***Check for Prime***

**What are prime numbers?**

* A prime number is a natural number greater than **1**, which is only divisible by 1 and itself. First few prime numbers are: 2 3 5 7 11 13 17 19 23…..



*Prime numbers*

* In other words, the prime number is a positive integer greater than 1 that has exactly two factors, 1 and the number itself.
* There are many prime numbers, such as 2, 3, 5, 7, 11, 13, etc.
* Keep in mind that 1 cannot be either prime or composite.
* The remaining numbers, except for 1, are classified as prime and composite numbers.

**Some interesting facts about Prime numbers:**

* Except for 2, which is the smallest prime number and the only even prime number, all prime numbers are odd numbers.
* Every prime number can be represented in form of **6n + 1** or **6n – 1** except the prime numbers 2 and 3, where n is a natural number.
* 2 and 3 are only two consecutive natural numbers that are prime.
* [Goldbach Conjecture:](https://en.wikipedia.org/wiki/Goldbach%27s_conjecture)Every even integer greater than 2 can be expressed as the sum of two primes.
* [Wilson Theorem](https://www.geeksforgeeks.org/wilsons-theorem/): Wilson’s theorem states that a natural number p > 1 is a prime number if and only if

(p - 1) ! ≡ -1 mod p

OR (p - 1) ! ≡ (p-1) mod p

* [Fermat’s Little Theorem](https://en.wikipedia.org/wiki/Fermat's_little_theorem): If n is a prime number, then for every a, 1 <= a < n,

an-1 ≡ 1 (mod n)

OR

an-1 % n = 1

* [Prime Number Theorem](https://en.wikipedia.org/wiki/Prime_number_theorem): The probability that a given, randomly chosen number n is prime is inversely proportional to its number of digits, or to the logarithm of n.
* [Lemoine’s Conjecture](https://www.geeksforgeeks.org/lemoines-conjecture/): Any odd integer greater than 5 can be expressed as a sum of an odd prime (all primes other than 2 are odd) and an even semiprime. A semiprime number is a product of two prime numbers. This is called Lemoine’s conjecture.

**Properties of prime numbers:**

* Every number greater than 1 can be divided by at least one prime number.
* Every even positive integer greater than 2 can be expressed as the sum of two primes.
* Except 2, all other prime numbers are odd. In other words, we can say that 2 is the only even prime number.
* Two prime numbers are always coprime to each other.
* Each composite number can be factored into prime factors and individually all of these are unique in nature.

**Prime numbers and co-prime numbers:**

It is important to distinguish between prime numbers and co-prime numbers. Listed below are the differences between prime and co-prime numbers.

* A coprime number is always considered as a pair, whereas a prime number is considered as a single number.
* Co-prime numbers are numbers that have no common factor except 1. In contrast, prime numbers do not have such a condition.
* A co-prime number can be either prime or composite, but its greatest common factor (GCF) must always be 1. Unlike composite numbers, prime numbers have only two factors, 1 and the number itself.
* **Example of co-prime:**13and 15 are co-primes. The factors of 13 are 1 and 13 and the factors of 15 are 1, 3 and 5. We can see that they have only 1 as their common factor, therefore, they are coprime numbers.
* **Example of prime:**A few examples of prime numbers are 2, 3, 5, 7 and 11 etc.

**How do we check whether a number is Prime or not?**

**Naive Approach:**A naive solution is to iterate through all numbers from 2 to sqrt(n) and for every number check if it divides n. If we find any number that divides, we return false.

Below is the implementation:

C++Java

// A school method based Java program to

// check if a number is prime

import java.lang.\*;

import java.util.\*;

class GFG {

// Check for number prime or not

static boolean isPrime(int n)

{

// Check if number is less than

// equal to 1

if (n <= 1)

return false;

// Check if number is 2

else if (n == 2)

return true;

// Check if n is a multiple of 2

else if (n % 2 == 0)

return false;

// If not, then just check the odds

for (int i = 3; i <= Math.sqrt(n); i += 2) {

if (n % i == 0)

return false;

}

return true;

}

// Driver code

public static void main(String[] args)

{

if (isPrime(19))

System.out.println("true");

else

System.out.println("false");

}

}

**Output**

**true**

**Time Complexity:** O(sqrt(n))  
**Auxiliary space:** O(1)

**Efficient approach:**To check whether  the number is prime or not follow the below idea:

*In the previous approach given if the size of the given number is too large then its square root will be also very large, so to deal with large size input we will deal with a few numbers such as 1, 2, 3, and the numbers which are divisible by 2 and 3 in separate cases and for remaining numbers, we will iterate our loop from 5 to sqrt(n) and check for each iteration whether that  (iteration) or (that iteration + 2) divides n or not. If we find any number that divides, we return false.*

Below is the implementation for the above idea:

C++Java

// Java program to check whether a number

import java.lang.\*;

import java.util.\*;

class GFG {

// Function check whether a number

// is prime or not

public static boolean isPrime(int n)

{

if (n <= 1)

return false;

// Check if n=2 or n=3

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

return true;

// Check whether n is divisible by 2 or 3

if (n % 2 == 0 || n % 3 == 0)

return false;

// Check from 5 to square root of n

// Iterate i by (i+6)

for (int i = 5; i <= Math.sqrt(n); i = i + 6)

if (n % i == 0 || n % (i + 2) == 0)

return false;

return true;

}

// Driver Code

public static void main(String[] args)

{

if (isPrime(11)) {

System.out.println("true");

}

else {

System.out.println("false");

}

}

}

**Output :**

true

**Time complexity:**O(sqrt(n))  
**Auxiliary space:** O(1)