# Number Theory - 5

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

(a/b)%m = ((a%m)\*((b^-1)%m))%m;

Today’s goal is to calculate **(b^-1)%m.**

**-> Modulo inverse of b.**

**Modulo Multiplicative Inverse.**

(a\*x)%n = 1

Then it is known as the modulo inverse of a w.r.t n = x = (a^-1).

And 1<=x<=(n-1)

And it will exist only when **gcd(a,n)=1;**

When ever you found gcd(a,n)=1 then you may write it like

a\*x+n\*y=1;

Here

x-> modulo inverse of a w.r.t n = (a^-1)%n;

Take modulo w.r.t n on LHS and RHS then

(a\*x)%n = 1

y-> modulo inverse of n w.r.t a;

Take modulo w.r.t a on LHS and RHS then

(n\*y)%a = 1

**Important Property**

**(a^ETF(p))%p = 1 (always)**

**Where a and p are coprime i.e gcd(a,p)=1;**

**If p is a prime number then ETF(p) = p-1;**

**(a^(p-1))%p = 1 (**Fermat's Little Theorem)

(a\*(a^(p-2))\*)%p = 1;

-> (a\*x)%p=1 where x=a^-1

So x = a^(p-2)

**[(a^(-1))%p = (a^(p-2))%p][**

**-> Final outcome->**

**modulo inverse of a w.r.t m((a^-1)%m) is equal to ((a^(m-2))%m; [m is prime number].**

**x=(a^-1)%m;** 1<=x<=(m-1)

**Last property-> (a^z)%m = (a^(z%ETF(m)))%m;**

Note-> In 99% of the cases you would find m as a prime number.

Find modulo with m = 10e9+7; (prime number);

**Q.1) Find (nCr%m) = fac[n]/(fac[r]\*fac[n-r]);**

**Where m = 1e9+7(prime number);**

const int max=1e6;

vector<long> fac(max+1);

fac[0]=1;

for(int i=1;i<=max;i++) fac[i]=fac[i-1]\*i%m;

Using above knowledge find (nCr%m);

nCr = fac[n]/(fac[r]\*fac[n-r]);

fac[n]\*(fac[r]^-1)\*(fac[n-r]^-1)%m;

fac[n]\*(fac[r]^(m-2))\*(fac[n-r]^(m-2))%m;

m=> prime number

**-> Function to calculate the modulo inverse**

long ModuloInverse(long a,long m){

//(a^-1)%m;

//(a^(m-2))%m;

//using binary exponentiation calculate value of (a^(m-2))%m and return ans;

Long ans = (a^(m-2))%m;

Return ans;

}

**Q.2)** <https://codeforces.com/problemset/problem/300/C>

a=2,b=3,n=10;

2222222333 -> 23

Total n digits -> i of them are a

-> n-i digits would be b

Sum = a\*i+(n-i)\*b

Total numbers that can be formed using i a’s and (n-i) b’s = nCi

Sol:-

#include<bits/stdc++.h>

#define int long long

using namespace std;

int fac[1000001];

int rem = 1e9+7;

void pre()

{

fac[0]=1;

for(int i=1;i<=1e6;i++){

fac[i] = fac[i-1]\*i; // (a\*b)%m = ((a%m)\*(b%m))%m

fac[i]%=rem;

}

}

int binExp(int x,int n)

{

int res=1;

while(n){

if(n%2==1){

res\*=x;

res%=rem;

}

n/=2;

x\*=x;

x%=rem;

}

return res;

}

int ncr(int n,int r)

{

int temp1 = fac[n];

int temp2 = fac[n-r]\*fac[r];

temp2%=rem;

int temp3 = binExp(temp2,rem-2); // temp3 is the inverse

temp1\*=temp3;

temp1%=rem;

return temp1;

}

bool check(int sum,int a,int b) // return 1 if sum is a good number else it returns 0;

{

for(int i=sum;i>0;i/=10){ //645 -> 64 -> 6-> 0 and 6 and 4 and 5

int r = i%10;

if(r!=a&&r!=b){

return 0; // number is not good

}

}

return 1; // number is good

}

int32\_t main()

{

int a,b,n;

cin>>a>>b>>n;

pre();

int ans=0;

for(int i=0;i<=n;i++){

int sum = a\*i+(n-i)\*b;

if(check(sum,a,b)==1){

//add nci to ans

ans+= ncr(n,i);

ans%=rem;

}

}

cout<<ans;

}

N -> (n-3) + (n-4)+(n-5)+...3

N-1 -> (n-4)+(n-5)+...3

n -> (n-1)+(n-3)