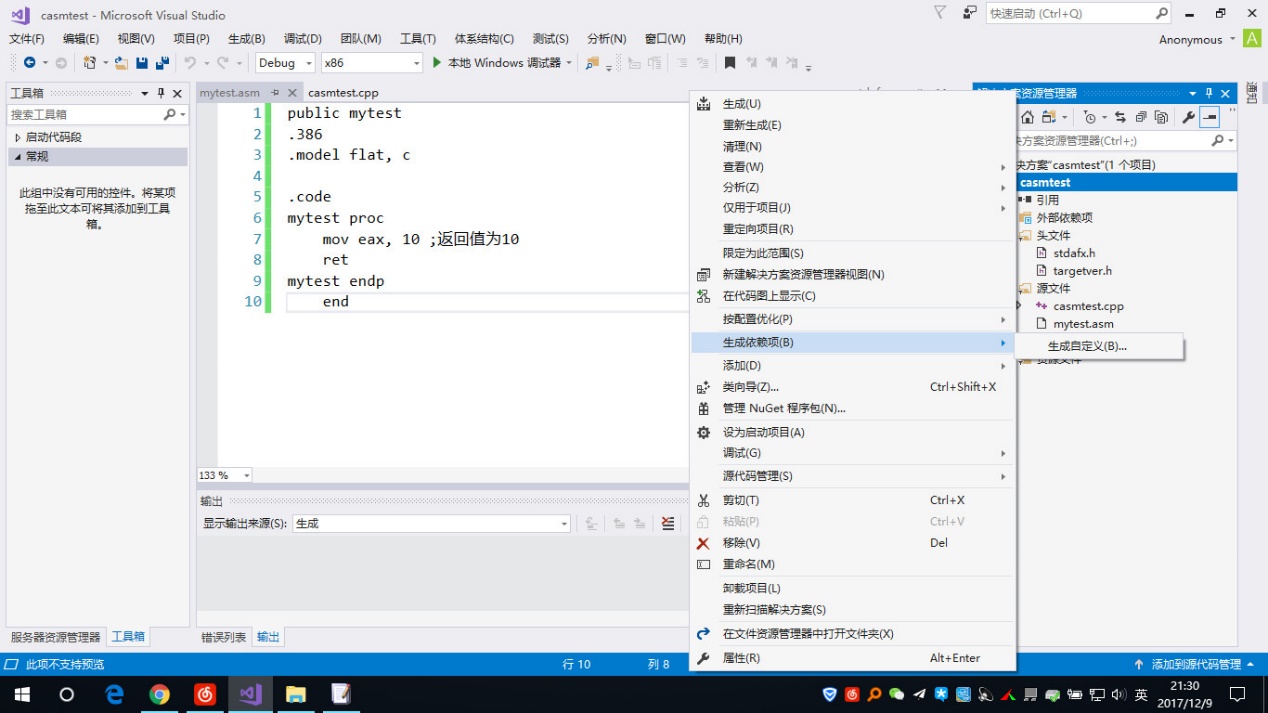
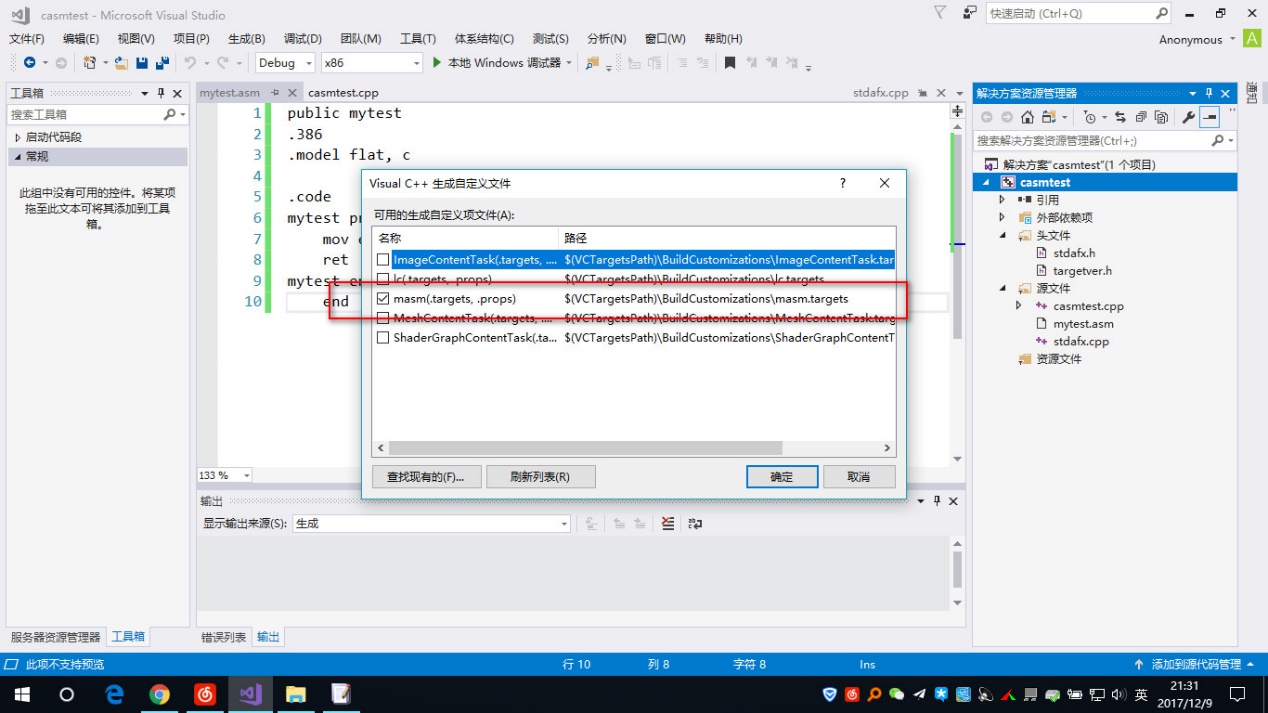
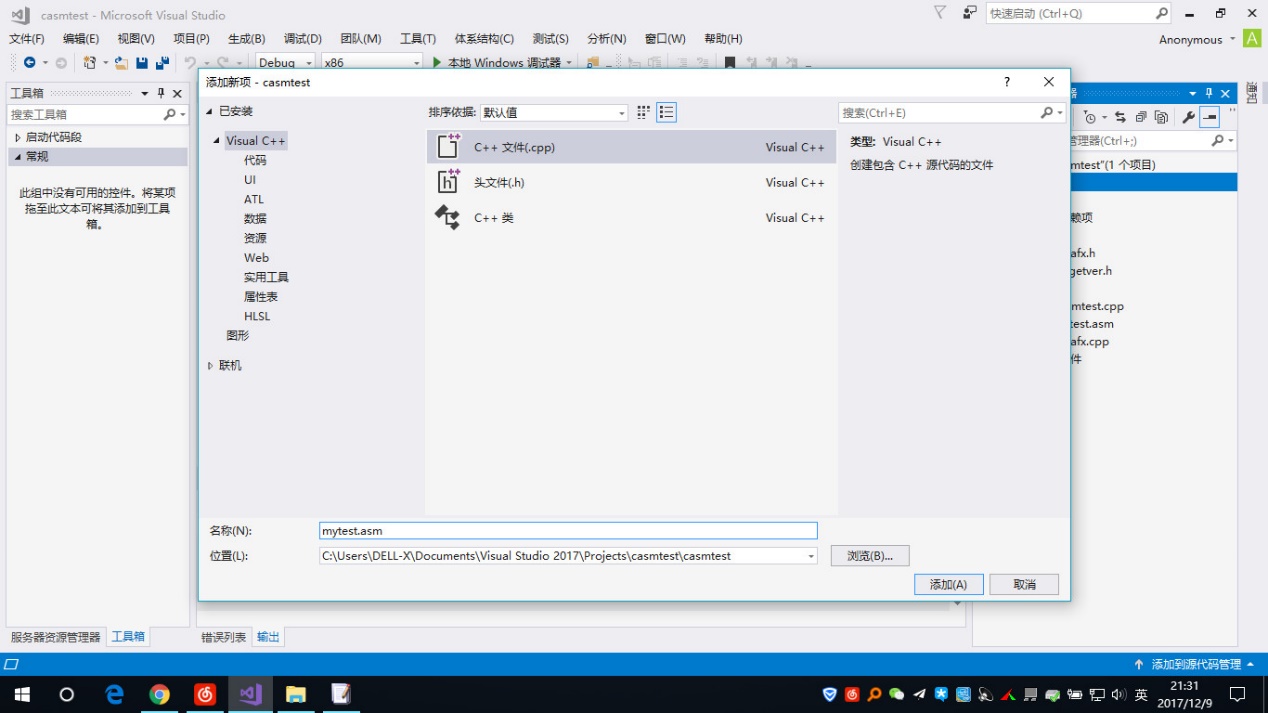
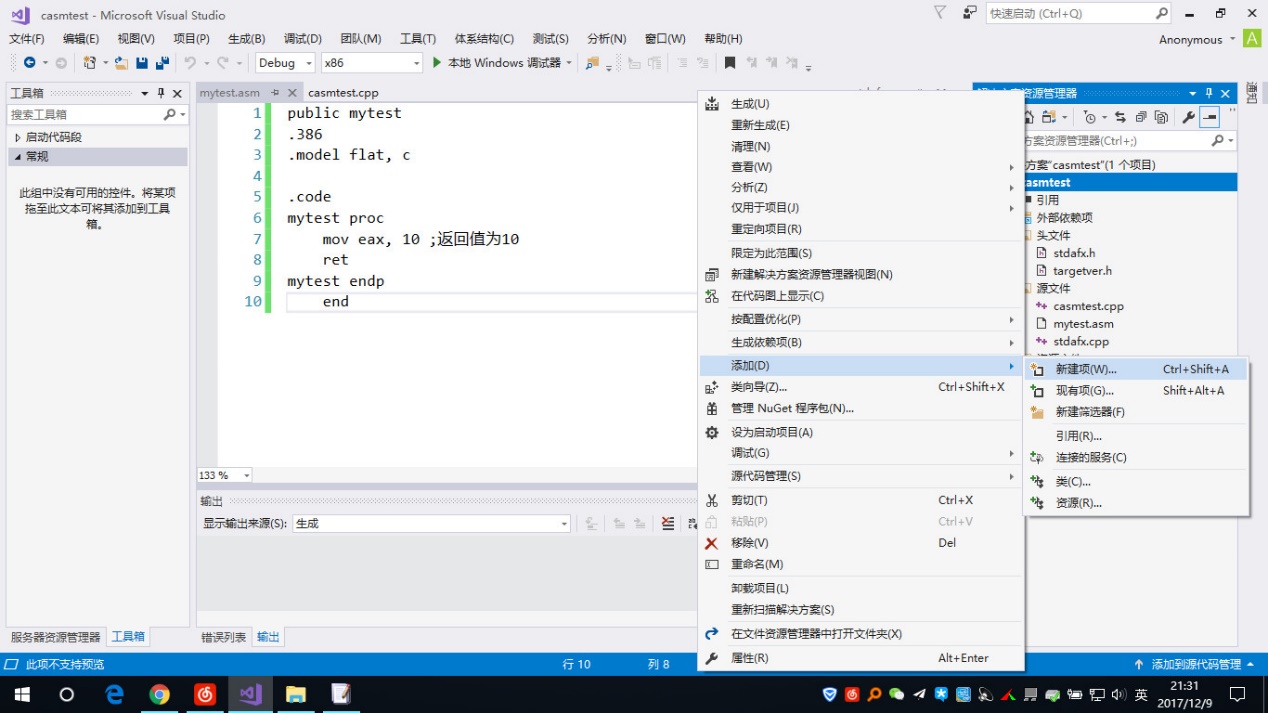
