Class 1:

Euter totient function: 1-n পর্যন্ত কো প্রাইম বের করে।

```
317^1 9817^1
17^2 89^1 509^1
2^18 3^8 5^4 7^2 11^1 13^1 17^1 19^1
328439^1 234884407^1
```

This is the prime factorization

Pollar p-1 method: 20 digit পর্যন্ত প্রাইম প্রেক্টোরাইজেশন করতে পারে।

Pollard Rho Algorithm :২৯ টা ডিজিট করতে পারে ।

12 has pairs (1 , 12) (2 , 6) , (3 , 4)

```
bool isPrime(int n)
{
        if(n == 1) return false;

        for(int i=2;i*i<=n;i++)
        {
                if(n % i == 0)
                return false;
        }
        return true;
}
```

Naive Approach

```
bool isPrime(int n)
{
        if(n == 1)
                return false;

        for(int i=2;i<n;i++)
        {
                if(n % i == 0)
                        return false;
        }
        return true;
}
```

Class 2:

S eive 10^6 কম হতে হবে।

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

Why we need Sieve?

Preprocessing TIme : O(Nlog(logN))
Answers Query : O(1)
Extra Space : O(N)

```
int is_prime[1000001];

void sieve()
{
    int maxN = 1000000;

    for(int i=1; i<=maxN; i++) is_prime[i] = 1;
    is_prime[0] = is_prime[1] = 0;

    for(int i=2; i*i<=maxN; i++)
    {
        if(is_prime[i])
        {
            for(int j=i*i; j<=maxN; j+=i)
                is_prime[j] = 0;
        }
    }
}
```

Int j=i*i কেন দিয়েছি , instead of i.

কারণ i*i er nicher gula i er nicher gula check hoye geche কালার করা।

Sieve: nlog(logn)

L02.1:

Apply sieve till 87000008
and instead of bool array use bitset

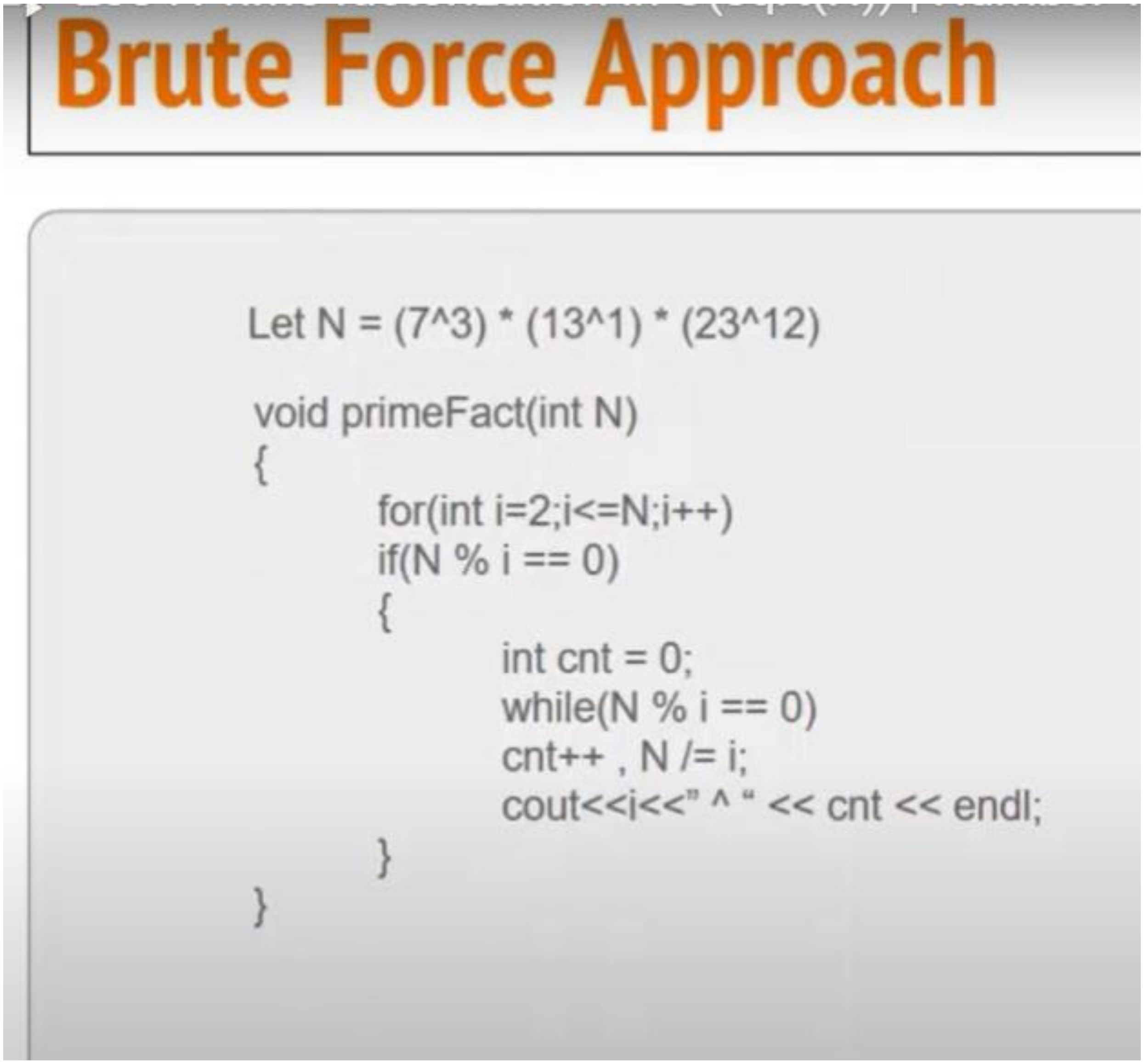
Bool use kore 1 byte ,

On the other hand int use 4 byte

- `bool`: Represents true/false values, occupies a small amount of memory (usually 1 byte).
- `int`: Represents integers within a certain range, occupies 4 or 8 bytes depending on the architecture.
- `long long`: Represents larger integers with a wider range than `int`, occupies 8 bytes.

L013:

Prime factorization :



Claim: if N is a composite number then there are at least 1 prime divisor of N below sqrt(N)

```
Let N = (7^3) * (13^1) * (23^2)

void primeFact(int N)
{
        for(int i=2;i * i<=N;i++)
        if(N % i == 0)
        {
                int cnt = 0;
                while(N % i == 0)
                cnt++ , N /= i;
                cout<<i<<" ^ " << cnt << endl;
        }
        if(N > 1)
        cout<<N<<" ^ "<<1<<endl;
}
```

Class L04: Binary exponentiation

Binary exponential to calculate a^n to O(logn) time complexity

Power যদি ওড হয় এক কমিয়ে দিয়ে res and base multiply korbo . যদি এভেন হয় power ke 2 dara vag korbo and base ke sqare korbo zoto kkon power 0 hbe na totokkonn colte thakbe

Class L05: prime factorization buji nai pore dekho nio . onno youtube channel e jeta log(n) time complexity .

Class L06: matrix exponentiaion

xor er জন্য আইডেন্টিটি এলিমেন্ট হলো 0 ।

2D matix function er modde pathaite hole obossoi obossi column nam bole dite hoy . arr[][n]

Neive approach M^3* N complexity

Batter approach M^3 log(n) time

M hocche dymention

Class L08: Euclid Algorithm

$$N1 = mod * q1 + r1$$
$$N2 = mod * q2 + r2$$

$$(N1 + N2) \% mod = (mod * q1 + r1 + mod * q2 + r2) \% mod$$
$$= (0 + r1 + 0 + r2) \% mod$$
$$= (r1 + r2) \% mod$$
$$= ((N1 \% mod) + (N2 \% mod)) \% mod$$

GCD means highest common factor . (HCF)

Euclid algorithm can be used to calculate GCD (HCF) of 2 numbers say A and B

$$\gcd(a, b) = \begin{cases} a, & \text{if } b = 0 \\ \gcd(b, a \bmod b), & \text{otherwise.} \end{cases}$$

```
int gcd (int a, int b) {
    if (b == 0)
        return a;
    else
        return gcd (b, a % b);
}
```

B always less than a

Making some observation :

1. GCD(A,0)=GCD(0,A)
2. GCD(A,B)=GCD(B,A)
3. GCD(A-B,B)=GCD(A,B-A)

4. Observation 3 : GCD(A , B) = GCD(A-B ,B) = GCD(A , B-A)

   $A = g * X$ $\qquad$ $B = g * Y$

   $A - B = g(X - Y)$
   $B - A = g(Y - X)$

g hocce highest gcd ekhn , it is clearly shown that a-b er kintu g dara divisible also B-A so. It will be work.

