



素数是数学中一个重要的基本概念,我们从小学就开始接触它。素数的定义是,一个大于1的自然数,如果只能被1和它自身整除,就叫作素数。任何一个大于1的自然数都可以分解成几个素数连乘积的形式,而且这种分解是唯一的。可以说,素数是构成整个自然数大厦的砖瓦。

在两千多年前,古希腊数学家欧几里得在《几何原本》这本著名的数学著作中对素数进行了详细的讨论,并巧妙地证明了“素数是无穷多个”的,但没有找到无穷多个素数的分布规律。公元前250年,古希腊数学家厄拉多塞创造了著名的古典筛法来寻找素数。

在探索素数的征途中,费马、欧拉、狄里克雷、高斯、哥德巴赫、陈景润等数学家承前启后、乐此不疲地投入对素数的研究中,各种数学方法和理论被发展,素数定理、哥德巴赫猜想、黎曼假设、陈氏定理等不断地给数学界注入新鲜血液。随着技术进步和数学家不懈地探索,素数的神秘密码也被数学家一点点地破译,但是素数依然有着无穷的奥秘等着我们去发现。

本章将介绍寻找素数的方法和寻找一些有趣的素数,内容如下:

- ◇ 厄拉多塞筛法
- ◇ 哥德巴赫猜想
- ◇ 梅森素数
- ◇ 孪生素数
- ◇ 回文素数
- ◇ 可逆素数

3.1 厄拉多塞筛法



问题描述

在两千多年前的古希腊,数学家厄拉多塞在写一本叫作《算术入门》的书。在写到“数的整除”部分时,他想:怎样才能找到一种最简单的、判断素数的方法呢?左思右想也没个结果,于是他就去郊外散步。他边走边思考,竟然走到了一家磨坊。磨坊的工人们正在忙碌着,有的搬运麦子,有的磨面,有的筛粉。厄拉多塞突然眼前一亮,是否可以用筛选的方法来挑选素数?把合数像筛粉一样筛掉,留下的肯定就是素数了。



厄拉多塞受此启发创造了这样一种与众不同的寻找素数的方法：先将 $2 \sim n$ 的各个自然数放入表中，然后在 2 的上面画一个圆圈，再划去 2 的其他倍数；第一个既未画圈又没有划去的数是 3，将它画圈，再划去 3 的其他倍数；现在既未画圈又没有划去的第一个数是 5，将它画圈，并划去 5 的其他倍数……以此类推，直到所有小于或等于 n 的各数都画了圈或被划去为止。这时，表中画了圈的以及未划去的那些数正好就是小于 n 的素数。这个简单而高效的寻找素数的方法被称作“厄拉多塞筛法”。请使用“厄拉多塞筛法”算法编写程序，找出自然数 1000 以内的所有素数。



编程思路

寻找素数的厄拉多塞筛法易于理解，据此编写程序实现筛选 1000 以内的自然数中的所有素数。该程序由入口程序和厄拉多塞筛法、各数入表、删除合数等模块组成。

该程序的核心是“厄拉多塞筛法”模块。在该模块中，先调用“各数入表”模块把待筛选的自然数放入“素数表”列表中，接着调用“删除合数”模块，把素数表中的合数都删除。如果当前要操作的素数的平方大于要筛选的最大数时，就可以结束筛选过程，因为当前素数后面没有被删除的数都是素数。



程序清单

程序清单见图 3-1 和图 3-2。



图 3-1 “厄拉多塞筛法”程序清单

其中，模块“删除合数”用于删除某个素数的其他倍数，即删除素数表中的部分合数。我们从列表“素数表”中删除某个素数的倍数时，由后往前删除，直至遇到该素数为止。如果是由前往后删除，则列表中的元素会重新排列，从而导致程序不能实现想要的结果。该模块的代码见图 3-2。

