****

**软件测试技术**

**实验报告**

**实验题目： 测试求两个整型最大公因数和最小公倍数**

**班级：\_\_\_\_软件1602班\_\_\_\_**

**学号：\_\_\_\_41609050229\_\_\_\_**

**姓名：\_\_\_\_\_\_黎红丽\_\_\_\_\_\_\_**

**实验日期：\_2018年12月14日\_**

1. **实验内容**

**本次实验将完成测试“求两个int型数最小公倍数和最大公因数代码”的实验，实验中首先测试输入值，然后测试“求最大公因数方法”，再测试“求最小公倍数方法”，最终对实验结果进行分析。**

1. **实验目的**

**通过本次实验，将学习测试的基本思想，掌握基本测试技能。**

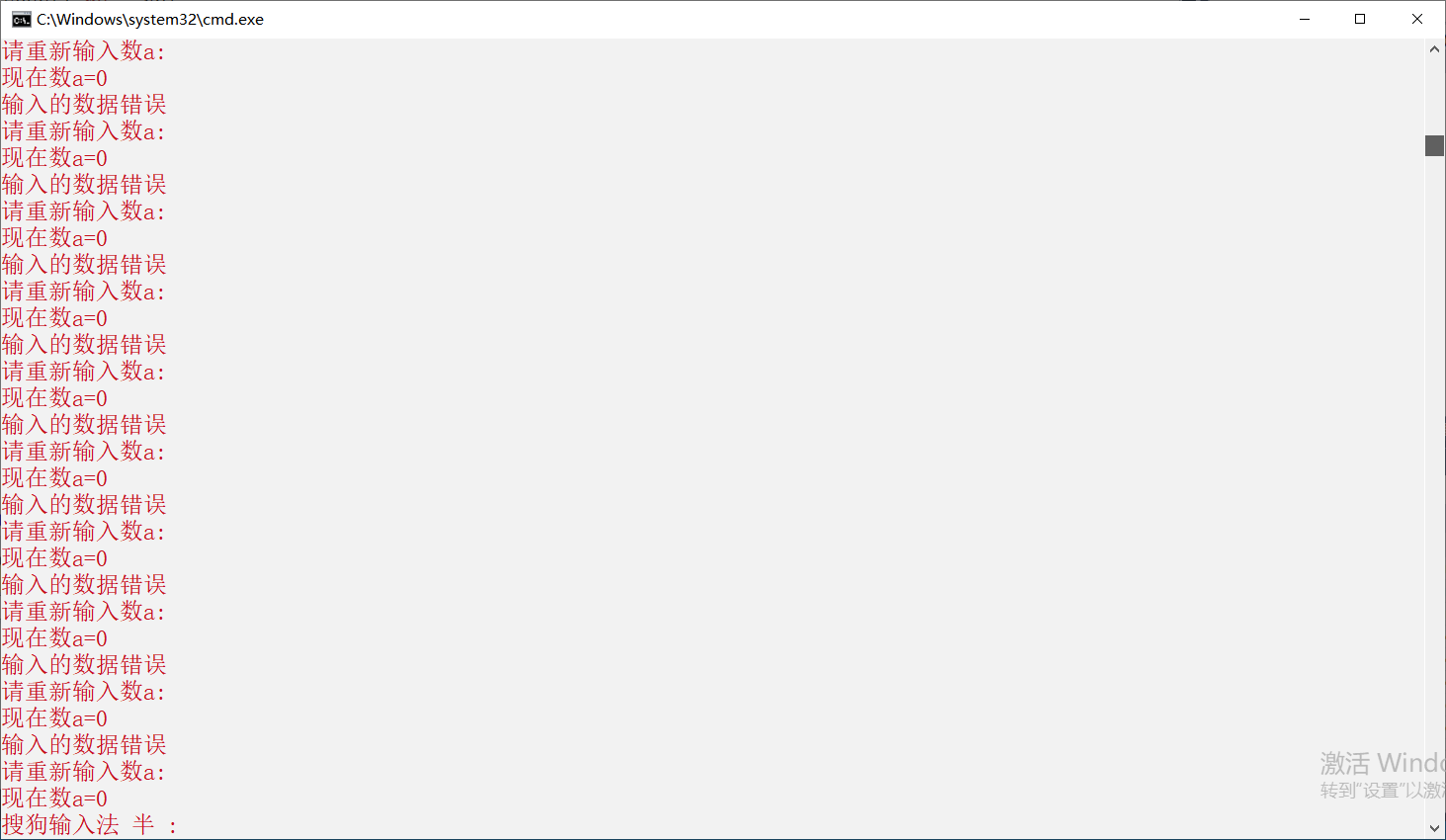
1. **实验方法**

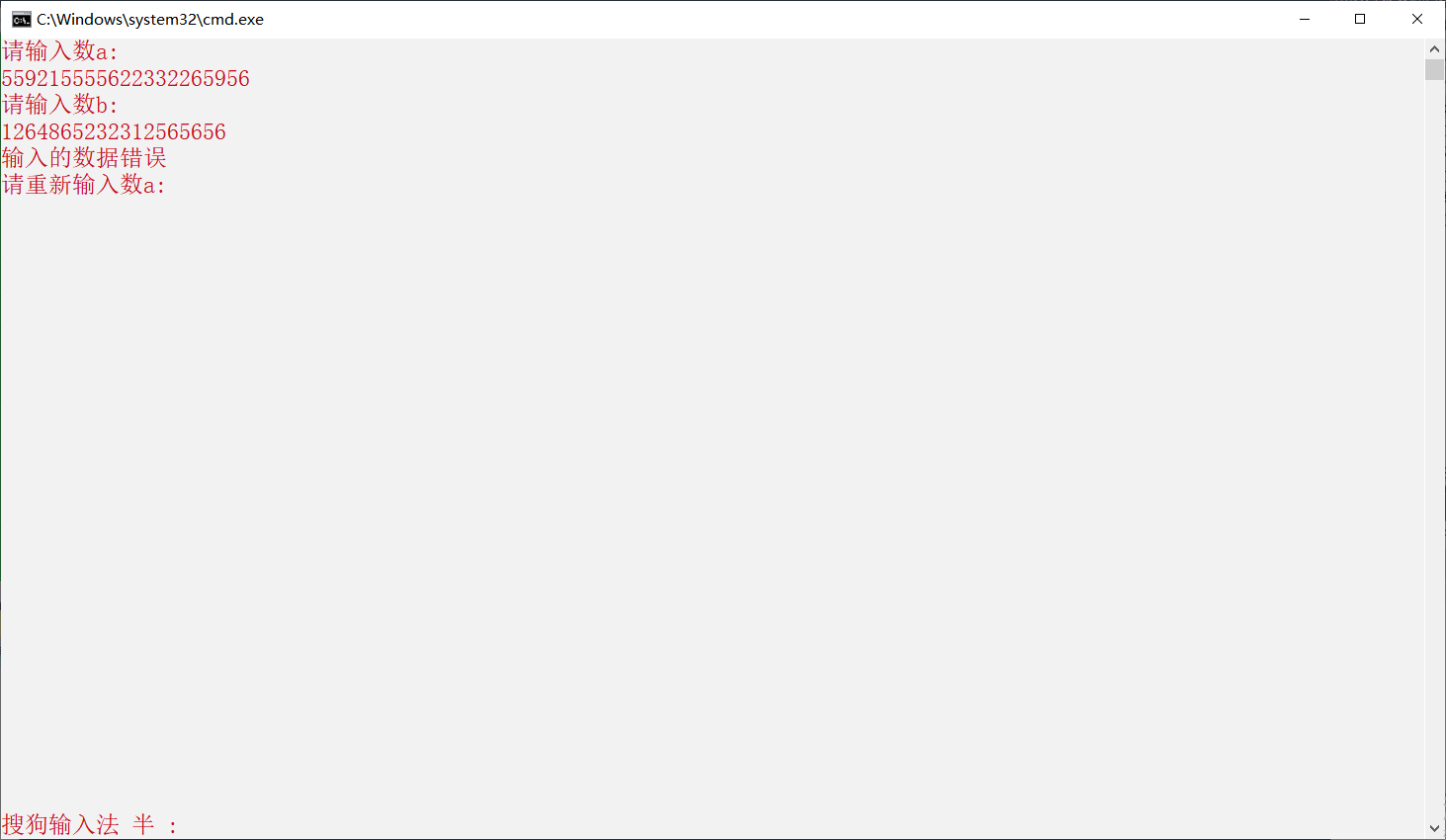
**实验中采用模拟/举例测试，在VS2013（Personal）运行待测代码，检验运行结果的正确性。**

