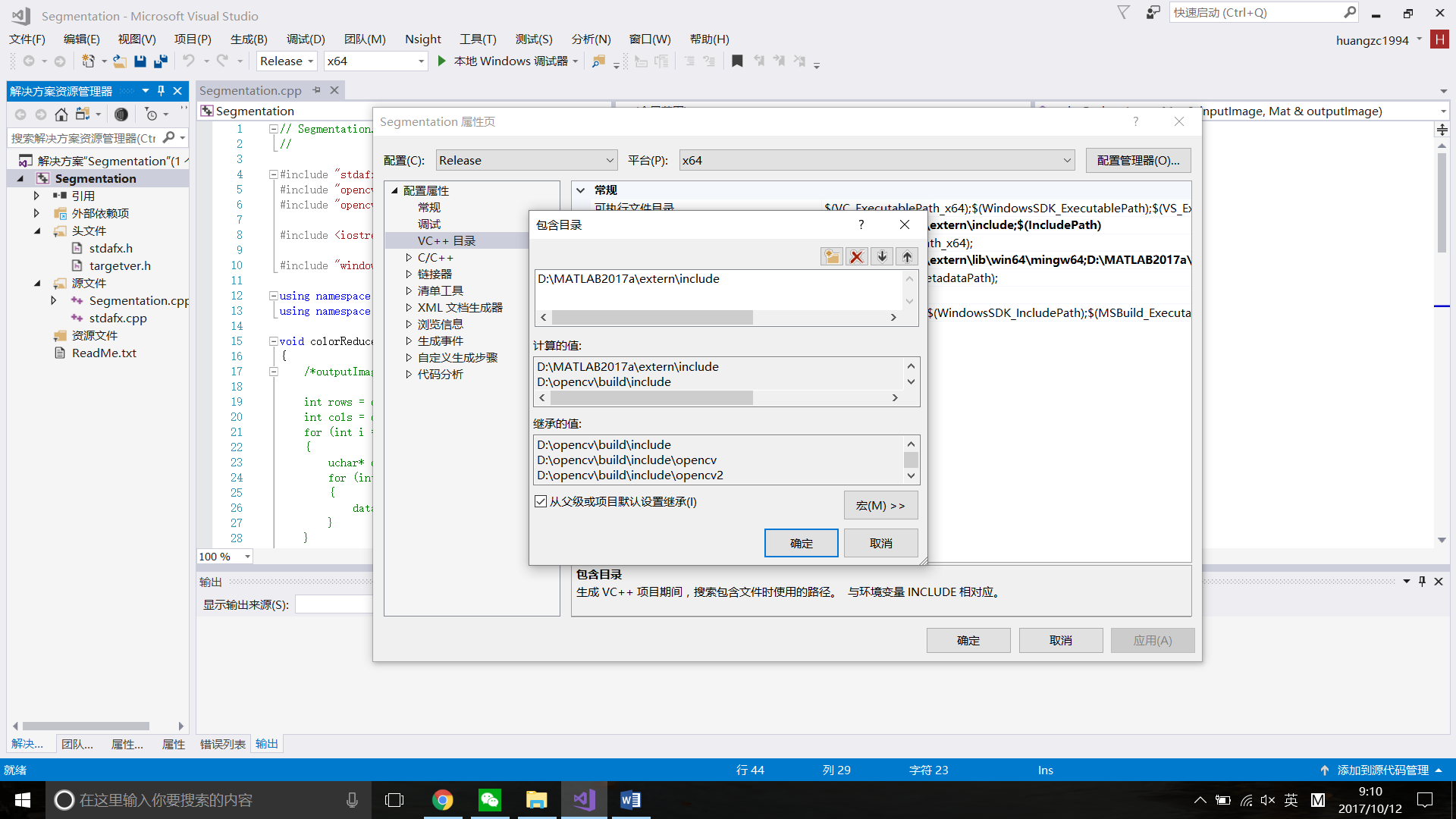
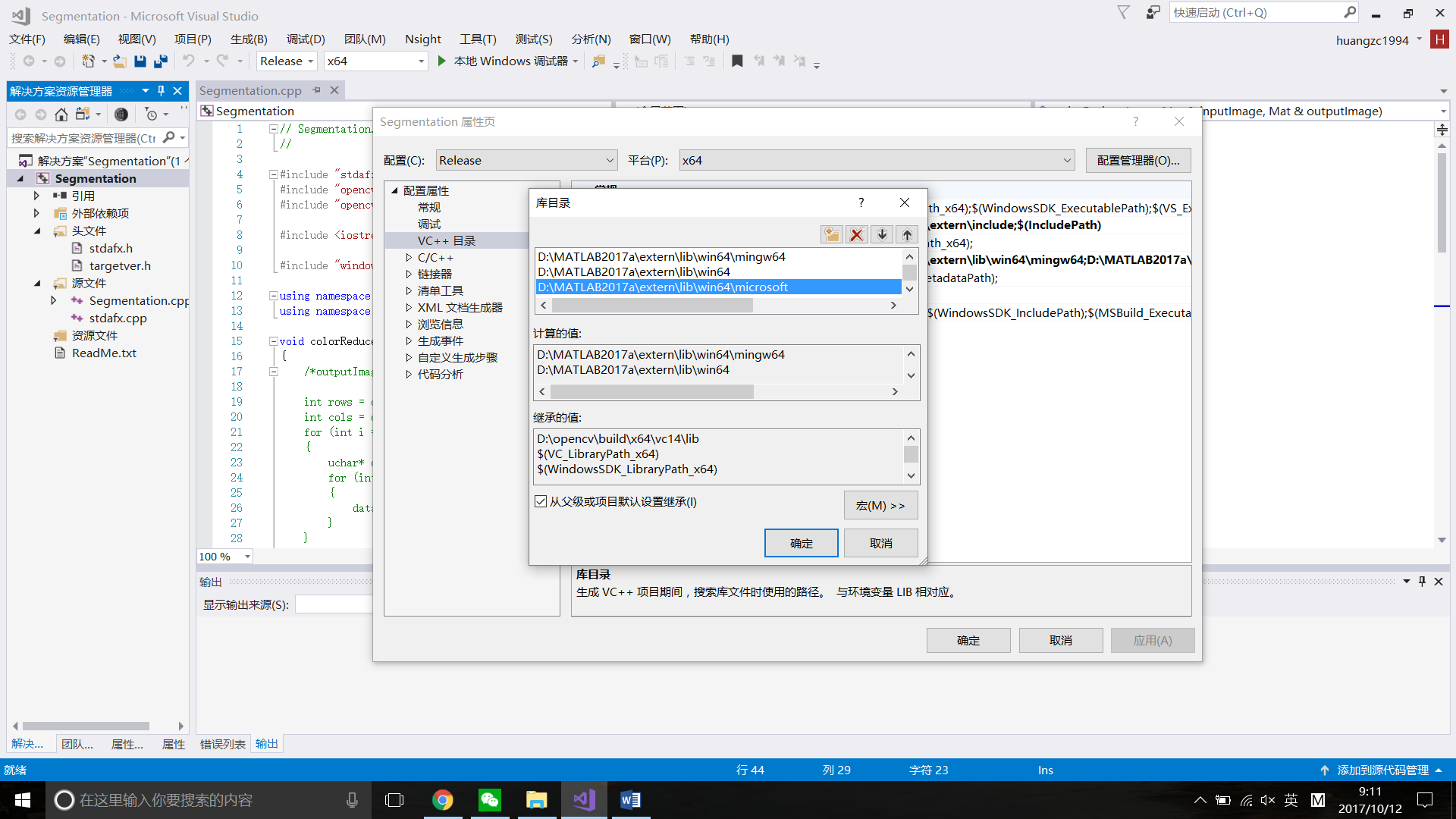
C++调用MATLAB

