

Topics :

1. Number System (Base Conversion)
2. Logarithm
3. Ceil & Floor
4. Non-Rounding Fractional Number
5. Number Series / Sequence
6. Big Integer
7. GCD & LCM
8. Bit manipulation
9. Misc

Number System (Base Conversion)

1. Base to Decimal :

Suppose there is number x in base B , then $x = d_n d_{n-1} \dots d_3 d_2 d_1 d_0$, where $0 \leq d_i < B$.

The number x in decimal will be $d_n \times B^n + d_{n-1} \times B^{n-1} + \dots + d_1 \times B^1 + d_0 \times B^0$.

For example :

(AB12) in Base 16 = (??) in base 10

$$\begin{array}{c} \text{A} \times 16^3 + \text{B} \times 16^2 + \text{1} \times 16^1 + \\ = (43794)_{10} \\ \text{2} \times 16^0 \end{array}$$

2. Decimal to Base :

$x \% B$ will give us the last digit of x in base B . We can get the remaining digits by repeatedly dividing x by B and taking its last digit until x becomes 0.

$$x = x : d_0 = x \% B$$

$$x = \lfloor \frac{x}{B} \rfloor : d_1 = x \% B$$

...

For example :

(14) in Base 10 = (??) in base 2

$$\begin{array}{r} 14 \\ \times 2 \\ \hline 28 \\ \times 2 \\ \hline 56 \\ \times 2 \\ \hline 112 \\ \times 2 \\ \hline 224 \\ \times 2 \\ \hline 448 \\ \times 2 \\ \hline 896 \\ \times 2 \\ \hline 1792 \\ \times 2 \\ \hline 3584 \\ \times 2 \\ \hline 7168 \\ \times 2 \\ \hline 14336 \\ \times 2 \\ \hline 28672 \\ \times 2 \\ \hline 57344 \\ \times 2 \\ \hline 114688 \\ \times 2 \\ \hline 229376 \\ \times 2 \\ \hline 458752 \\ \times 2 \\ \hline 917504 \\ \times 2 \\ \hline 1835008 \\ \times 2 \\ \hline 3670016 \\ \times 2 \\ \hline 7340032 \\ \times 2 \\ \hline 14680064 \\ \times 2 \\ \hline 29360128 \\ \times 2 \\ \hline 58720256 \\ \times 2 \\ \hline 117440512 \\ \times 2 \\ \hline 234881024 \\ \times 2 \\ \hline 469762048 \\ \times 2 \\ \hline 939524096 \\ \times 2 \\ \hline 1879048192 \\ \times 2 \\ \hline 3758096384 \\ \times 2 \\ \hline 7516192768 \\ \times 2 \\ \hline 15032385536 \\ \times 2 \\ \hline 30064771072 \\ \times 2 \\ \hline 60129542144 \\ \times 2 \\ \hline 120259084288 \\ \times 2 \\ \hline 240518168576 \\ \times 2 \\ \hline 481036337152 \\ \times 2 \\ \hline 962072674304 \\ \times 2 \\ \hline 1924145348608 \\ \times 2 \\ \hline 3848290697216 \\ \times 2 \\ \hline 7696581394432 \\ \times 2 \\ \hline 15393162788664 \\ \times 2 \\ \hline 30786325577328 \\ \times 2 \\ \hline 61572651154656 \\ \times 2 \\ \hline 123145302309312 \\ \times 2 \\ \hline 246290604618624 \\ \times 2 \\ \hline 492581209237248 \\ \times 2 \\ \hline 985162418474496 \\ \times 2 \\ \hline 1970324836948992 \\ \times 2 \\ \hline 3940649673897984 \\ \times 2 \\ \hline 7881299347795968 \\ \times 2 \\ \hline 15762598695591936 \\ \times 2 \\ \hline 31525197391183872 \\ \times 2 \\ \hline 63050394782367744 \\ \times 2 \\ \hline 126100789564735488 \\ \times 2 \\ \hline 252201579129470976 \\ \times 2 \\ \hline 504403158258941952 \\ \times 2 \\ \hline 100880631657898384 \\ \times 2 \\ \hline 201761263315796768 \\ \times 2 \\ \hline 403522526631593536 \\ \times 2 \\ \hline 807045053263187072 \\ \times 2 \\ \hline 1614090106526374144 \\ \times 2 \\ \hline 3228180213052748288 \\ \times 2 \\ \hline 6456360426105496576 \\ \times 2 \\ \hline 12912720852210993152 \\ \times 2 \\ \hline 25825441704421986304 \\ \times 2 \\ \hline 51650883408843972608 \\ \times 2 \\ \hline 103301766817687945216 \\ \times 2 \\ \hline 206603533635375890432 \\ \times 2 \\ \hline 413207067270751780864 \\ \times 2 \\ \hline 826414134541503561728 \\ \times 2 \\ \hline 