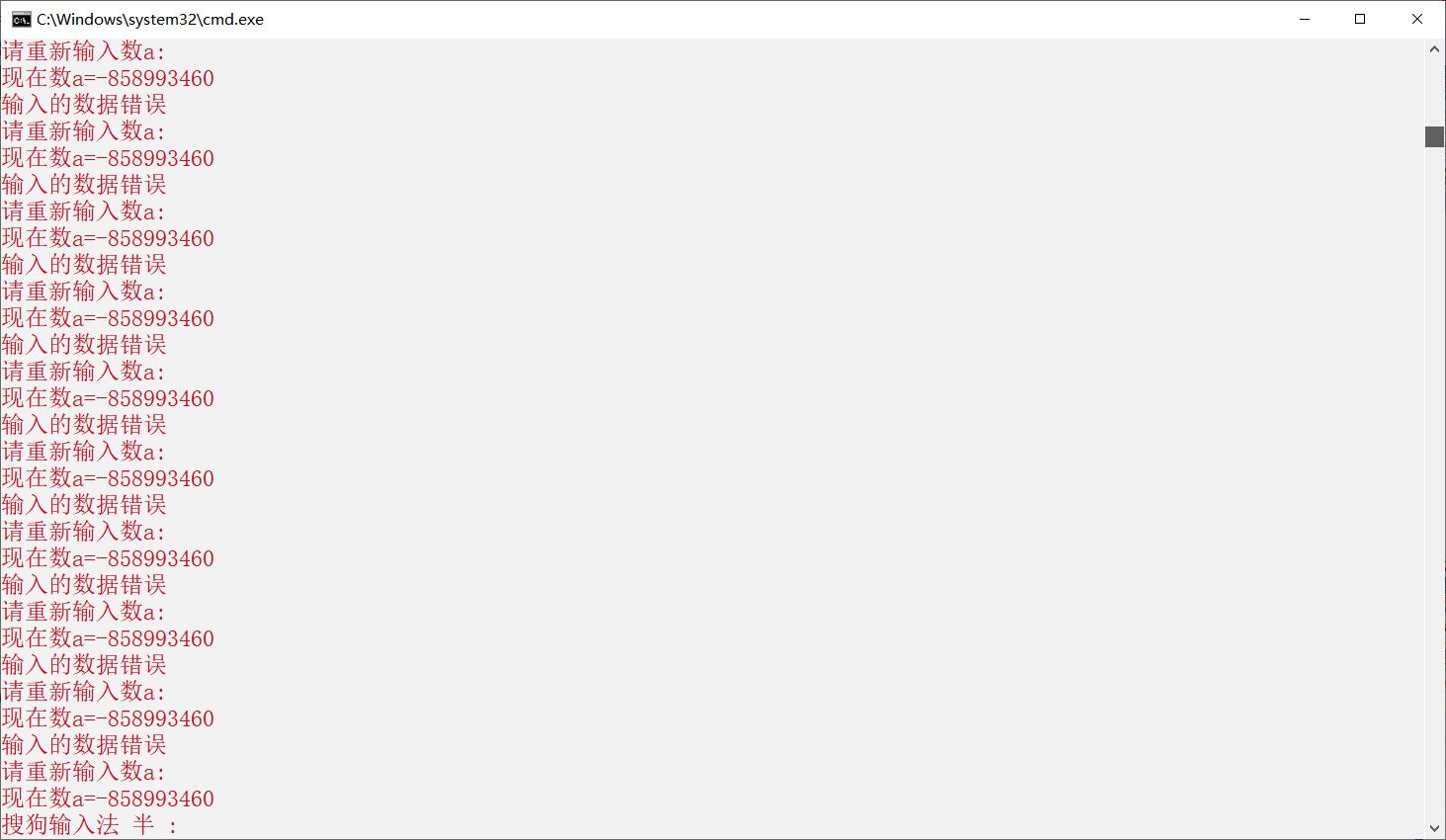
1. **实验结果**

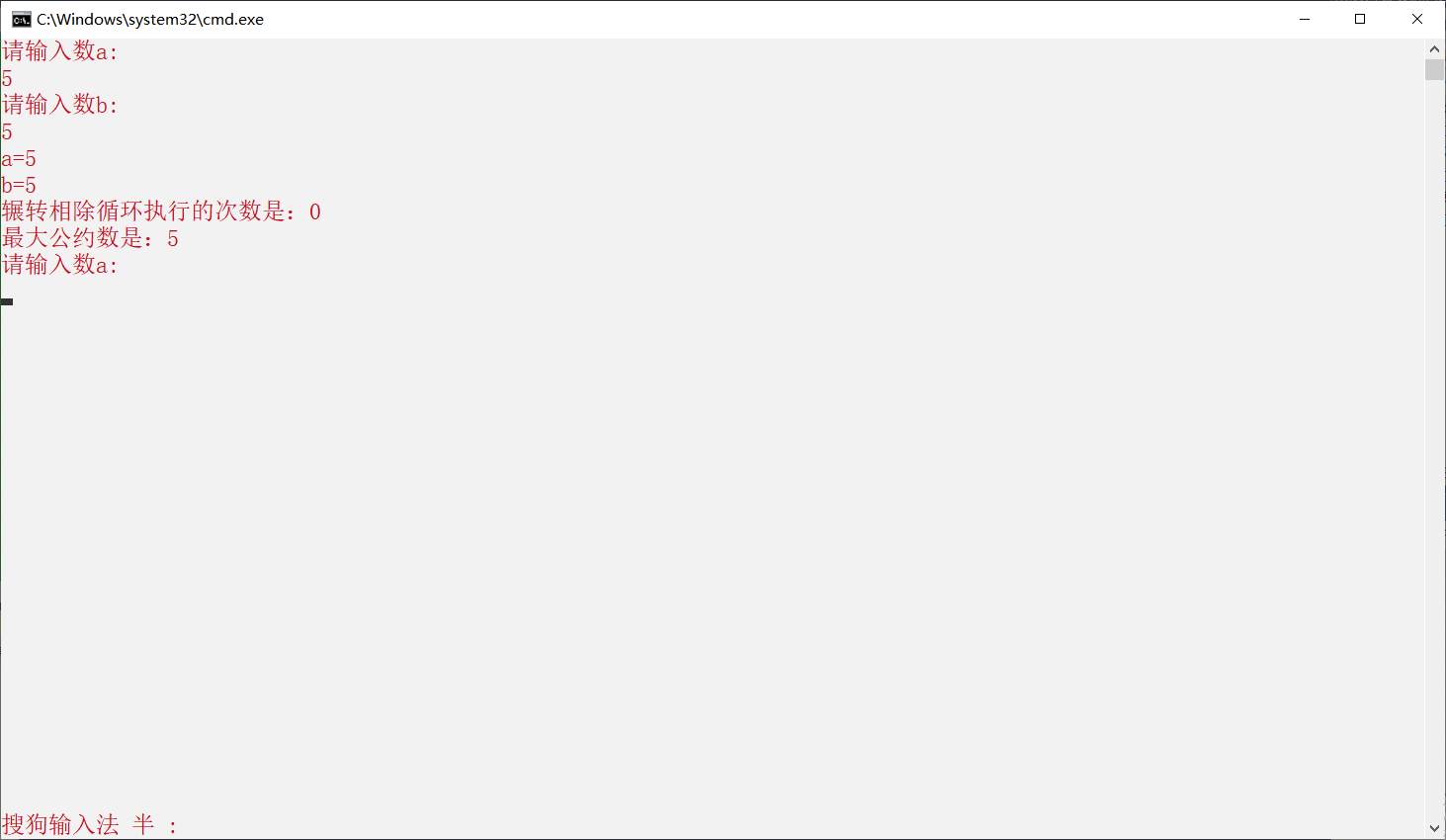
**本次实验运行了VS2013（Personal），对目的代码进行了测试，形成的结果如下：**

