How to Check if a Number Is Prime

Prime numbers are numbers that are divisible only by themselves and 1 - other numbers are called *composite* numbers. When it comes to testing whether a given number is prime, numerous options exist. Some of these methods are relatively simple but prove impractical for large numbers. Other tests that are often used for large numbers are actually *probabilistic* algorithms that can sometimes falsely characterize a number as prime or composite. See Step 1 below to start learning how to test a number for primality.

Method 1 of 4: Using Trial Division

Trial division is by far the simplest test for primality. For small numbers, it is usually also the quickest test available. This test is based on the definition of a prime number: a number is prime if it has no factors that divide evenly into it other than itself and one.

1 **Set *n* as the number you want to test.** In the trial division method of primality testing, you divide your given number n by all of its possible integer factors. For large values of n, like n=101, it's extremely impractical to divide by *every* integer below n. Luckily, several tricks exist to whittle down the number of factors you must test.

2 **Determine whether *n* is even.** All even numbers are evenly divisible by 2. Because of this, if n is even, you can automatically say that **n is composite (not prime)**. To quickly determine whether a number is even, pay attention only to its last digit. If its last digit is a 2, 4, 6, 8, or 0, the number is even and thus is not prime.

- The sole exception to this rule is the number 2 itself, which, because it is evenly divisible only by itself and 1, is prime. 2 is the only even prime number.

3 **Divide *n* by each number between 2 and n-1.** Since a prime number has no factors other than itself and 1 and since whole number factors are necessarily smaller than their product, checking all of the numbers less than n and greater than 2 for

even divisibility will determine whether n is prime. We start after 2 because even numbers (which are multiples of 2) are not primes. This is *far from* the most efficient way to test, and, as we'll see below, a number of streamlining strategies exist.

- For example, if we were to use this method to test whether 11 is prime or not, we would divide 11 by 3, 4, 5, 6, 7, 8, 9, and 10, each time looking for a whole number answer with no remainder. Since none of these numbers divide evenly into 11, we can say that 11 is **prime**.

4 **To save time, test only up to sqrt(*n*), rounded up.** Testing a number n by all numbers between 2 and n-1 can quickly become prohibitively time-consuming. For instance, if we wanted to check whether 103 is a prime number in this way, we would have to divide by 3, 4, 5, 6, 7 ... and so on, all the way to 102! Luckily, it's not necessary to test by every single possible factor. It's actually only necessary to test the factors between 2 and the square root of n. If the square root of n isn't a whole number, round up to the nearest whole number and test up to this number instead. See below for an explanation:

- Let's examine the factors of 100. 100 = 1 × 100, 2 × 50, 4 × 25, 5 × 20, 10 × 10, 20 × 5, 25 × 4, 50 × 2, and 100 × 1. Notice that after 10 × 10, the factors are the same as those before 10 × 10, only reversed. In general cases, we can ignore the factors of n greater than the sqrt(n) because they are just rearrangements of factors smaller than the sqrt(n).
- Let's look at an example problem. If n = 37, we don't need to test all of the numbers 3 through 36 to determine whether n is prime. Instead, we can test only the numbers between 2 and sqrt(37), rounded up.
    - sqrt(37) = 6.08 - we'll round up to 7.
    - 37 is not evenly divisible by 3, 4, 5, 6, and 7, so we can say confidently that it is **prime**.

5 **To further save time, use only prime factors.** It's possible to make the trial division process even shorter by eliminating factor choices that aren't prime numbers. By definition, every composite number can be expressed as the product of two or more primes. So, dividing our number n by a composite number is redundant - it's basically the same as dividing it by primes multiple times. Thus, we can further narrow down our list of possible factors to only prime numbers less than sqrt(n).

- This means that all even factors, as well as all factors that are multiples of prime numbers, can be omitted.
- For example, let's try determining whether 103 is prime or not. The square root of 103 rounded up is 11. The prime numbers between 2 and 11 are 3, 5, 7, and 11. 4, 6, 8, and 10 are even and 9 is a multiple of 3, a prime number, so we can omit them. By doing this, we've whittled our list of possible factors down to just four numbers!
    - Neither 3, 5, 7 or 11 divide evenly into 103, so we know that 103 is **prime**.

Method 2 of 4: Using Fermat's Little Theorem

In 1640, the French mathematician Pierre de Fermat first described a theorem (now named for him) that can be of great use when deciding whether a number is prime. Technically, Fermat's test is a test for compositeness, rather than for primeness. This is because the test can determine whether a number is composite with absolute certainty, but can only tell whether a number is *very likely* to be prime.[1] Fermat's Little Theorem is useful in situations where trial division is impractical and when a list of numbers that produce exceptions to the theorem is available.

