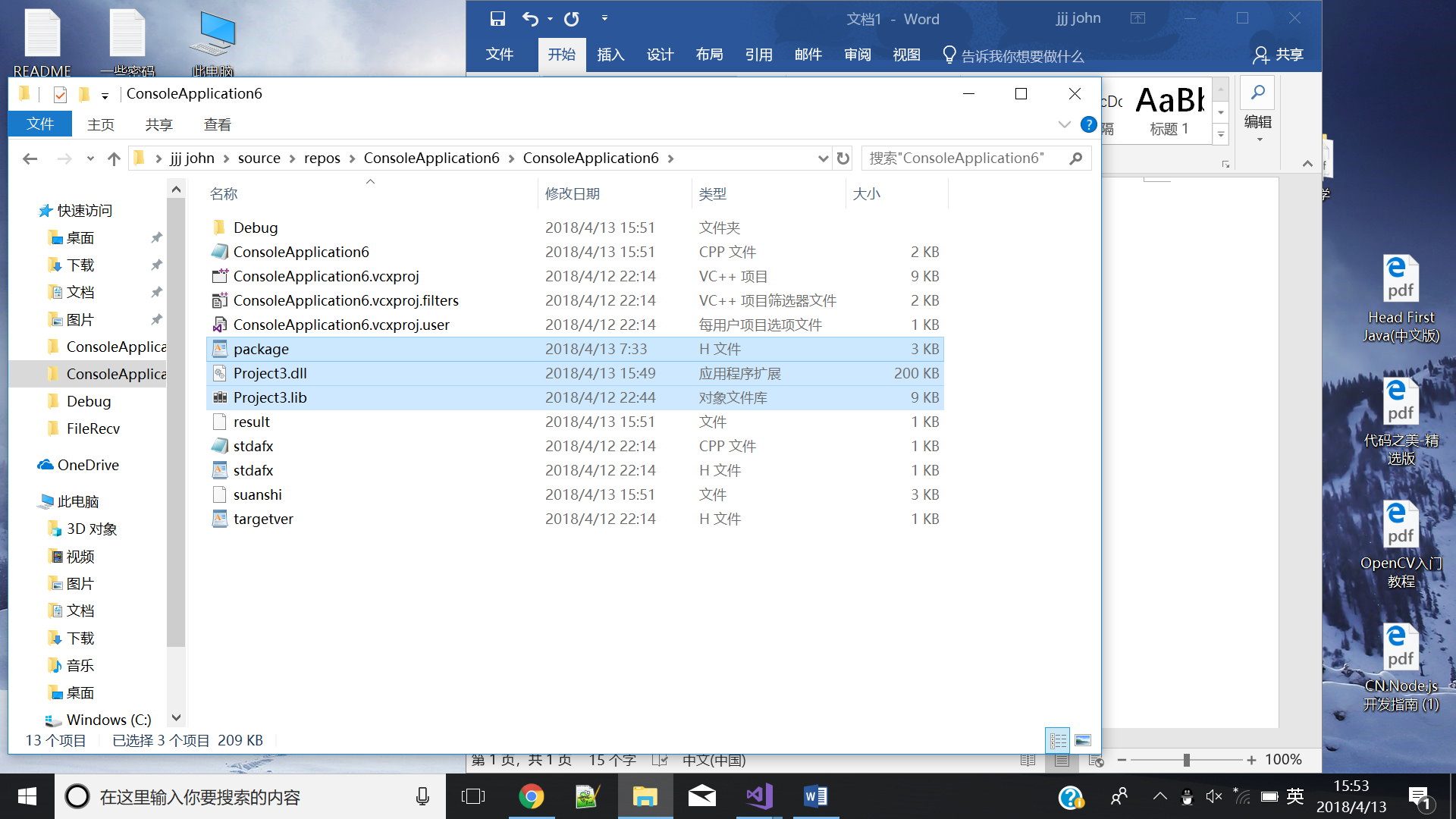
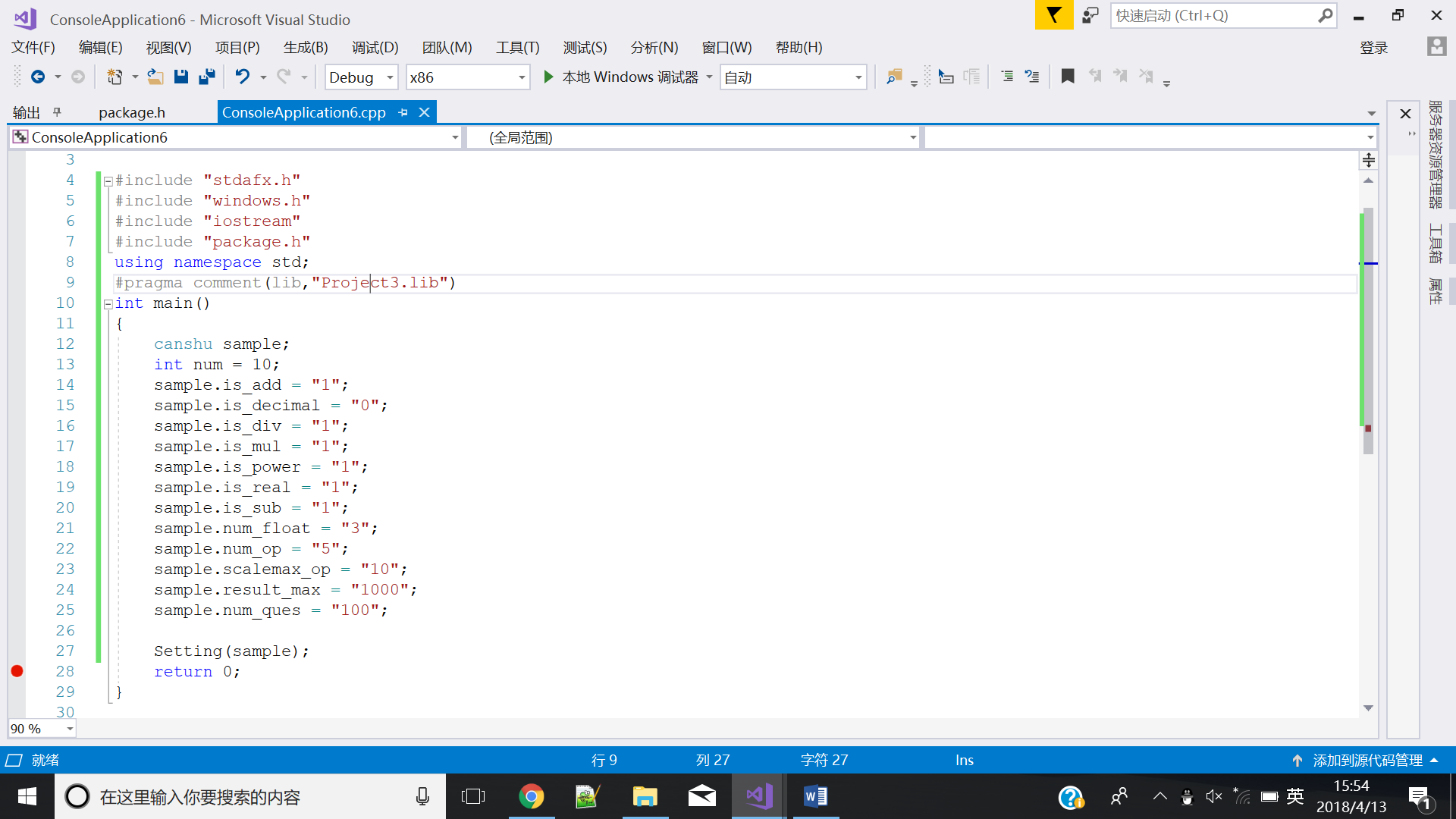
我们的四则运算生成算法怎么用：