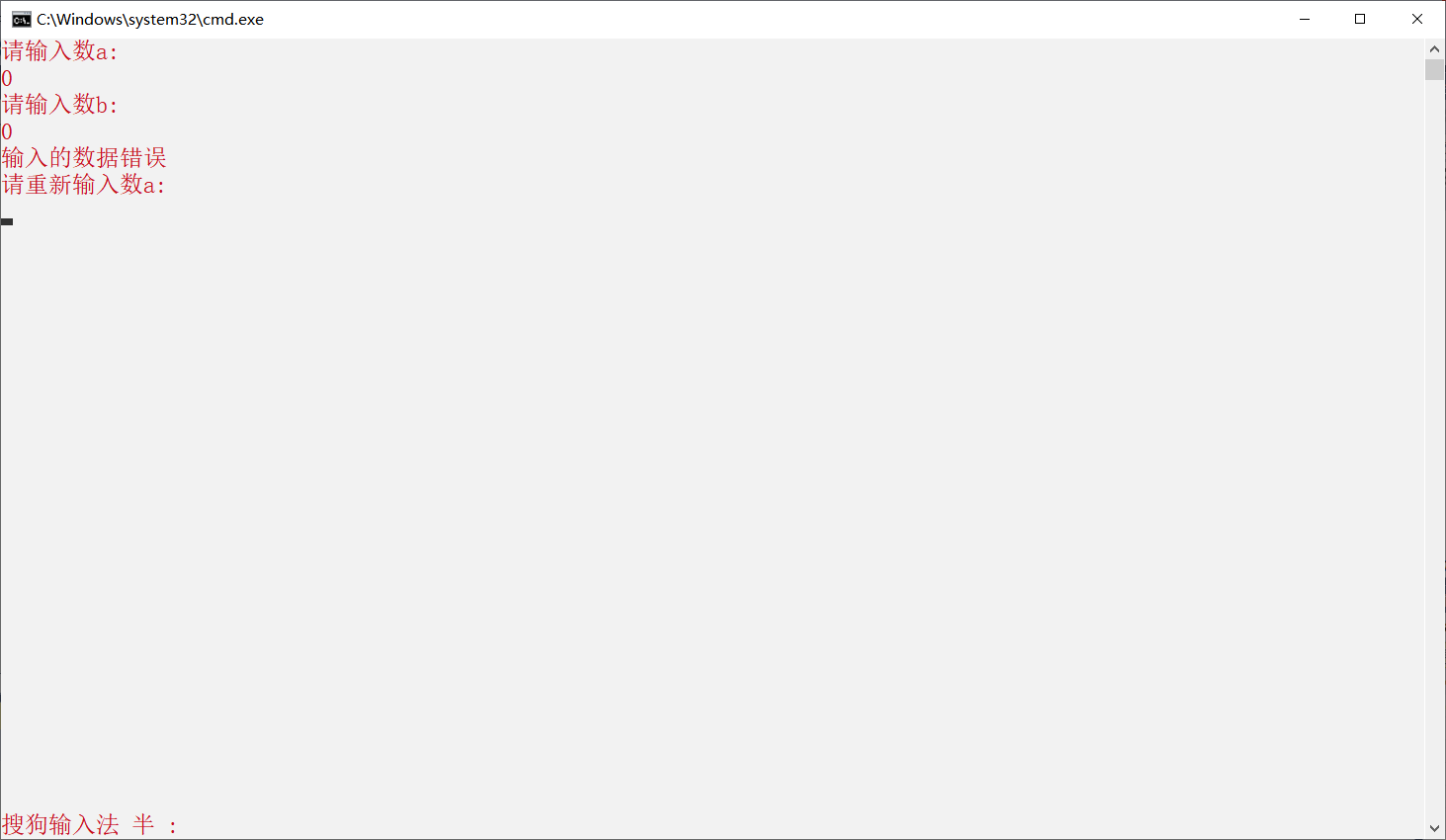
1. **实验截图**

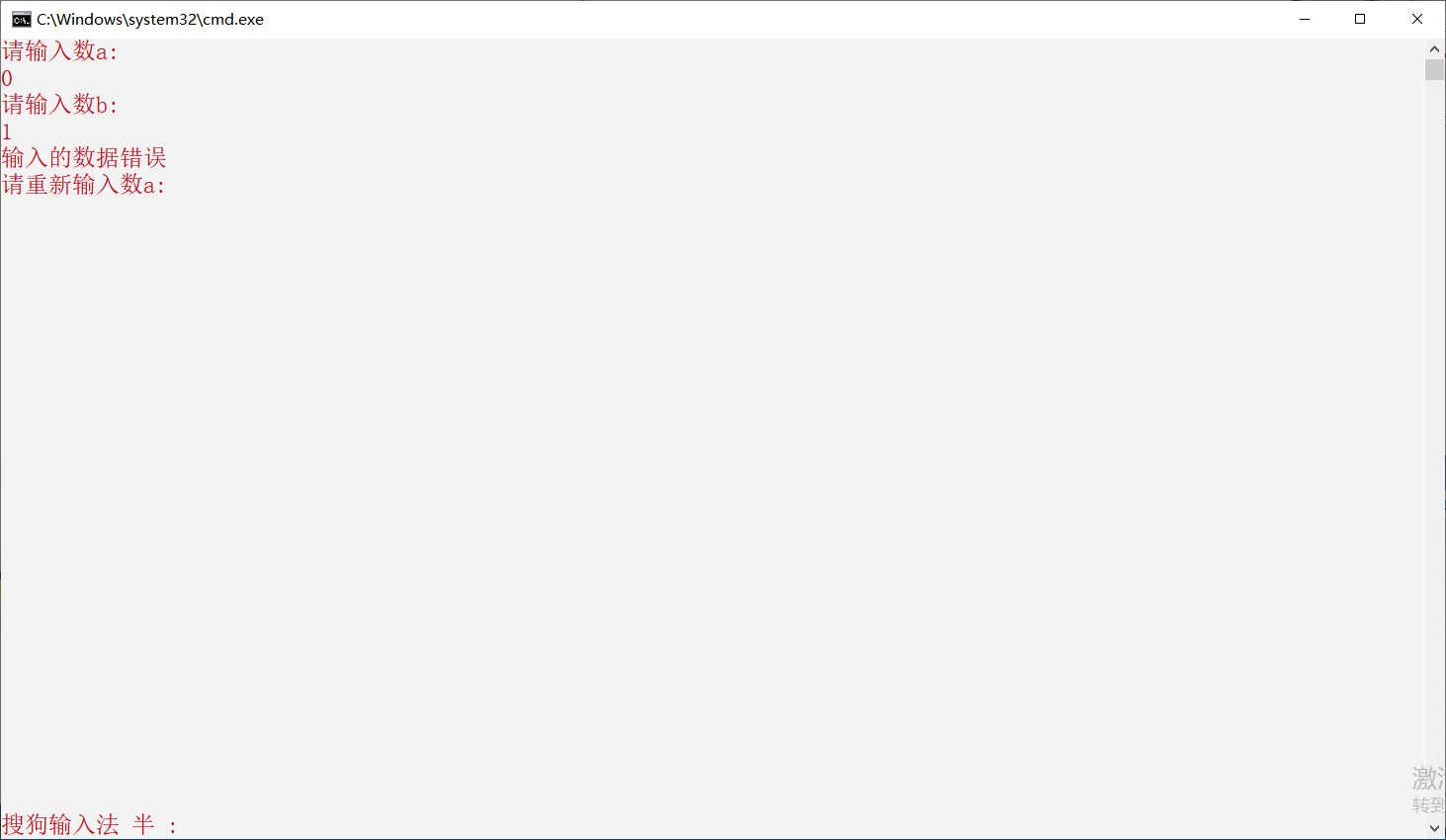
****

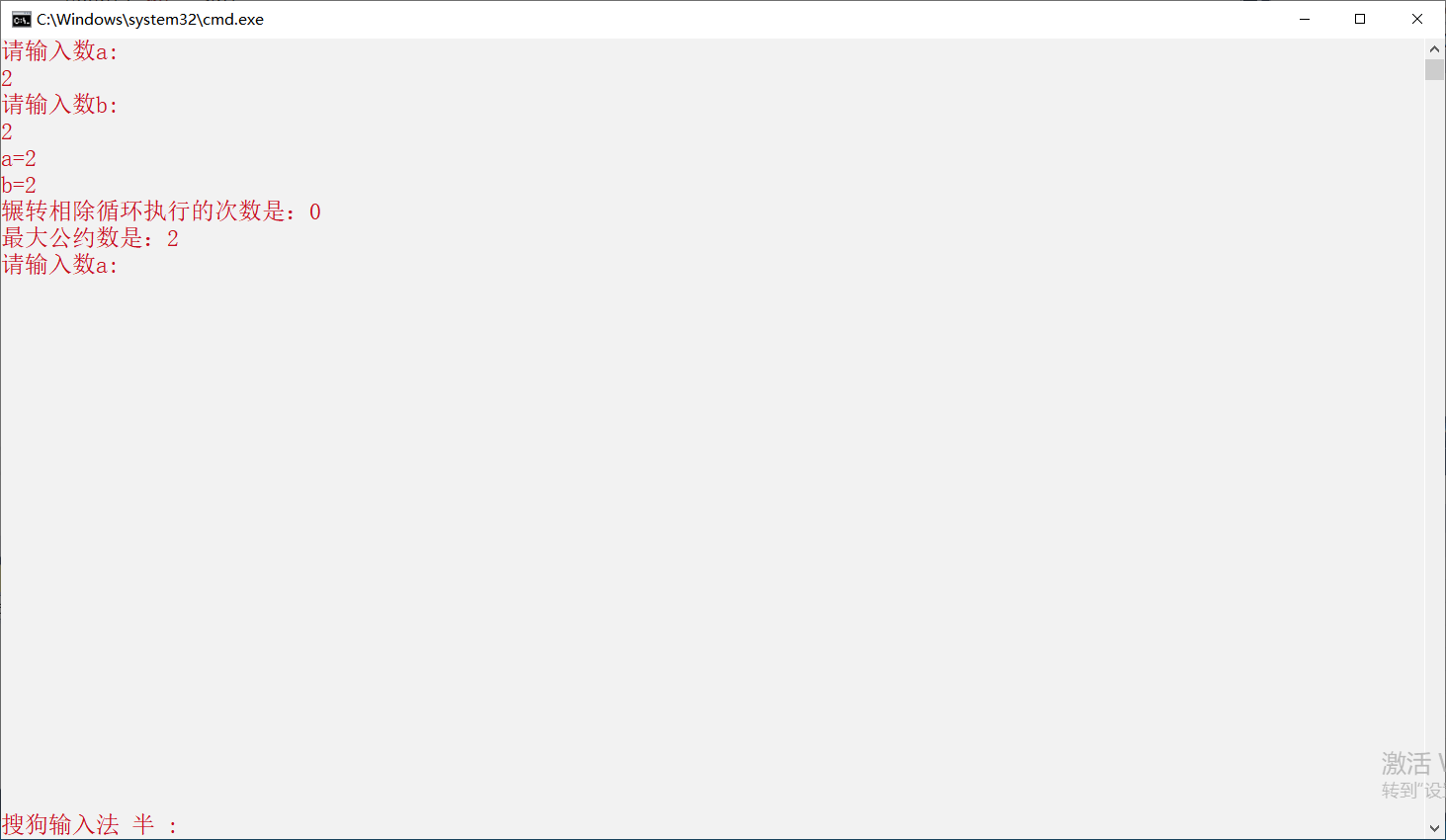
****

****

****

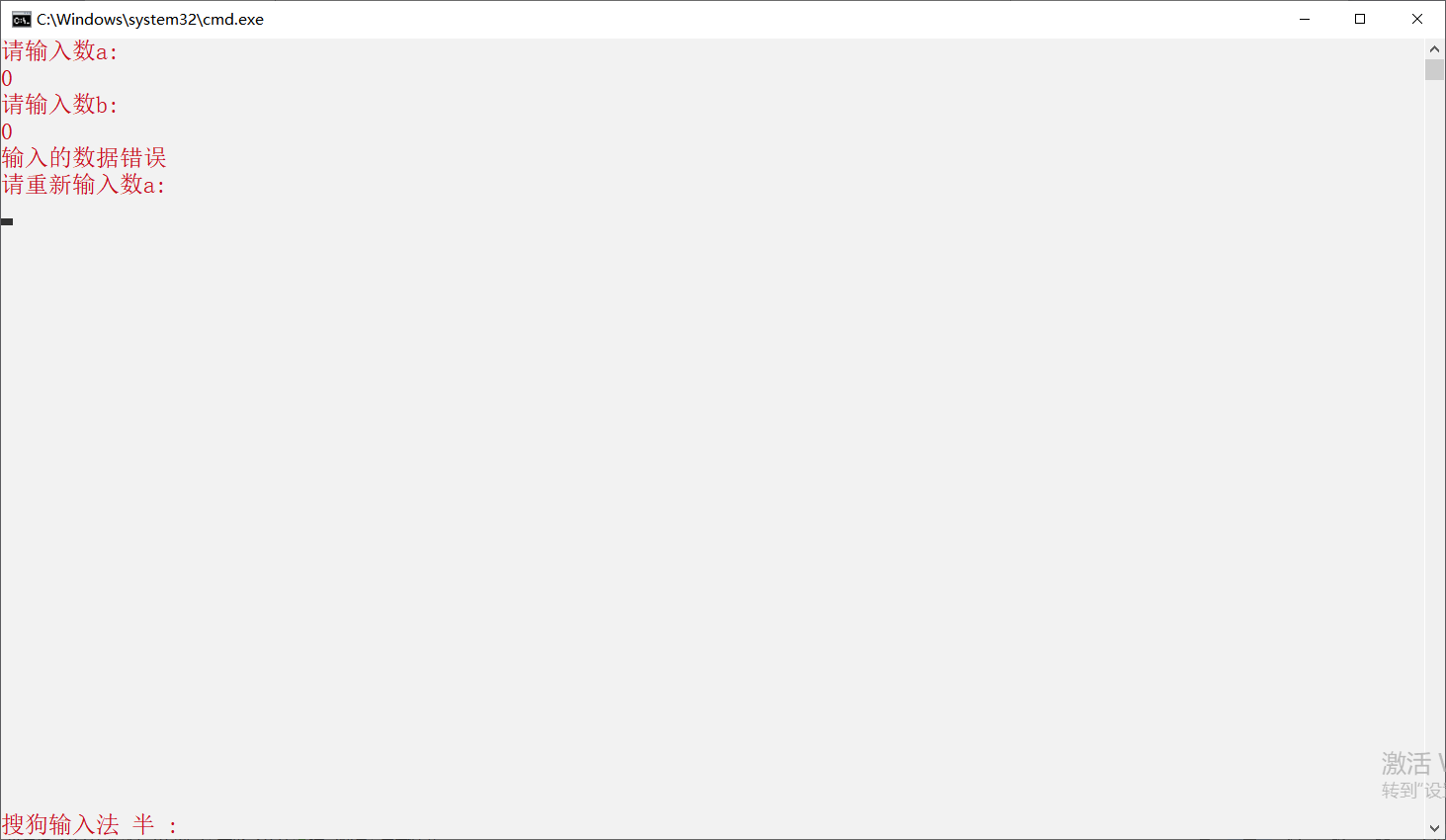
****

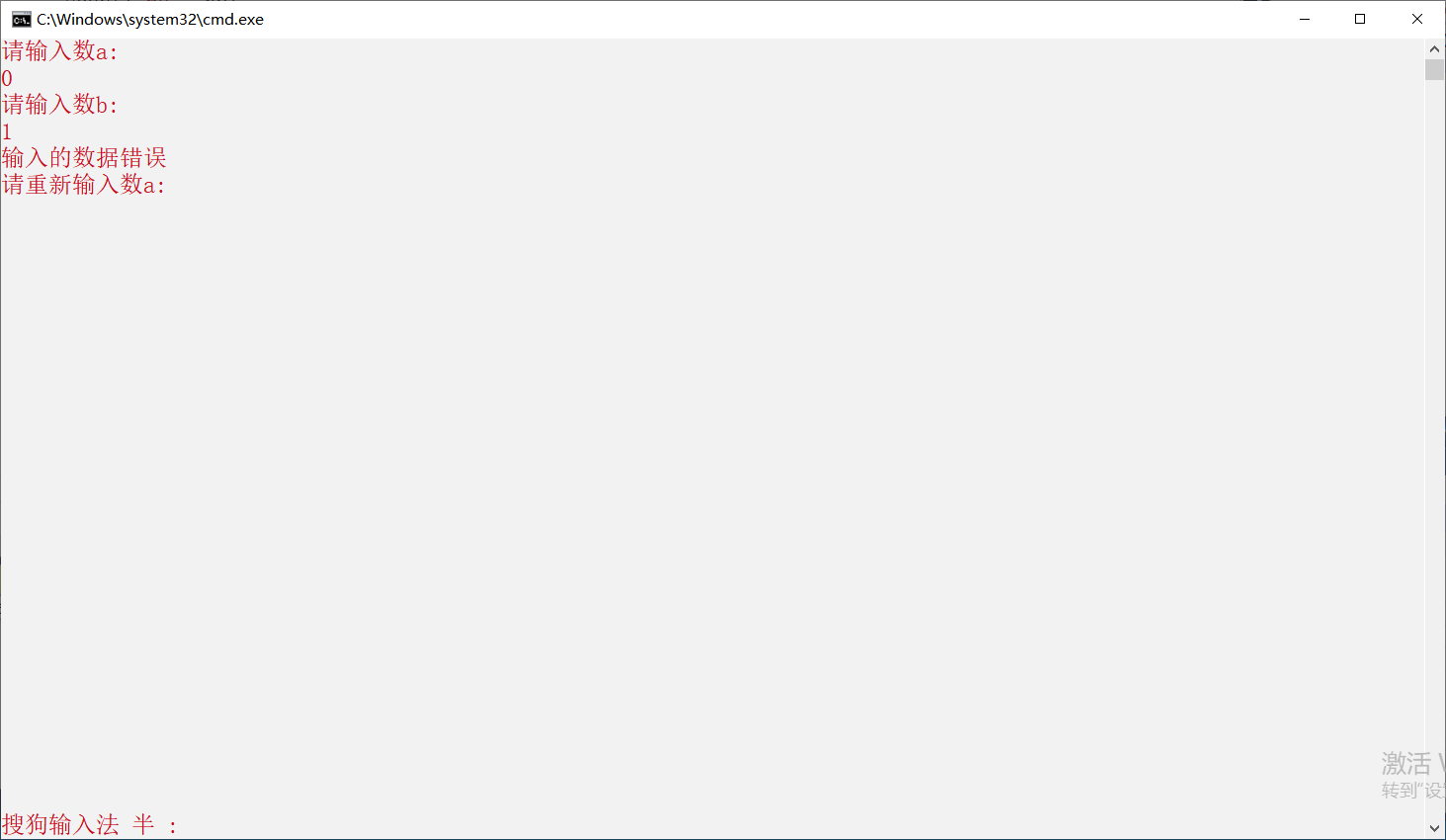
****

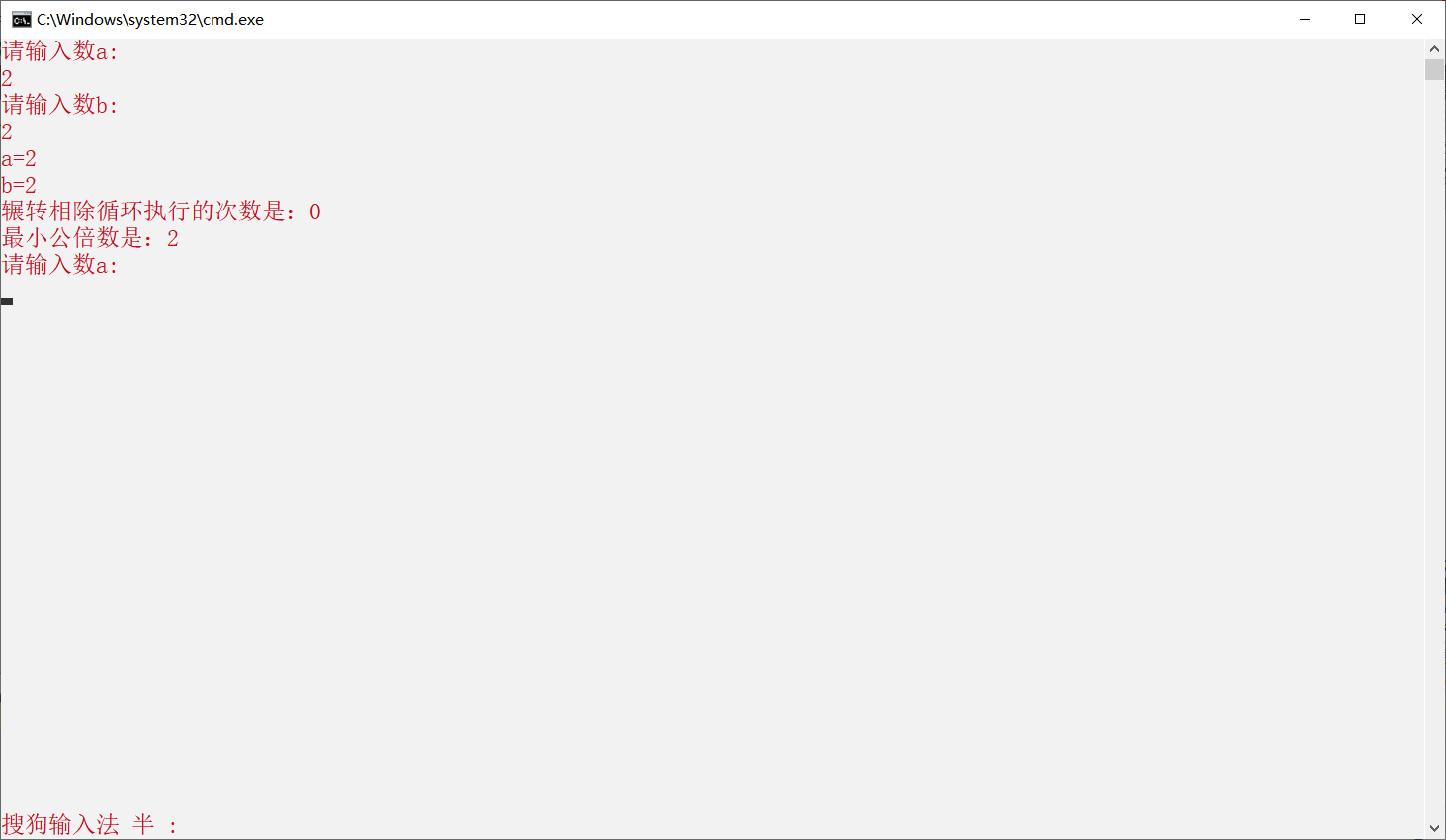
****

****

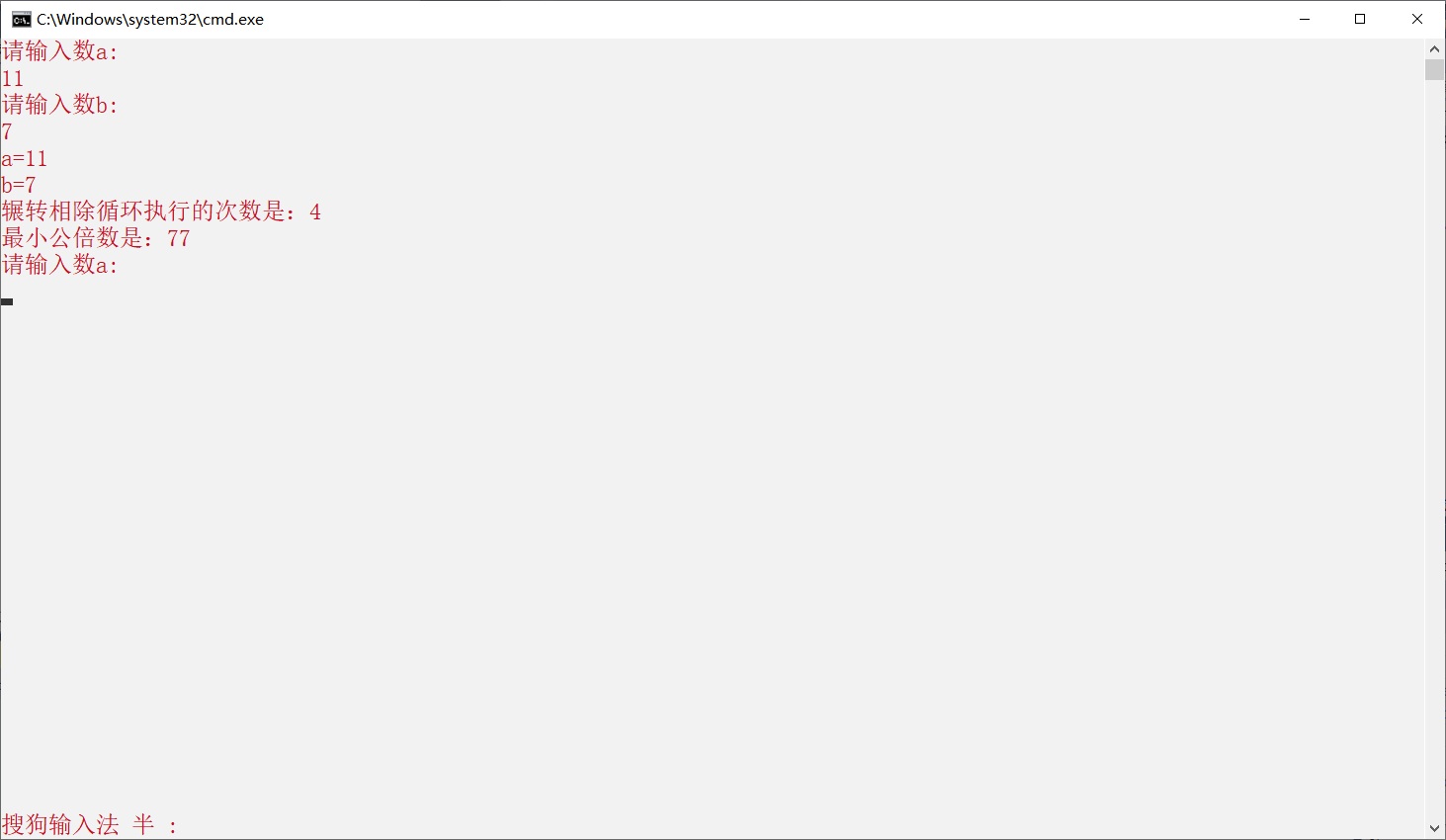
****

****

****

****

****

****

1. **实验数据**

**输入值测试**

**A.输入浮点型：0.5**

**B.输入值超出int范围**

**C.输入非法值（如空值、符号等）：！**

**D.输入正常int值：5 5**

**测试求最大公约数**

**A.0 0**

**B.0 1**

**C.两个数相同：2 2**

**D.两个数互有倍数关系：2 4**

**E.两个质数：11 7**

**测试求最小公倍数**

**A.0 0**

**B.0 1**

**C.两个数相同：2 2**

**D.两个数互有倍数关系：2 4**

**E.两个质数：11 7**

1. **程序代码**

**#include<stdio.h>**

**int gcd(int a, int b)**

**{**

**int temp;**

**int i; i = 0;**

**if (a<b)**

**{**

**temp = b; b = a; a = temp;**

**}**

**printf("a=%d\n", a);**

**printf("b=%d\n", b);**

**if (a == b)**

**{**

**printf("辗转相除循环执行的次数是：%d\n", i);**

**return a;**

**}**

**while (b != 0)**

**{**

**i = i + 1;**

**temp = a%b;**

**a = b;**

**b = temp;**

**}**

**printf("辗转相除循环执行的次数是：%d\n", i);**

**return a;**

**}**

**int icm(int a, int b)**

**{**

**return a\*b / gcd(a, b);**

**}**

**int main()**

**{**

**int a;**

**int b;**

**while (1)**

**{**

**printf("请输入数a:\n");**

**scanf("%d", &a);**

**printf("请输入数b:\n");**

**scanf("%d", &b);**

**while (a <= 0)**

**{**

**printf("输入的数据错误\n");**

**printf("请重新输入数a:\n");**

**scanf("%d", &a);**

**printf("现在数a=%d\n", a);**

**}**

**while (b <= 0)**

**{**

**printf("输入的数据错误\n");**

**printf("请重新输入数b:\n");**

**scanf("%d", &b);**

**printf("现在数b=%d\n", b);**

**}**

**printf("最大公约数是：%d\n", gcd(a, b));**

**printf("最小公倍数是：%d\n", icm(a, b));**

**}**

**return 0;**

**}**

1. **实验分析**