1 **Let *n* be the number to test for primality.** This primality test is used to help determine whether a given number n is prime. However, as noted above, the Theorem occasionally falsely identifies certain composite numbers as prime. It's important to know this and to be prepared to verify your answer, as we'll learn below.

2 **Pick any integer *a* between 2 and *n*-1 (inclusive).** The precise integer you pick isn't important. Since the parameters for a are inclusive, 2 and n-1 themselves are valid choices.

- As a running example, let's try to determine whether 100 is prime or not. Let's use **3** as our a value - it's between 2 and n-1, so it will work fine.

3 **Compute a^n (mod *n*).** Computing this expression requires some knowledge of a mathematical system called *modular arithmetic*. In modular arithmetic, numbers "wrap around" back to zero upon reaching a certain value, called the *modulus*. Think of this like a clock: an hour after noon, it's 1 o'clock, not 13 o'clock - the time has "wrapped around" back to its starting point. The modulus is specified via the notation (mod *n*). Thus, for this step, calculate a^n with a modulus of n.

- Another way to think of this is to calculate a^n, then divide by n and use the remainder as your answer. Specialized calculators with a modulus function[2] can be extremely useful here, as they can instantly calculate the remainder of division problems involving large numbers.
- If we use such a calculator for our example, we can see that $3^{100}/100$ has a remainder of 1. Thus, 3^{100} (mod 100) is **1**.

4 **If solving by hand, use exponent notation as a shortcut.** If you don't have access to a calculator with modulus functions, use exponent notation to make the process of determining the remainder easier. See below:

- In our example, we would calculate 3^{100} with a modulus of 100. 3^{100} is a very, very large number - 515,377,520,732,011,331,036,461,129,765,621,272,702,107,522,001 - so large, in fact, that it's difficult to work with. Rather than use the 48-digit answer for 3^{100}, let's instead represent it in exponent notation as $(((((((3^2)*3)^2)^2)^2)*3)^2)^2$. Remember that taking the exponent of an exponent has the effect of multiplying the exponents ($(x^y)^z = x^{yz}$).

- Now, let's determine the remainder. Start solving $(((((((3^2)*3)^2)^2)^2)*3)^2)^2$ at the innermost set of parentheses and continue outwards, dividing by 100 after each step. Once we get a remainder, we'll use it for the next step rather than the actual answer. See below:
  - $(((((((9)*3)^2)^2)^2)*3)^2)^2$ - 9/100 has no remainder, so let's continue.
  - $((((((27)^2)^2)^2)*3)^2)^2$ - 27/100 has no remainder, so let's continue.
  - $(((((729)^2)^2)*3)^2)^2$ - 729/100 = 7 R 29. Our remainder is 29. We'll perform the next step on this, rather than on 729.
  - $((((29^2=\mathbf{841})^2)*3)^2)^2$ - 841/100 = 8 R 41. We'll use our remainder 41 again in the next step.
  - $(((41^2 = \mathbf{1681})*3)^2)^2$ - 1681/100 = 16 R 81. We'll use our remainder 81 in the next step.
  - $((81*3 = \mathbf{243})^2)^2$ - 243/100 = 2 R 43. We'll use our remainder 43 in the next step.
  - $(43^2 = \mathbf{1849})^2$ - 1849/100 = 18 R 49. We'll use our remainder 49 in the next step.
  - $49^2 = \mathbf{2401}$ - 2401/100 = 24 R 1. Our final remainder is 1. In other words, 3^{100} (mod 100) = 1. Notice that this is the same answer we got with a calculator in the previous step!

5 **Check whether a^n (mod n) = a (mod n).** If not, n is composite. If true, n is **likely, (but not certainly)** prime. Repeating the test with different values for a can increase your confidence in the outcome, though there are rare composite numbers that satisfy the Fermat condition for *all* values of a. These are called the Carmichael numbers - the smallest of such numbers is 561.

- In our example, 3^{100} (mod 100) = 1 and 3 (mod 100) = 3. $1 \neq 3$, so we can say that 100 is **composite**.

6 **Use Carmichael number resources as insurance.** Knowing which numbers are Carmichael numbers ahead of time can save you the headache of worrying about whether your number is *actually* prime or not. In general, Carmichael numbers are the product of distinct primes where for all primes p dividing n, p-1 also divides n-1.[3] Online lists of Carmichael numbers can be extremely useful when using Fermat's Little Theorem to determine a number's primality.

Method 3 of 4: Using the Miller-Rabin Test

The Miller-Rabin test works similarly to Fermat's Little Theorem but handles

pathological cases like Carmichael numbers better.[4]

1 **Let *n* be an odd number to test for primality.** As in the above methods, n will be our variable for the number whose primality we wish to determine.