把我们提供的有黑色的三个文件：project3.dll,project3.lib,package.h，放到你们的项目中去。



接下来分别一一介绍他们的功能和注意事项：

首先include好我们的头文件： #include "package.h"

然后，连接一下对象文件库： #pragma comment(lib,"Project3.lib")

然后，注意有一个叫canshu的结构体，结构体定义如下：

typedef struct canshu {

string num\_ques;

string scalemax\_op;

string num\_op;

string num\_float;

string is\_real;

string is\_decimal;

string is\_power;

string is\_add;

string is\_sub;

string is\_mul;

string is\_div;

string result\_max;

}canshu;

这些在头文件中都被定义好了，泥萌只要对它们进行赋值就好了。它们都要被赋好值，不赋值的话按照默认的标准值来进行。接下来一一介绍它们传参的注意事项，接口检查我们已经帮你们做好了。

首先：他们**都是string的类型，并且内容只能为数字字符，不能有其他字符，**不然会返回错误信息。

canshu sample;//初始化一个结构体，结构体定义见上面

sample.is\_add = "1";

sample.is\_sub = "1";

sample.is\_mul = "1";

sample.is\_div = "1";

sample.is\_power = "1";

//五种运算符的支持与否，他们之间没有任何制约，能够支持任意N种运算结合。

它们的**输入参数只能为字符串"1"或者"0"，否则是非法输入，返回错误信息。**

sample.is\_real = "1";

sample.is\_decimal = "0";

//3种模式的支持与否：

is\_real = "1"，is\_decimal = "1"; 支持小数运算。

is\_real = "1"，is\_decimal = "0"; 支持分数运算。

is\_real = "0"，is\_decimal = "1"; 支持小数运算。

is\_real = "0"，is\_decimal = "0"; 支持整数运算。

它们的**输入参数只能为字符串"1"或者"0"，否则是非法输入，返回错误信息。**

sample.num\_float = "3";

// 浮点数的小数位数 **在5以内，否则报错**。必须是数字字符串。

sample.num\_op = "5";

// 操作符的数量，也就是算式长度  **小于等于10，否则为非法输入，报错**。

sample.scalemax\_op = "10";

//操作数最大范围 **支持乘方（乘法）的时候只能20以内，没有的话500，否则为非法输入。**

sample.result\_max = "1000";

//结果的最大值

sample.num\_ques = "100";