2. **实验数据分析**

**测试用例中，测试结果与预期想要的到的结果一样，代码中的两个方法可以实现预期功能，不存在问题。**

1. **实验代码分析**

**利用辗转相除法求得最大公因数**

**int gcd(int a, int b)**

**{**

**int temp;**

**int i; i = 0;**

**if (a<b)**

**{**

**temp = b; b = a; a = temp;**

**}**

**printf("a=%d\n", a);**

**printf("b=%d\n", b);**

**if (a == b)**

**{**

**printf("辗转相除循环执行的次数是：%d\n", i);**

**return a;**

**}**

**while (b != 0)**

**{**

**i = i + 1;**

**temp = a%b;**

**a = b;**

**b = temp;**

**}**

**printf("辗转相除循环执行的次数是：%d\n", i);**

**return a;**

**}**

**利用求得的最大公约数求得最小公倍数**

**int icm(int a, int b)**

**{**

**return a\*b / gcd(a, b);**

**}**

1. **实验过程分析**

**按照软件测试计划书要求，逐步进行测试，过程执行正常。**

**软件测试计划书（转下页）**

软件测试计划书

目录

[1. 引言 1](#_Toc272916920)

[1.1. 目的 1](#_Toc272916921)

[1.2. 背景 1](#_Toc272916922)

[1.3. 范围 1](#_Toc272916923)

[1.4. 定义 1](#_Toc272916924)

[1.5. 参考资料 1](#_Toc272916925)

[2. 测试内容 1](#_Toc272916926)

[3. 测试规则 1](#_Toc272916927)

[3.1. 进入准则 1](#_Toc272916928)

[3.2. 暂停/退出准则 1](#_Toc272916929)

[3.3. 测试方法 2](#_Toc272916930)

[3.4. 当完成模块测试后进行整个系统的功能测试测试手段 2](#_Toc272916931)

[3.5. 测试要点 2](#_Toc272916932)

[3.6. 测试工具 2](#_Toc272916933)

[4. 测试环境 2](#_Toc272916934)

[4.1. 硬件环境 2](#_Toc272916935)

[4.2. 软件环境 2](#_Toc272916936)

[4.3. 安全性环境要求 2](#_Toc272916937)

[5. 项目任务 2](#_Toc272916938)

[5.1. 测试规划 2](#_Toc272916939)

[5.2. 测试设计 2](#_Toc272916940)

[5.3. 测试执行准备 2](#_Toc272916941)

[5.4. 测试执行 2](#_Toc272916942)

[5.5. 测试总结 2](#_Toc272916943)

[6. 实施计划 3](#_Toc272916944)

[6.1. 工作量估计 3](#_Toc272916945)

[6.2. 人员需求及安排 3](#_Toc272916946)

[6.3. 进度安排 3](#_Toc272916947)

[6.4. 可交付工件 3](#_Toc272916948)

[7. 风险管理 3](#_Toc272916949)

1. 引言
   1. 目的

测试代码“输入两个int型数a,b，求两个数的最大公约数跟最小公倍数”是否能实现其功能，并测试是否存bug。预期达到能够使该代码求出任意两整型数字的最大公约数和最小公倍数。