## 1.首先 生成自定义(顺序很重要)



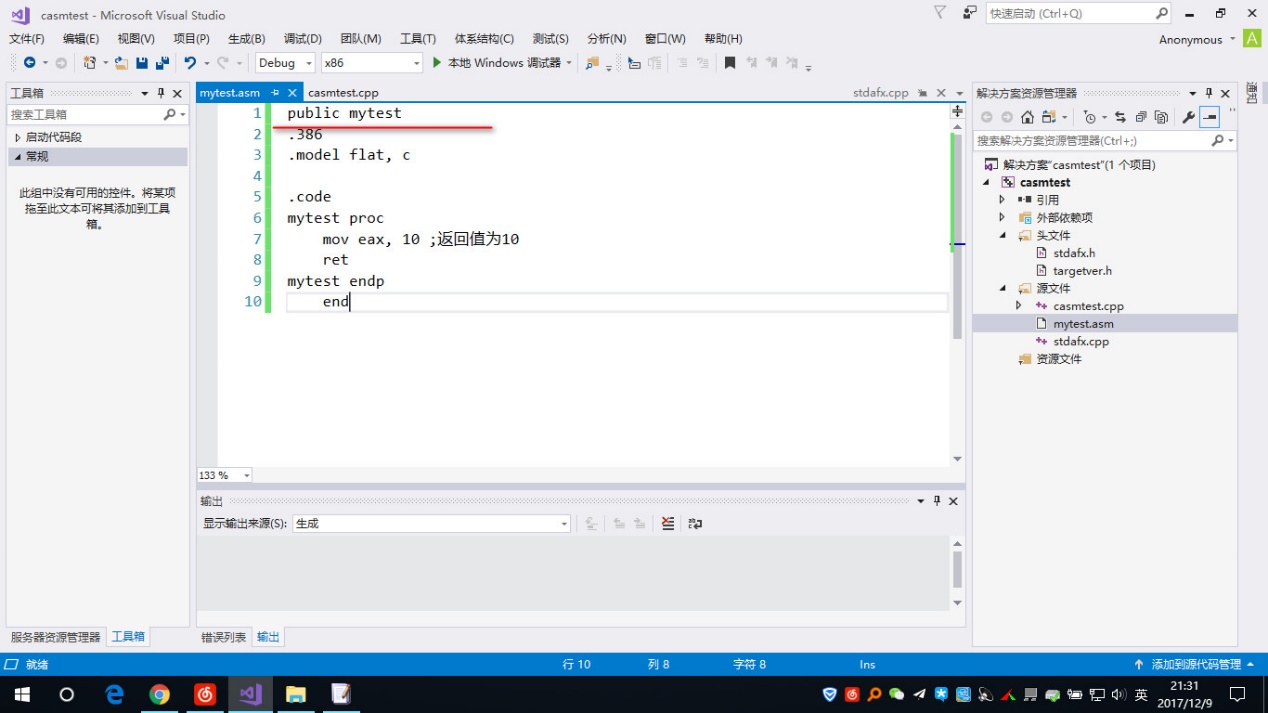
## 2.选中masm编译选项



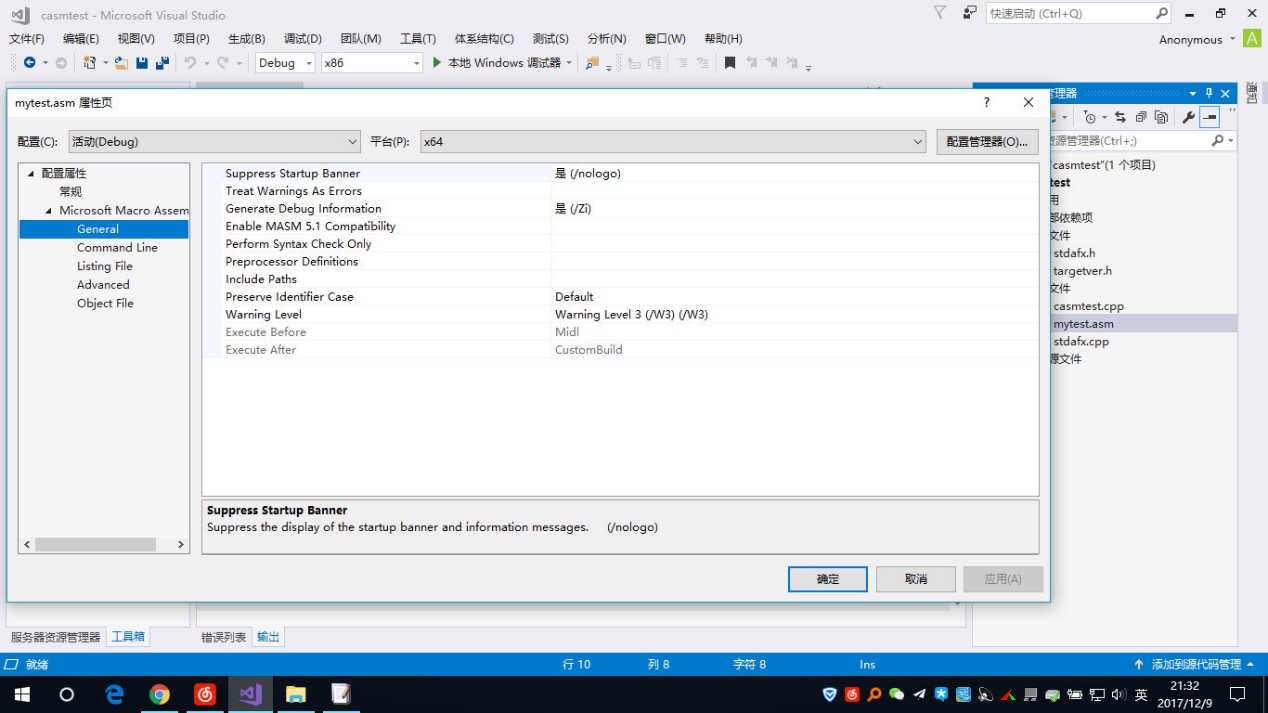
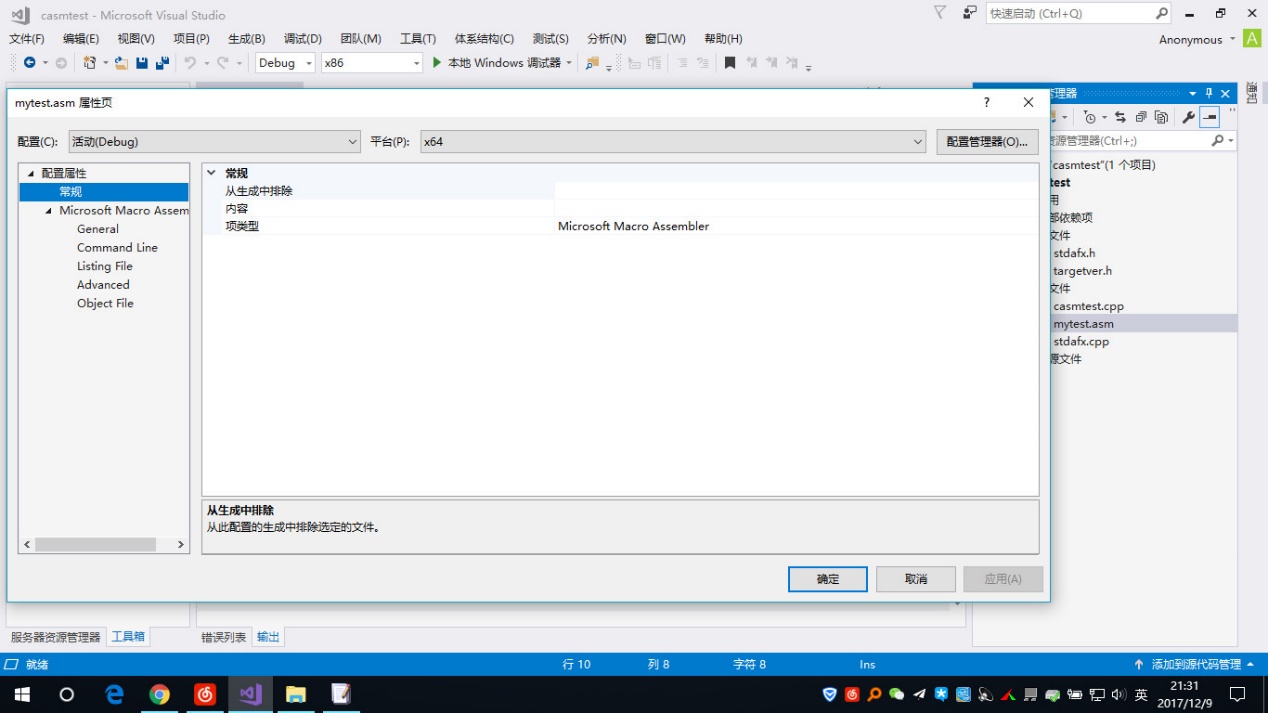
## 3.