1.属性-配置：

VC++目录，包含目录设置到matlab的include目录下，如图所示



库目录中包含MATLAB中lib目录下的win64（64位系统）及其子文件夹目录，如下图所示



链接器-附加依赖项中增加如下内容，如下图所示。

mclmcr.lib

mclmcrrt.lib

mclbase.lib

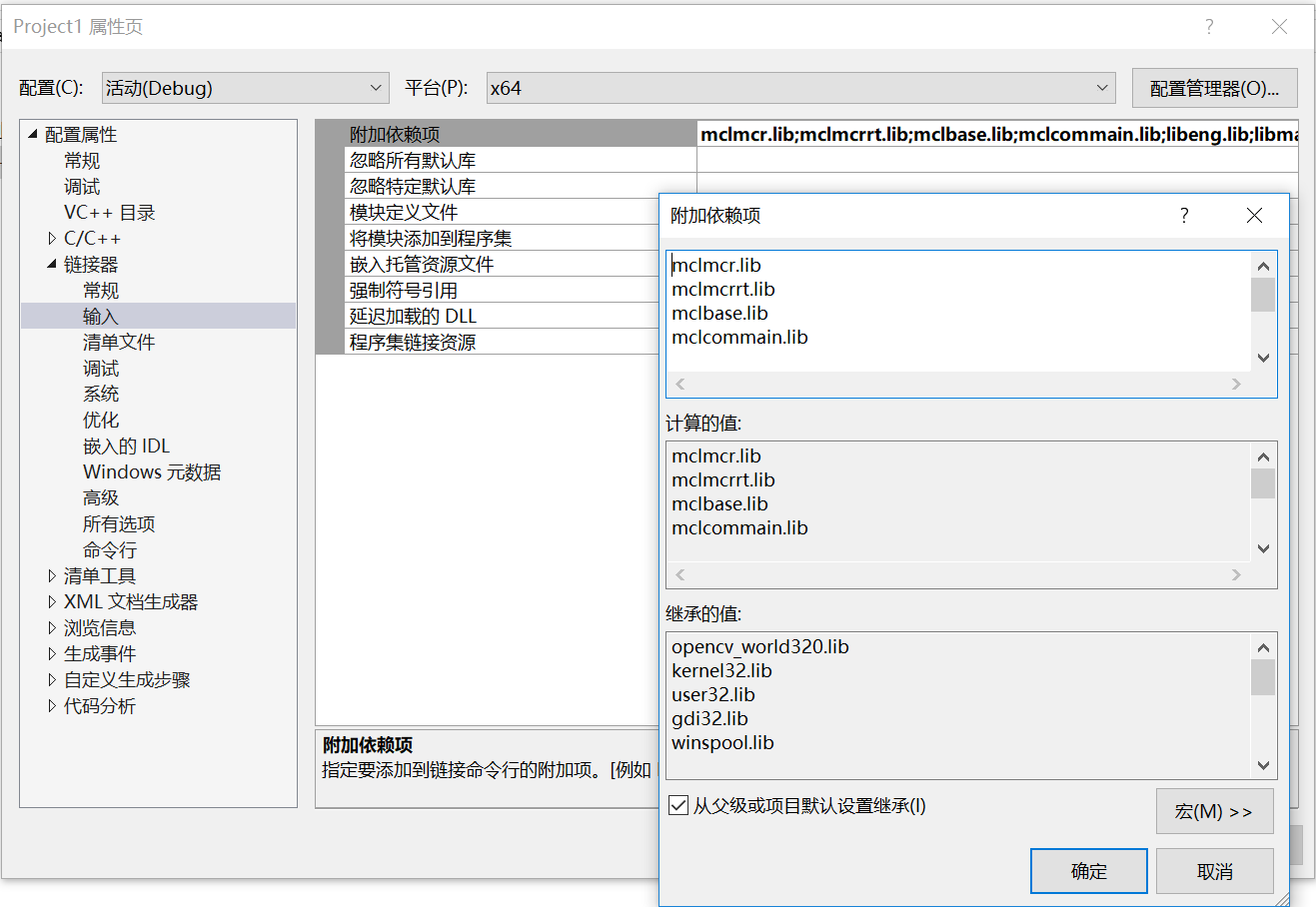
mclcommain.lib

libeng.lib

libmat.lib

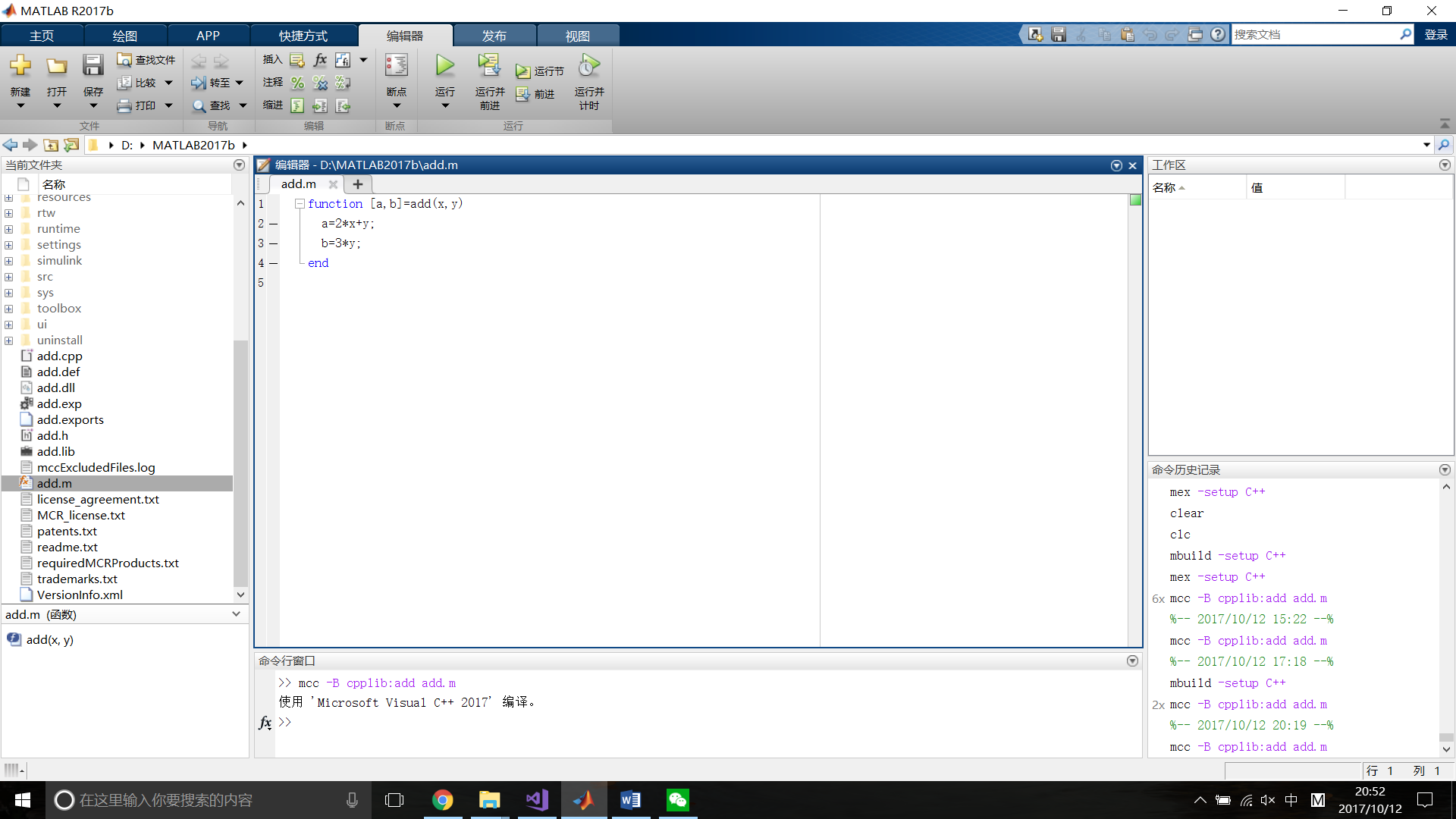
libmx.lib

libmex.lib



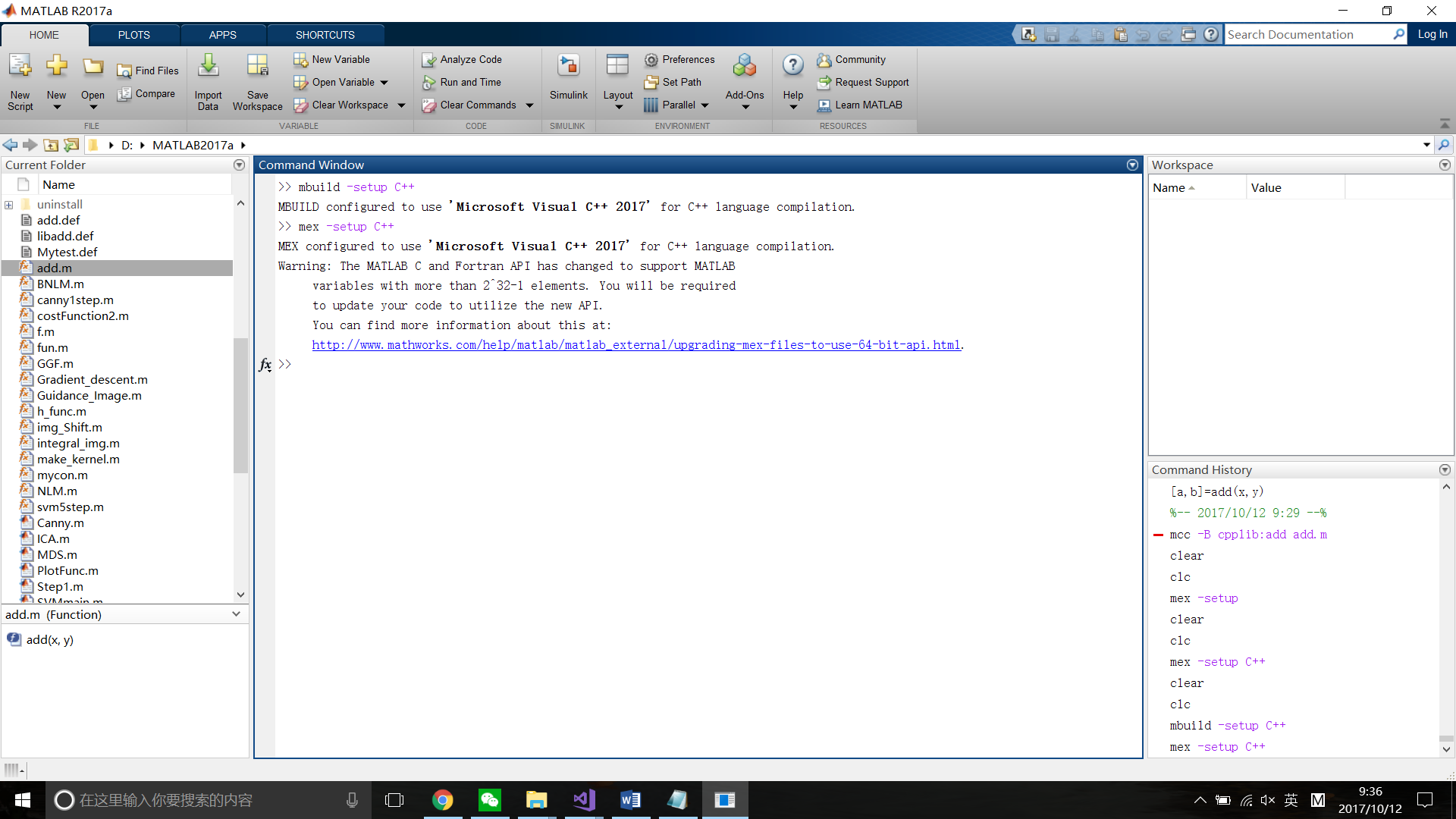
