目标：

编写MFC程序实现：一个可以实现整数加减乘除四则运算的简单计算器。

要求在图形界面上至少存在：

(1) 2个用来接受操作数输入的编辑框，

(2) 1个用来输出计算结果的编辑框，

(3) 加减乘除四个按钮。

要求该计算器能正确进行整数加减乘除运算。如果计算有误或输入非法，则应在图形界面上给出提示(自行确定具体的提示方式)。

提高要求：编写有理数的加减乘除计算器。

原理：

编写构件库程序，定义一个计算类，以及一个有理数结构体，其中用bool类型变量表示数的符号，用unsigned long long格式表示分子和分母。计算类中用两个有理数结构体分别代表两个参与运算的有理数，用enum类型的变量来表示运算符号（加减乘除中的某一种），并且编写了通过CString类来获取有理数并且进行运算同时检查输入和计算有效性的函数。具体见下方所示：

struct RationalNum

{

bool Positive;

unsigned long long int numerator, denominator;

RationalNum(bool P1, unsigned long long int n1, unsigned long long int n2) : Positive(P1), numerator(n1), denominator(n2)

{

};

RationalNum()

{};

};

enum Operations { O\_plus, O\_minus, O\_multi, O\_divide };

class FH\_Cal

{

public:

FH\_Cal();

~FH\_Cal();

CString Get\_Nums(CString CS1, CString CS2);

void SetOp(Operations);

private:

RationalNum Num1, Num2, TempAnsNum;

Operations Op;

CString TempAns;

bool Cstring\_To\_Nums(CString CS, RationalNum& Num\_CS);

bool Check\_Cstrings(CString CS);

int Calcu();

unsigned long long Get\_Max\_Co(unsigned long long a, unsigned long long b);

};

编写MFC程序，包含上述构件库的类，为对话框增加变量，同时提供了MFC窗口事件响应的相关代码，通过按钮的点击事件以及编辑框的内容更改事件来进行计算操作以及显示结果的操作。