1652828269083007123456 \\ \times 2 \\ \hline 3305656538166014246912 \\ \times 2 \\ \hline 6611313076332028493824 \\ \times 2 \\ \hline 13222626152664056967648 \\ \times 2 \\ \hline 26445252305328113935296 \\ \times 2 \\ \hline 52890504610656227870592 \\ \times 2 \\ \hline 105781009221312455741184 \\ \times 2 \\ \hline 211562018442624911482368 \\ \times 2 \\ \hline 423124036885249822964736 \\ \times 2 \\ \hline 846248073770499645929472 \\ \times 2 \\ \hline 1692496147540999298858944 \\ \times 2 \\ \hline 3384992295081998597717888 \\ \times 2 \\ \hline 6769984590163997195435776 \\ \times 2 \\ \hline 13539969180327994390871536 \\ \times 2 \\ \hline 27079938360655988781743072 \\ \times 2 \\ \hline 54159876721311977563486144 \\ \times 2 \\ \hline 10831975344262395512692288 \\ \times 2 \\ \hline 21663950688524791025384576 \\ \times 2 \\ \hline 43327901377049582050769152 \\ \times 2 \\ \hline 86655802754099164101538304 \\ \times 2 \\ \hline 173311605508198328203076608 \\ \times 2 \\ \hline 346623211016396656406153216 \\ \times 2 \\ \hline 693246422032793312812306432 \\ \times 2 \\ \hline 1386492844065586625624612864 \\ \times 2 \\ \hline 2772985688131173251249225728 \\ \times 2 \\ \hline 5545971376262346502498451456 \\ \times 2 \\ \hline 11091942732524893049988929112 \\ \times 2 \\ \hline 22183885465049786099977858224 \\ \times 2 \\ \hline 44367770930099572199955716448 \\ \times 2 \\ \hline 88735541860199144399911432896 \\ \times 2 \\ \hline 177471083720398288799822665792 \\ \times 2 \\ \hline 354942167440796577599645331584 \\ \times 2 \\ \hline 709884334881593155199290663168 \\ \times 2 \\ \hline 1419768669763186310388581326336 \\ \times 2 \\ \hline 2839537339526372620777162652672 \\ \times 2 \\ \hline 5679074679052745241554325305344 \\ \times 2 \\ \hline 11358149358105490483108650610688 \\ \times 2 \\ \hline 22716298716210980966217301221376 \\ \times 2 \\ \hline 45432597432421961932434602442752 \\ \times 2 \\ \hline 90865194864843923864869204885504 \\ \times 2 \\ \hline 181730389729687847729738409770008 \\ \times 2 \\ \hline 363460779459375695459476819540016 \\ \times 2 \\ \hline 726921558918751390918953639080032 \\ \times 2 \\ \hline 1453843117837522781837907278160064 \\ \times 2 \\ \hline 2907686235675045563675814556320128 \\ \times 2 \\ \hline 5815372471350091127351629112640256 \\ \times 2 \\ \hline 1163074494270018225470325822528512 \\ \times 2 \\ \hline 2326148988540036450940651645057024 \\ \times 2 \\ \hline 4652297977080072901881303290114048 \\ \times 2 \\ \hline 9304595954160145803762606580228096 \\ \times 2 \\ \hline 18609191908320291607525213160456192 \\ \times 2 \\ \hline 37218383816640583215050426320912384 \\ \times 2 \\ \hline 74436767633281166430100852641825768 \\ \times 2 \\ \hline 148873535266562332860201705283651536 \\ \times 2 \\ \hline 297747070533124665720403410567303072 \\ \times 2 \\ \hline 595494141066249331440806821134606144 \\ \times 2 \\ \hline 1190988282132496662881613642269212288 \\ \times 2 \\ \hline 2381976564264993325763227284538425576 \\ \times 2 \\ \hline 4763953128529966651526454569076851152 \\ \times 2 \\ \hline 9527906257059933303052909138153702304 \\ \times 2 \\ \hline 19055812540119866606105818276307404608 \\ \times 2 \\ \hline 38111625080239733212211636552614809216 \\ \times 2 \\ \hline 76223250160479466424423273105229618432 \\ \times 2 \\ \hline 152446500320958932848846546210559236864 \\ \times 2 \\ \hline 304893000641917865697693092421118473728 \\ \times 2 \\ \hline 609786001283835731395386184842236947456 \\ \times 2 \\ \hline 121957200256767146279077236968447895912 \\ \times 2 \\ \hline 243914400513534292558154473936895791824 \\ \times 2 \\ \hline 487828801027068585116308947873791583648 \\ \times 2 \\ \hline 975657602054137170232617895747583167296 \\ \times 2 \\ \hline 195131520410827434046523579149516634592 \\ \times 2 \\ \hline 390263040821654868093047158299033269184 \\ \times 2 \\ \hline 780526081643309736186094316598066538368 \\ \times 2 \\ \hline 156105216328661947237218863319613307736 \\ \times 2 \\ \hline 312210432657323894474437726639226615472 \\ \times 2 \\ \hline 624420865314647788948875453278453230944 \\ \times 2 \\ \hline 1248841730629155577897709006556866461888 \\ \times 2 \\ \hline 2497683461258311155795418013113733123776 \\ \times 2 \\ \hline 4995366922516622311590836026227466247552 \\ \times 2 \\ \hline 999073384503324462318167205245493249504 \\ \times 2 \\ \hline 199814676900664892463634401049098649808 \\ \times 2 \\ \hline 399629353801329784927268802098197297616 \\ \times 2 \\ \hline 799258707602659569854537604196394595232 \\ \times 2 \\ \hline 1598517415205319139709073208392789190464 \\ \times 2 \\ \hline 3197034830410638279418146416785578380928 \\ \times 2 \\ \hline 6394069660821276558836292833571156761856 \\ \times 2 \\ \hline 1278813932164255311767258566714231353712 \\ \times 2 \\ \hline 2557627864328510623534517133428462707424 \\ \times 2 \\ \hline 5115255728657021247069034266856925414848 \\ \times 2 \\ \hline 10230511457344042494138068533713850829696 \\ \times 2 \\ \hline 20461022914688084988276137067427701659392 \\ \times 2 \\ \hline 40922045829376169976552274134855403318784 \\ \times 2 \\ \hline 81844091658752339953104548269710806637568 \\ \times 2 \\ \hline 163688183317504679906209096539421613275136 \\ \times 2 \\ \hline 327376366635009359812418193078843226540272 \\ \times 2 \\ \hline 654752733270018719624836386157686453080544 \\ \times 2 \\ \hline 1309505466540037439249672772315372906161088 \\ \times 2 \\ \hline 2619010933080074878499345544630745812322176 \\ \times 2 \\ \hline 5238021866160149756998691089261491624644352 \\ \times 2 \\ \hline 1047604373232029551399738217852298324928864 \\ \times 2 \\ \hline 2095208746464059102799476435704596649857728 \\ \times 2 \\ \hline 4190417492928118205598952871409193297755456 \\ \times 2 \\ \hline 8380834985856236411197905742818386595510912 \\ \times 2 \\ \hline 1676166997171247282239581148563677319102080 \\ \times 2 \\ \hline 3352333994342494564479162297127354638204160 \\ \times 2 \\ \hline 6704667988684989128958324594254709276408320 \\ \times 2 \\ \hline 13409335977369788257916649188509418532816640 \\ \times 2 \\ \hline 26818671954739576515833298377018837065633280 \\ \times 2 \\ \hline 53637343909479153031666596754037674131266560 \\ \times 2 \\ \hline 10727468781895830606333193508807334826253120 \\ \times 2 \\ \hline 21454937563791661212666387017614669652506240 \\ \times 2 \\ \hline 42909875127583322425332774035229339305012480 \\ \times 2 \\ \hline 85819750255166644850665548070458678610024960 \\ \times 2 \\ \hline 171639500510333289701331096140917357220049920 \\ \times 2 \\ \hline 343279001020666579402662192281834714440099840 \\ \times 2 \\ \hline 686558002041333158805324384563669428880199680 \\ \times 2 \\ \hline 1373116004082666317610648769127338857760399360 \\ \times 2 \\ \hline 2746232008165332635221297538254677715520798720 \\ \times 2 \\ \hline 5492464016330665270442$$