* 1. 背景

基于一段现有代码进行简单的白盒测试。代码如下：

#include<stdio.h>

int gcd(int a, int b)

{

int temp;

int i; i = 0;

if (a<b)

{

temp = b; b = a; a = temp;

}

printf("a=%d\n", a);

printf("b=%d\n", b);

if (a == b)

{

printf("辗转相除循环执行的次数是：%d\n", i);

return a;

}

while (b != 0)

{

i = i + 1;

temp = a%b;

a = b;

b = temp;

}

printf("辗转相除循环执行的次数是：%d\n", i);

return a;

}

int icm(int a, int b)

{

return a\*b / gcd(a, b);

}

int main()

{

int a;

int b;

while (1)

{

printf("请输入数a:\n");

scanf("%d", &a);

printf("请输入数b:\n");

scanf("%d", &b);

while (a <= 0)

{

printf("输入的数据错误\n");

printf("请重新输入数a:\n");

scanf("%d", &a);

printf("现在数a=%d\n", a);

}

while (b <= 0)

{

printf("输入的数据错误\n");

printf("请重新输入数b:\n");

scanf("%d", &b);

printf("现在数b=%d\n", b);

}

printf("最大公约数是：%d\n", gcd(a, b));

printf("最小公倍数是：%d\n", icm(a, b));

}

return 0;

}

* 1. 范围

主要测试该代码能否达到预期功能，是否存在问题。对各个边界值进行详细测试，记录测试结果，对结果进行分析处理。测试时对“求两个整型的最大公约数”和“求两个整型的最小公倍数”分别进行测试，对所有可能的取值都要进行测试，测试过程中对存在的问题加以分析，然后提交测试记录。最后，对代码存在的问题进行全面分析，并给予记录。在测试的过程中需要列出可能会影响测试设计、开发、或实施的所有风险或意外事件。列出可能会影响测试设计、开发或实施的所有约束。

* 1. 定义

gcd(),为求最大公约数函数

icm（）为求最小公倍数函数

变量i来存储循环体执行的次数

* 1. 参考资料

编写本计划及测试整个过程中所要参考的文件、资料

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **资料名称** | **作者** | **日期** | **出版单位** |
| 1 | 《软件测试入门与提高》 | 张成明 | 2008.6 | 清华大学出版社 |
| 2 | 《软件测试基础教程》 | 刘建宇 | 2007.3 | 邮电大学出版社 |

1. 测试内容

测试代码“输入两个int型数a,b，求两个数的最大公约数跟最小公倍数”是否能实现其功能，并测试代码是否存在bug。

1. 测试规则

3.1.进入准则

在VS2013中插入代码，进行各项测试。

3.2.暂停/退出准则

代码在编译过程中出现错误或警告时退出程序并记录，检测输入值、结果与预期不同或者出现警告、窗口提示、自动退出等状态时要暂停或退出程序并加以记录，正常执行完整体操作时退出程序并记录。

3.3.测试方法

本次测试采用白盒测试方法，将代码划分为“求两个整型的最大公约数”和“求两个整型的最小公倍数”两个模块分别进行测试。分别对输入值类型和取值范围进行检测、各个特殊取值进行检测。

## 3.4.当完成模块测试后进行整个代码的功能测试测试手段

当分模块测试进行结束后，同时调用两种方法进行组合测试，看该代码是否依旧满足预期要求。

3.5.测试要点

主要测试该代码能否实现预期要求，代码是否存在bug等。

3.6.测试工具

VS2013

4.测试环境

4.1.硬件环境

处理器：Intel i7 7500U或更高

内存：4GB 以上

硬盘空间：1TB 以上

显卡：NVIDA 940M显示适配器

4.2.软件环境

VS2013(Personal)

4.3.安全性环境要求

操作系统的安全性，测试工具的安全性，测试软件的安全性。

5.项目任务

5.1.测试规划

按测试用例逐一进行测试，记录结果，最终分析。

5.2.测试设计

将正常取值、特殊取值、异常取值均进行测试，测试不同取值状况下代码是否正常执行。

5.3.测试执行准备

保证整体测试环境的稳定及流畅，保证测试用例全面，尽量减少测试过程中被外界因素中断。

5.4.测试执行

按测试用例逐一进行测试。

5.5.测试总结

结合测试得出的结果进行错误整合以及原因分析。

1. 实施计划
   1. 工作量估计

三小时之内可以完成测试整体流程。

* 1. 人员需求及安排

一个人进行整个测试工作。

* 1. 进度安排

尽量确保三个小时内将整体流程进行完毕。

* 1. 可交付工作

无需交付工作，自身进行后续分析处理等。

1. 风险管理

将风险划分如下三个等级：L=Low(风险与处理的优先级为低)；M=Middle(风险与处理的优先级为中)；H=High(风险与处理的优先级为高)。按照测试过程中具体情况进行风险评估。

测试用例编写

1. 输入值测试
2. 输入浮点型：0.5
3. 输入值超出int范围
4. 输入非法值（如空值、符号等）：！
5. 输入正常int值：5 5
6. 测试求最大公约数
7. 0 0
8. 0 1
9. 两个数相同：2 2
10. 两个数互有倍数关系：2 4
11. 两个质数：11 7
12. 测试求最小公倍数