测试1，调用动态链接库DLL文件的形式：

在matlab中新建函数add

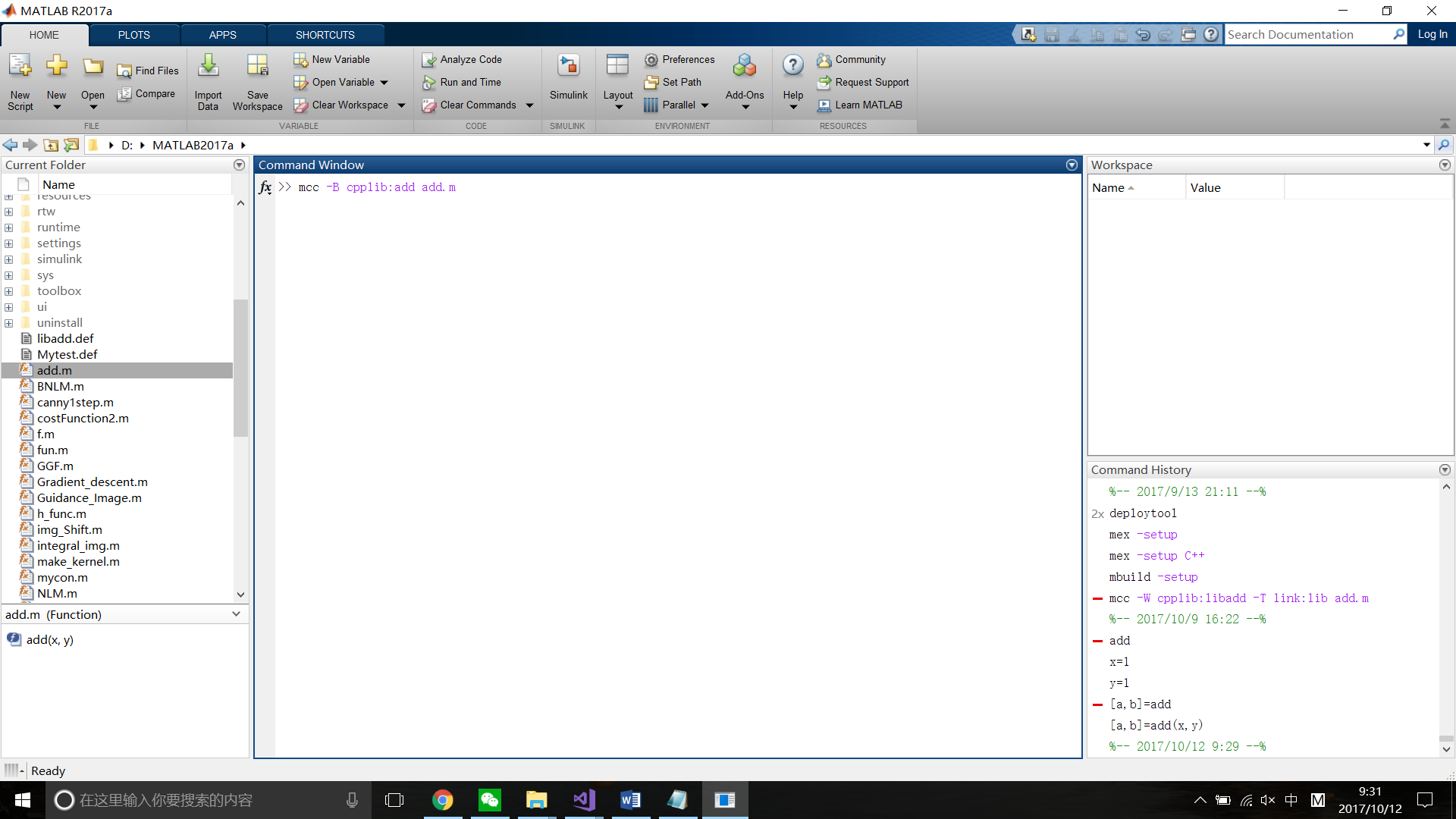


生成dll文件:

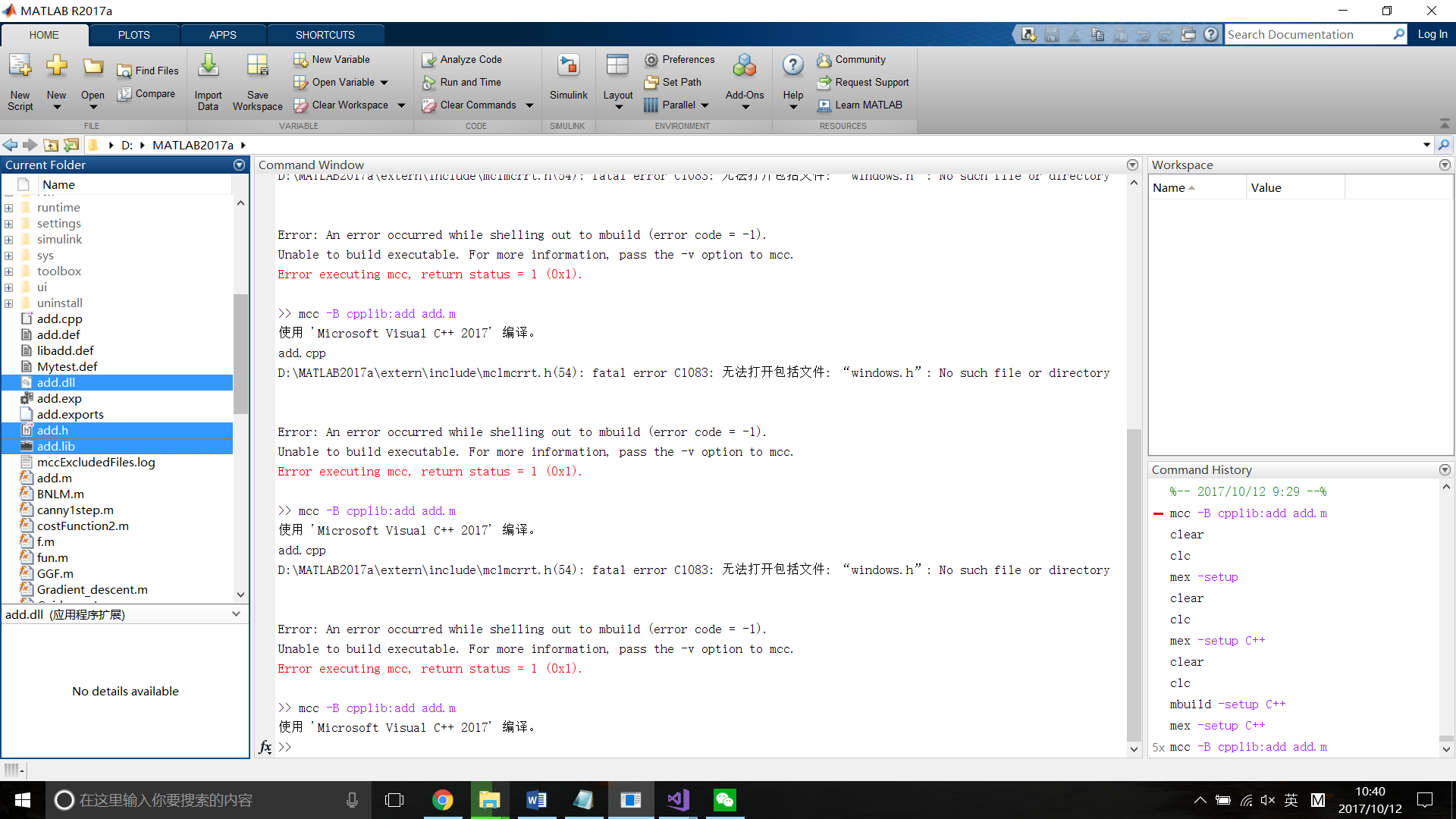
命令窗口输入mbuild –setup C++和mex–setup C++命令，选择VS2017编译器