Code

1. Base to Decimal :

$X = (A B | L)_{16}$



```
1 int baseToDecimal ( string x, int base )
2 {
3     int res = 0;
4     int len = x.length();
5
6     int coef = 1;
7     for ( int i = len - 1; i >= 0; i-- ) {
8         int d;
9         if(x[i] >= '0' && x[i] <= '9') d = x[i] - '0';
10        else d = (x[i] - 'A') + 10;
11        res += d * coef;
12        coef *= base;
13    }
14    return res;
15 }
```

2. Decimal to Base :



```
1 string symbol = "0123456789ABCDEF";
2
3 string decimalTobase(int x, int base)
4 {
5     string res = "";
6     while(x)
7     {
8         int r = x % base;
9         res = res + symbol[r];
10        x/=base;
11    }
12    if(res=="")
13        res = symbol[0];
14    reverse(res.begin(), res.end());
15 }
16 }
```



Logarithm

If $b^y = x$, then $y = \log_b x$

Base Change Rule,

$$\log_b(a) = \frac{\log_x(a)}{\log_x(b)}$$

$$\log_3(a) = \frac{\log_{10}(a)}{\log_{10}(3)}$$

$\text{ceil}(x)$ -> rounds up (returns double value)

$$f, x = \frac{a}{b}$$
$$a + [b-1]$$

$$\frac{5}{2} = 2$$

$\log_{10}(x)$ -> returns double/ long double value.

'x' can be int, float, double, long double, long long...

$$x = 10$$

ceil & Floor

$\text{floor}(x)$ -> rounds down (returns double value)

$$\sqrt[3]{2.5} = 2$$
$$\frac{5+2}{3}$$

Non-Rounding Fractional Number

```
● ● ●  
1 double num = 2020.99998888888;  
2 char buffer[50];  
3 int n;  
4 sprintf(buffer, "%f", num); // sprintf(char *str, format, variable)  
5 printf("%s\n", buffer);  
6  
7 sscanf(buffer, "%d", &n); // string to number  
8 printf("%d\n", n);
```