创建asm文件利用现有的cpp文件更改后缀名



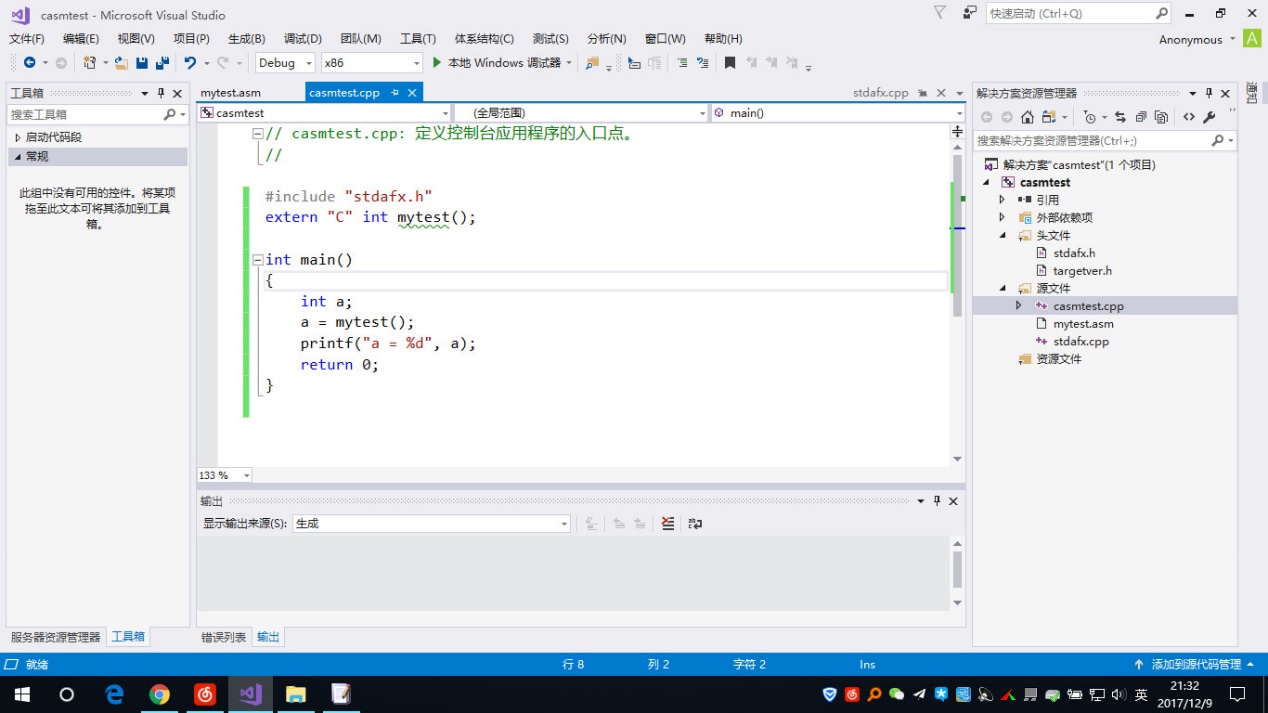
## 4.在汇编文件中将函数进行导出



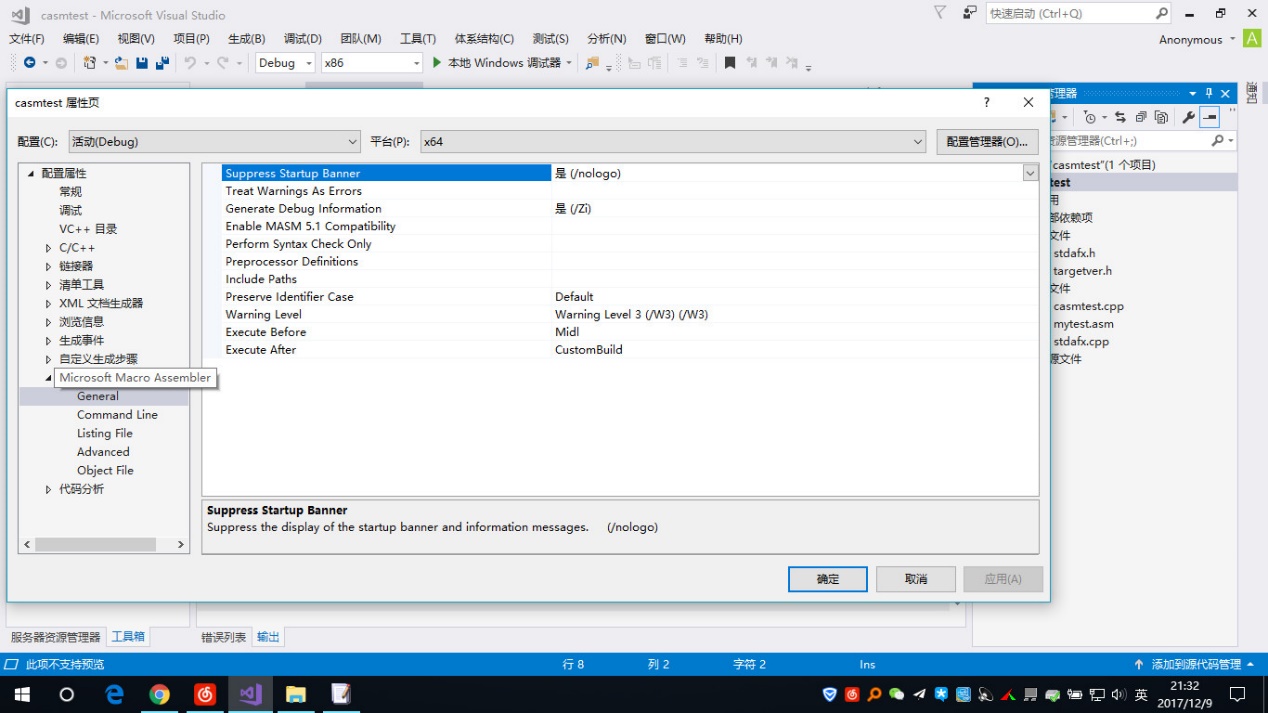
## 5.右键汇编文件设置编译属性



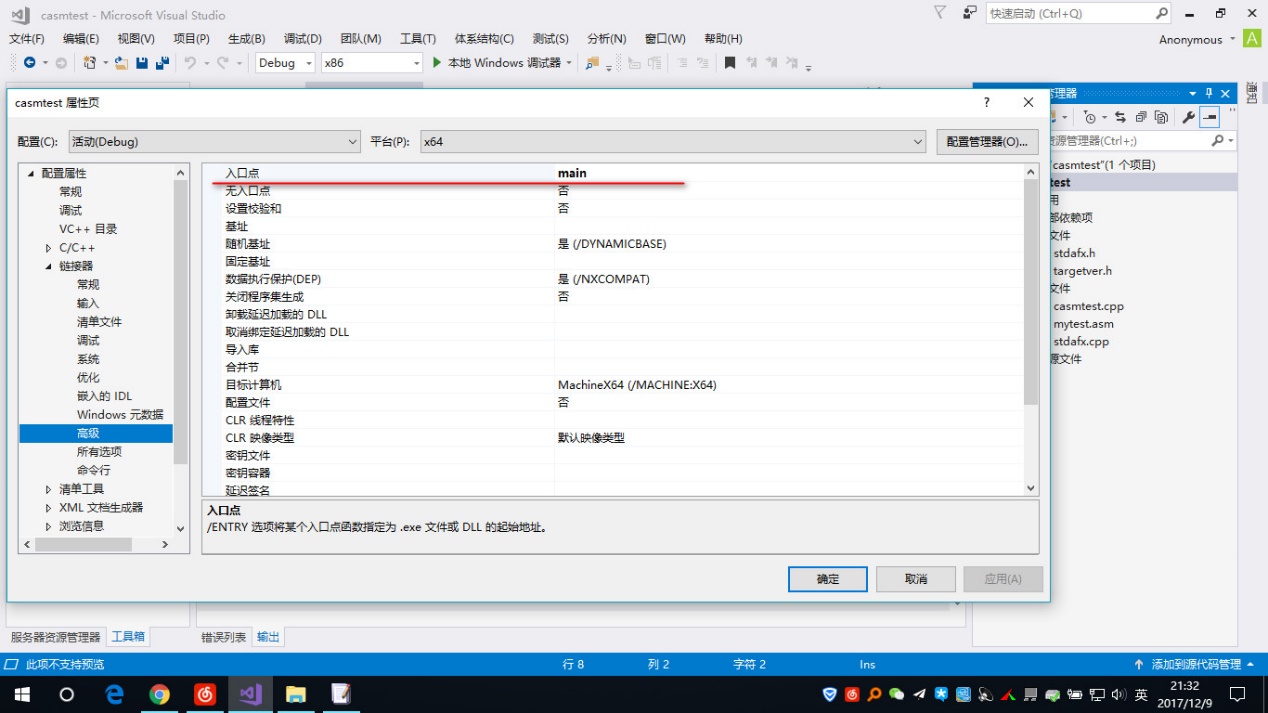