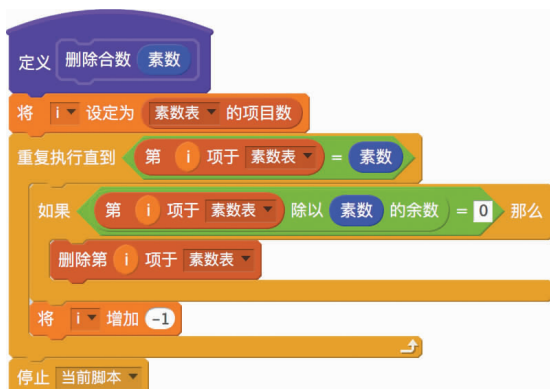


图 3-2 “删除合数”模块

单击绿旗运行程序,瞬间就能找出 2~1000 的素数。



通过修改“厄拉多塞筛法”模块的调用参数,寻找 1000~2000 的素数。

3.2 哥德巴赫猜想



哥德巴赫猜想是指任何大于 2 的偶数都可以写成两个素数之和。例如, $8=3+5$, $12=5+7$, $16=3+13$,……这是德国数学家哥德巴赫在 1742 年提出的一个猜想,它被称为世界近代三大数学难题之一。

哥德巴赫自己无法证明这个猜想,曾写信请教赫赫有名的大数学家欧拉帮忙证明。但是终其一生,欧拉也没能给出严格的证明。哥德巴赫猜想被提出后吸引了全世界数学家和数学爱好者的目光,它被人们称为数学皇冠上一颗可望而不可即的“明珠”。时至今日,哥德巴赫猜想依然没有解决,目前最好的成果(陈氏定理)是 1966 年由我国数学家陈景润取得的。

请编写验证“哥德巴赫猜想”的程序,对“1000 以内大于 2 的正偶数都能分解为两个素数之和”进行验证。



将一个偶数 n 分解为 j 和 $n-j$ 两部分,再判断如果 j 和 $n-j$ 都是素数,那么该偶数就验证通过。该程序的代码见图 3-3。

在该程序中,用到一个名为“是否素数”的模块(见图 3-4),它用于判断一个自然数是否为素数。在本章的其他程序中也用到这个判断素数的模块,将不再单独列出。



程序清单

程序清单见图 3-3 和图 3-4。



图 3-3 “哥德巴赫猜想”程序清单



图 3-4 “是否素数”模块

单击绿旗运行程序,1000 以内通过验证的正偶数被记录到“哥德巴赫猜想”列表中。



试一试

一个正偶数可能会有多种分解方法,该程序中只记录其中一种分解方法。另外,该程序中判断素数的方法不是高效的,在数据量少时尚可使用。如果你对此有兴趣,可以尝试



先建立一个素数表,再通过素数表来判断一个数是否为素数,这样效率更高。

请你试一试,使用上面的程序,继续验证 1000~10000 的正偶数是否符合“哥德巴赫猜想”。

3.3 梅森素数



问题描述

马林·梅森是一位法国科学家,他为科学事业做了很多有益的工作,被选为“100 位在世界科学史上有重要地位的科学家”之一。

由于梅森是最早系统而深入地研究 $2^p - 1$ 型数的人,因而数学界就把这种数称为“梅森数”,并以 M_p 记之(其中 M 为梅森姓名的首字母),即 $M_p = 2^p - 1$ 。如果梅森数为素数,则称之为“梅森素数”(即 $2^p - 1$ 型素数)。

已经证明了,如果 $2^p - 1$ 是素数,则幂指数必须是素数;然而,反过来并不对,当 p 是素数时, $2^p - 1$ 不一定是素数。

是否存在无穷多个梅森素数是数论中未解决的著名难题之一。目前仅发现 49 个梅森素数,最大的是 $2^{74207281} - 1$ (即 2 的 74207281 次方减 1),有 22338618 位数。由于这种素数珍奇而迷人,因此被人们誉为“数海明珠”。自梅森提出其断言后,人们发现的已知最大素数几乎都是梅森素数,因此寻找新的梅森素数的历程也就几乎等同于寻找新的最大素数的历程。请编写程序找出指数 p 在 $[2, 20]$ 中的梅森素数。



编程思路

先以 $M_p = 2^p - 1$ 为模型求出梅森数,再判断该梅森数是否为素数。



程序清单

程序清单见图 3-5。



图 3-5 “梅森素数”程序清单