# 第1次编程练习报告

姓名：刘修铭 学号：2112492 班级：信安一班

##### 一、编程练习1——Eratosthenes筛法

* **源码部分：**

#include<iostream>

using namespace std;

#define n 1000000

int main()

{

long int prime[n]={};

long int count = 0;

for (long int i = 0; i < n; i++)

{

prime[i] = 1;

}

for (long int i = 2; i < sqrt(n); i++)

{

if (prime[i] == 1)

{

for (long int j = i \* i; j < n; j += i)

{

prime[j] = 0;

}

}

}

for (long int i = 2; i < n; i++)

{

if (prime[i] == 1)

{

cout << i << ",";

count++;

}

}

cout << endl;

cout << "Total:" << count << endl;

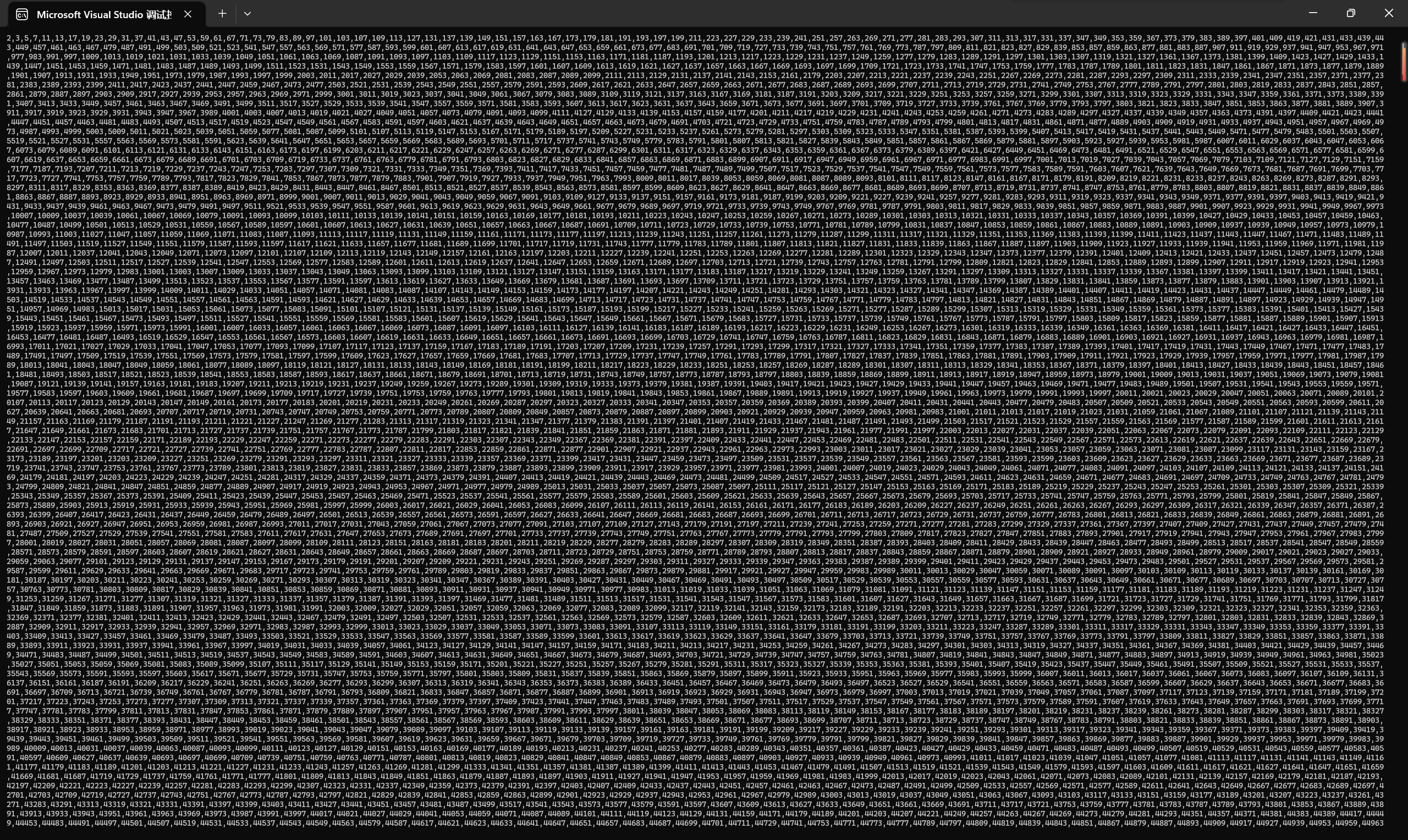
return 0;