GCD(44 , 12) = GCD(44 - 12 , 12) = GCD(32 , 12)

GCD(32 , 12) = GCD(20 , 12)

GCD(20 , 12) = GCD(8 , 12)

GCD(8 , 12) = GCD(12 , 8)

GCD(12 , 8) = GCD(4 , 8)

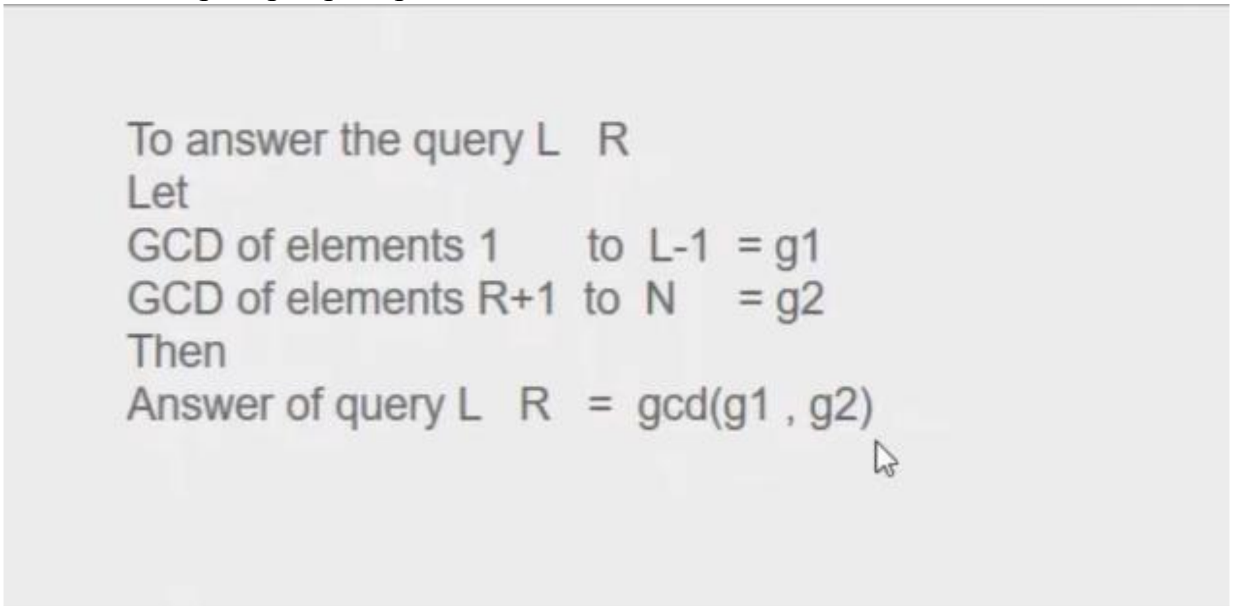
GCD(4 , 8) = GCD(8 , 4)

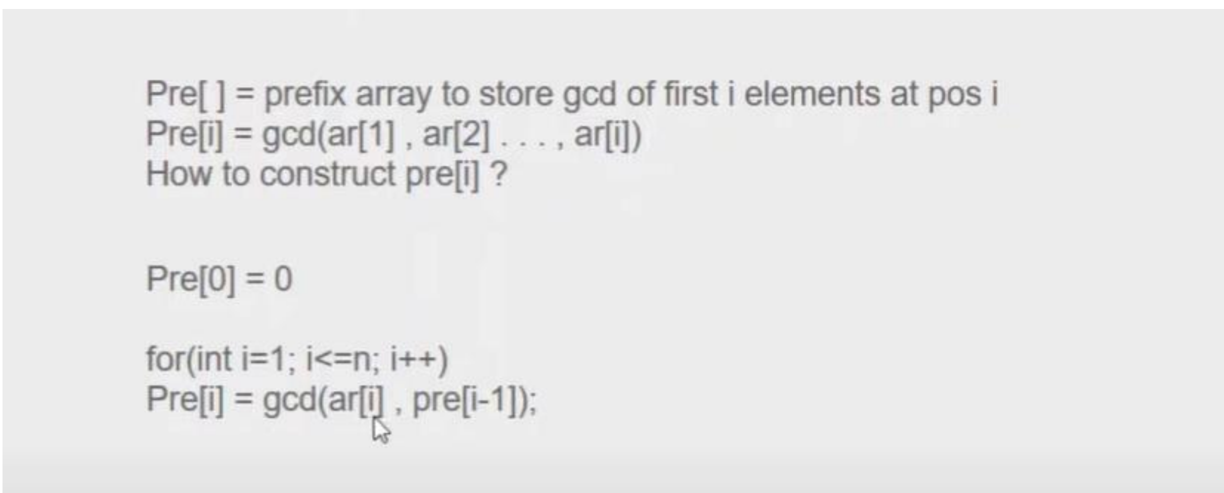
GCD(8 , 4) = GCD(4 , 4)