2 **Express *n*-1 in the form $2^s \times d$ where *d* is odd.** For n to be prime, it must be odd. So, n - 1 must be even. Because n - 1 is even, it can be represented as some power of 2 times an odd number - $4 = 2^2 \times 1$, $80 = 2^4 \times 5$, and so on. Express n - 1 for your value of n in this way.

- Let's say that we want to know whether n = 321 is prime. 321 - 1 = 320, which we might express as $\mathbf{2^6 \times 5}$.
  - In this case, n = 321 is a convenient number. n - 1 for a more inconvenient number, like n = 371, can require a large value for d, which complicates the process later. 371 - 1 = 370 = $2^1 \times$ **185**

3 **Pick a random number *a* between 2 and *n*-1.** The precise number you pick isn't important - it just has to be less than n and greater than 1.

- In our n = 321 example, let's pick a = **100**.

4 **Compute $a^d \pmod n$.** If $a^d = 1$ or -1 (mod *n*), then *n* passes the Miller-Rabin test and is *probably* prime. Like Fermat's Little Theorem, this test can't pinpoint primes with absolute certainty with only one test.

- In our n = 321 example, $a^d \pmod n = 100^5 \pmod{321}$. $100^5 = 10{,}000{,}000 \pmod{321} =$ **313**. We would use a specialized calculator or the exponent shortcut described earlier to find the remainder of $100^5/321$.
  - Since we didn't get 1 or -1, we can't say that n is probably prime yet. However, there's still more to do - see below.

5 **If your result doesn't equal 1 or -1, compute a^{2d}, a^{4d}, ... and so on to $a^{2^{s-1}d}$.** Calculate a to the power of d times powers of 2 up to 2^{s-1}. If one of these equals 1 or -1 (mod *n*), then *n* passes the Miller-Rabin test and is probably prime. If you find that n does pass the test, check your answer (see the step below). If n doesn't pass any of these tests, it is **composite**.

- As a reminder, in our example, our value for a is 100, our value for s is 6, and our value for d is 5. We would continue to test as below:
  - $100^{2d = 10} = 1 \times 10^{20}$.
    - $1 \times 10^{20} \pmod{321} = 64$. $64 \neq$ 1 or -1. We'll keep going.
  - $100^{4d = 20} = 1 \times 10^{40}$.
    - $1 \times 10^{40} \pmod{321} = 244$. $244 \neq 1$ or -1.
  - At this point, we can stop. s - 1 = 6 - 1 = 5. We've reached $4d = 2^2$ and there are no more powers of 2 times d under 5d. Since none of our calculations gave 1 or -1, we can say confidently that n = 321 is

composite.

6 **If *n* passes the Miller-Rabin test, repeat for alternate values of *a*.** If you find that your value for n may be a prime, try again with another random value for a to improve the confidence in the outcome of the test. If n is in fact prime, it will pass with any value of a. If n is composite, it will fail for at least three quarters of the values of a. This allows greater certainty than with Fermat's Little Theorem, in which certain composite numbers (the Carmichael numbers) can pass for any value of a.

Method 4 of 4: Using the Chinese Remainder Theorem

1 **Choose two numbers.** One of the numbers is not prime and the second number is the number that needs to be tested for primality.

- "Prime1" = 35
- Prime2 = 97

2 **Choose two datapoints that are greater than zero and less than prime1 and prime2 respectfully.** They can't equal each other.

- Data1 = 1
- Data2 = 2

3 **Calculate MMI (Mathematical Multiplicative Inverse) for Prime1 and Prime2**

- Calculate MMI
  - MMI1 = Prime2 ^ -1 Mod Prime1
  - MMI2 = Prime1 ^ -1 Mod Prime2
- For Prime Numbers only (it will give a number for non-prime numbers but it won't be its MMI):
  - MMI1 = (Prime2 ^ (Prime1-2)) % Prime1
  - MMI2 = (Prime1 ^ (Prime2-2)) % Prime2
- e.g
  - MMI1 = (97 ^ 33) % 35
  - MMI2 = (35 ^ 95) % 97