（1）0 0

（2）0 1

（3）两个数相同：2 2

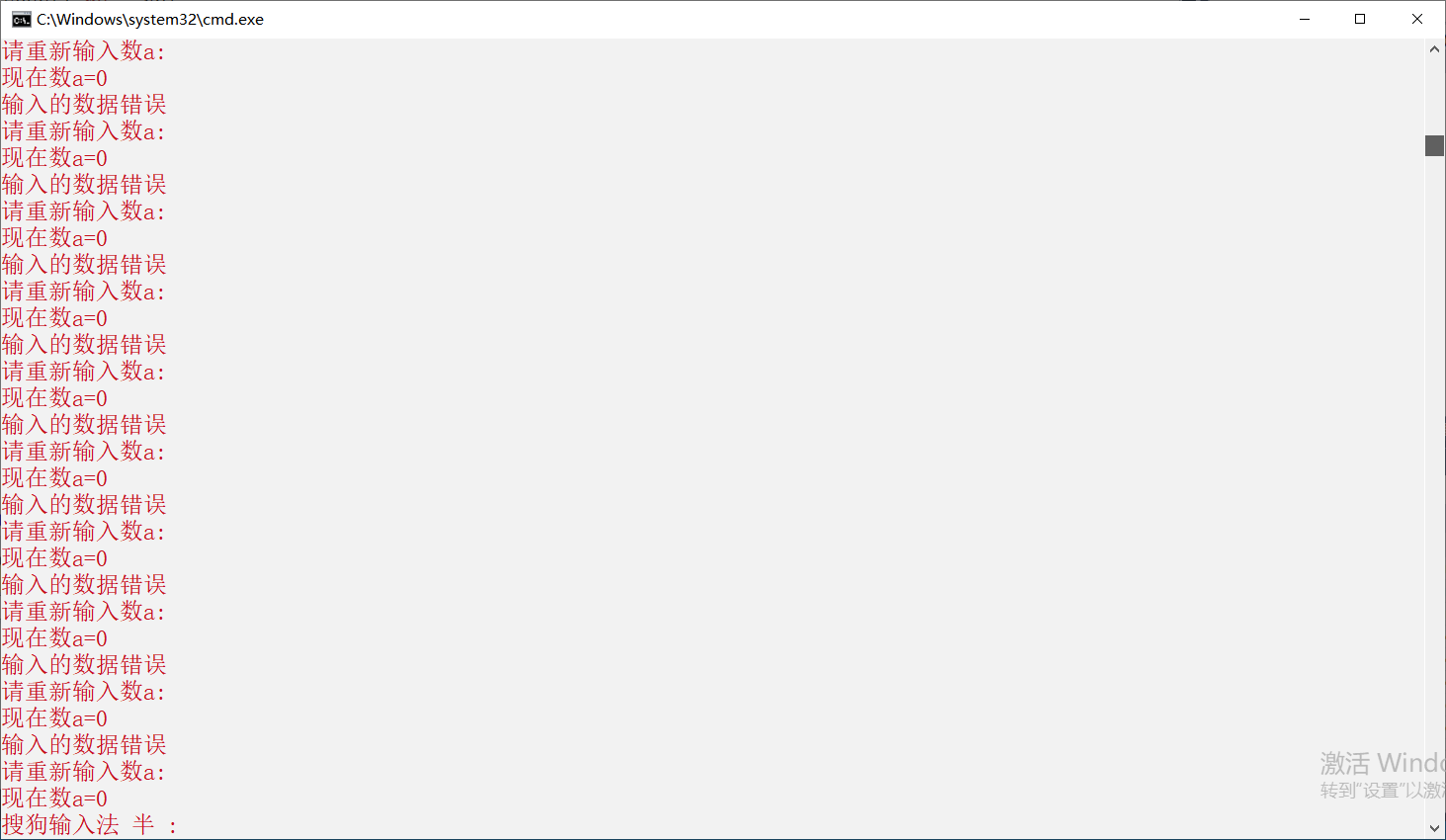
（4）两个数互有倍数关系：2 4

（5）两个质数：11 7

测试结果记录及分析

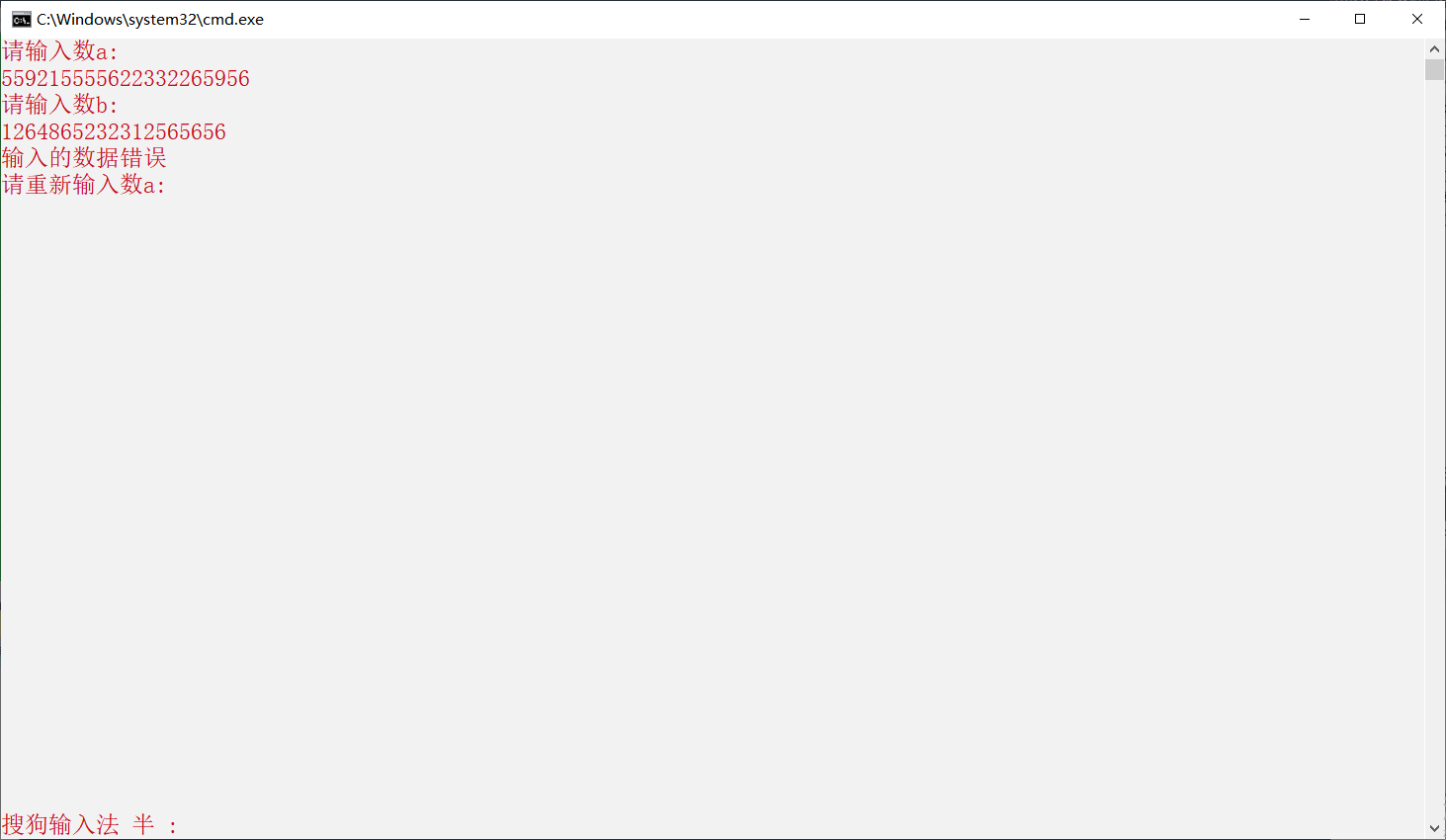
1.输入值测试

（1）输入浮点型：0.5



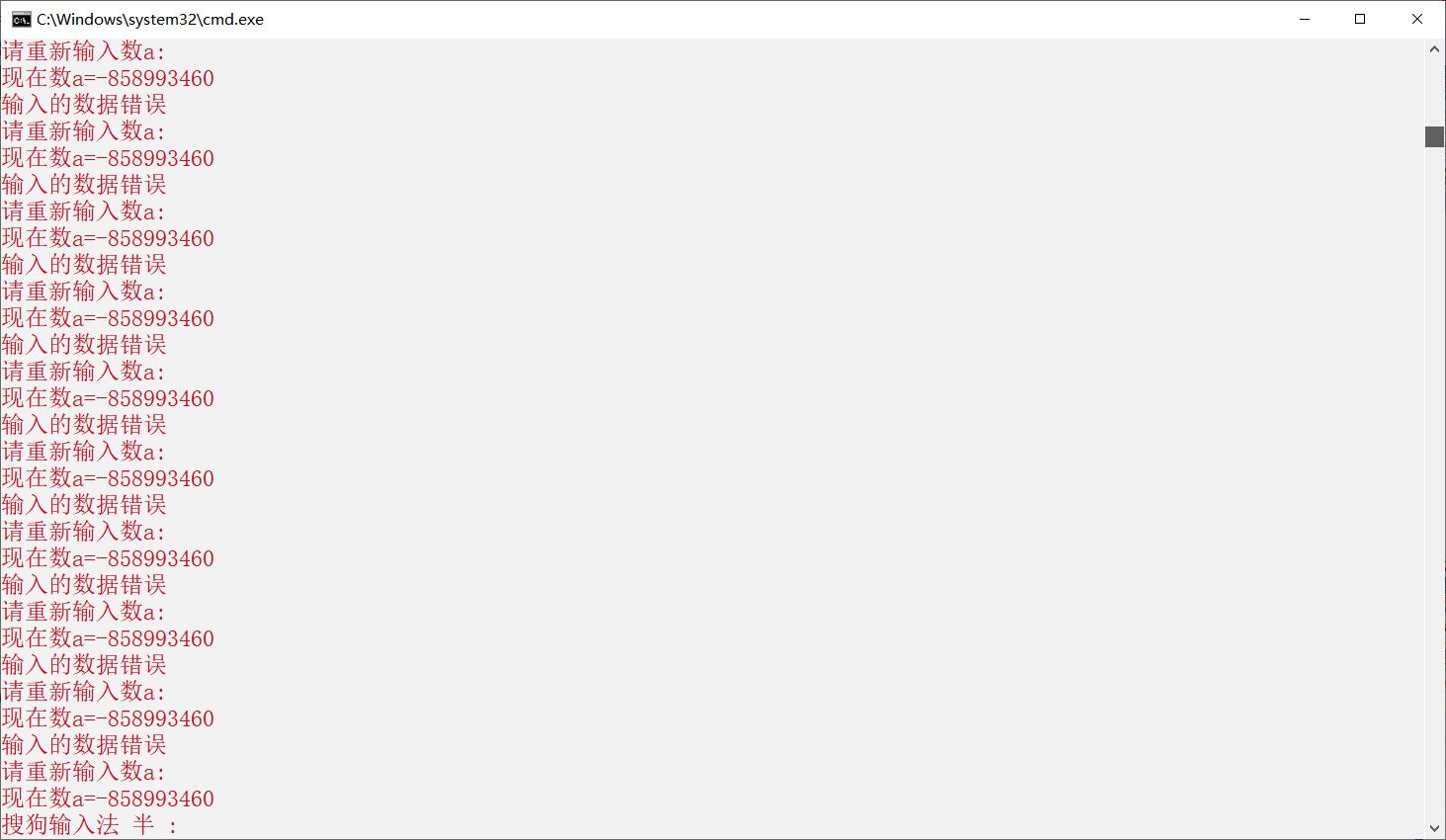
输入类型错误时控制台提示输入错误，与预期结果相同

1. 输入值超出int范围：



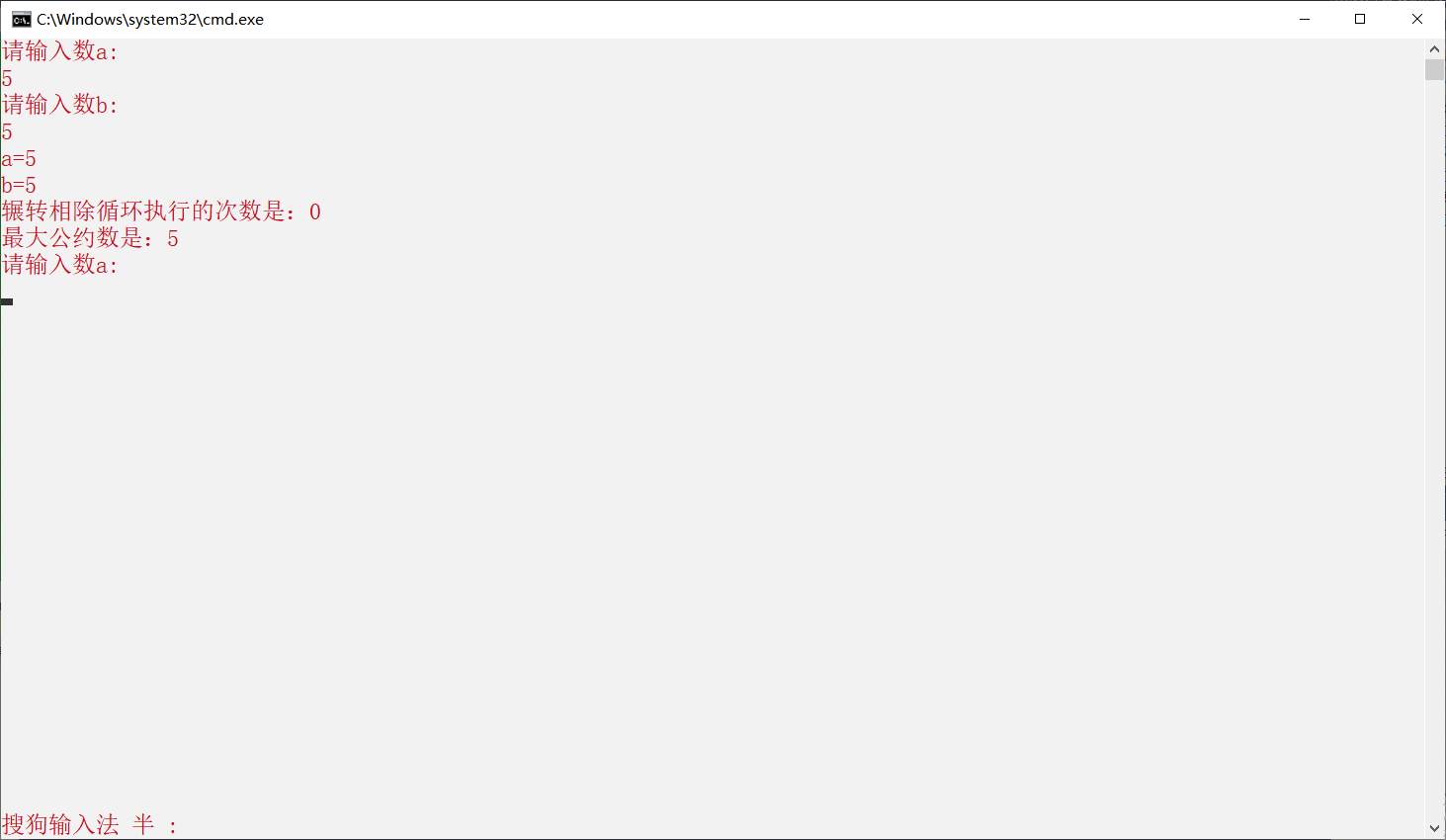
输入类型错误时控制台提示输入错误，与预期结果相同