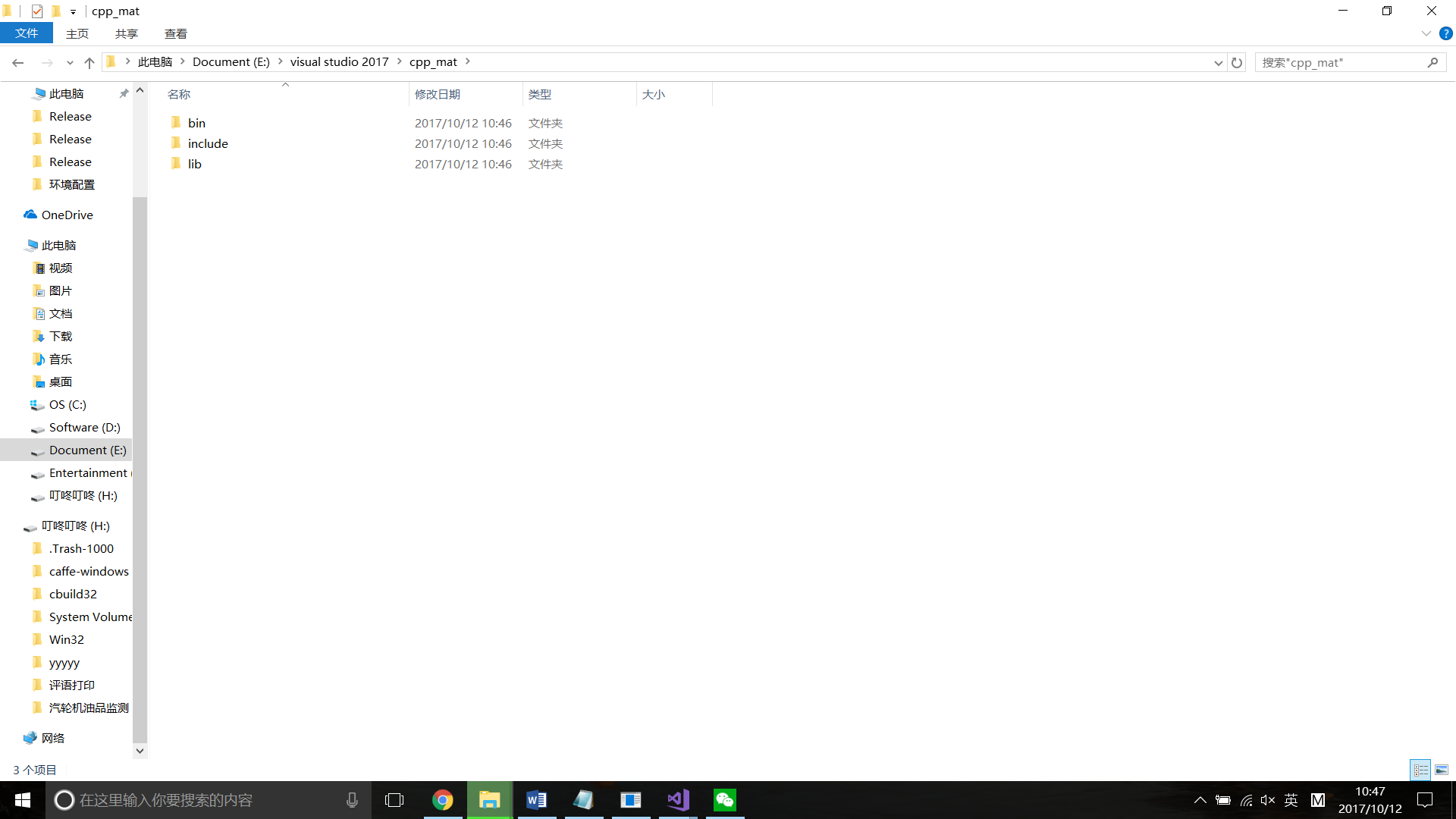
输入命令：mcc -B cpplib:add add.m或mcc -W cpplib:add -T link:lib add.m



提取add.h，add.dll，add.lib 共3个文件



新建文件夹存放生成的上述三个文件，分别命名为bin、include、lib对应.dll、.h、.lib文件，将其如上述所示添加到VS的属性配置（包含目录，库目录，附加依赖项）中



在VS中写入测试代码

#include "stdafx.h"

#include "add.h"

#include<iostream>

using namespace std;

int main() {

if (!addInitialize()) {

cout << "initilize failed!!!" << endl;

return -1;

}

mwArray a(1, 1, mxDOUBLE\_CLASS);

mwArray b(1, 1, mxDOUBLE\_CLASS);

a(1, 1) = 20;

b(1, 1) = 30;

mwArray x(1, 1, mxDOUBLE\_CLASS);

mwArray y(1, 1, mxDOUBLE\_CLASS); //输出参数个数，输出参数，输入参数

add(2,x,y,a,b);

double \*i=new double;

double \*j=new double;

x.GetData(i,1);

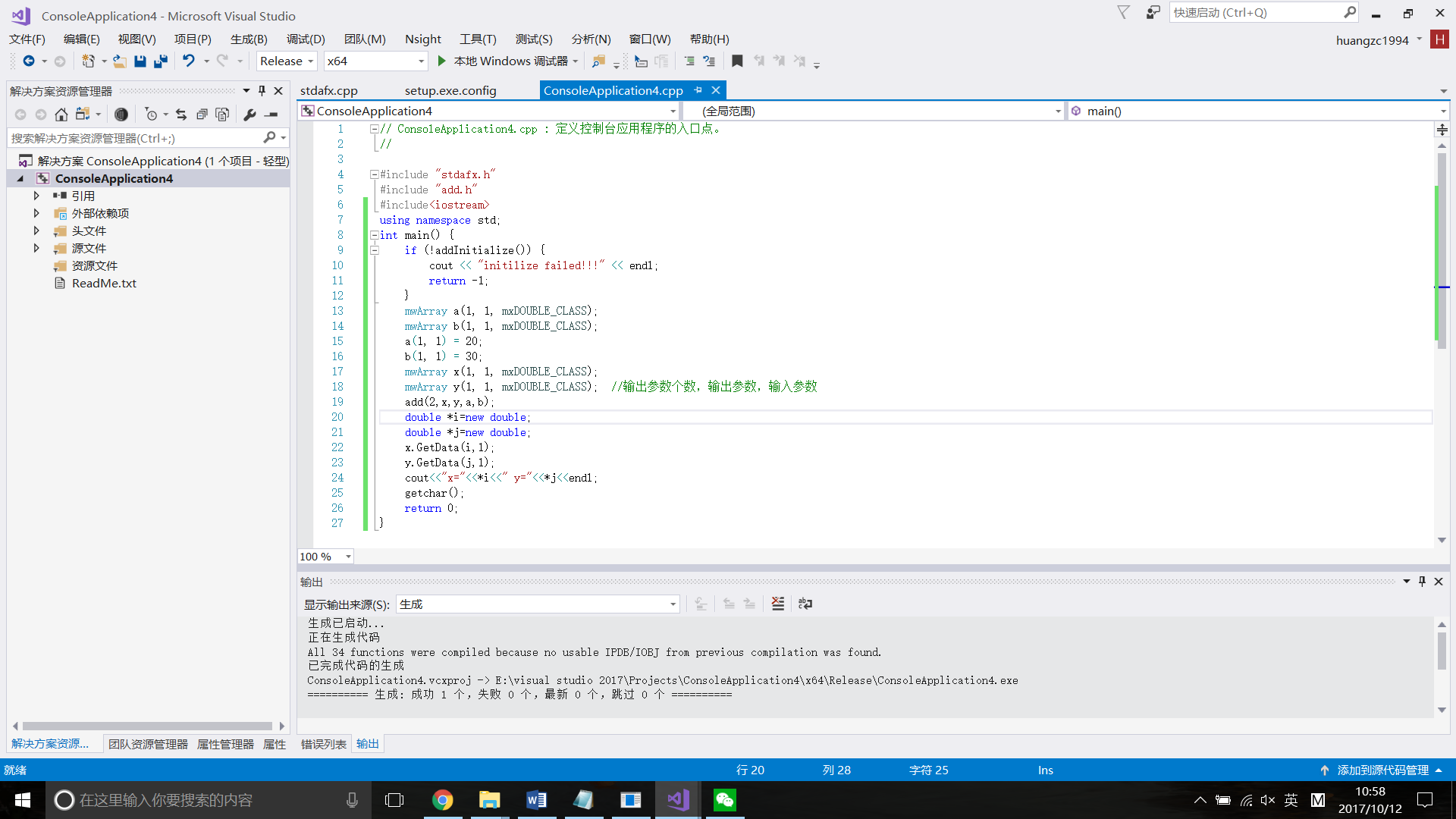
y.GetData(j,1);

cout<<"x="<<\*i<<" y="<<\*j<<endl;

