目标：

编写C++程序来达到：接收输入正整数a与b并且计算正整数a和b的最大公约数，并且编写程序自动验证方法的正确性。

原理：

编写头文件和源文件，用面向对象的方法编程，通过对软件构件库代码、构件库测试代码、构件库应用代码的编写实现程序的目标并且进行测试。同时编写程序用样例测试数据（包括边界数据）验证程序的正确性，用越界数据验证程序的容错性。

**如何编程和运行：**

* 软件构件库部分：

在头文件中定义类和类的构造、析构函数，声明outStart函数，定义变量。

在源文件中include头文件并且对头文件中声明的outStart函数编写代码：首先对传入的数据进行处理，若不同时满足正整数的条件就输出错误信号（输出0）。如果数据合理则用辗转相除法进行计算，函数返回计算结果。

在定义变量的时候，均使用long型。（包括函数返回类型）

* 构件库测试部分：

编写头文件，（自动测试程序）声明测试函数（一个整体测试主函数，一个使用质数进行测试的函数）。（手动输入程序不包含用质数测试的函数。）

（自动测试程序）在源文件中对测试函数的代码进行编写。首先构造类的对象，之后调用软件构件库内的函数进行计算操作。使用样例测试和质数测试程序测试两种方法，输入包括两部分：一部分正常数据测试，另一部分越界及边界程序测试（测试软件构件库函数中容错性以及测试函数能计算的最大范围）。

质数测试部分中利用以下事实进行测试：对于任意正整数a与两不同质数b和c，ab与ac的最大公约数就是a。所以使用不同的质数组合对它们的倍数进行遍历测试直到超出函数所能接受的范围。

（手动测试部分）输入正整数a和b，同时检查输入数据的有效性（是否是符合要求的、可以被正确读取的正整数），若输入出错则要求重新输入，直到输入的数据合法并且读入正确。

* 构件库应用部分：

在该部分的源程序中编写main函数，其中调用构件库测试部分中的函数。（手动输入部分中为了实现循环测试编写了部分代码（多次调用测试代码达到循环测试的效果））。

结果：

自动测试程序运行结果：

------------------------------------------------------------

第1类测试开始！此测试使用已知数据测试！

第1类测试结束，共测试6组数据。其中0组数据错误！

------------------------------------------------------------

第2类测试开始！此测试使用两质数293和97的倍数测试！

此时使用的一个数据为7329296\*293的结果，但计算为-2147483568，已溢出！

第2类测试结束，共测试7329295组数据。其中0组数据错误！

------------------------------------------------------------

第3类测试开始！此测试使用两质数29333和29947的倍数测试！

此时使用的一个数据为71710\*29947的结果，但计算为-2147467926，已溢出！

第3类测试结束，共测试71709组数据。其中0组数据错误！

------------------------------------------------------------

第4类测试开始！此测试使用两质数49993和49999的倍数测试！

此时使用的一个数据为42951\*49999的结果，但计算为-2147460247，已溢出！

第4类测试结束，共测试42950组数据。其中0组数据错误！

------------------------------------------------------------

第5类测试开始！此测试使用两质数9982499和9522827的倍数测试！

此时使用的一个数据为216\*9982499的结果，但计算为-2138747512，已溢出！

第5类测试结束，共测试215组数据。其中0组数据错误！

------------------------------------------------------------

第6类测试开始！此测试使用越界数据测试程序容错性！包括负数和0！此时应当输出0！

第6类测试结束，共测试6组数据。其中0组数据错误！

------------------------------------------------------------

第7类测试开始！此测试使用边界数据测试有效数据范围！

计算得到2147483646与1073741823的结果为1073741823,结果应该为1073741823，此输出正确！

计算得到2147483647与4294967294的结果为0,结果应该为2147483647，此输出错误！

计算得到2147483648与1073741824的结果为0,结果应该为1073741824，此输出错误！

计算得到2147483647与1的结果为1,结果应该为1，此输出正确！

计算得到2147483647与2147483647的结果为2147483647,结果应该为2147483647，此输出正确！

第7类测试结束，共测试5组数据!

------------------------------------------------------------

总用时109ms！请按任意键继续. . .

其中Debug X64版本用时531ms，Debug X86版本用时500ms，Release X64版本用时109ms，Release X86版本用时125ms。输出结果除时间外均相同，可见在测试用的64位计算机中，32位与64位对数据有效范围没有影响。

自动测试程序验证结果报告：

* 对于64位的计算机

当输入从1到2147483647的整数对时，都可以正确输出运算结果。

当输入的某一个整数超出此范围，将输出0。

对于手动输入程序，若输入不符合规则将要求重新输入。

结论：

该C++程序运行没有问题，使用的编写模式结构清晰，便于重用，体现了面向对象编程的特点。