**# Properties of Modular Arithmetic**

1. **Addition**:-

(a+b)%m = (a%m + b%m)%m

1. **Subtraction**:-

(a-b)%m = (a%m - b%m + m)%m

1. **Multiplication**:-

(a.b)%m = ((a%m).(b%m))%m

1. **Exponentiation**:-

If (a%m) = b, then (%m)= (%m), for any positive integer k.

1. **Division**:-

(a/b)%m = (a\*(mod inverse b))%m = ((a%m)\*((mod inverse b)%m))%m

**Modular multiplicative inverse:**

The modular multiplicative inverse of (a%m) is an integer ‘x’ such that: (a.x)%m = 1 . ( The inverse exists iff a and m are relatively prime )

The idea is to use [Extended Euclidean algorithms](https://www.geeksforgeeks.org/basic-and-extended-euclidean-algorithms/) that takes two integers ‘a’ and ‘b’, finds their gcd and also find ‘x’ and ‘y’ such that ax + by = gcd(a,b) .

Here b = m, and also gcd(a,m) = 1, as a and m are co prime.

ax + my = 1

Taking modulo on both side

1. x)%m = 1

**How does Extended Euclidean Algorithm Work?**



On Comparing the coefficient of a and b in equation (1) and (2)



X and y can be find out by recursion, for this base case is when a==0

For a==0, b%a == 0 if b%a==0 then gcd will be b.

We have to return b and value of x = 0 and y =1

Code to find x, y and gcd by using Extended Euclidean Algorithm:

<https://ide.codingblocks.com/s/123169>

Code for Modular Multiplicative Inverse:

<https://ide.codingblocks.com/s/123187>

Code for Modular Arithmetic (Divison):

<https://ide.codingblocks.com/s/123193>