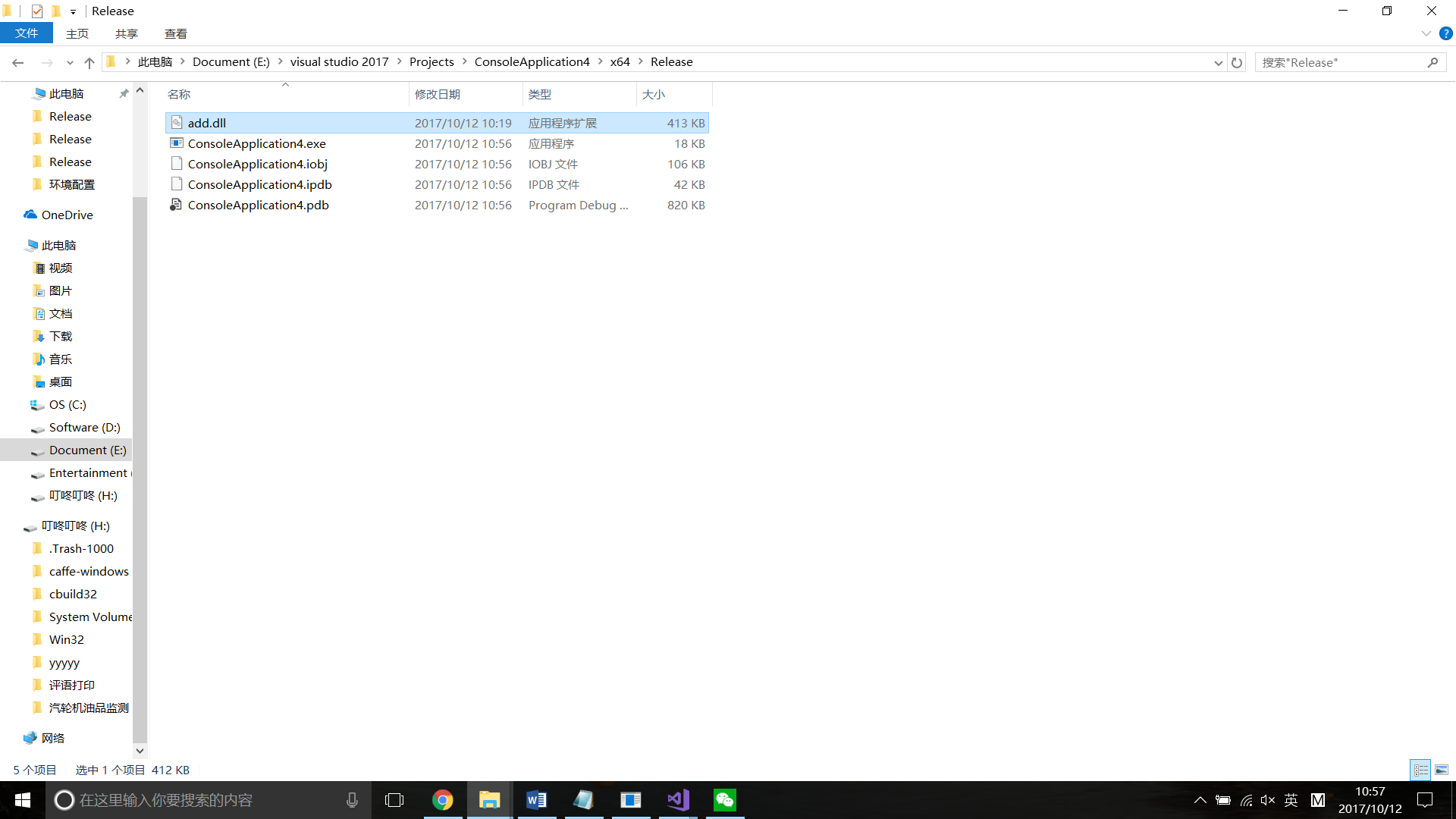
getchar();

return 0;

