# 第1次编程练习报告

姓名：李潇逸 学号：2111454 班级：信安法班

##### **编程练习1——Eratosthenes筛法**

* **源码部分：**

#include <iostream>

#include <cmath>

using namespace std;

bool su(double a){

int aa;

int b;

bool flag;

aa = int(a);

b = int(sqrt(a)) + 1;

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

if(aa == 2){

flag = true;

break;

}

else if(aa != 2 && aa % i == 0){

flag = false;

break;

}

else{

flag = true;

}

}

return flag;

}

void Eratosthenes(bool \*a,int n){

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

if(su(i)){

for(int k = 2;k \* i <= n;k++){

a[k \* i] = false;

}

}

}

}

int main()

{

int n;

int m;

char k;

int total = 0;

cout<<"请输入范围（格式：m-n）：";

cin>>m;

cin>>k;

cin>>n;

bool \* a = new bool[n-m+1];

for(int i = m;i <= n;i++){

a[i] = true;

}

a[1] = false;

Eratosthenes(a,n);

bool flag = true;

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

if(a[i]){

}

if(a[i] && !flag){

cout<<","<<i;

total++;

}

else if(a[i] && flag){

cout<<i;

flag = false;

total++;

}

}

cout<<endl<<"质数共计"<<total<<"个"<<endl;

}

* **说明部分：**

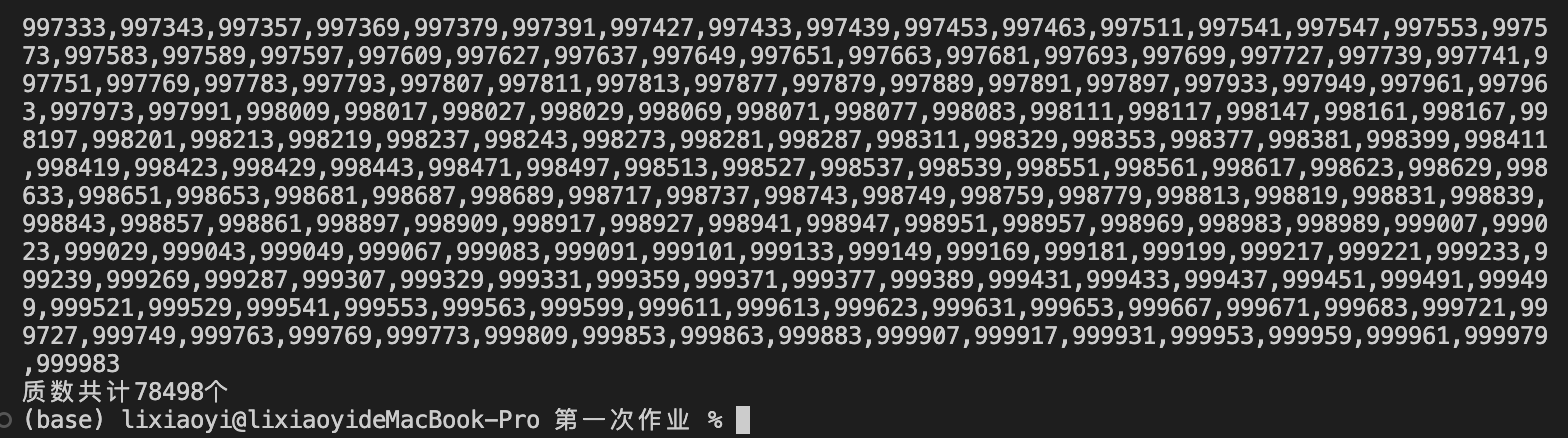
初始化。依据输入的第二个数据n初始化一个n+1个元素的一维bool类型的数组a，并将除小于输入的第一个数据m的元素设置为false，其余的元素设置为true。

构建su函数。该函数的作用为判断传入数据是否为素数。

构建Eratosthenes函数。所需传入的参数有一维数组a，输入的第二个数据n。使用for循环，使之遍历[2，n]。循环内进行判断：若遍历时出现素数i，则将所有小于n的i的倍数多所对应的数组元素设置为false；反之则不进行操作。

书写main函数。输入m,n;初始化数组；带入Eratosthenes函数。之后进行一次循环，计算共有多少个质数（也就是数组中还有多少true）并将元素为true的index输出。

* **运行示例：**//截图



**其他：**

1. 对比筛法与普通算法的性能差异；

普通算法时间复杂度为*O(n^2)*

埃氏筛的时间复杂度为*O(n√n / logn)*

性能显著提升。

1. 递归调用该算法求更大范围素数进行优化；

调用递归

c.求更大的素数该方法是否适用？会引入哪些新的问题？

不再适用。会重复标记一个数不是素数的倍数

##### **编程练习2——算术基本定理**

* **源码部分：**

#include<iostream>

#include<cmath>

using namespace std;

bool su(double a){

int aa;

int b;

bool flag;

aa = int(a);

b = int(sqrt(a)) + 1;