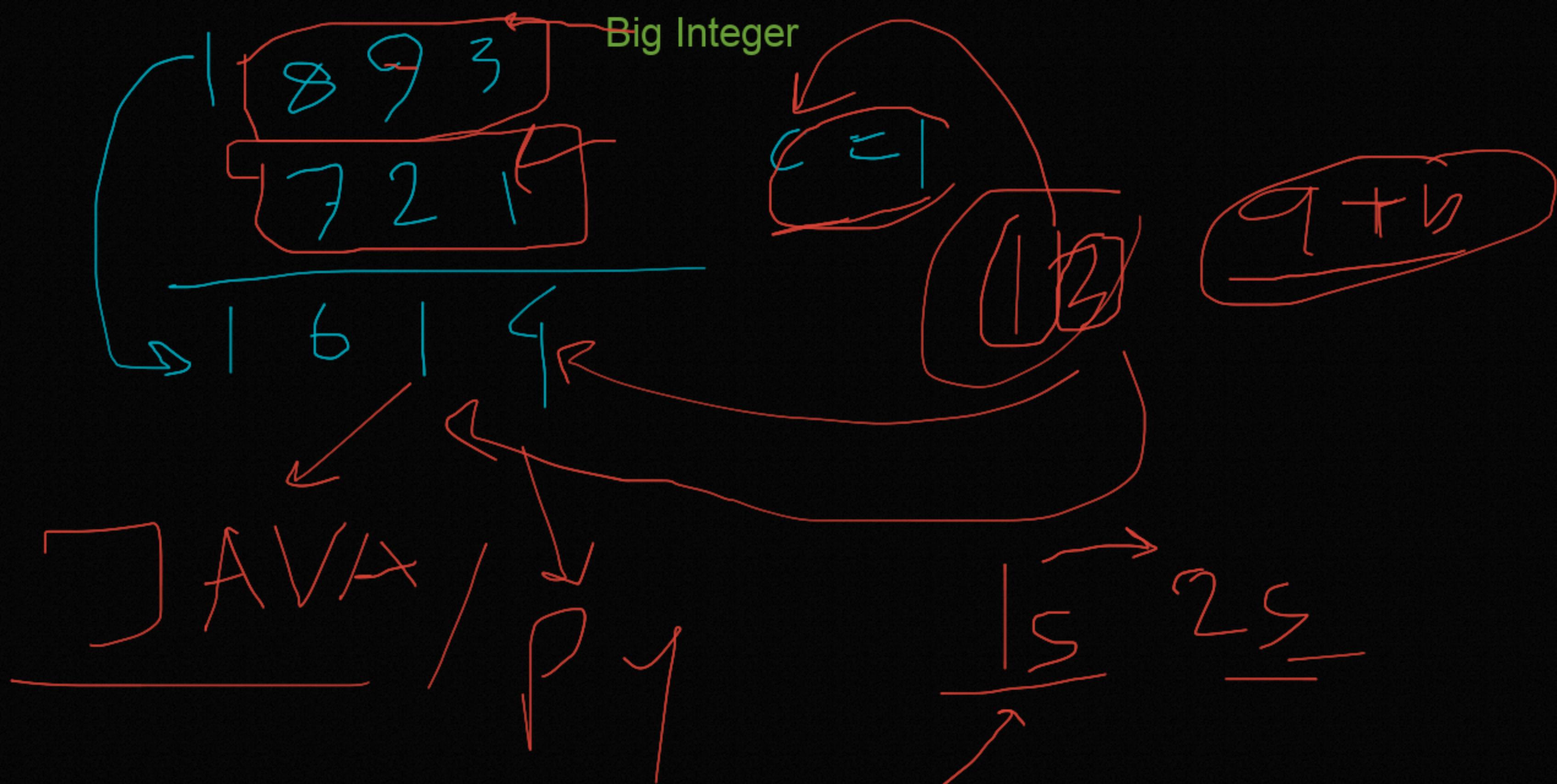


Number Series & Sequence

visit site : <https://oeis.org/>

:p

1. Arithmetic progression
2. Geometric progression



GCD & LCM

10

```
1 int gcd(int a, int b)
2 {
3     if (a == 0)
4         return b;
5     return gcd(b % a, a);
6 }
```

三

$$a * b = \gcd(a, b) * \text{lcm}(a, b)$$

Basic Euclidean Algorithm for GCD

18 15

$$18 = 12 \text{ } \boxed{3} \text{ } 6 \mid 3$$

15-185 15

$\log(n)$

Bit Manipulation

OR:

$$\begin{array}{r} \text{S} = 101 \\ \text{B} = 010 \\ \hline \text{R} = 111 \end{array}$$

XOR:

$$\begin{array}{r} \text{S} = 101 \\ \text{B} = 011 \\ \hline \text{L} = 110 \end{array}$$

AND:

$$\begin{array}{r} \text{S} = 101 \\ \text{B} = 011 \\ \hline \text{L} = 001 \end{array}$$

$$\begin{array}{r} \text{S} = 101 \\ \text{B} = 101 \\ \hline \text{L} = 001 \end{array}$$