由于整数也属于有理数的范畴，所以整数的运算可以完全按照上述操作进行，只是需要更改提示语言以及输入合法性检测部分。

**如何编程和运行：**

* 软件构件库部分：

编写原理中所述函数的具体实现。

* 构件库测试部分：

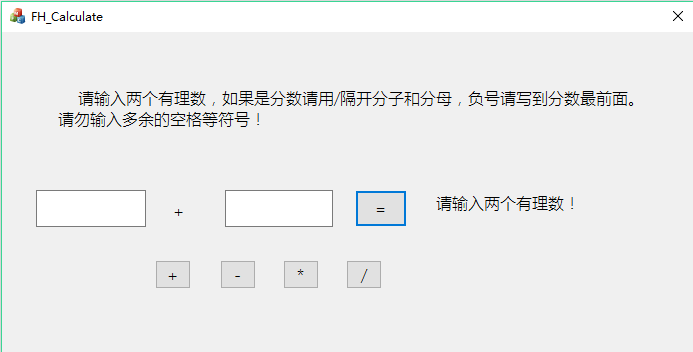
编写所需要的测试函数，并且提供MFC程序直接运行的入口。

* 构件库应用部分：

建立工程，生成exe可执行文件。

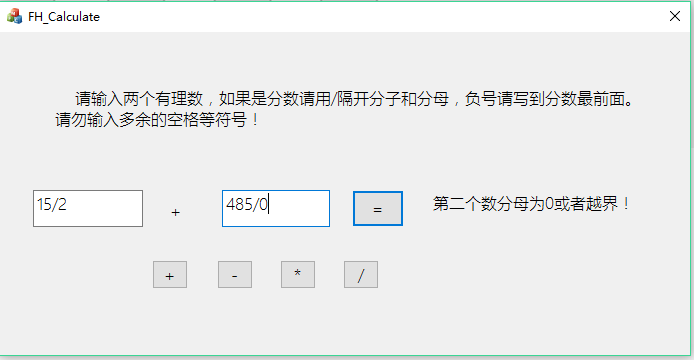
结果：

以下测试结果均在64位操作系统下用VS 2017 测试得到。

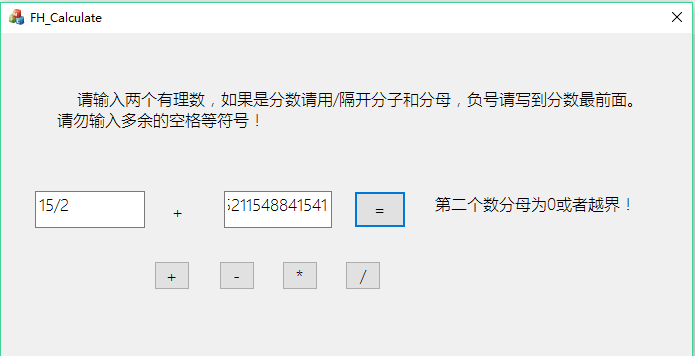




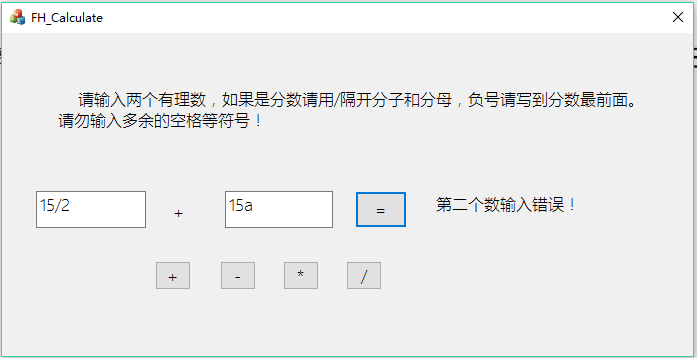
初始界面



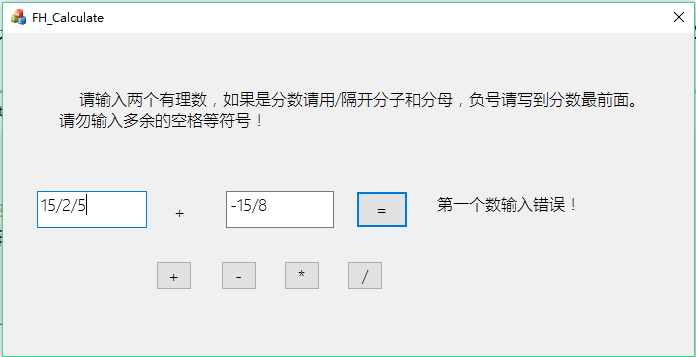
当输入错误时的提示1



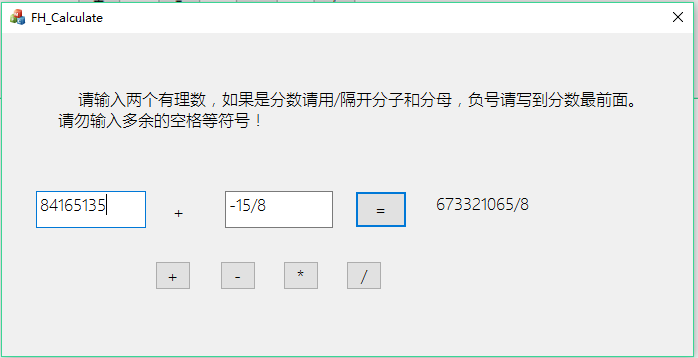
当输入错误时的提示2（此时第二个编辑框输入的内容为48584165211548841541）