1. 输入非法值（如空值、符号等）：！



输入类型错误时控制台提示输入错误，与预期结果相同

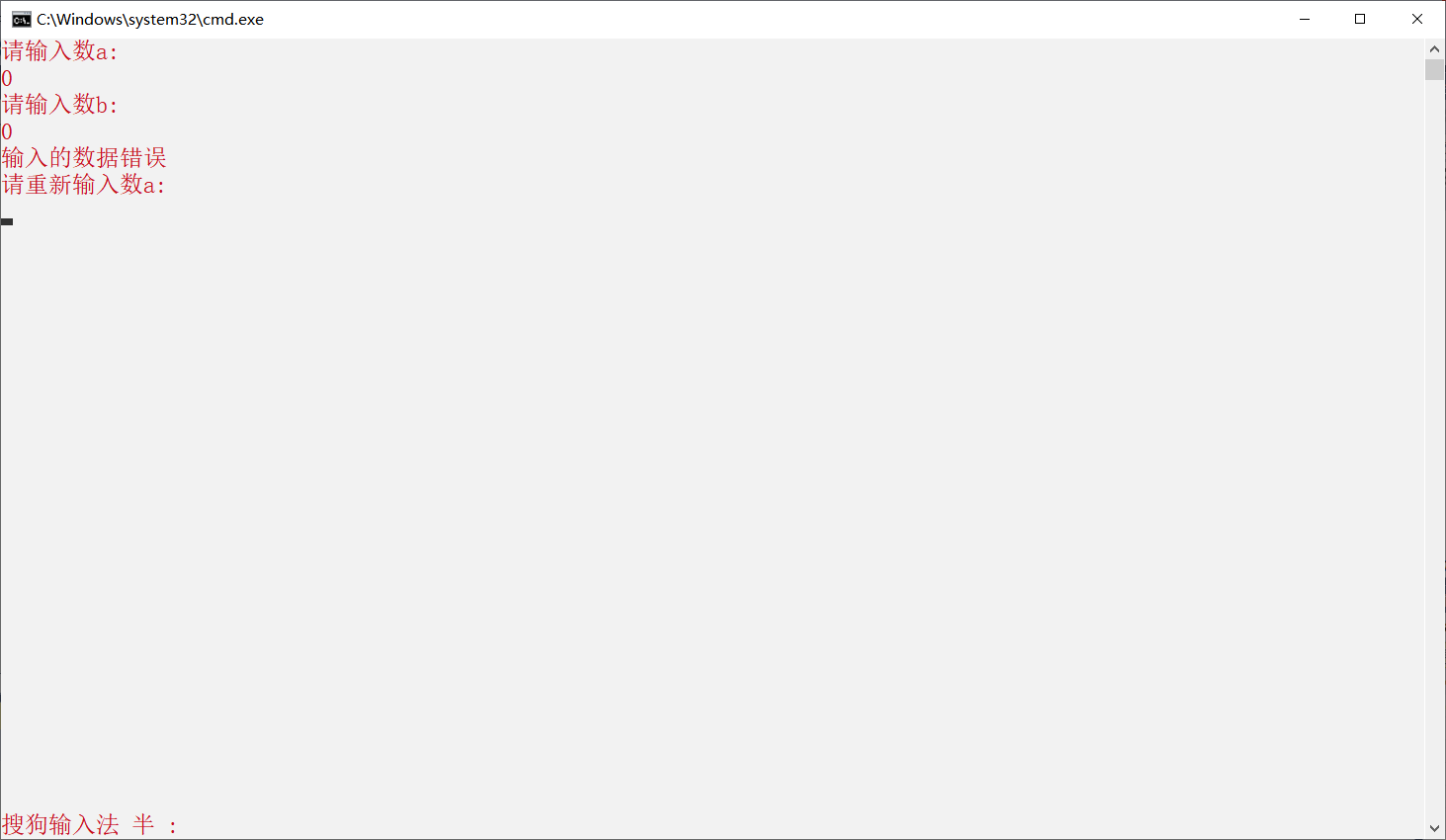
1. 输入正常int值：5 5



输入正常int值，程序正常运行，与预期结果相同

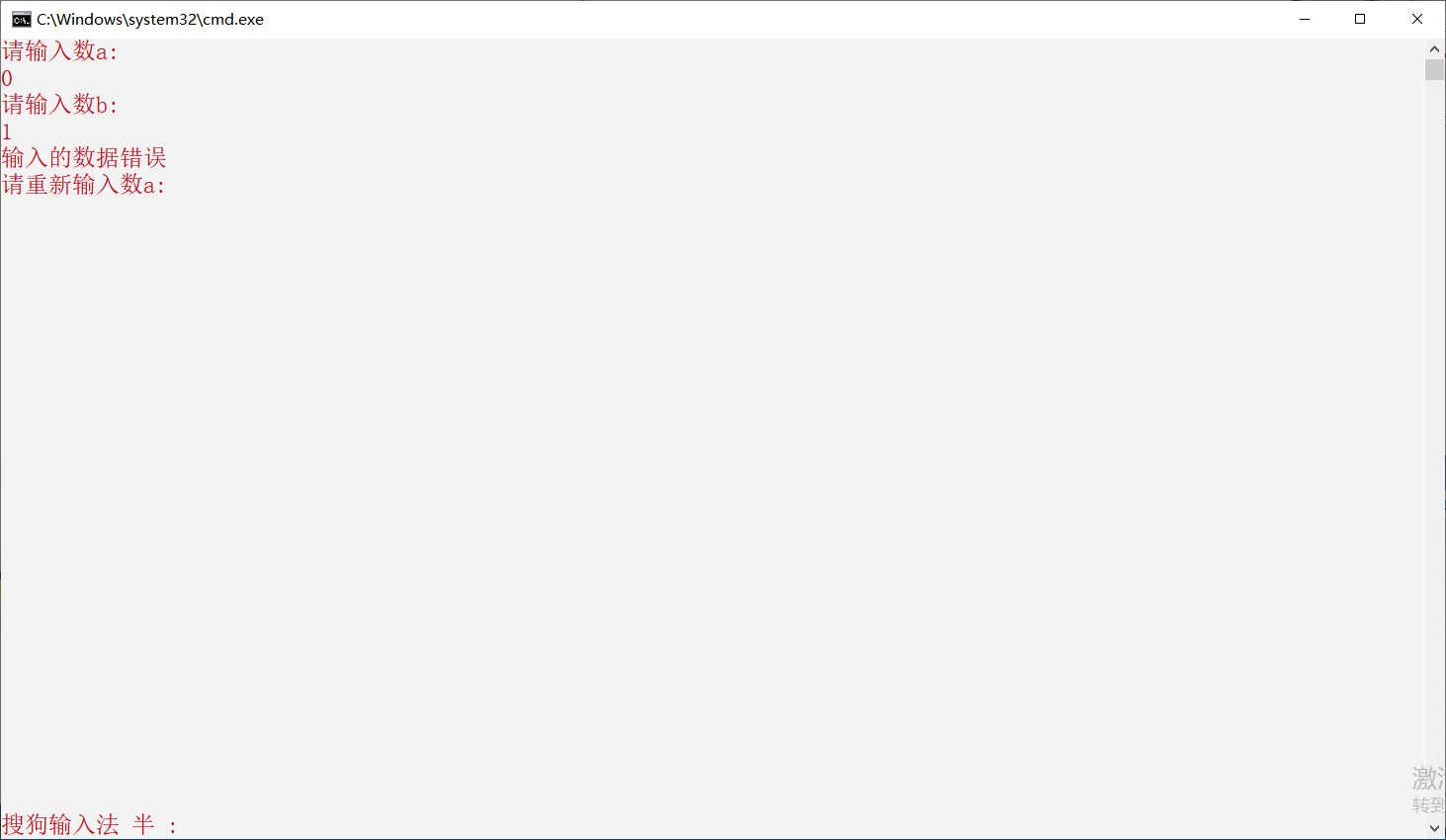
2.测试求最大公约数

（5）0 0



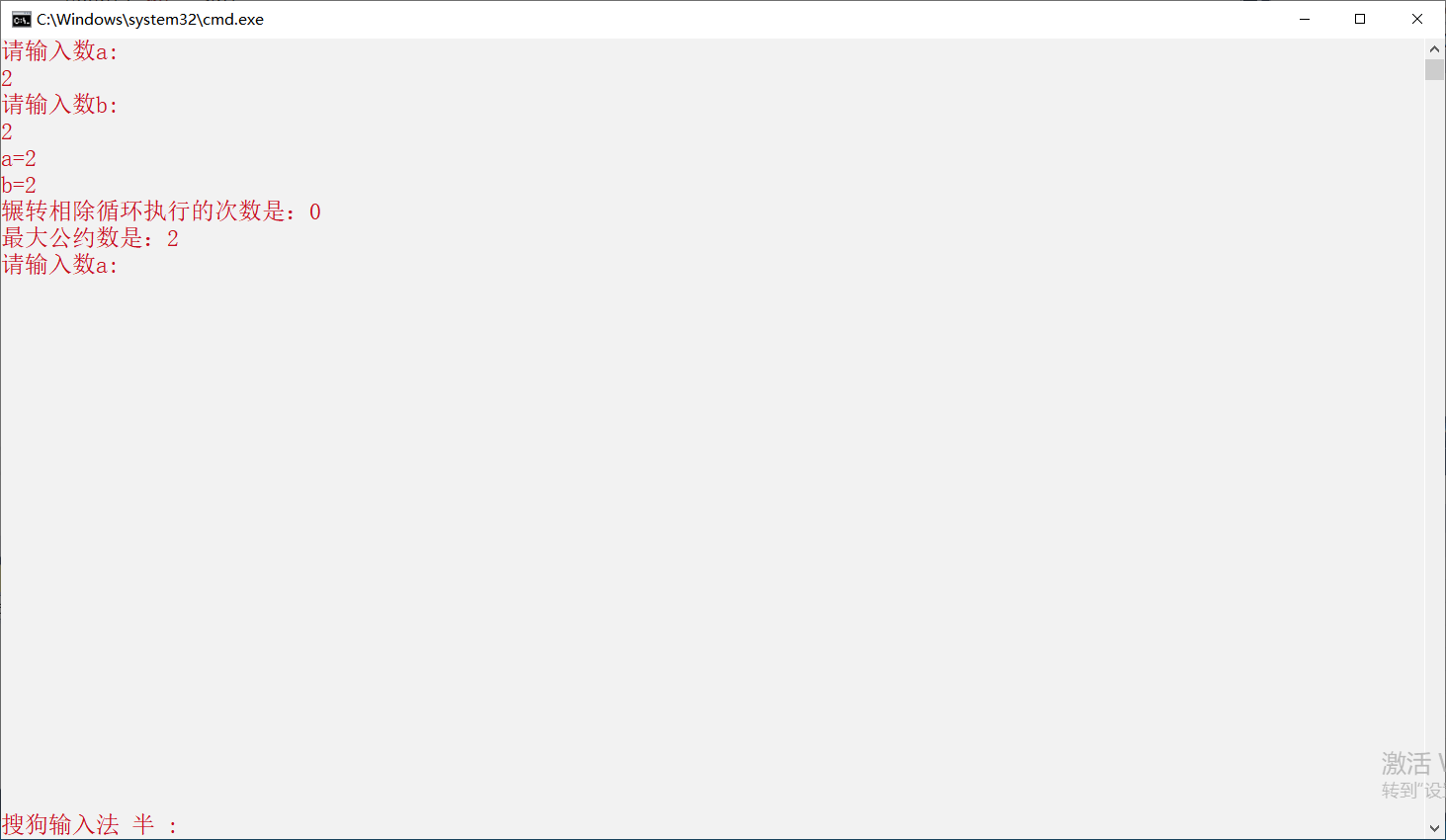
与预期结果相同

1. 0 1



与预期结果相同

1. 两个数相同：2 2



与预期结果相同

1. 两个数互有倍数关系：2 4



与预期结果相同

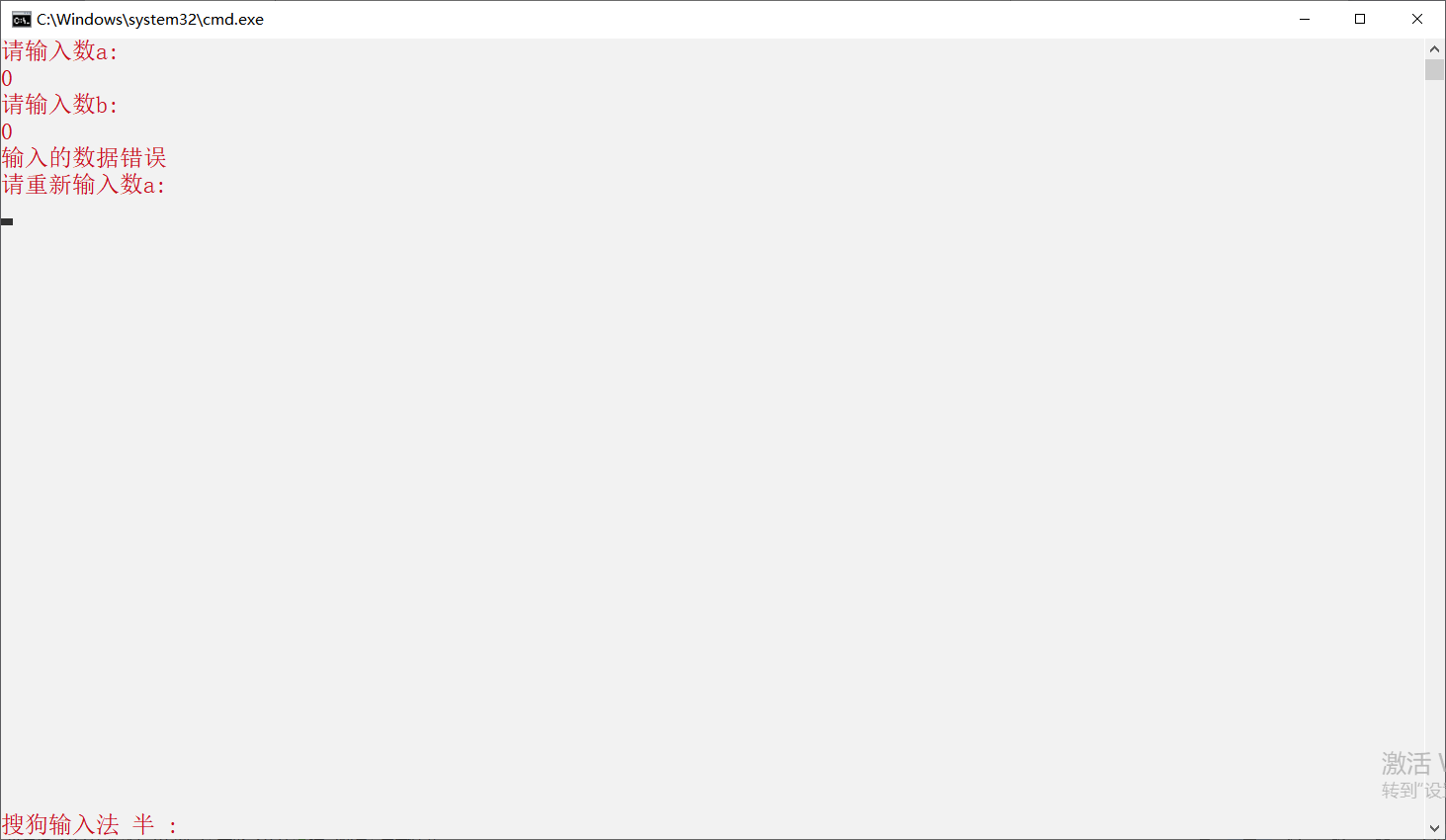
