Read Me

一、 函数：

Answer set(int ExpNumber, int operator\_number, int Accuracy, bool fraction, bool decimal, int min, int max, int operator\_mode)

温馨提示：Answer的结构体：

Struct Answer{

string\* consequence;

string\* express;

};

二、 函数调用方法：

ExpNumber:表达式的数目，必须是正整数且小于等于1000

operator\_number:操作符的数目，必须是正整数且小于等于1000

Accuracy:所有数字的精确度，保留x位小数则Accuracy为x，且x大于等于0小于等于3

Fraction:表示是否支持分数 true为支持，false为不支持

Decimal:表示是否支持小数 true为支持，false为不支持

min:产生数的最小值， 必须是非负整数

max:产生操作数的最大值， 必须是非负整数

operator\_mode:选择操作符的模式，仅五种模式可选

    温馨提示：

    1·小数和分数同时为true时，返回设置出错，并且将其都设置为默认值false；

    2·min>max时会将min置成0；min<0或者min>1000时会将min置成0；max<0或者max>1000时会将max置成1000；

    3·ExpNumber、operator\_number非法时会被置成1000；

    4·Accuracy非法会被置成1；

    5·operator\_mode

     0：只有加法

     1：只有加减法

     2：只有加减法和乘法

     3：只有四则运算

     4：有四则运算和乘方

     若非法则会被置成0.

三、 Eg:

想要产生12道题，每道题运算符数目不超过34个，精确度为小数点后面0位，不支持分数，支持小数，运算数的范围是0~100，含有四则运算，则

只要set(12,34,0,false,true,0,100,3);即可

四、输出：

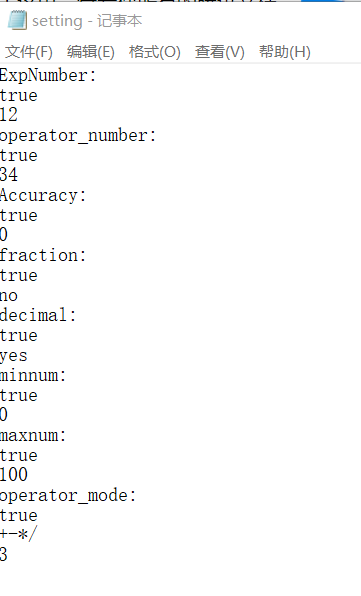
采用文件输出

setting.txt 、consequence.txt、expression.txt

其中分数线为// ，乘方符号为^

setting.txt:

（设置情况）



说明：

表达式个数：// true/false（设置成功/设置失败）// 生成个数

操作符个数：// true/false（设置成功/设置失败）//生成个数

保留小数位数：// true/false（设置成功/设置失败）//保留的小数位数

是否支持分数：// true/false（设置成功/设置失败）//no/yes（不支持/支持）

是否支持小数：// true/false（设置成功/设置失败）//no/yes（不支持/支持）

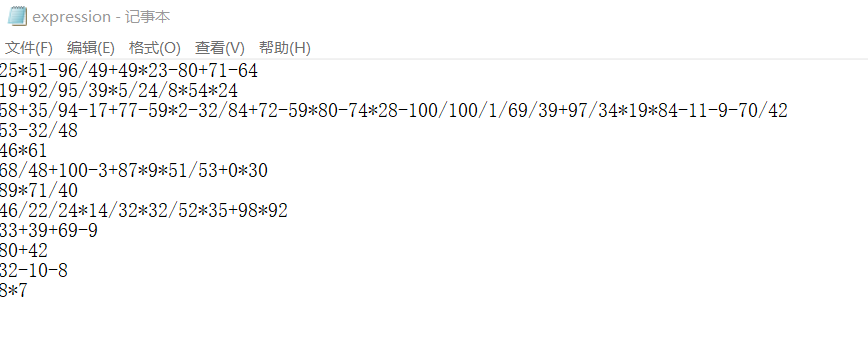
设置下限：// true/false（设置成功/设置失败）// 0（下限）

设置上限：// true/false（设置成功/设置失败）// 0（下限）

设置操作符模式：// true/false（设置成功/设置失败）//+-\*/（操作符）// 3（模式）

expression.txt:

（表达式）



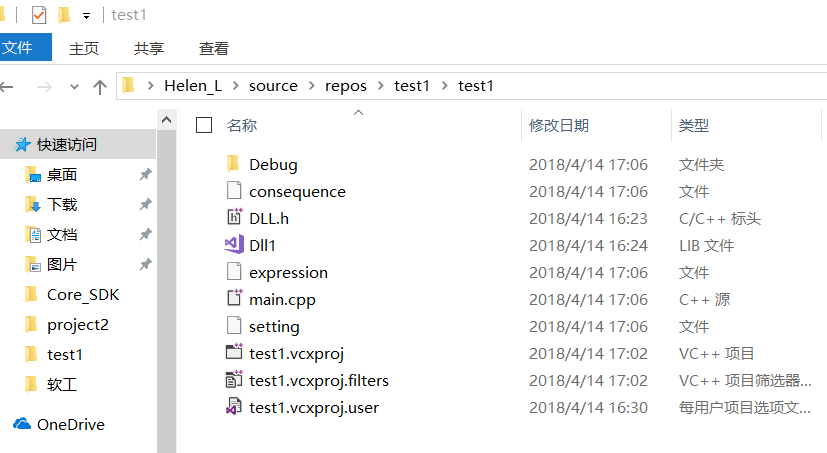
consequence.txt:

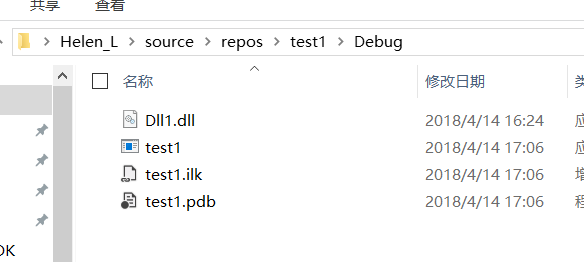
（结果）

五：dll调用方法例举：

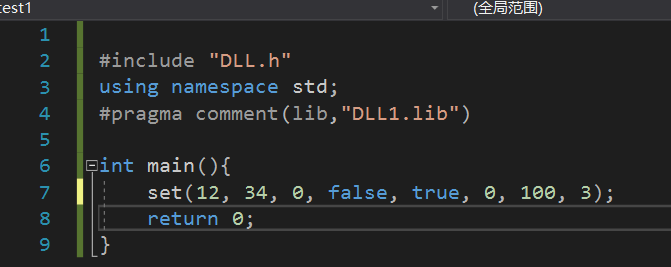
1.将我们提供的DLL.h, Dll1.lib,Dll1.dll添加到你们的工程当中

示例：





2.main 函数示例：



上图所示方法会得到三个文件

或者：

Answer a=set(12,34,0,false,true,0,100,3);

这样子同样会有三个文件，结果的两个指针也在a这个结构体中方便读取内存组的使用