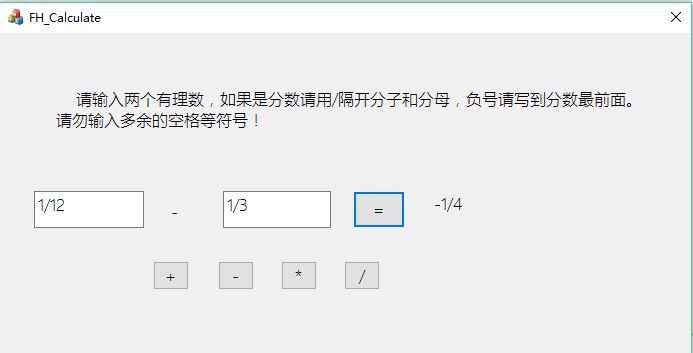
当输入不合法字符时的提示



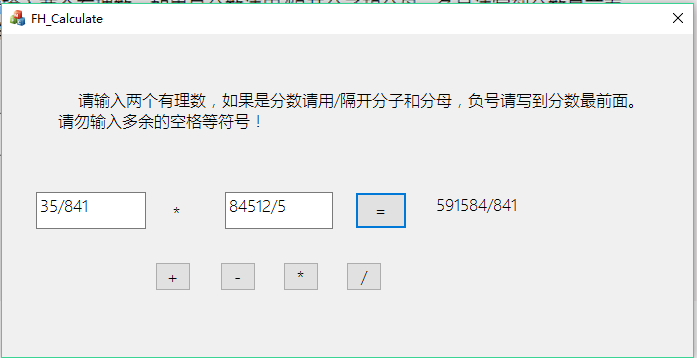
当输入不符合要求时的提示



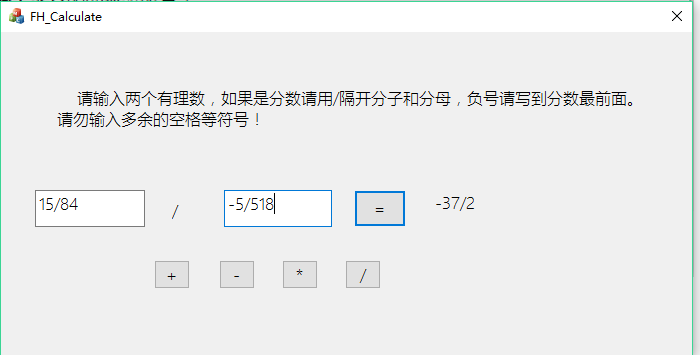
一次正确输入的结果（加法）



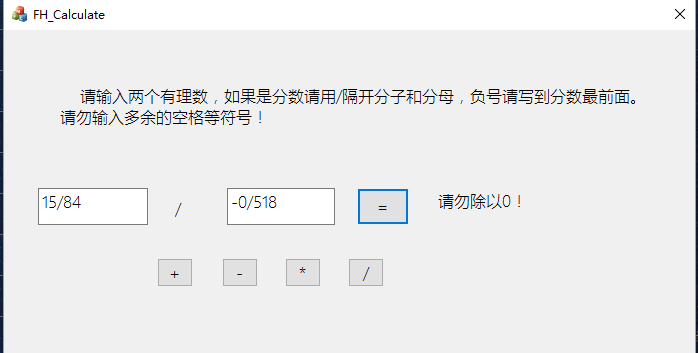
一次正确输入的结果（减法）



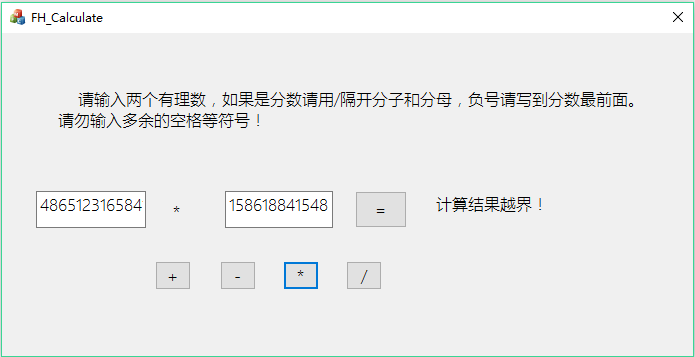
一次正确输入的结果（乘法）



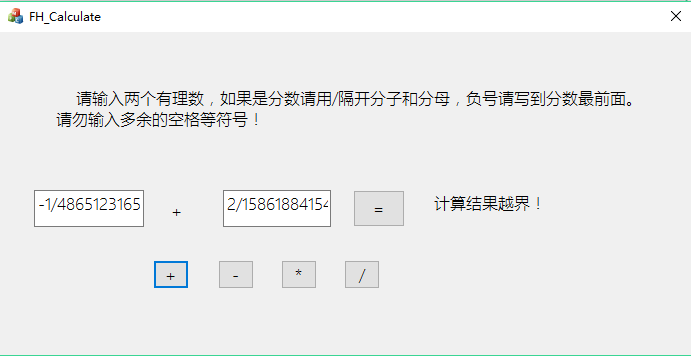
一次正确输入的结果（除法）



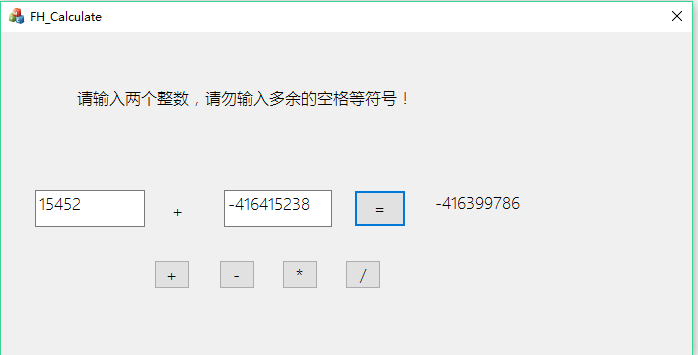
除以0错误的提示



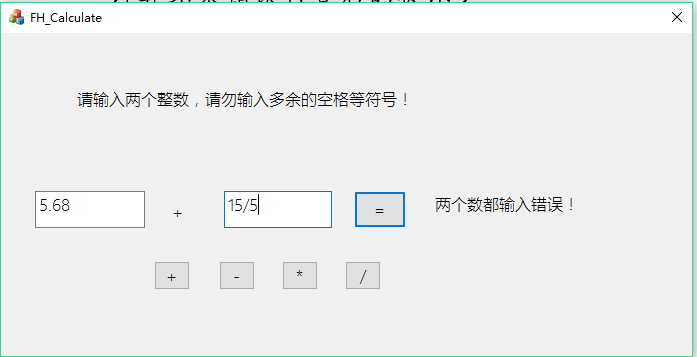