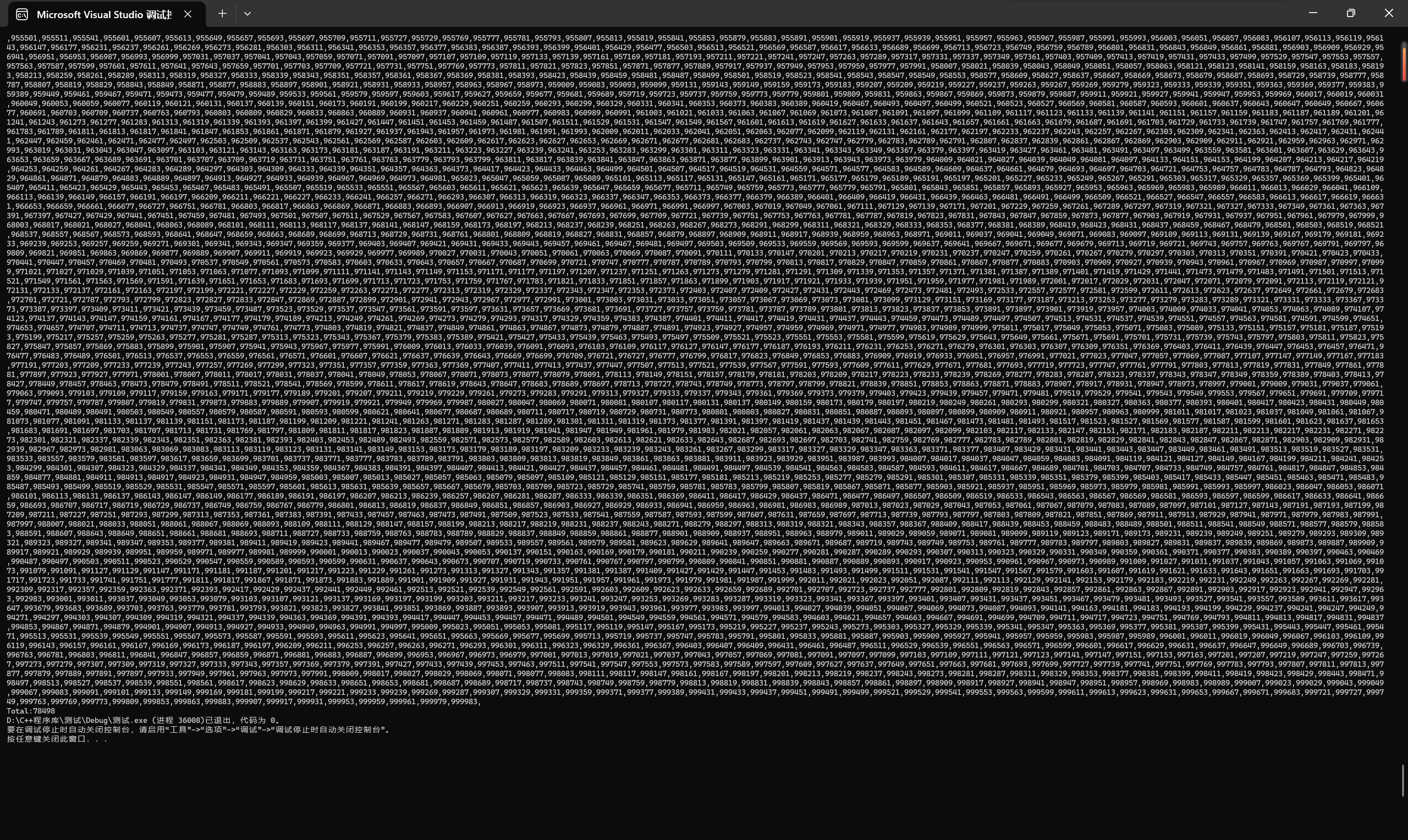
}

* **说明部分：**

设置一个prime数组，数组的每个位置值为0或1，用来存储该下标是否为素数。最后以此作为判定条件，输出所有素数。

* **运行示例：**





* **其他：**

1.性能差异：普通算法为O（n\*sqrt（n）），Eratosthenes筛法为O（n\*log logn）

2.递归调用：对于较大的n进行开方处理。

3.求更大素数：不再适用。此方法会占用大量内存空间，导致溢出。

##### 二、编程练习2——计算最大公因数和最小公倍数

* **源码部分：**

#include<iostream>

using namespace std;

int gcd(int x, int y)

{

int z = y;

while (x % y != 0)

{

z = x % y;

x = y;

y = z;

}

return z;

}

int lcm(int a, int b)

{

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

}

void main() {

int a, b;

cout << "a=";

cin >> a;

cout << "b=";

cin >> b;

cout << "gcd(a,b)=" << gcd(a, b) << endl;