//问题的数量：**10000以内**

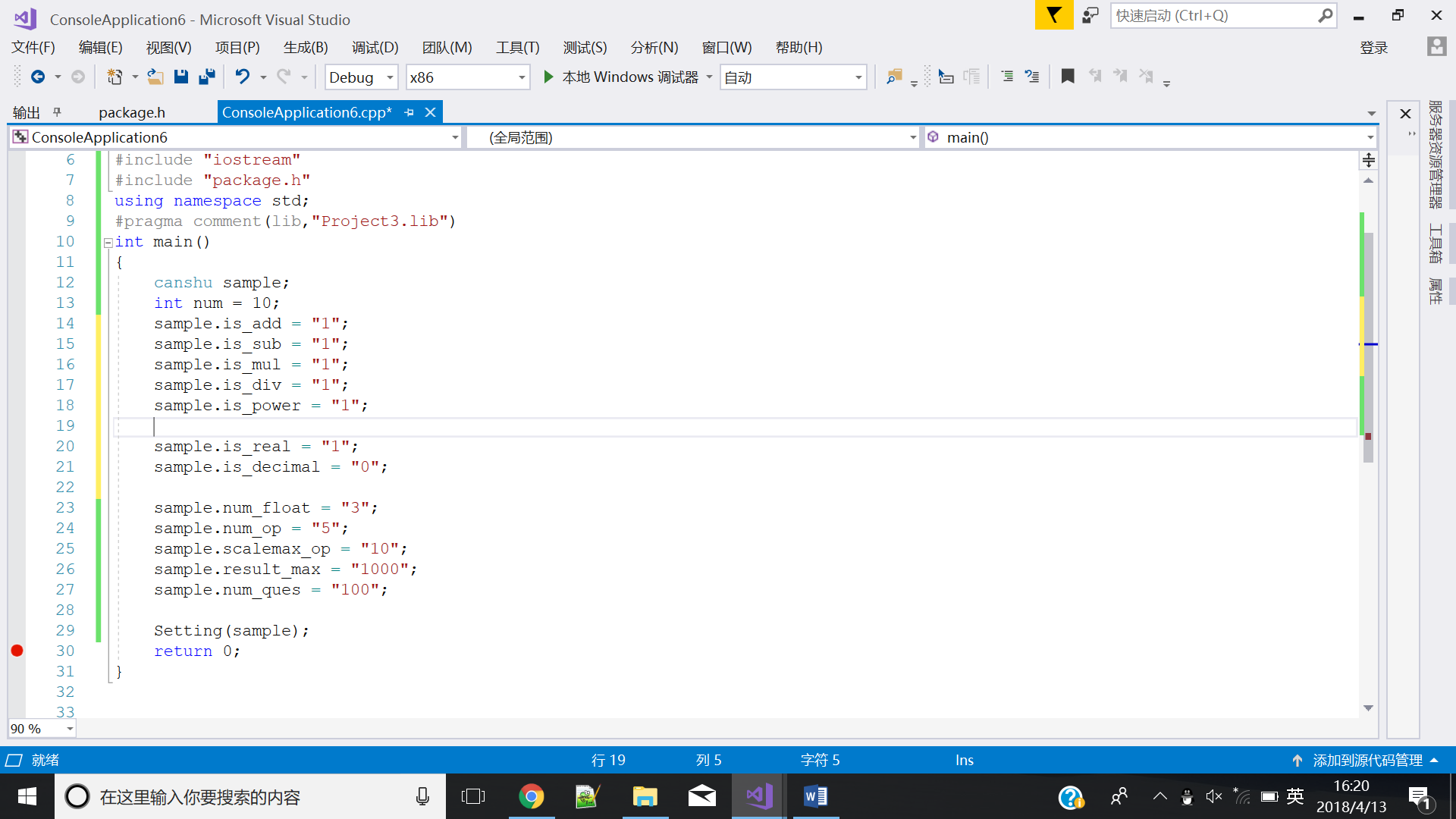
好，这样就可以用我们的核心函数了。

Setting(sample);