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

if(aa == 2){

flag = true;

break;

}

else if(aa != 2 && aa % i == 0){

flag = false;

break;

}

else{

flag = true;

}

}

return flag;

}

void t(double a,int \*b){

int aa = int(a);

int c = int(sqrt(a))+1;

for(int i = 2;i <= c;i++){

if(su(i)){

if(su(aa)){

b[aa]++;

break;

}

else{

while(aa % i == 0){

aa /= i;

b[i]++;

}

}

}

}

}

int main()

{

double a;

int \*b;

cin>>a;

int aa = int(a);

b = new int[aa+1];

for(int i = 0;i <= aa;i++){

b[i] = 0;

}

t(a,b);

bool flag = false;

for(int i = 2;i <= aa;i++){

if(b[i] != 0 && !flag){

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

flag = true;

}

else if(b[i] != 0 && flag){

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

}

}

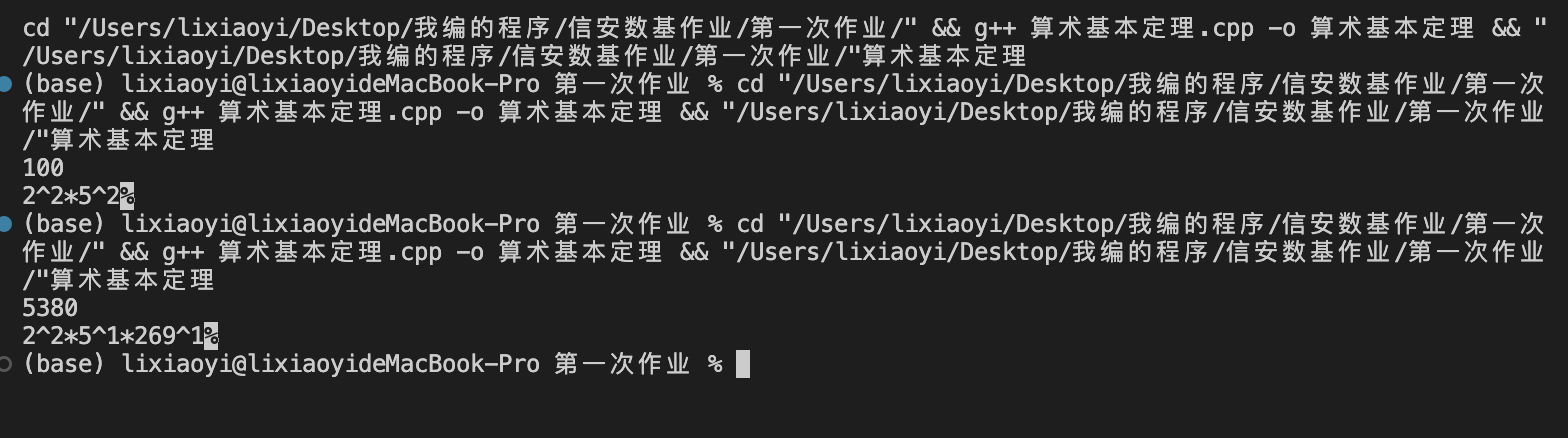
}

* **说明部分：**

构建su函数。该函数的作用为判断传入数据是否为素数。

构建t函数。传入输入的数字a，构建好的数组b（用于储存每个素数有几个）。从2开始直到a进行循环，每个素数与a做mod运算，等于零的让a/=i，并使计数器b[i]++。最后进行输出即可。

* **运行示例：**//截图



* **其他：无**

##### **编程练习3——gcd与lcm**

* **源码部分：**

#include<iostream>

using namespace std;

int gcd(int a,int b){

int k = 0;

for(int i = 1;i <= a;i++){

if(a % i == 0 && b % i == 0){

k = i;

}

}

return k;

}

int lcm(int a,int b,int k){

int p = a / k;

int q = b / k;

return p \* q \* k;

}

int main()

{

int m;

int n;

int p;

cin>>m;

cin>>n;

cout<<"a="<<m<<endl;

cout<<"b="<<n<<endl;

if(n < m){

p = m;

m = n;

n = p;

}

int k = gcd(m,n);

