Fermat’s little theorem

[Fermat’s little theorem](https://en.wikipedia.org/wiki/Fermat%27s_little_theorem) states that if p is a prime number, then for any integer a, the number a p – a is an integer multiple of p.

*Here p is a prime number****ap ≡ a (mod p).***

**Special Case:** If a is not divisible by p, Fermat’s little theorem is equivalent to the statement that a p-1-1 is an integer multiple of p.

*ap-1 ≡ 1 (mod p)  
OR  
ap-1% p = 1  
Here a is not divisible by p.*

**Take an Example How Fermat’s little theorem works**  
Examples:

P = an integer Prime number

a = an integer which is not multiple of P

Let a = 2 and P = 17

According to Fermat's little theorem

2 17 - 1  ≡ 1 mod(17)

we got 65536 % 17 ≡ 1

that mean (65536-1) is an multiple of 17

**Use of Fermat’s little theorem**

If we know m is prime, then we can also use Fermats’s little theorem to find the inverse.

am-1 ≡ 1 (mod m)  
If we multiply both sides with a-1, we get

a-1 ≡ a m-2 (mod m)  
Below is the Implementation of above

* C++
* Java
* Python3
* C#
* PHP

filter\_none

edit

play\_arrow

brightness\_5

|  |
| --- |
| // C++ program to find modular inverse of a  // under modulo m using Fermat's little theorem.  // This program works only if m is prime.  #include <bits/stdc++.h>  using namespace std;    // To compute x raised to power y under modulo m  int power(int x, unsigned int y, unsigned int m);    // Function to find modular inverse of a under modulo m  // Assumption: m is prime  void modInverse(int a, int m)  {      if (\_\_gcd(a, m) != 1)          cout << "Inverse doesn't exist";        else {            // If a and m are relatively prime, then          // modulo inverse is a^(m-2) mode m          cout << "Modular multiplicative inverse is "               << power(a, m - 2, m);      }  }    // To compute x^y under modulo m  int power(int x, unsigned int y, unsigned int m)  {      if (y == 0)          return 1;      int p = power(x, y / 2, m) % m;      p = (p \* p) % m;        return (y % 2 == 0) ? p : (x \* p) % m;  }    // Driver Program  int main()  {      int a = 3, m = 11;      modInverse(a, m);      return 0;  } |

Output :

Modular multiplicative inverse is 4