**实验八 计算素个数**

**一、实验目的和要求**

输入两个正整数m和n（1<=m, n <=100）,统计并输出m~n之间的素数的个数。要求定义并调用函数isPrime(n)判断n是否是素数。

**二、实验内容**

1. 实验准备
   1. 理论知识介绍
      1. 数据的输入：

要从用户依次读入int类型的变量a,n，可以使用如下语句:

cin >> a>>n;

* + 1. 数据的输出

要将变量的内容显示在显示器上，可以使用cout和流插入运算符。Cout还可以输出表达式的执行结果，比如:

cout << a << ‘+’ << b << ‘=’ << a+b << endl;

* + 1. 算术表达式

C++中，算术运算里有+（加法），-（减法），/（除法），%（取模），但是没有取平方算数运算。

* + 1. 数据类型 （整型）：

C++中一个整型变量可以储存一个整数，可以直接通过cin和cout输入和输出。

* + 1. 数据类型转换

C++中可以隐式地转换数据类型，也可以显式地转换数据类型，比如可以利用下述的方法将浮点数转换为一个整型数(a是一个浮点数)

**(int) a**

* + 1. 关系表达式

关系表达式用于比较两个值的大小。C++提供了6个关系运算符: <, <=, >, >=,

==, !=。关系运算符左结合。

* + 1. 逻辑表达式

关系表达式只能表示简单的情况，当要表示更复杂的情况时，需要用到逻辑表达式。C++定义了3个逻辑运算符，即! (逻辑非)、&& (逻辑与)和|| (逻辑或)。

* + 1. for语句

**for (i = 0; i < n; ++i)**

**需要重复执行的语句**

* + 1. if 语句

**if （条件） 语句**

**if （条件） 语句1 else 语句2**

* + 1. 函数

编写一个实现某个功能的函数称为函数定义。一旦定义了一个函数，在程序中就可以反复调用这个函数。函数的定义形式如下：

**类型名 函数名 (形式参数表)**

**{**

**变量定义部分**

**语句部分**

**}**

1. 实验项目
   1. 分析
      1. 实验要求定义isPrime(n)判断n是否是素数, 所以除了主函数以外还要再定义一个函数isPrime(n)。
      2. 判断一个数是否是素数可以利用for循环来对这个数进行因数的检查，如果不存在因数这个数就是素数
      3. 素数的检查要考虑特殊的情况，比如1不是素数, 2、3是素数。
      4. 通过数学分析可以得出，素数的因数检查的因数可以只检验到这个数的算数平方根为止。
   2. 方案
      1. 利用cin读取m,n的数值
      2. 利用for循环来检查m~n的每一个数
      3. 针对每一个数利用isPrime(n)函数判断是否为素数
      4. 对这个数是1,2,3的情况分类讨论
      5. isPrime(n)函数只要检验到n的算术平方根为止
   3. 源程序

#include<iostream>

using namespace std;

bool isPrime(int n) {

int i;

if (n == 1)

return false;

if (n == 2 || n == 3)

return true;

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

if (n % i == 0) {

return false;

}

}

return true;

}

int main() {

int m, n, i, count;

count = 0;

cout << "Please input m, n: ";

cin >> m >> n;

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

if (isPrime(i)) {

++count;

}

}

cout << "Count = " << count << endl;

return 0;

}

* 1. 测试数据
  2. 1 100
  3. 2 50
  4. 3 20
  5. 1 10
  6. 运行结果

1. Count = 25
2. Count = 15
3. Count = 7
4. Count = 4
   1. 异常现象

输入 1 10 时，输出的结果为2，其他的情况也比实际的素数个数少2.

* 1. 分析

最初的程序没有考虑到被检查的数为2和3情况，而且对因数的考察多余了，只用考察到n的算数平方根就够了。

* 1. 总结

判断素数的要考虑到几个特殊的情况：1,2,3，否则总数会相差几个,而且素数的考察只用遍历到其算数平方根即可，这样可以提高代码的效率。

**三、实验小结**

1. 完成情况

考虑了多种不同 的情况并对其进行分类讨论之后得到正确的结果。

1. 重点

函数的定义和调用。

1. 难点

素数的判断中的特殊情况和因数判断中for语句的遍历范围。

1. 解决方法

分别考虑1,2,3的 情况，因数判断中的 for语句的遍历范围应从2到n的算术平方根。

1. 有待改进之处

分离的if判断语句可以用连续的if else语句代替。

1. 收获、体会

函数给我们重复利用代码带来了极大的便利性，我们可以利用函数来极大地简化我们的代码。