|  |
| --- |
|  |
| Mersenne Primes  Write Up |
| |  |  |  | | --- | --- | --- | | Boe, Jessica M. | Dr. Carrigan | MAT 178 | |

A Mersenne prime, coined after 17th century French mathematician Marin Mersenne, where the exponent n is also prime. In variable form, this looks like the following, where *Mn* is the prime. Mersenne himself proved the first four numbers in the Mersenne set, *M2 =3, M3 = 7, M5 =31, and M7 =127.*

*Mn* = 2*n*−1.

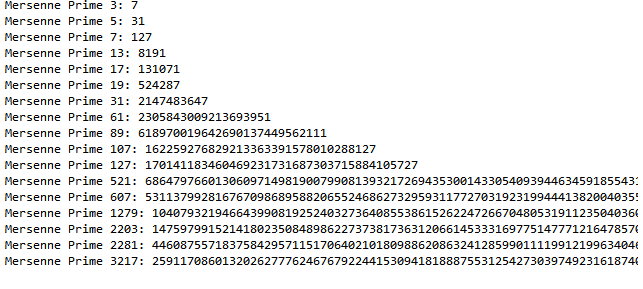
Checking for these factors is a very difficult task as the numbers get very big extremely quickly, and without the computational power that we take for granted today, rendering primality testing a very daunting assignment. Not only that, but there wasn’t an efficient method for testing primality at this point either. The next contribution to Mersenne primes was from Euler himself, who proved that *M31* was prime around the 1770s. Euler showed any prime divisor of the Mersenne number must be congruent to 1, or 62 mod 248. However, the amount of division you’d need to do by hand to check this wasn’t extremely long, so he found the next prime by applying a form of the Mersenne factor theorem.

This was a groundbreaking discovery, but it wasn’t until 100 years later that more progress was contributed by Edouard Lucas, who verified that *M31*, the largest Mersenne prime known before the advent of computers, was really prime (OEIS). In the early 1900s, Derek Henry Lehmer revised Lucas’ method with binary arithmetic. It relies mainly on the fact that if a Mersenne number has any chance of being prime, then the exponent 2 is being raised to has to be prime too. This doesn’t necessarily work on the converse. This primality test is defined below (Great Internet Mersenne Prime Search (GIMPS)):

For p an odd prime, the Mersenne number 2p -1 is prime if and only if 2p -1 divides S(p-1) where S(n+1) = S(n)2-2, and S(1) = 4

By using this simple algorithm in conjunction with a primality test, we can find Mersenne primes.

**Program Output:**



**\*Note:** For the program, I wasn’t sure whether we were supposed to be factoring to the 41st M, which would be a crazy insane number, or stop between M31(2147483647) and M61 (2305843009213693951). I’m hoping it’s the latter, because my computer could ‘only’ get up to Mp 4423, which is:

285542542228279613901563566102164008326164238644702889199247456602284400390600653875954571505539843239754513915896150297878399377056071435169747221107988791198200988477531339214282772016059009904586686254989084815735422480409022344297588352526004383890632616124076317387416881148592486188361873904175783145696016919574390765598280188599035578448591077683677175520434074287726578006266759615970759521327828555662781678385691581844436444812511562428136742490459363212810180276096088111401003377570363545725120924073646921576797146199387619296560302680261790118132925012323046444438622308877924609373773012481681672424493674474488537770155783006880852648161513067144814790288366664062257274665275787127374649231096375001170901890786263324619578795731425693805073056119677580338084333381987500902968831935913095269821311141322393356490178488728982288156282600813831296143663845945431144043753821542871277745606447858564159213328443580206422714694913091762716447041689678070096773590429808909616750452927258000843500344831628297089902728649981994387647234574276263729694848304750917174186181130688518792748622612293341368928056634384466646326572476167275660839105650528975713899320211121495795311427946254553305387067821067601768750977866100460014602138408448021225053689054793742003095722096732954750721718115531871310231057902608580607

I also used Stack Overflow for help with the Lucas Lehmer algorithm, because I wasn’t sure where to start. I put the link on Github.

# References

Great Internet Mersenne Prime Search (GIMPS). "Mathematics and Research Strategy." n.d. *Mersenne.com.*

OEIS. "Lucas-Lehmer Primality Test." n.d. *OEIS.*