目标：

**构件库部分:**

改写第3讲第54~56个幻灯片的接受整数输入例程。定义一个会抛出异常的接受整数输入函数。该函数接受来自控制台窗口输入的字符串，并分析该字符串的格式，看它是否符合整数的表示格式。如果输入或格式有误，则抛出异常。请采用不同的值表示不同的输入错误状态，具体的值及其含义请自行确定。如果格式无误，则该函数返回该字符串所对应的整数。

定义另一个函数。该函数调用上面的函数，并对上面函数的各种异常都进行了处理，即该函数不会抛出异常。如果上面函数抛出表明输入结束的异常，则该函数直接返回0；否则，该函数不断接受输入，直到接收到正常输入并且格式正确的整数，并返回该整数。

**应用验证部分:**

提供测试案例比较改写后的程序与原程序的健壮性，并形成对比验证报告。

原理：

编写构件库程序，定义一个输入类与一个输入函数，具体如下所示：

class FH\_Int\_Input

{

private:

string Inputed;

int OutPut;

bool InputState;

public:

int Start\_Input();

void Get\_Error(exception e);

bool Get\_State();

int Get\_Int();

};

extern int GetIntMain();

其中在 GetIntMain() 中调用 Start\_Input() 函数，而 Start\_Input() 函数可以接受输入，并且根据要求的判断抛出异常。在函数运行中使用星号（\*）表示输入结束，同时会导致 GetIntMain() 函数返回0。异常的catch处理均在 GetIntMain() 函数中，如果出现别的异常会使用文字形式表示出来，并且要求继续输入。

**如何编程和运行：**

* 软件构件库部分：

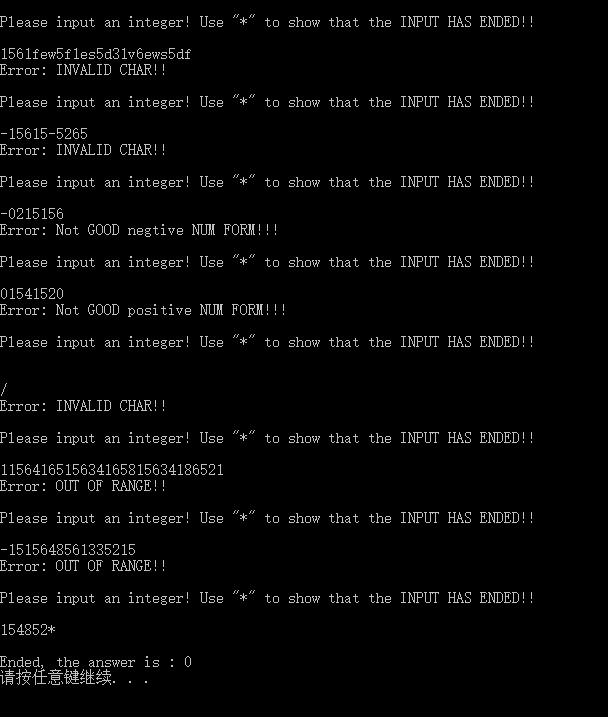
编写原理中所述函数的具体实现。

* 构件库测试应用部分：

建立工程，编写 main() 函数并在其中调用 GetIntMain() 函数。

结果：

以下测试结果均在64位操作系统下用VS 2017 测试得到。



两种模式下除法计算结果的表示

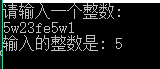
程序验证结果报告：

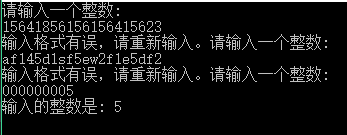
* 对于64位的计算机

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。其中包括数字格式错误（即以0开始的数，比如说 -023 这种不合适的输入）、含有非法字符、越界（包括向上向下越界）以及输入失败（在上面的截图中没有显示）。

与课件中程序的对比：

（在课件的程序中，存在一个bug（使用 cin.sync( ); 并不能够清空缓存区），所以首先添加了cin.ignore( ) 。）

课件中程序运行结果：



可以看出：

1. 课件中程序的容错性较低，在输入错误时不会提示具体的错误类型；
2. 课件中程序在输入1a15类的错误输入时会在非数字字符处截止，只读取前面的数字，而不会直接提示错误；
3. 课件中程序整体架构不清晰，在函数中直接进行错误处理，阅读比较困难。

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。