GCD(4 , 4) = GCD(0 , 4)

Dekhon subtract na kore tumi kintu ekbar I reminder i kintu likte partam . we can directly calculate reminder instead of subtract .
The complexity of
Log(max(a,b))

Class L08 1: query base gcd ber korte hbe exlude L,R
Firstly gcd1=1---(L-1)
Second gcd2=(R+1,N)
Then calculate gcd=gcd(gcd1,gcd2)





Suff[ ] = suffix array to store gcd of elements from i to N
Suff[i] = gcd(ar[i] , ar[i+1] , ar[i+2] . . . , ar[N])
How to construct Suff[i] ?

Suff[N+1] = 0

```
for(int i=N; i>=1; i--)
Suff[i] = gcd(ar[i] , Suff[i+1]);
```

L09 class :

A and B ke congruent বলা হয় I if এদের সেম রিমাইন্ডার থাকে যেটা N dara mod korbo .

Understanding Modular Congruences

a and b are said to be congruent to each other under modulo N , if they leave same remainder when divided by N

$a \equiv b \pmod{N}$

$13 \equiv 41 \pmod{7}$

13 mod 7 = 6
41 mod 7 = 6

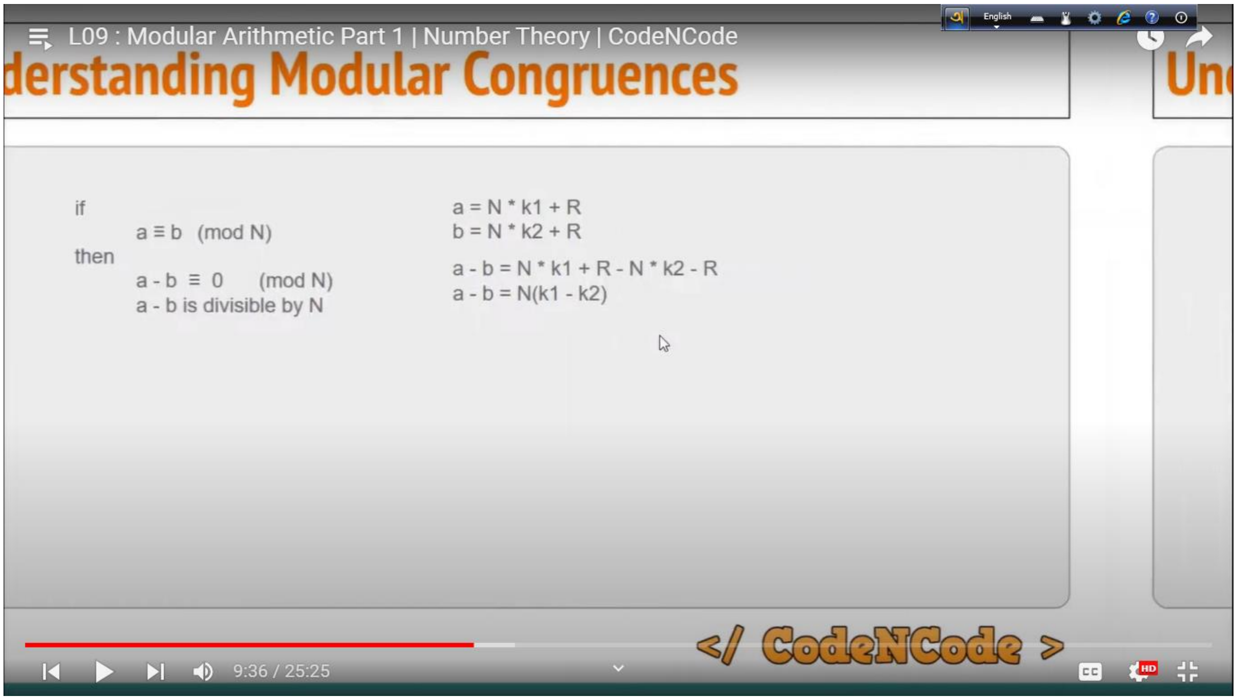
Eta khub helpful karon tumi ektar viporite arek ta likte paro .