计算结果错误自查后的提示1



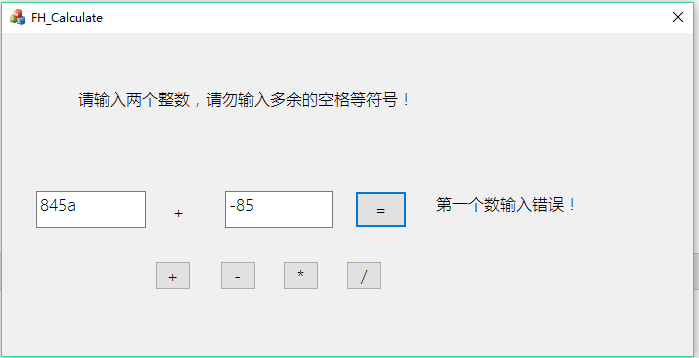
计算结果错误自查后的提示2



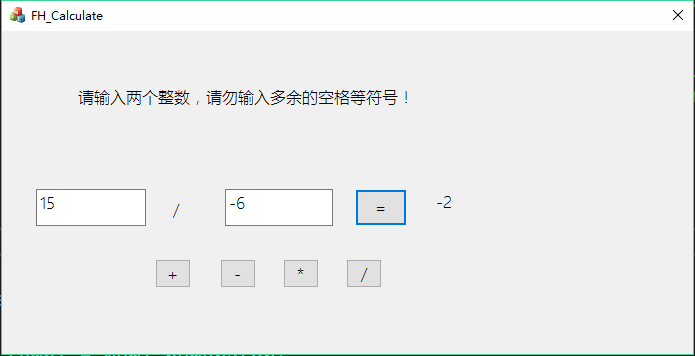
整数计算模式下一次计算结果

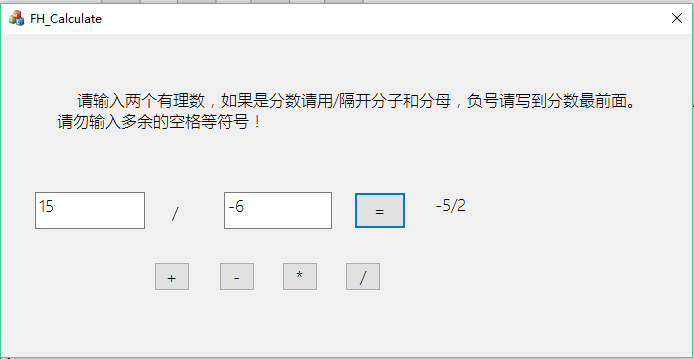


整数计算器，会检测整数输入的合法性，阻止小数分数的输入



检测非法输入





两种模式下除法计算结果的表示

程序验证结果报告：

* 对于64位的计算机

在上述的程序运行过程中未发现错误。如果输入不合规范，会给出具体的提示。在计算过程中，如果越界或者计算错误也会发出提示。

有理数计算过程中，计算结果会以最简分式的形式表示出来。而整数计算过程中，计算结果会直接取整（不会四舍五入）。

结论：

程序能较好完成设计目的要求，在使用的过程中未发现错误。