# 第1次编程练习报告

姓名：孙蕗 学号：2112060 班级：信安一班

##### **编程练习1——Eratosthenes筛法**

* **源码部分：**

#include<iostream>

using namespace std;

int main()

{

bool p[1000001]; //p[i]=true 表示i是素数,p[i]=false表示i不是素数

int total = 0;//记录素数总个数

int a, b;//接收cin的两个数字

char temp;

cout << "Please input the range: ";

cin >> a >> temp >> b;//a=1,b=1000000

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

{

p[i] = true;//先全置为true

}

p[0] = false;

p[1] = false;//01不是素数

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

{

for (int j = i \* 2; j <= b; j += i)//i的相应整数倍数都不是素数置为false

{

p[j] = false;

}

}

for (int i = a; i <= b; i++)//遍历输出范围内的素数并统计素数总个数

{

if (p[i])

{

cout << i << ",";

total++;

}

}

cout << endl;

cout <<"Total: "<< total << endl;

return 0;

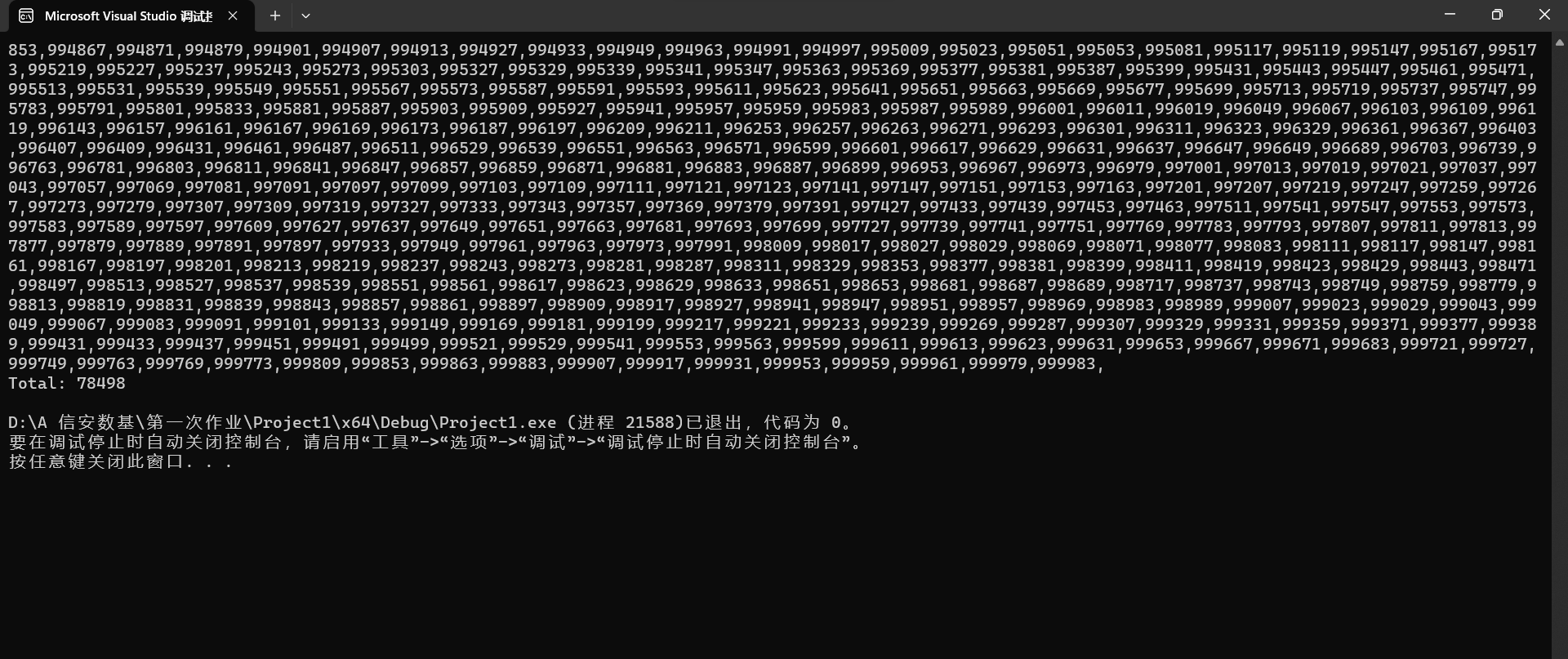
}

* **说明部分：**

取bool数组p[1000001]，当p[i]=true时表示i是素数；p[i]=false表示i不是素数。2为第一个素数，从2开始遍历，先将数组中每个数都置为true。从2开始两层for循环遍历，i为第一层循环，j为第二层循环，j表示范围内i的整数倍。i相应整数倍数j都不是素数，将p[j]置为false。最后遍历输出范围内的素数并统计素数总数。

* **运行示例：**





* **其他：**

a: Eratosthenes筛法的时间复杂度为O(nlogn)，欧拉算法时间复杂度为O(n)，普通筛选素数算法的时间复杂度为O(n)。

b: 优化，可将i<=b优化成i\*i<=b；i\*i之前的合数一定是某个小于i的素数的倍数，在前面的循环中已经被筛掉了，所以j可以直接从i \*i开始循环

c: 超过数组可以表示的范围

##### **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 m, int n)

{

if (m < n)

{

int temp;

temp = m;