1. 两个质数：11 7



与预期结果相同

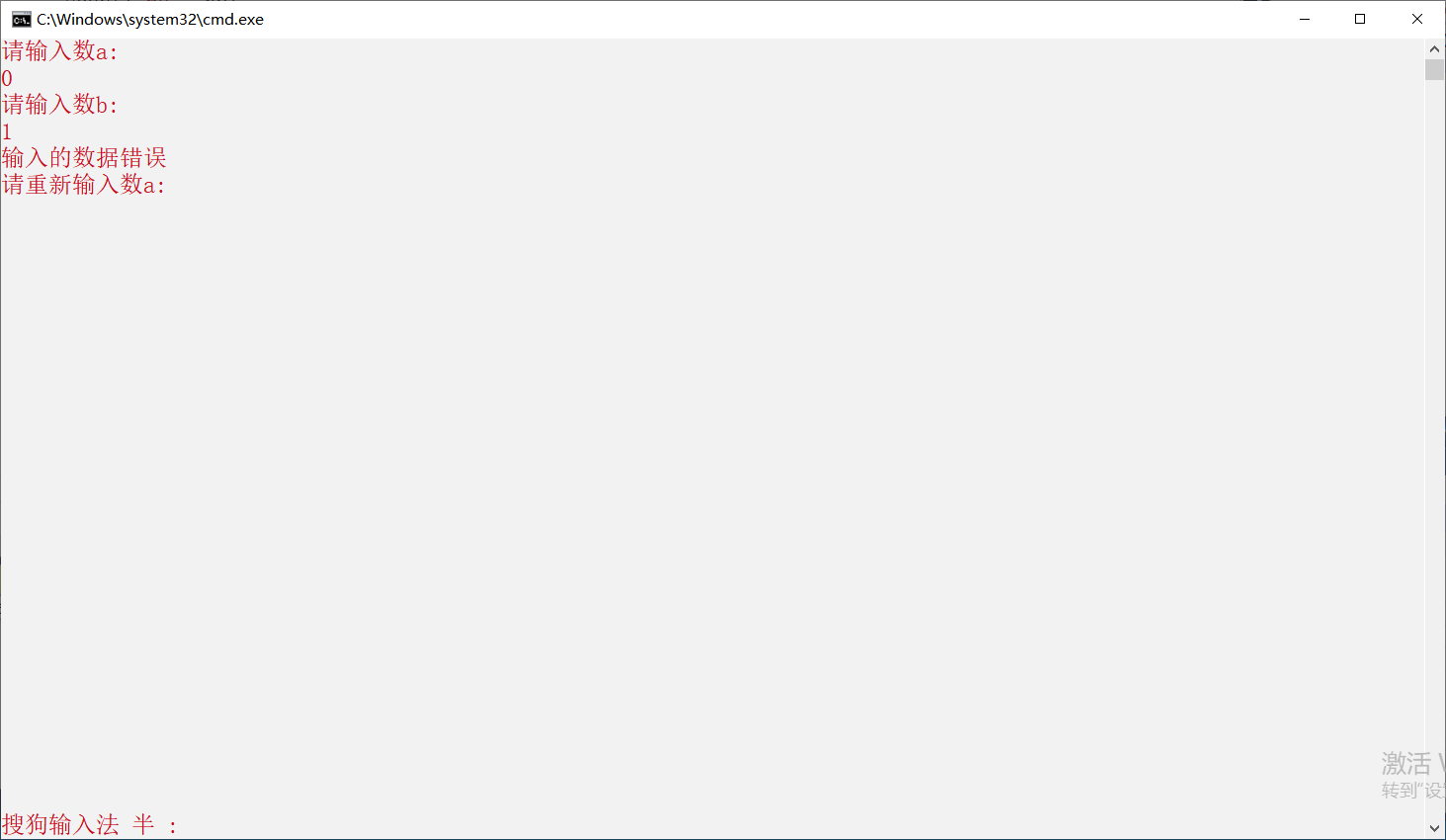
3.测试求最小公倍数

（10）0 0



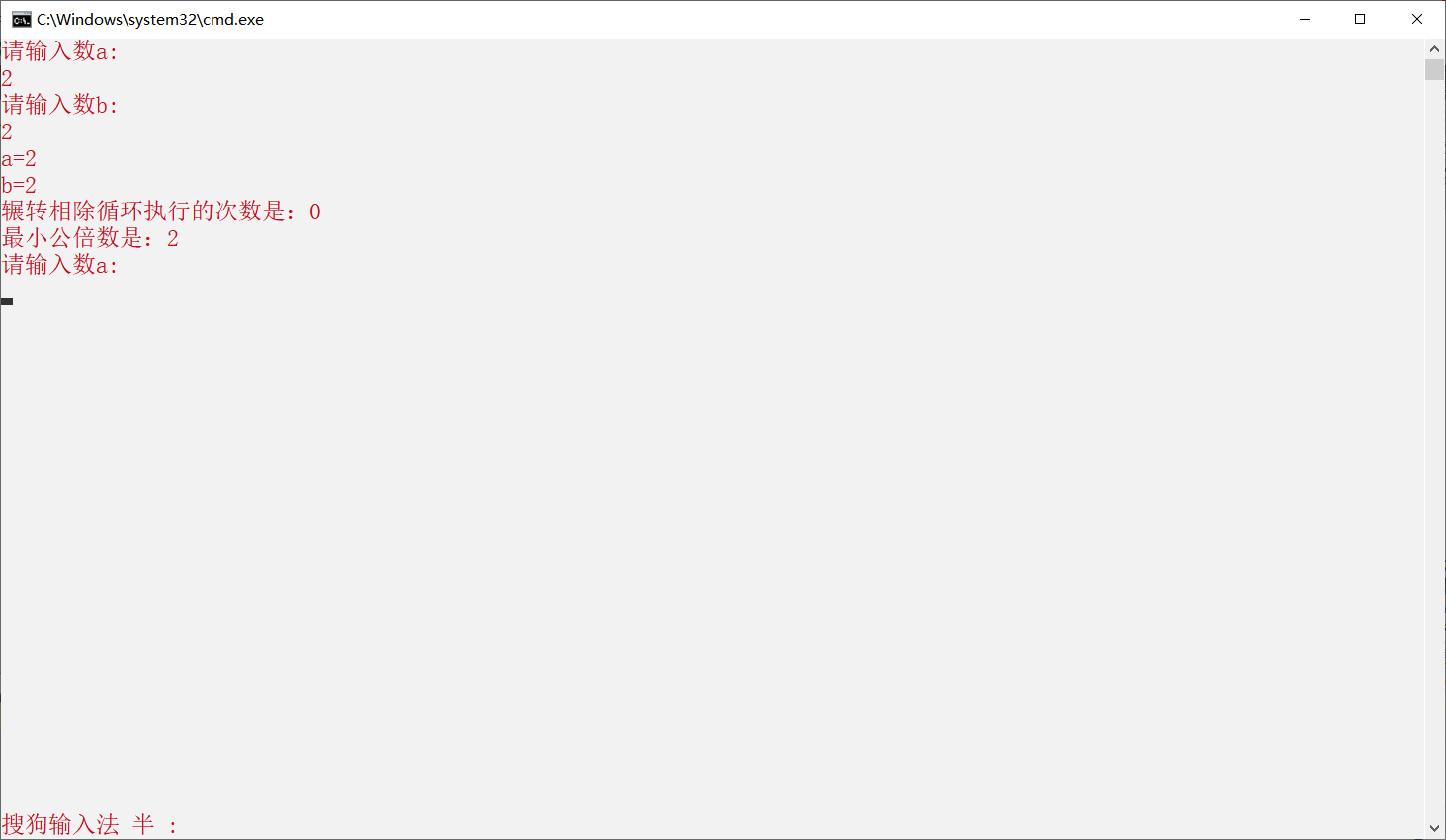
与预期结果相同

（11）0 1



与预期结果相同

1. 两个数相同：2 2



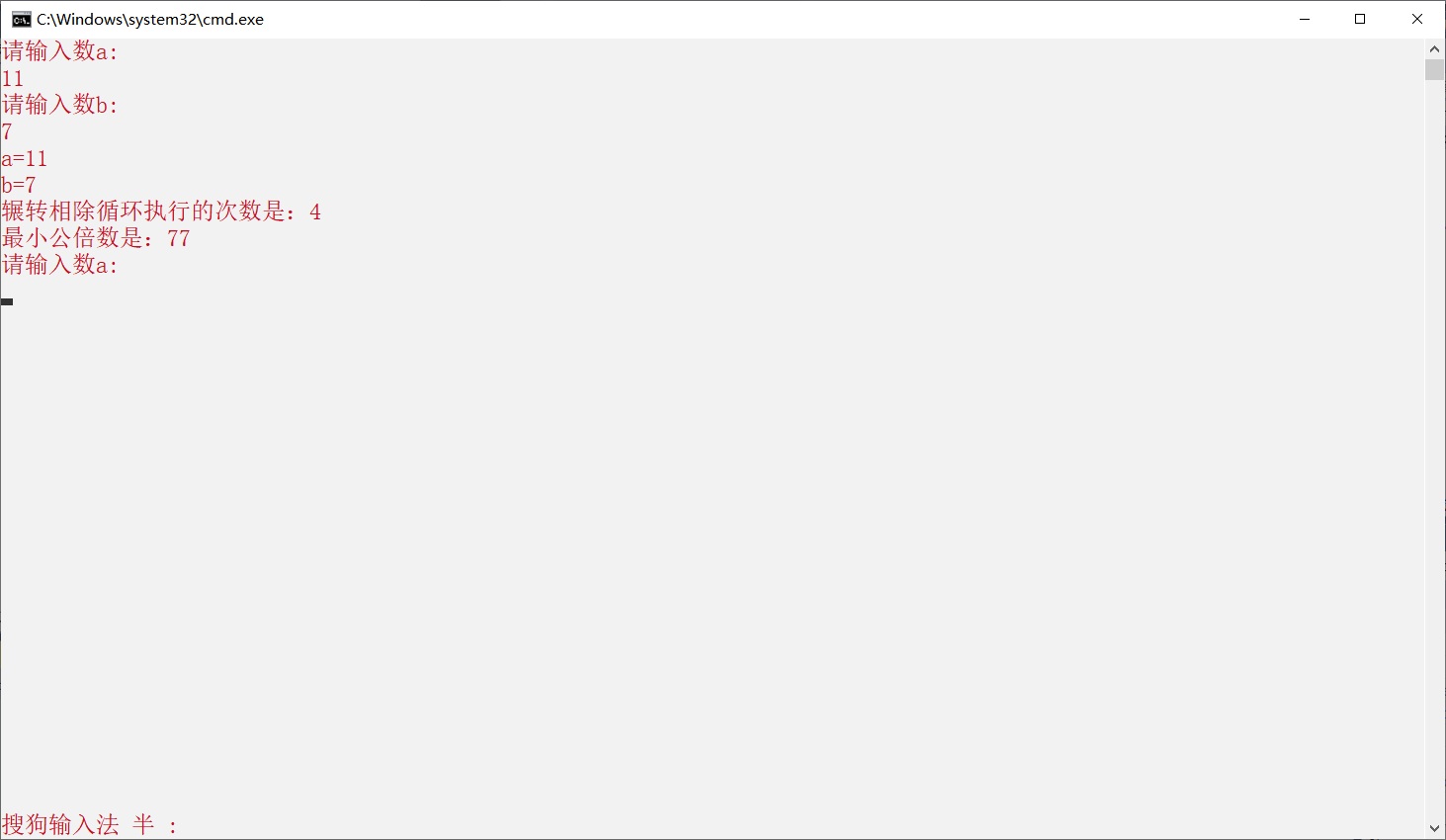
与预期结果相同

1. 两个数互有倍数关系：2 4



与预期结果相同

1. 两个质数：11 7



与预期结果相同

总结

该代码可以实现预期求两个整型的最大公因数和最小公倍数功能，对于输入错误等都有完备的操作错误提示。