**这样就在项目目录里面生成了两个文件，一个叫result，记录他的结果，另一个叫suanshi，记录他的算式。**

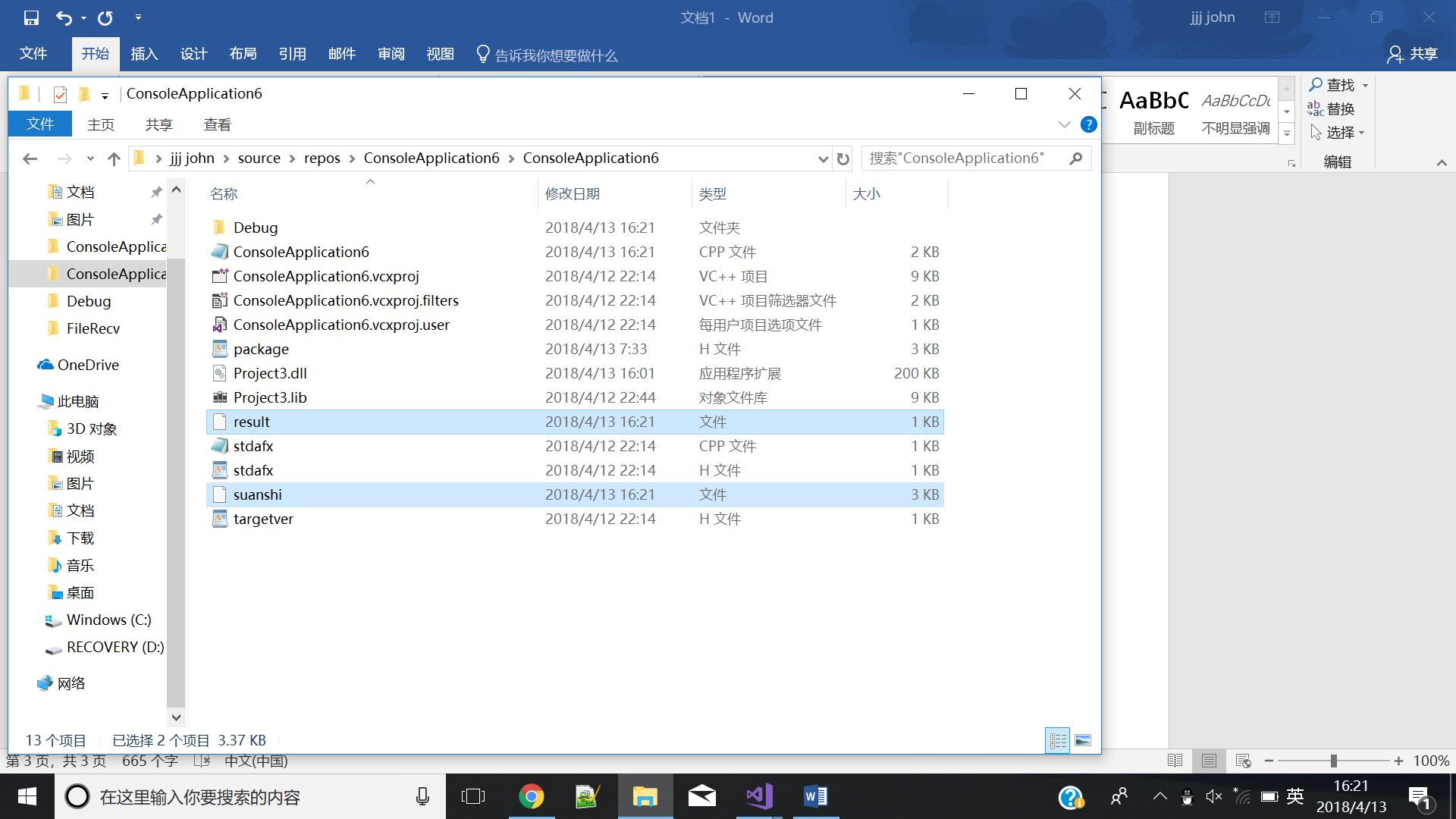
**可以展示一下：**

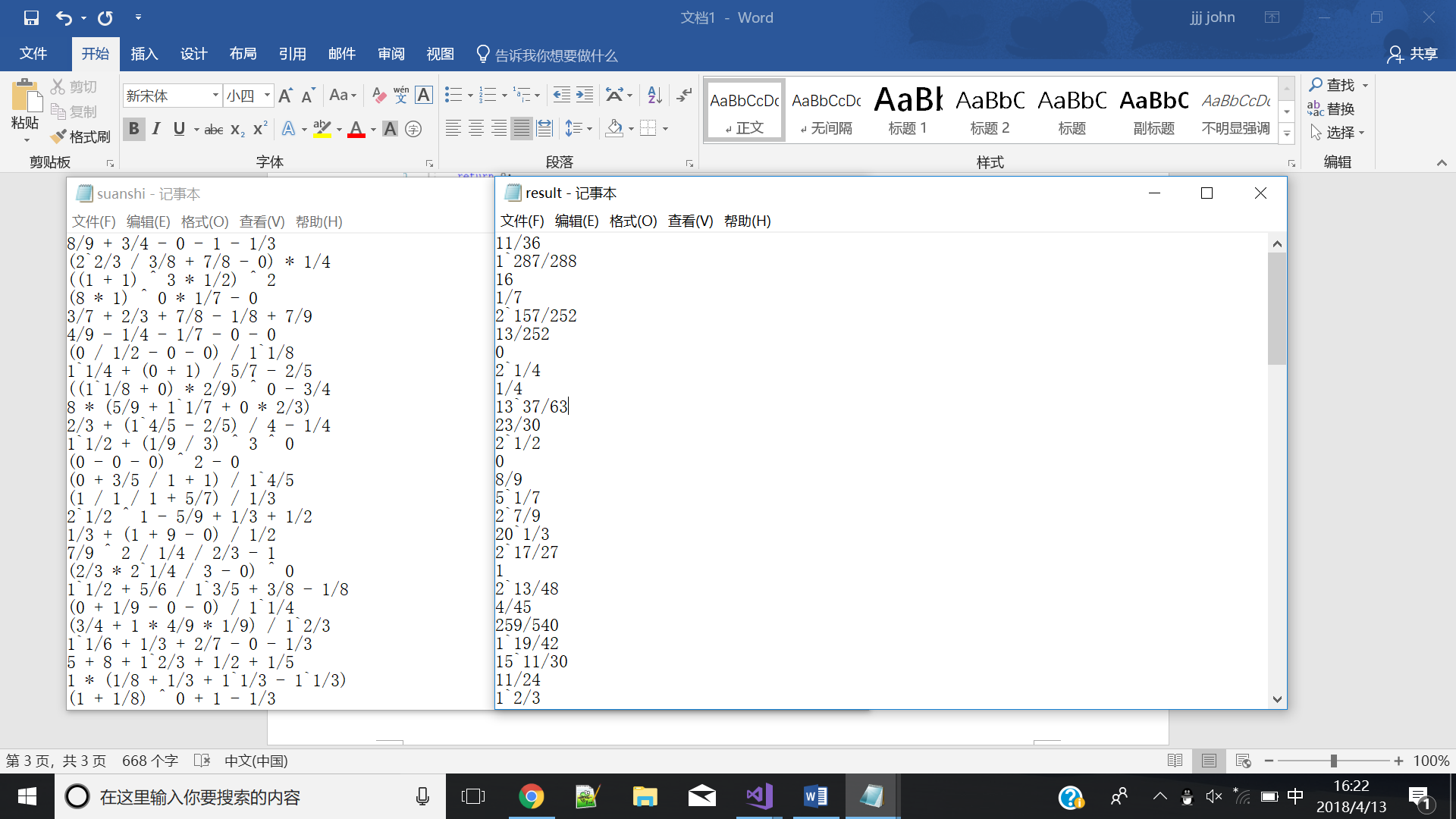