## 6.编写调用汇编文件的c++文件



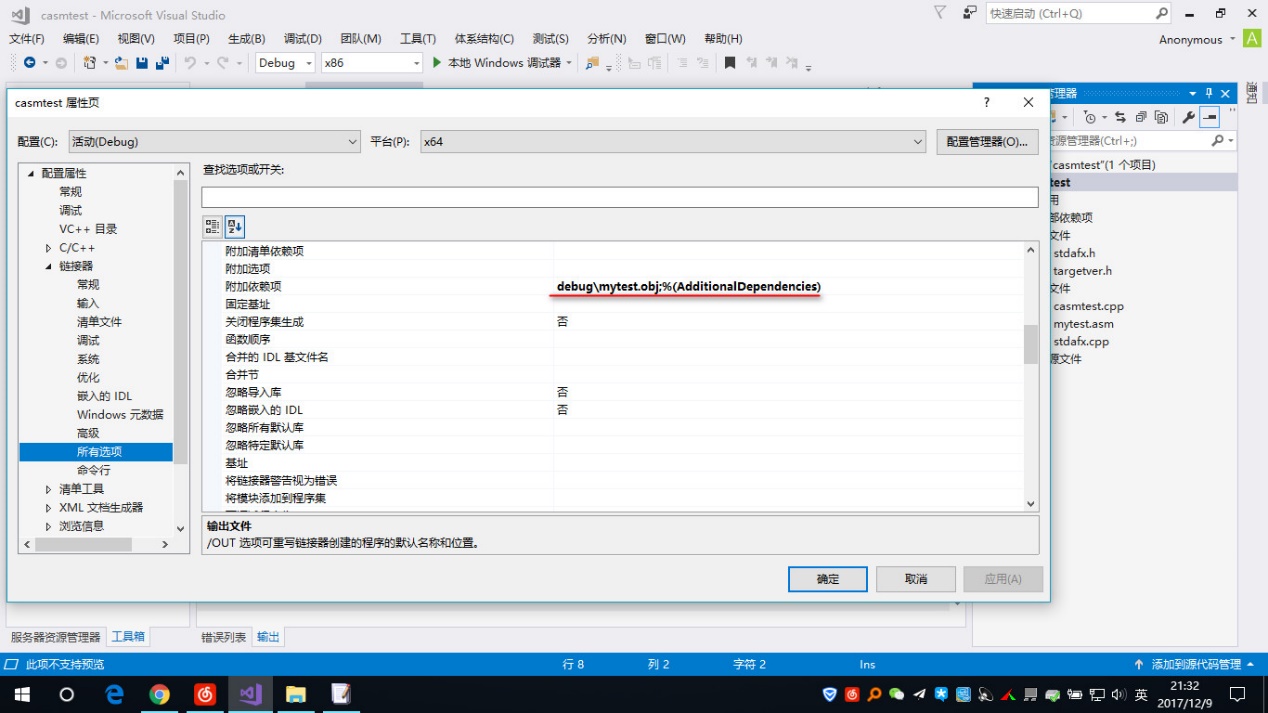
## 7.右键项目设置c++项目的编译选项



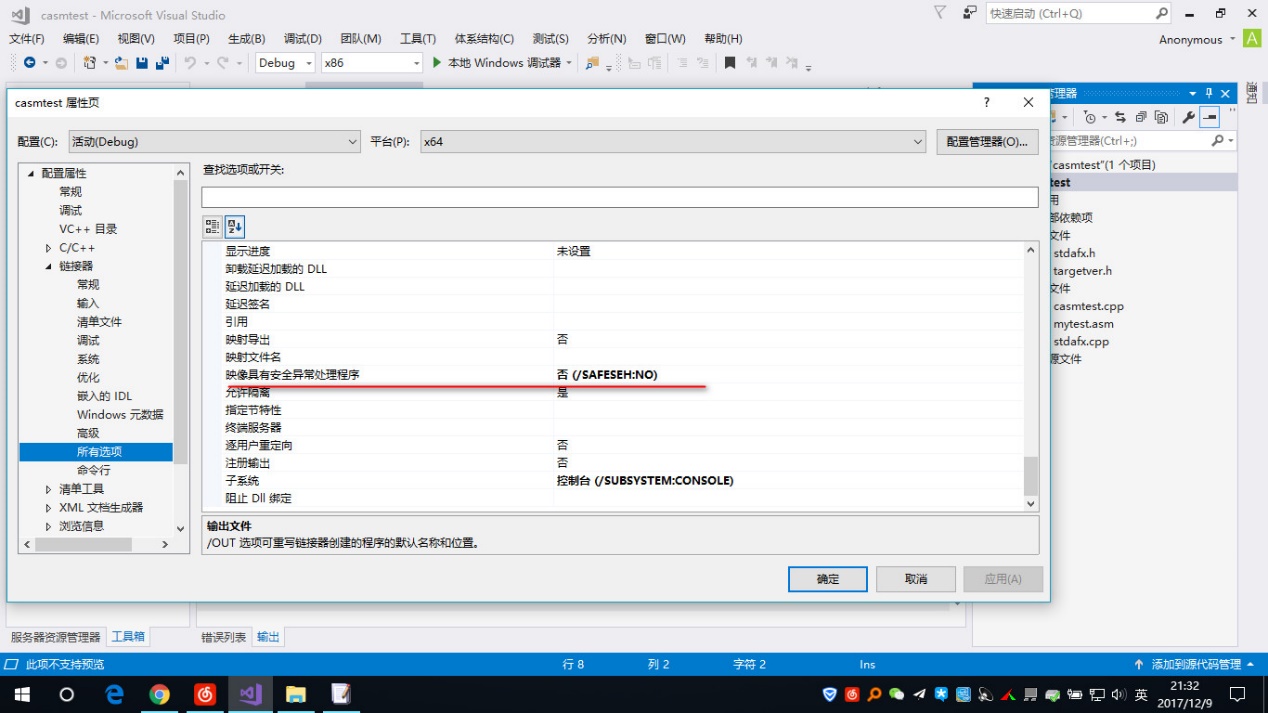
### 7.1 手动指明文件入口



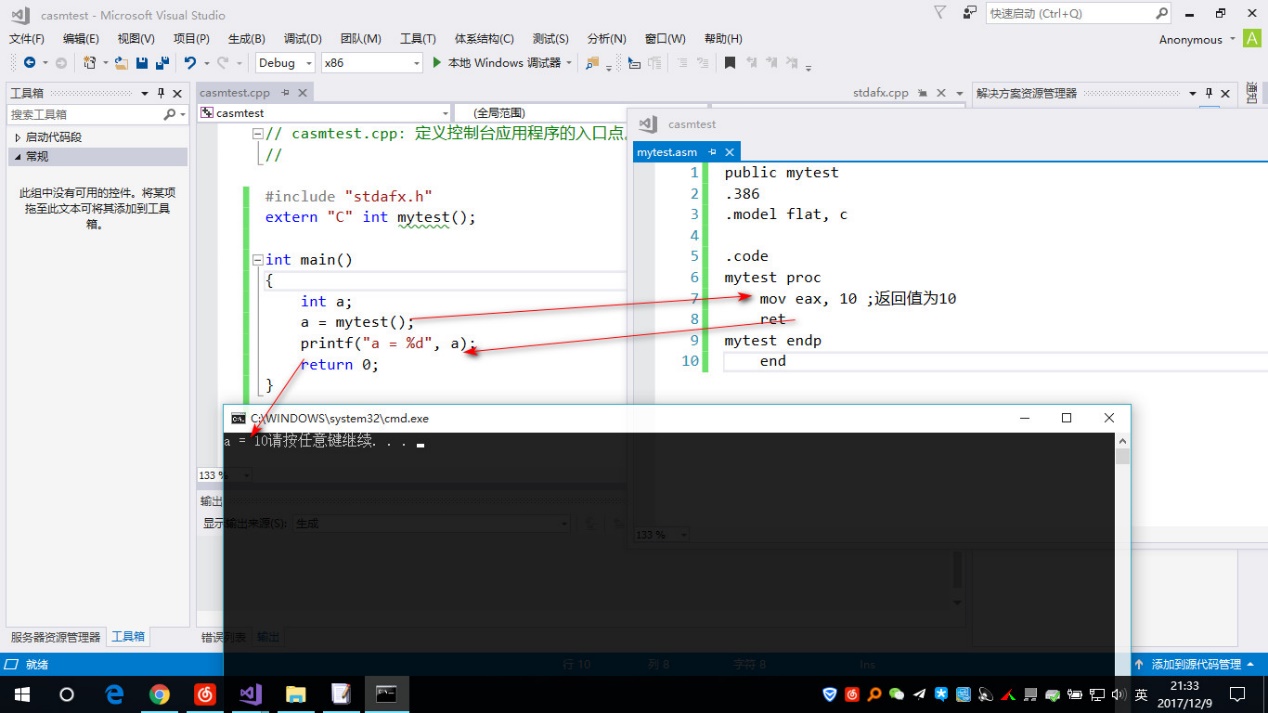