4 **Create a binary table for each MMI up to Log2 of the Modulus**

- For MMI1
  - F(1) = Prime2 % Prime1 = 97 % 35 = 27
  - F(2) = F(1) * F(1) % Prime1 = 27 * 27 % 35 = 29
  - F(4) = F(2) * F(2) % Prime1 = 29 * 29 % 35 = 1
  - F(8) = F(4) * F(4) % Prime1 = 1 * 1 % 35 = 1
  - F(16) =F(8) * F(8) % Prime1 = 1 * 1 % 35 = 1
  - F(32) =F(16) * F(16) % Prime1 = 1 * 1 % 35 = 1
- Calculate the binary of Prime1 - 2
  - 35 -2 = 33 (10001) base 2
  - MMI1 = F(33) = F(32) * F(1) mod 35
  - MMI1 = F(33) = 1 * 27 Mod 35
  - MMI1 = 27
- For MMI2
  - F(1) = Prime1 % Prime2 = 35 % 97 = 35
  - F(2) = F(1) * F(1) % Prime2 = 35 * 35 mod 97 = 61
  - F(4) = F(2) * F(2) % Prime2 = 61 * 61 mod 97 = 35
  - F(8) = F(4) * F(4) % Prime2 = 35 * 35 mod 97 = 61
  - F(16) = F(8) * F(8) % Prime2 = 61 * 61 mod 97 = 35
  - F(32) = F(16) * F(16) % Prime2 = 35 * 35 mod 97 = 61
  - F(64) = F(32) * F(32) % Prime2 = 61 * 61 mod 97 = 35
  - F(128) = F(64) * F(64) % Prime2 = 35 * 35 mod 97 = 61
- Calculate the binary of Prime2 - 2
  - 97 - 2 = 95 = (1011111) base 2
  - MMI2 = (((((F(64) * F(16) % 97) * F(8) % 97) * F(4) % 97) * F(2) % 97) * F(1) % 97)
  - MMI2 = (((((35 * 35) %97) * 61) % 97) * 35 % 97) * 61 % 97) * 35 % 97)
  - MMI2 = 61

5 **Calculate (Data1 * Prime2 * MMI1 + Data2 * Prime1 * MMI2) % (Prime1 * Prime2)**

- Answer = (1 * 97 * 27 + 2 * 35 * 61) % (97 * 35)
- Answer = (2619 + 4270) % 3395
- Answer = 99

6 **Verify that "Prime1" is not Prime**

- Calculate (Answer - Data1) % Prime1
- 99 -1 % 35 = 28
- Since 28 is greater than 0, 35 is not prime

7 **Check if Prime2 is Prime**

- Calculate (Answer - Data2) % Prime2
- 99 - 2 % 97 = 0

- Since 0 equals 0, 97 is potentially prime

8 **Repeat steps 1 through 7 at least two more times.**

- If step 7 is 0:
  - Use a different "prime1" where prime1 is a non-prime
  - Use a different prime 1 where prime 1 is an actual prime. In this case, steps 6 and 7 should equal 0.
  - Use different data points for data1 and data2.
- If step 7 is 0 every time, there is an extremely high probability that prime2 is prime.
- Steps 1 though 7 are known to fail in certain cases when the first number is a non-prime number and the second prime is a factor of the non-prime number "prime1". It works in all scenarios where both numbers are prime.
- The reason why steps 1 though 7 are repeated is because there are a few scenarios where, even if prime1 is not prime and prime2 is not prime, step 7 still works out to be zero, for one or both the numbers. These circumstances are rare. By changing prime1 to a different non-prime number, if prime2 is not prime, prime2 will rapidly not equal zero in step 7. Except for the instance where "prime1" is a factor of prime2, prime numbers will always equal zero in step 7.

Tips

- The 168 prime numbers under 1000 are: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509,

521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601, 607, 613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683, 691, 701, 709, 719, 727, 733, 739, 743, 751, 757, 761, 769, 773, 787, 797, 809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863, 877, 881, 883, 887, 907, 911, 919, 929, 937, 941, 947, 953, 967, 971, 977, 983, 991, 997 [5]

- While trial division is slower than more sophisticated methods for large numbers, it is still quite efficient for small numbers. Even for primality testing of large numbers, it is not uncommon to first check for small factors before switching to a more advanced method when no small factors are found.

Things You'll Need

- ☐ Working out tools, such as paper and pen or a computer

Sources and Citations

1. ↑ Mathworld, Fermat's Little Theorem
2. ↑ http://www.javascripter.net/math/calculators/100digitbigintcalculator.htm
3. ↑ http://mathworld.wolfram.com/CarmichaelNumber.html
4. ↑ Mathworld, Rabin-Miller Strong Pseudoprime Test
5. ↑ Online Encyclopedia of Integer Sequences, A000040

- Topcoder.com - sample source code and documentation for methods discussed here
- Online Prime Number Checker - check numbers with up to 5000 digits

Article Info

Categories: Featured Articles | Mathematics

Featured Article

In other languages:

Español: saber si un número es primo, Italiano: Riconoscere un Numero Primo, Português: Determinar se um Número é Primo, Русский: проверить, является ли число простым, Français: tester la primalité d'un nombre, Deutsch: Überprüfen ob eine Zahl eine Primzahl ist, Nederlands: Controleren of een getal een priemgetal is