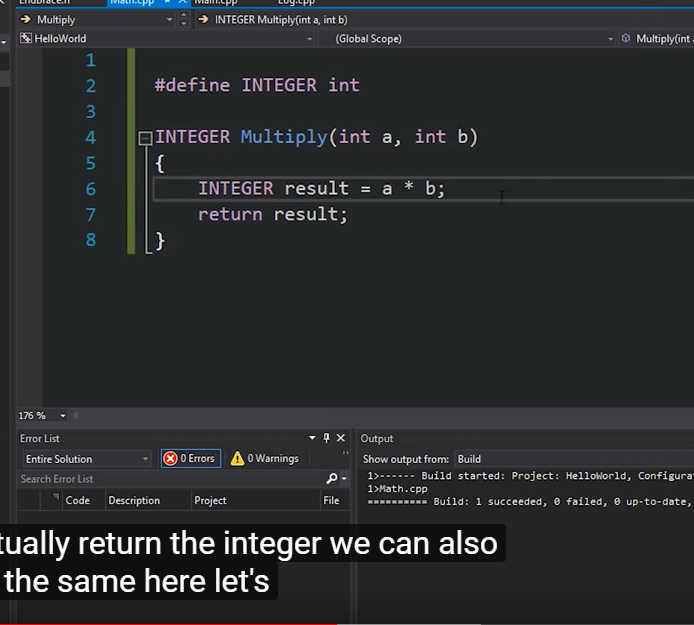
然后compile成功

说明include .h 实际上就是把.h里的代码完全复制进来

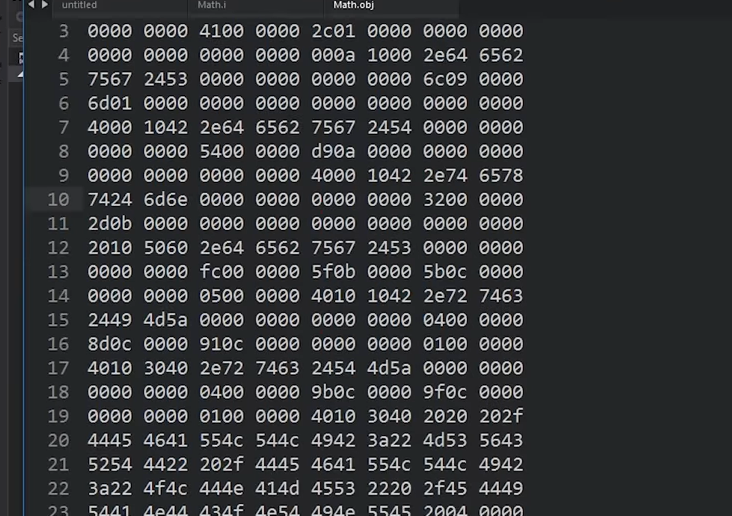


平时的一个小小 Include iostream

实际上复制了50000行代码



Define 你想要的字符 被替代的字符



实际生成的.obj文件，也就是machine code

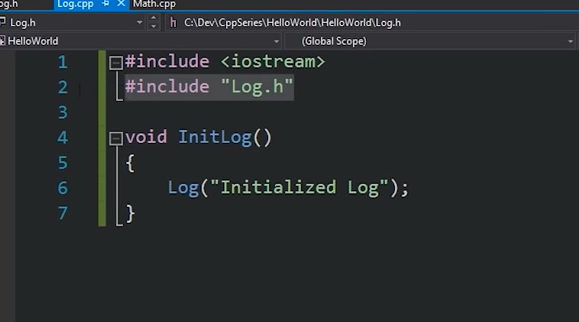
2.object file through linking->executable

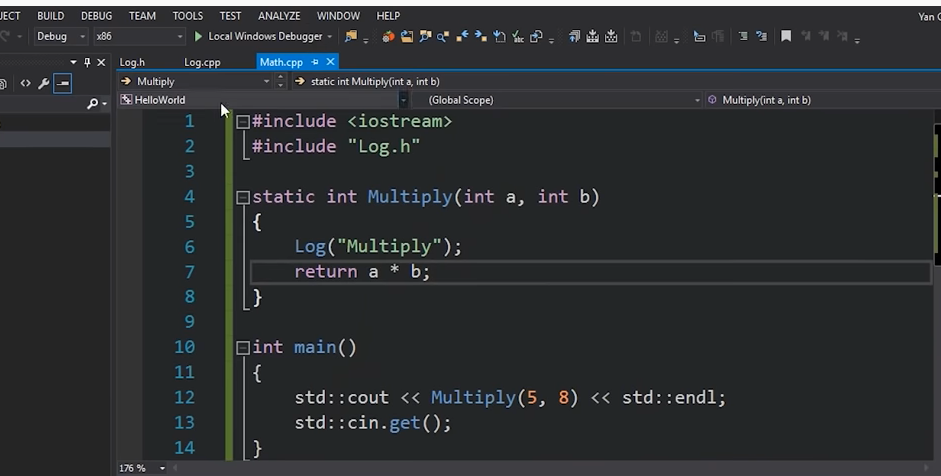
按f7生成.obj file，不会link

按f5生成.exe 会link

如果没有 main() function，会产生linking error,F5 FAIL

Linking实际上就是把所有function method class啥的整合到一起，所以如果有两个cpp file 描述了相同的function，就会产生Linking error，因为他不知道该整合哪个function



j

假设这里两个都include 了log.h

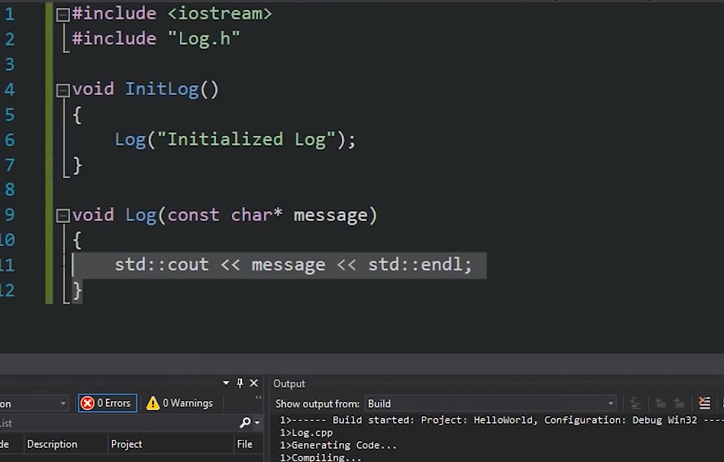
会产生linking error

为什么

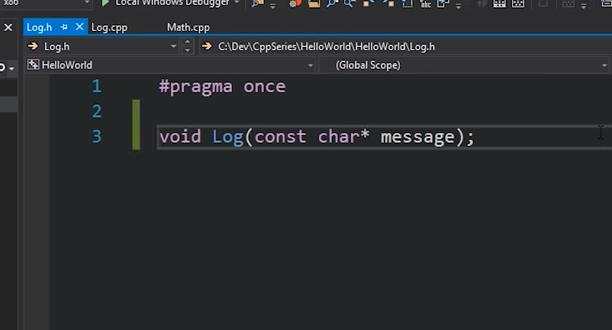
因为log.h没有加if def

所以log.h实际上是把这个Log function 完全复制下来了，就define了两次，就有Linking error

最好的方法



Cpp 描述



.h declare