### 7.2 指明汇编文件后的编译格式 obj文件



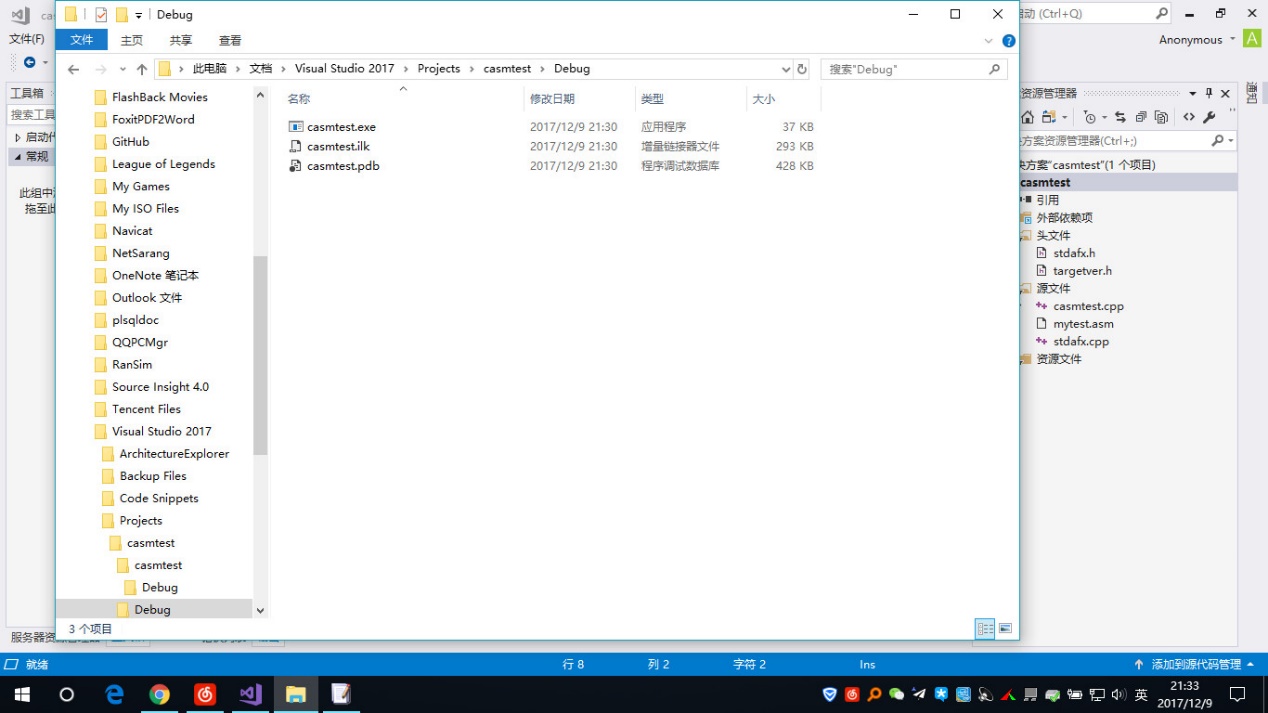
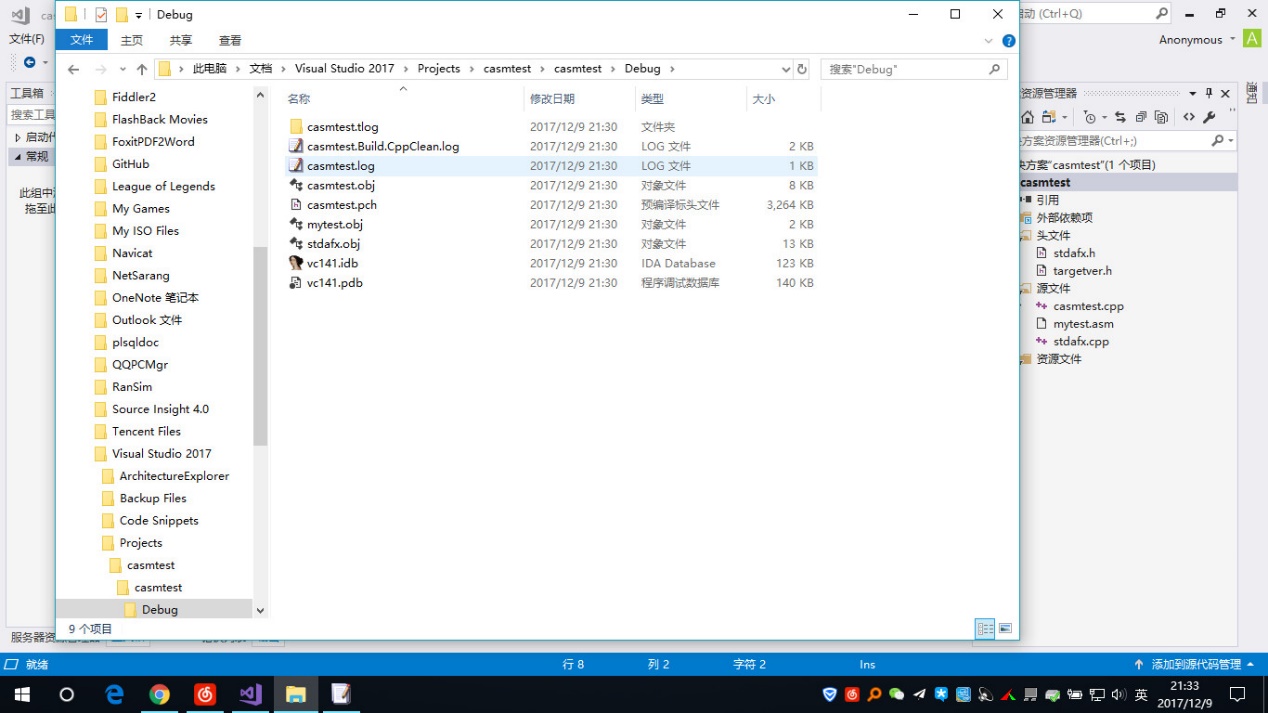
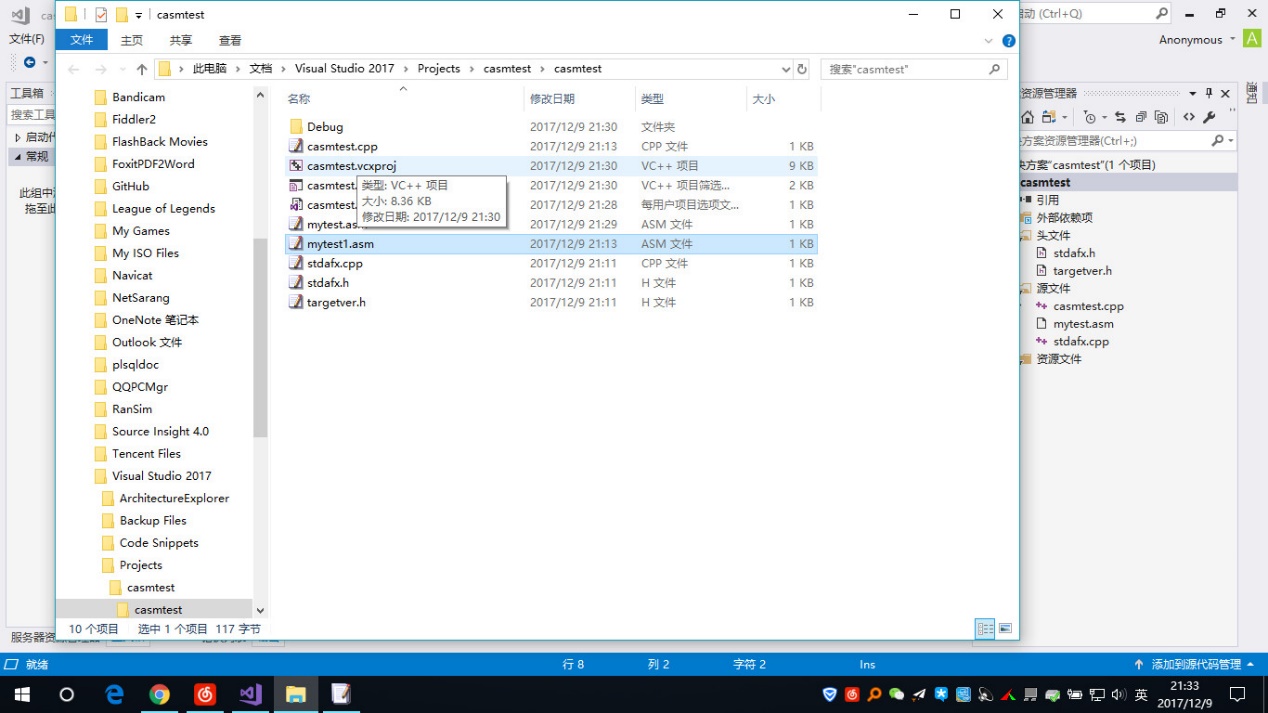
### 7.3 关闭 映像保护



## 8. 直接生成项目 ctrl+f5 查看结果



## 9.项目磁盘文件结构说明



## 总结

1. 不可以先创建asm文件,然后添加自定义依赖,否则会出现编译的asm 后的obj文件路径不对,以及不可以编译asm文件的异常
2. 在asm右键的时候,可以选择自定义的编译选项 如masm编译器,也可以使用vs提供的默认的microsoft 编译选项
3. 在进行项目整体编译的时候需要指明asm编译后的路径和文件名
4. 通过附加依赖项指明obj文件的添加路径
5. 如果出现 无法打开 mytest.obj 的异常说明,在设置的项目时的操作顺序出现了问题
6. 如果出现 \_main 无法解析符号,表示obj文件没有正常引入,可能是编译选项出现了问题,也可能是没有将导出函数声明为public
7. C++中引用asm中的函数应该使用extern关键字 进行函数头的声明
8. 从磁盘上的文件路径可以看出编译文件的顺序是在源文件的deubg目录下进行区域编译,然后再项目的debug目录下输出一个exe文件
9. Vs2017很智能 知道先编译asm文件,然后在编译项目,说明extern关键字起到了预编译的作用
10. 编译环境说明 vs2017+ win10+ win64