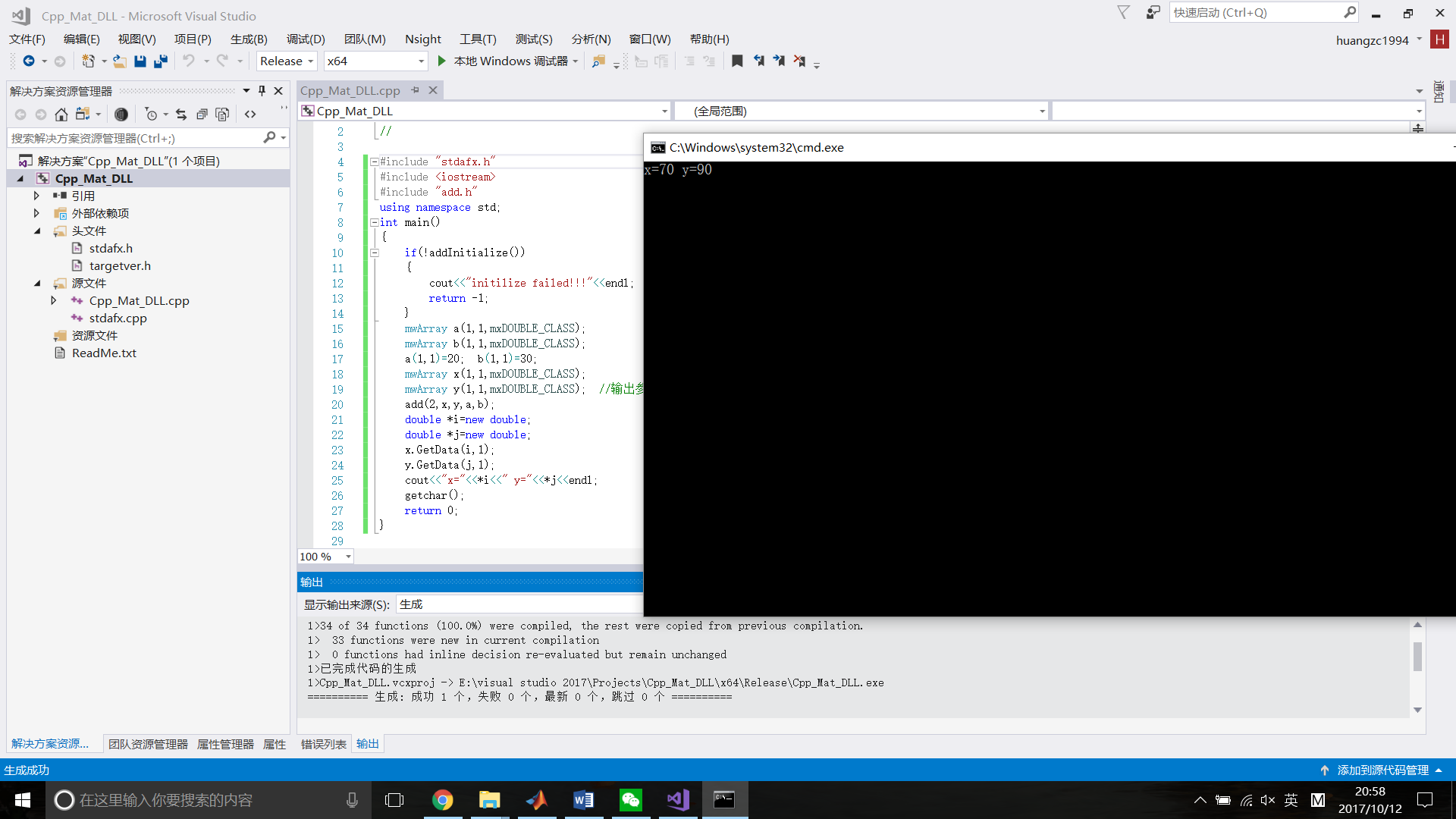
}



开始执行不调试，报错无法找到可执行文件，把.dll文件拷贝到可执行文件目录下



再次执行，成功显示计算结果。



测试2，直接调用MATLAB引擎

输入如下代码：

#include "stdafx.h"

#include<iostream>

#include<Engine.h>

using namespace std;

int main() {

Engine\* m\_pEngine;

m\_pEngine = engOpen(NULL);

if (m\_pEngine == NULL) {

cout << "error!" << endl;

exit(-1);

}

engEvalString(m\_pEngine, "x=0:0.01:2\*pi;");

engEvalString(m\_pEngine, "y=sin(x);");

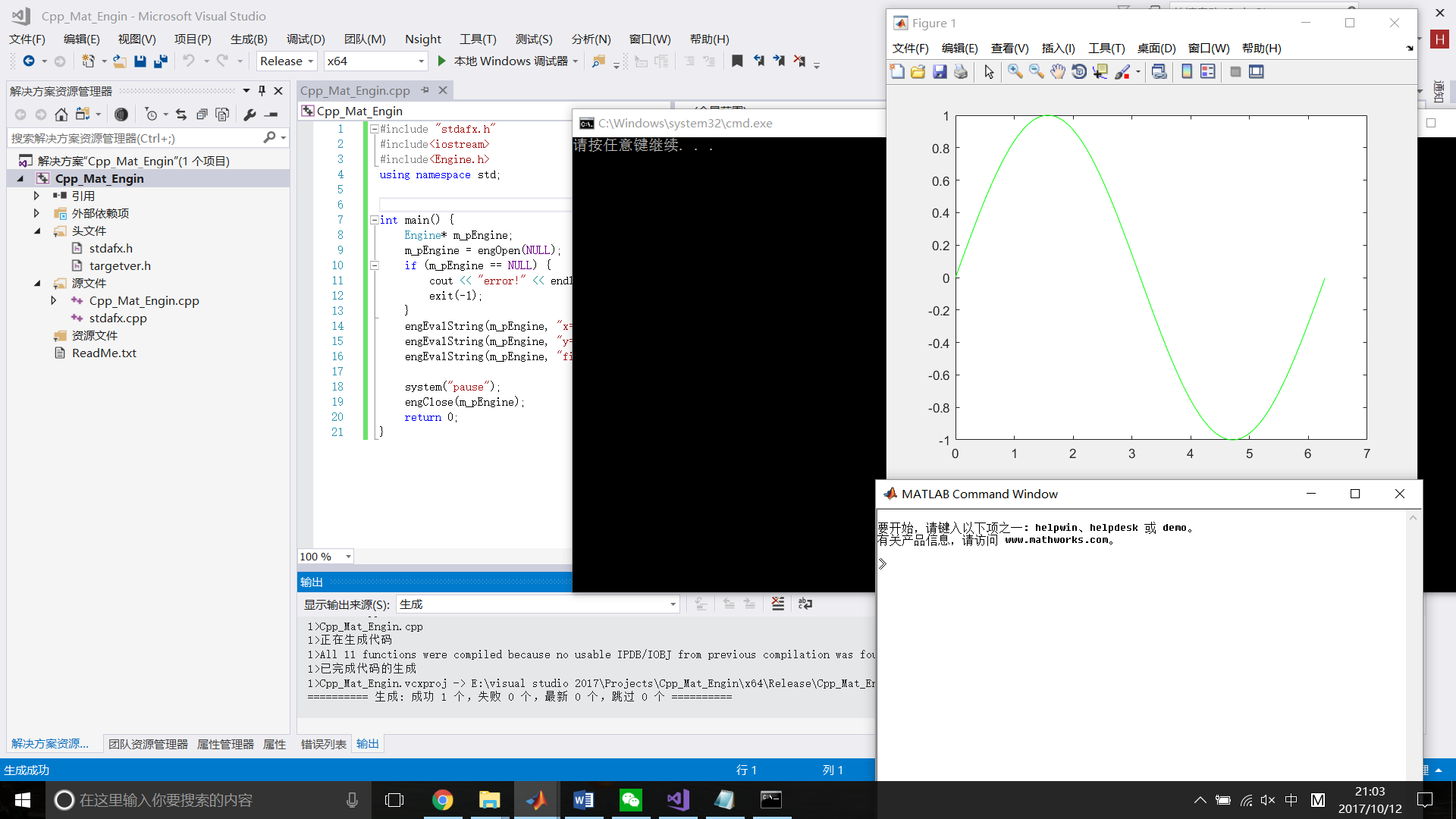
engEvalString(m\_pEngine, "figure; plot(x,y,'g');");

system("pause");

engClose(m\_pEngine);

return 0;