m = n;

n = temp;

}

while (n != 0)

{

int r = m % n;

m = n;

n = r;

}

return m;

}

int lcm(int m, int n)

{

return m \* n / gcd(m, n);

}

int 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;

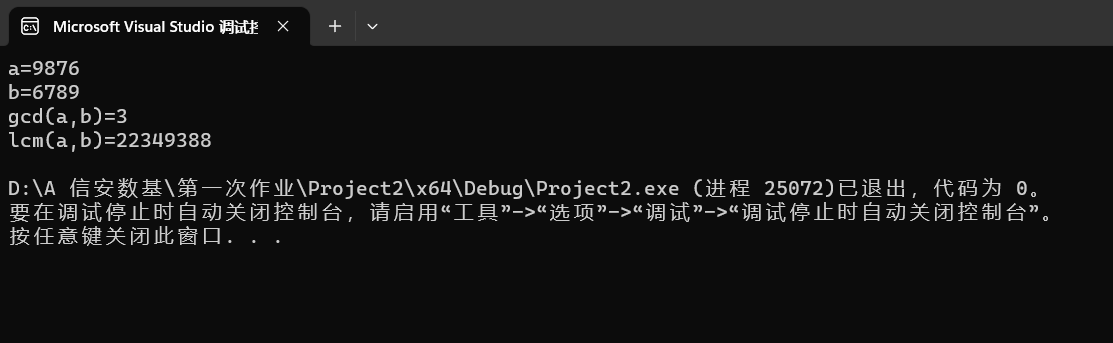
return 0;

}

* **说明部分：**

辗转相除法求两个数的最大公因数；最小公倍数通过两个数的乘积除以两个数的最大公因数得到。

* **运行示例：**



##### **编程练习3——实现算数基本定理**

* **源码部分：**

#include<iostream>

using namespace std;

bool Isprime(int m)//素数判断

{

if (m <= 1)

{

return false;

}

else if (m == 2)

{

return true;

}

else

{

for (int i = 2; i \* i <= m; i++)

{

if (m % i == 0)

{

return false;

}

}

return true;

}

}

int main()

{

int num, t;

int sum = 1;

bool count = true;

cout << "请输入数字: ";

cin >> num;

t = num;

int time[1000] = { 0 };//该数组用来记录1000以内的素数出现的次数

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

{

if (Isprime(i))

{

while (t % i == 0)

{

sum = sum \* i;

if (sum < num)

{

time[i]++;

}

else if (sum == num)

{

time[i]++;

}

t = t / i;

}

}

}

cout << num << "=";

for (int i = 2; i < 1000 ; i++)

{

if (Isprime(i) && time[i] != 0)

{

if (!count)

{

cout << "\*" << i << "^" << time[i];

}

else

{

cout << i << "^" << time[i];

count = false;

}

}

}

return 0;

}

* **说明部分：**

**先判断是不是素数，再计算该素数在输入数字的算数基本定理展开中最多可出现几次。最后按照格式输出。**

* **运行示例：**