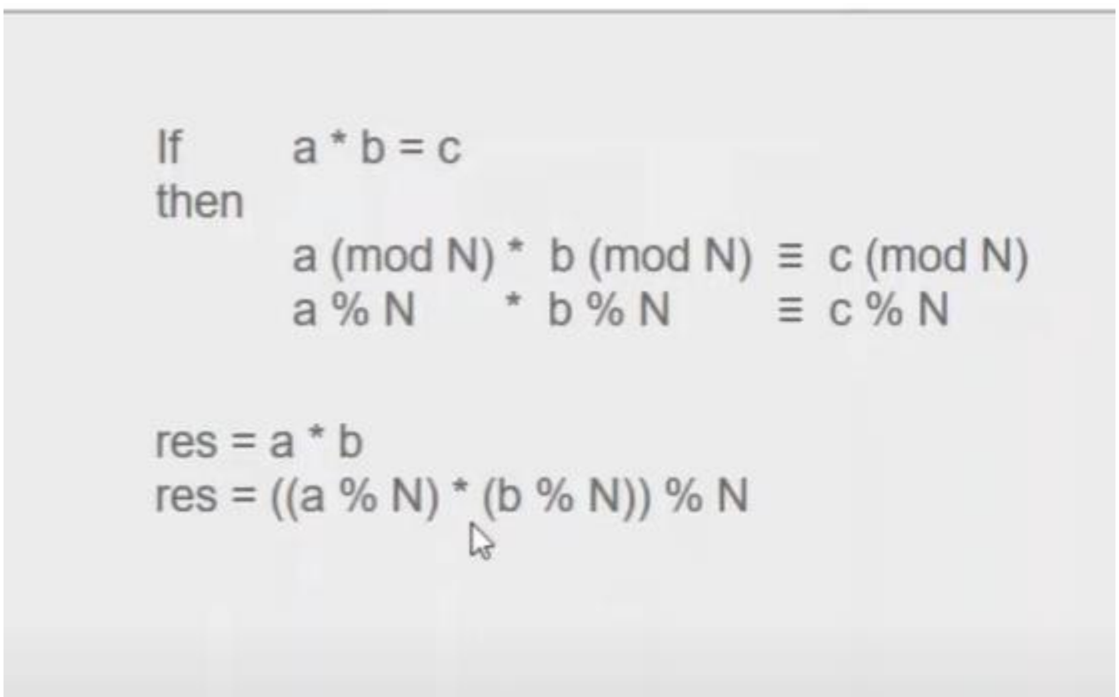
$13 \equiv 41 \pmod{7}$

(13 + 35 + 5) % 7 = (53) % 7 = 4

(41 + 35 + 5) % 7 = (81) % 7 = 4

Eta multiplication er jonno khate





Overflow er jonno use korbo.

Find last digit of 2573 * 34268 ?

To find last digit

(2573 * 34268) % 10

(3 * 8) % 10

(24) % 10 = 4

Modular Arithmetic part 2:

একটা নাম্বার ৯ অথবা ৩ দ্বারা ডিভিজিবিলিটি কি না কীভাবে চেক করব?
সেটা হচ্ছে সব গুলো যোগ করে ৩ অথবা ৯ দ্বারা যোগ করে ভাগ দিলেই হবে। কিন্তু এটা কেন???

১২৩৪৫%৯

(1*10^4+2*10^3+3*10^2+4*10^1+5*10^0)%9

=(1*(9999+1)+2*(999+1)+3*(99+1)+4*(9)+5*1)%9

=1*(0+1)+2*(0+1)+3(0+1)+2*(0+1)+1*(0+1)%9 /// (a+b)%m=(a%m+b%m)%m

=(1+2+3+4+5)%9

Check whether number 4819250393285 is divisible by 9

12345 % 9 = (1 * 10^4 + 2 * 10^3 + 3 * 10^2 + 4 * 10^1 + 5 * 10^0) % 9

(1 * (9999 + 1) + 2 * (999 + 1) + 3 * (99 + 1) + 4 * (9 + 1) + 5 *(1)) % 9

(1 * (1) + 2 * (1) + 3 * (1) + 4 * (1) + 5 *(1)) % 9

(1 + 2 + 3 + 4 + 5) % 9

Exponentiation in modular arithmetic:

If $a \equiv b \pmod N$ then
$a^k \equiv b^k \pmod N$

If $16 \equiv 1 \pmod 3$ then
$16^5 \equiv 1^5 \pmod 3$

(29^10)mod(3)
(29^10)%3=(2^10)%3
1024%3
Ans

Let's solve some problems

Find 2^123456789 (mod 7)

123456789 ≡ 0 (mod 3)

2^123456789 = (2^3)^41152263

(8 ^ 41152263) % 7

8 ≡ 1 (mod 7)

(1 ^ 41152263) % 7 = 1

Class L11: Modular GCD_jodi number beshi boro hoy:

Example Input

```
2
10 1 1
9 1 5
```

Example Output

```
1
2
```

Explanation

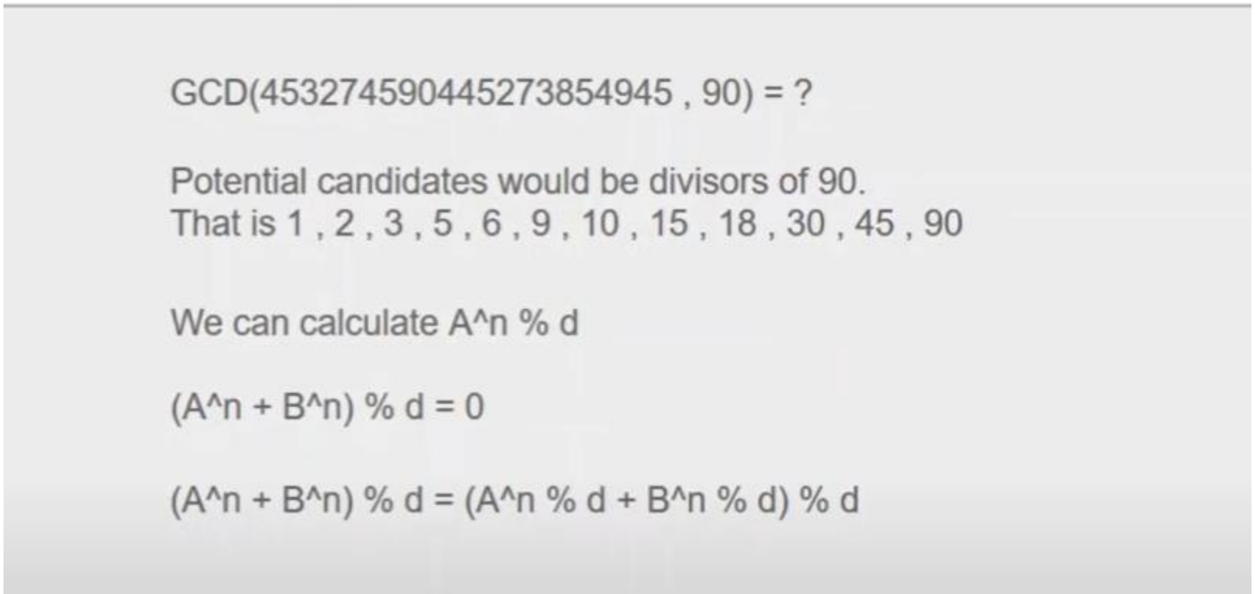
Example case 1: $GCD(10^1 + 1^1, 10 - 1) = GCD(11, 9) = 1$

Example case 2: $GCD(9^5 + 1^5, 9 - 1) = GCD(59050, 8) = 2$

If you need to calculate GCD(X , Y) where X is a very huge number but Y is smaller then We would find potential GCD candidates and apply modulo arithmetic to find GCD

GCD(453274590445273854945 , 90) = ?

Potential candidates would be divisors of 90.
That is 1 , 2 , 3 , 5 , 6 , 9 , 10 , 15 , 18 , 30 , 45 , 90



এখানে একটা টেকনিক খাটিয়েছি , এখানে overflow hbe , tai divisor diye chceck korbo arki

Ekhn dekho ami jodi abs(a-b) er divisor gula jodi a^n+a^b dara vig di zodi shunno pai
তাহলে সেটার থেকে মেক্সিমাম টা নিলেই কিন্তু হইতেছে বুঝছোস?
এটাই করব ।
কিন্তু তখন ডিভিসর গুলা দিয়ে মোড করতে হবে ।
https://github.com/NazrulIslamSajib/number_theroy/blob/main/too_big_gcd.cpp