**输入：**



**支持加减乘除乘方，分数运算。**

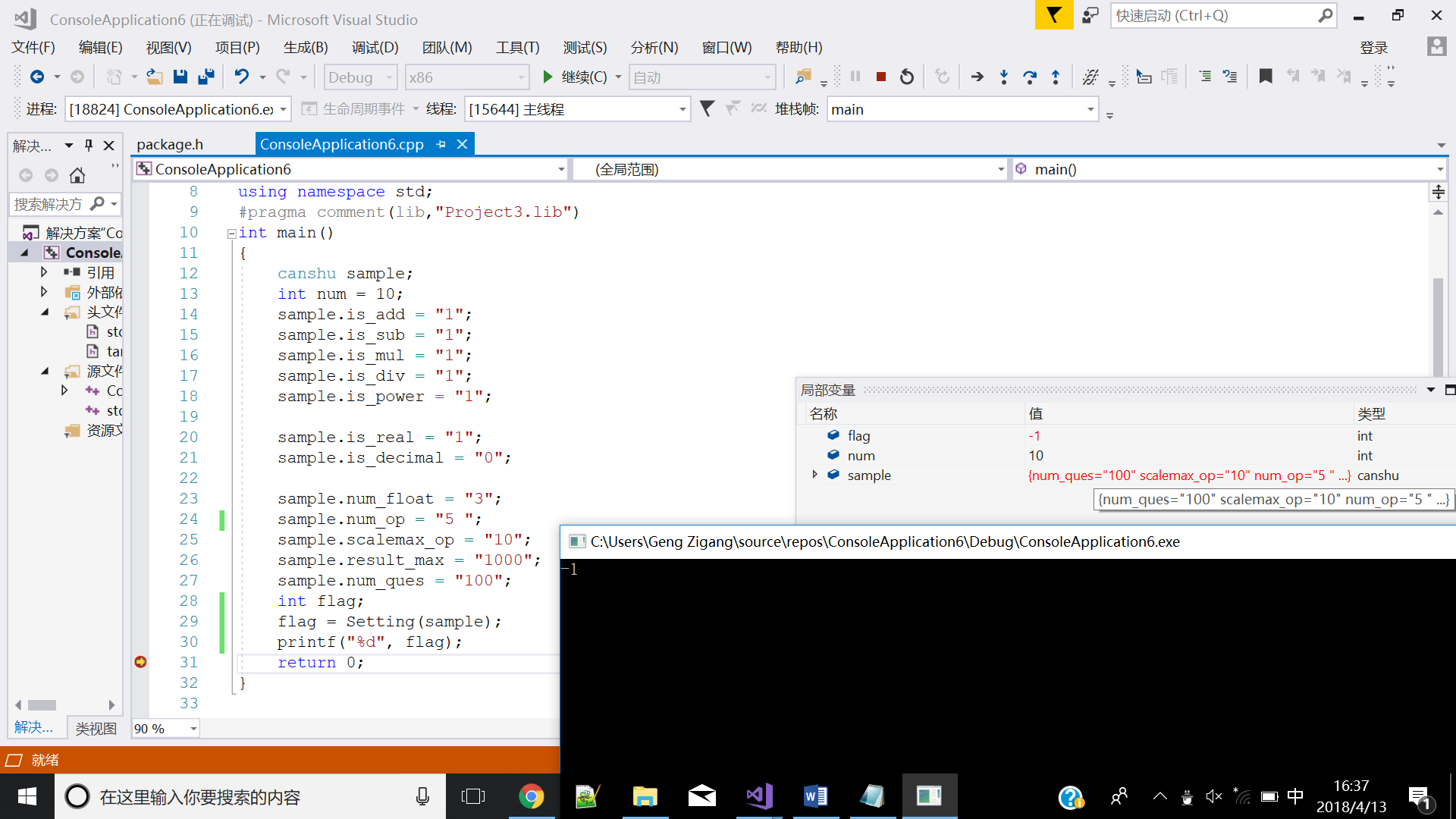
**输出：**





**注意：我们生成的算术txt文件中，每个运算符前后都有空格，如果遇到没有空格的‘/’符号，说明这是一个分数！（推荐读取时使用getline函数进行读取）**

**但是如果输入为这样：**



注意这里，为非法输入，因为（箭头所指出）5后面出现空格，他不是纯数字，他还有其他空格。

**所以setting的返回值就是-1，这就是非法输入的错误信息。**

**如果一切正常，flag应为0。**

还有其他功能你萌自己探索吧。出了问题一定要及时联系我们，共同努力让程序变得更完善。

联系方式：1025489007 欧阳炳濠

397601889 耿子刚