}



测试完成。

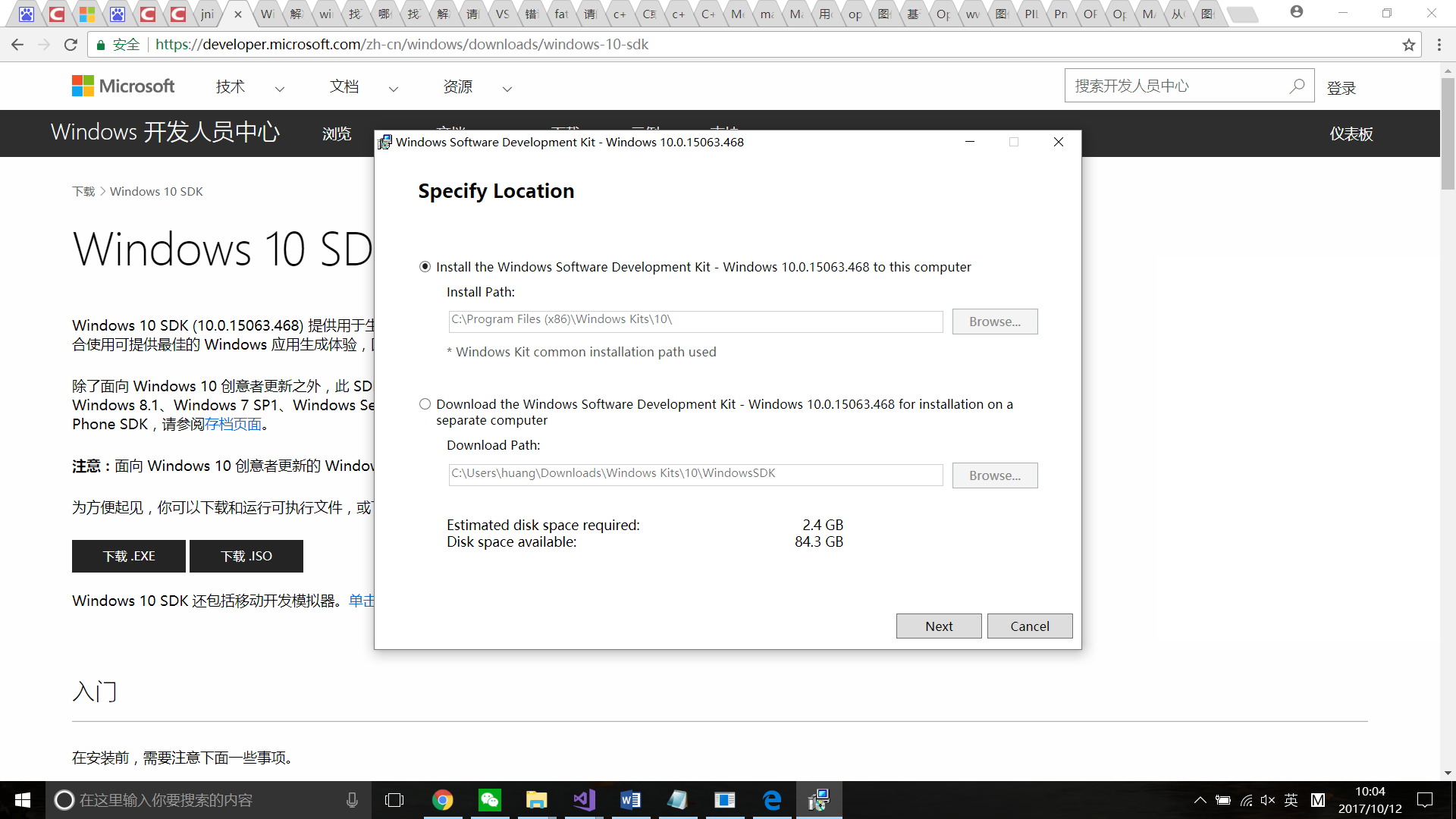
补充，期间如过出现VS SDK不匹配问题，请进行更新。

下载地址：

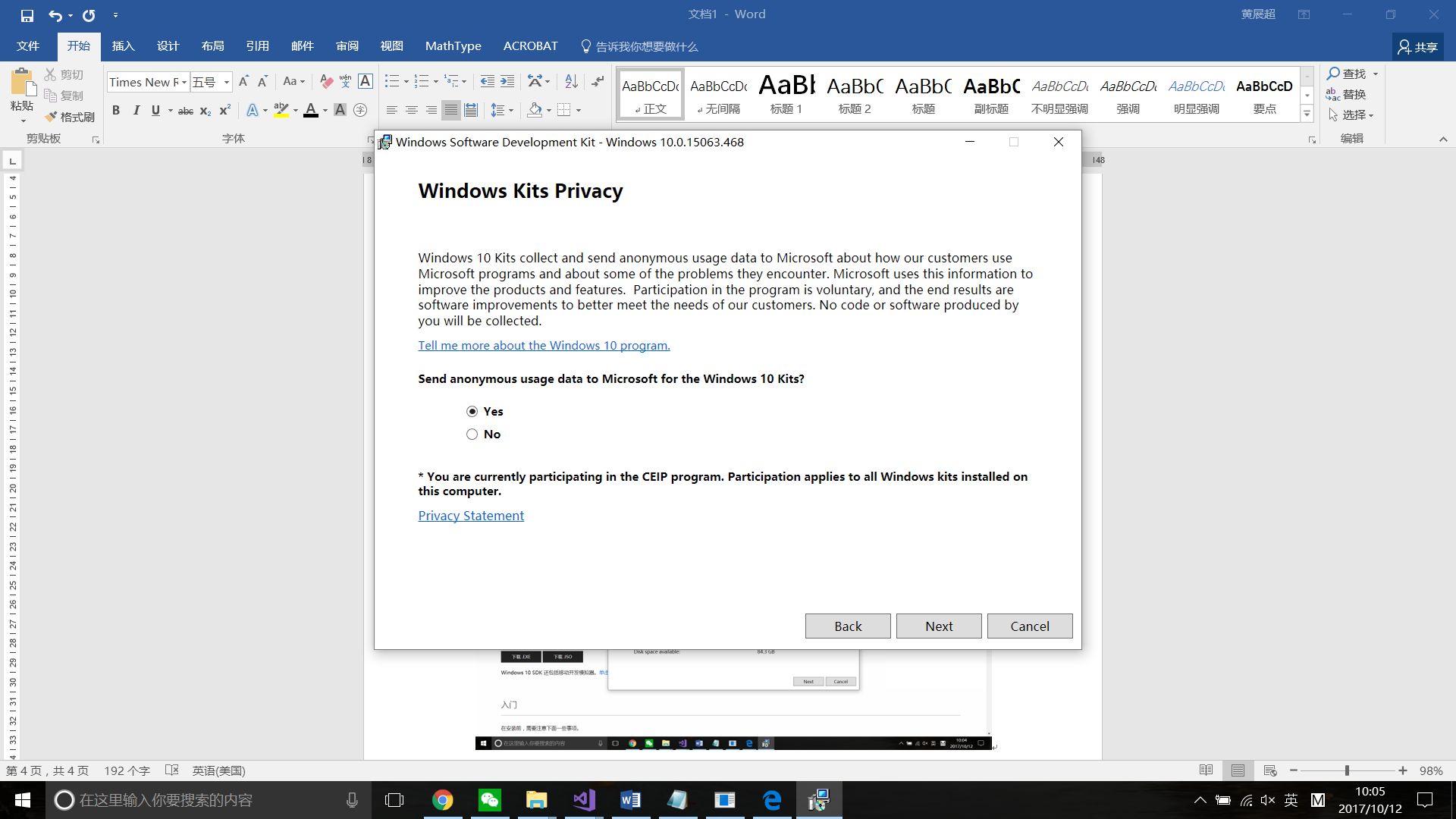
https://developer.microsoft.com/zh-cn/windows/downloads/windows-10-sdk



运行exe



Next-完成



accept