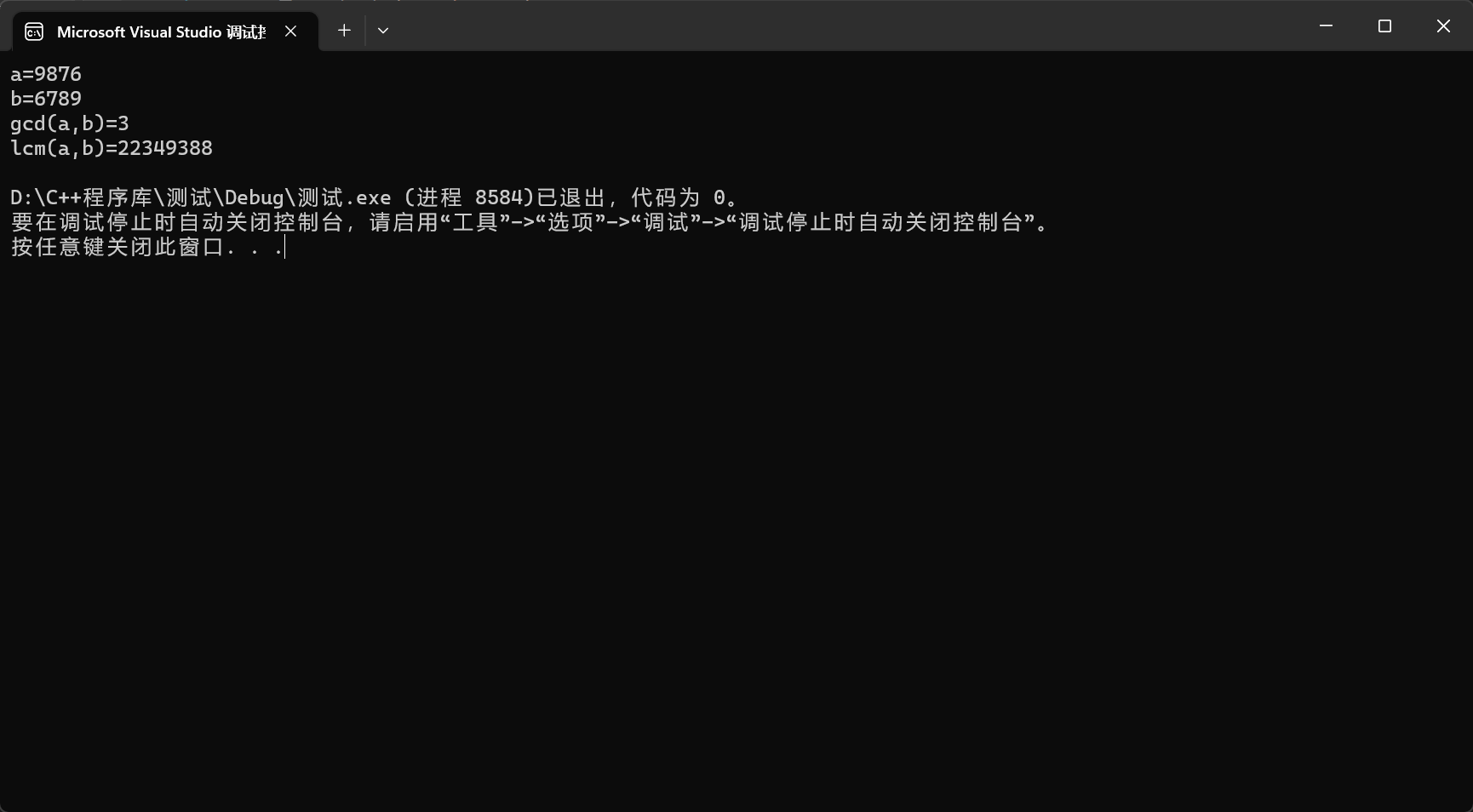
cout << "lcm(a,b)=" << lcm(a, b) << endl;

}

* **说明部分：**

此代码利用辗转相除法求最大公因子，然后利用最小公倍数的计算式，调用前面已求得的最大公因子得到最小公倍数。

* **运行示例：**



##### 三、编程练习3——实现算术基本定理

* **源码部分：**

#include<iostream>

#include<math.h>

using namespace std;

int main()

{

int num;

int sum = 1;

bool flag;

cin >> num;

cout << num << "=";

int t = num;

for (int i = 2; i < num; i++)

{

flag = true;

for (int m = 2; m < sqrtf(i); m++)

{

if (i % m == 0)

{

flag = false;

break;

}

}

if (flag)

{

while (t % i == 0)

{

sum = sum \* i;

if (sum < num)

{

cout << i << "\*";

}

else if (sum == num)

cout << i << endl;

t = t / i;

}

}

}

return 0;

}

* **说明部分：**

从2开始到sqrt（n）逐个进行取模运算，若为0，则进行flag标记，然后进行乘法计算，直到二者相等。

* **运行示例：**