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

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

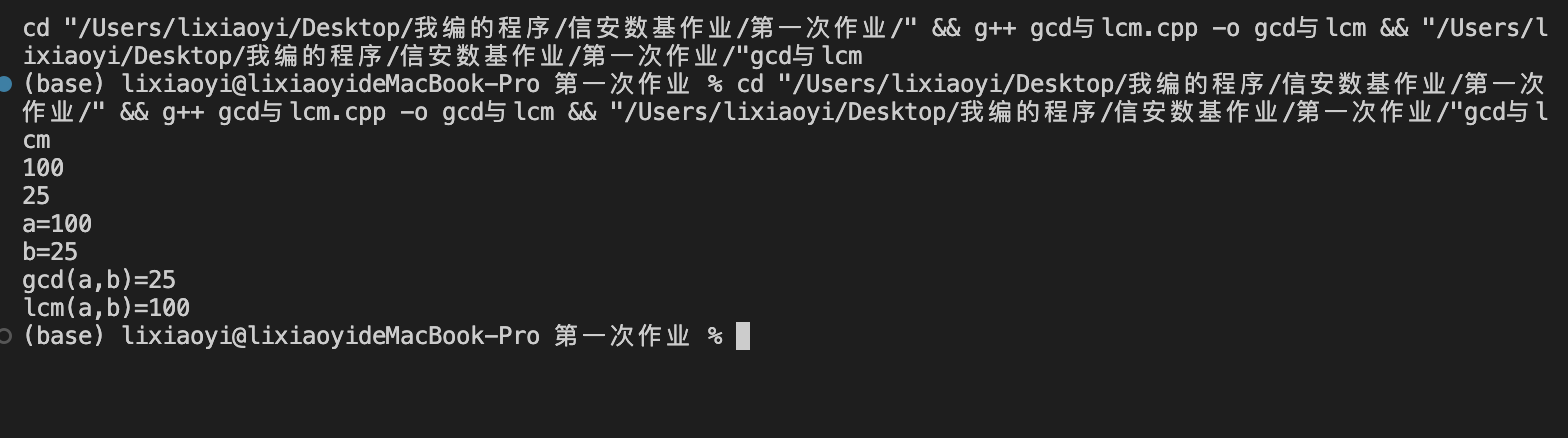
}

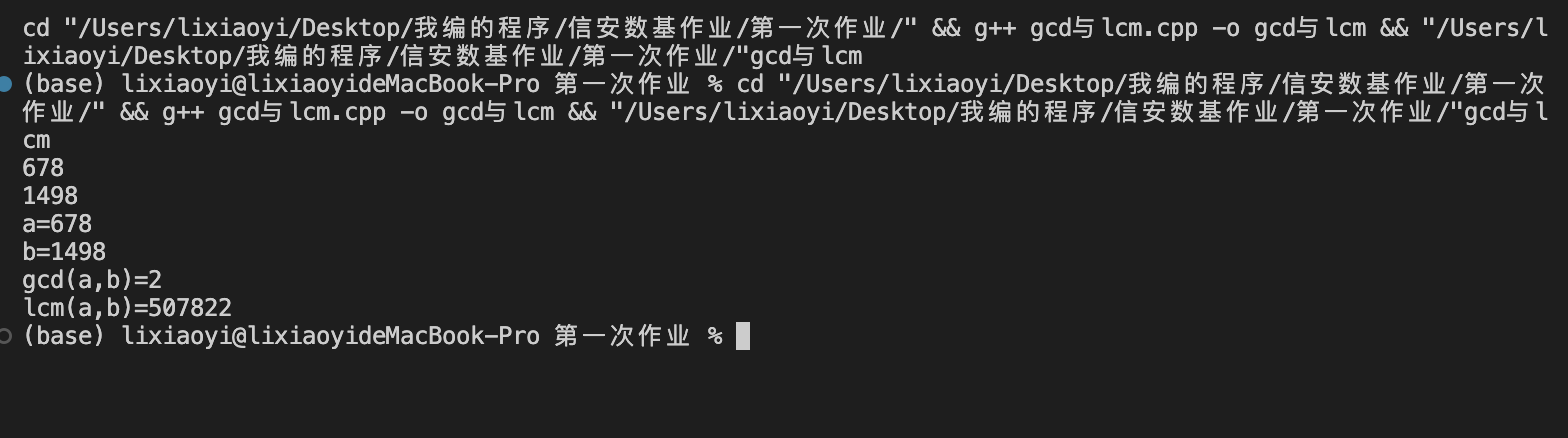
* **说明部分：**

构造gcd函数。从2开始循环直到a（传入函数之前已经进行了排序，a为较小的一个），直到最大的a mod k = 0,b mod k = 0,返回k。

构造lcm函数。只需使（a/gcd(a,b)）\*b即可。

* **运行示例：**//截图





* **其他：无**