(A + B)

(A & B)

1. $a[l] \wedge a[l+1] \wedge a[l+1] \wedge \dots \wedge a[r] = b[r] \wedge b[l-1]$ (here $b[i]$ is prefix xor till ' i ')
2. $(A + B) = (A \wedge B) + 2 * (A \& B)$

Misc

$$1. (3 + 6 - 2 + 5) = ??$$

$$2. (2 + 6 / 2 + 4) = ??$$

3. double pow (double base, double exponent);

↑

↙
call (Pow(5, 2)

$$5^2 = 25$$

5000 |

6000 |

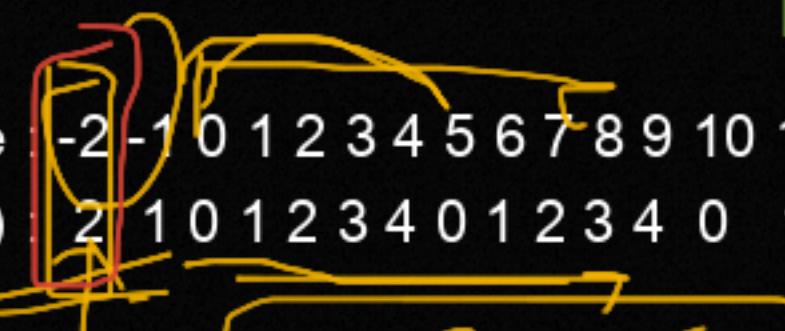
24.9999

Topics :

1. Addition
2. Subtraction
3. Multiplication
4. Division (Inverse MOD)
5. Exponentiation (Big MOD)

Modular Addition

Number Line



Number Line (mod 5)



$$5 \pmod{3} \equiv 8 \pmod{3}$$

$$-5 \pmod{2} \equiv 1 \pmod{2}$$

not "-1"

$$(-5) \% 2 = -1 + 2 = 1$$

$$(a + b) \% \text{MOD} = (a \% \text{MOD} + b \% \text{MOD}) \% \text{MOD}$$

$$(a + b + c + \dots) \% \text{MOD}$$

$$= (a \% \text{MOD} + b \% \text{MOD} + c \% \text{MOD} + \dots) \% \text{MOD}$$

If we use mod n then we will get number 0 to n-1.



```
1 inline void norm(ll &a) { a %= mod; (a < 0) && (a += mod); }
```



```
1 inline ll modAdd(ll a, ll b) { a %= mod, b %= mod; norm(a), norm(b); return (a + b) % mod; }
```

Modular Subtraction

$$(a - b) \% \text{MOD} = (a \% \text{MOD} - b \% \text{MOD}) \% \text{MOD}$$



```
1 inline ll modSub(ll a, ll b) { a %= mod, b %= mod; norm(a), norm(b); a -= b; norm(a); return a; }
```

Modular Multiplication

$$(a * b) \% \text{MOD} = (a \% \text{MOD} * b \% \text{MOD}) \% \text{MOD}$$



```
1 inline ll modMul(ll a, ll b) { a %= mod, b %= mod; norm(a), norm(b); return (a * b) \% mod; }
```

$$\begin{array}{|c|} \hline 3 \\ \hline \end{array} \times \begin{array}{|c|} \hline 4 \\ \hline \end{array}$$

$$\begin{array}{|c|} \hline 4 \\ \hline \end{array} + \begin{array}{|c|} \hline 4 \\ \hline \end{array} + \begin{array}{|c|} \hline 4 \\ \hline \end{array}$$

Modular Division

$$\frac{5}{2} = 2 \cdot 5$$

$$\left(\frac{a}{b} \right) \not\equiv 0 \pmod{d}$$

$$\begin{aligned} \left(\frac{a}{b} \right) \not\equiv 0 \pmod{d} &= \left(ab^{-1} \right) \not\equiv 0 \pmod{d} \\ &= \left(a \not\equiv 0 \pmod{d} \right) * \left(b^{-1} \not\equiv 0 \pmod{d} \right) \end{aligned}$$

[From modular multiplication]

Fermat's little theorem

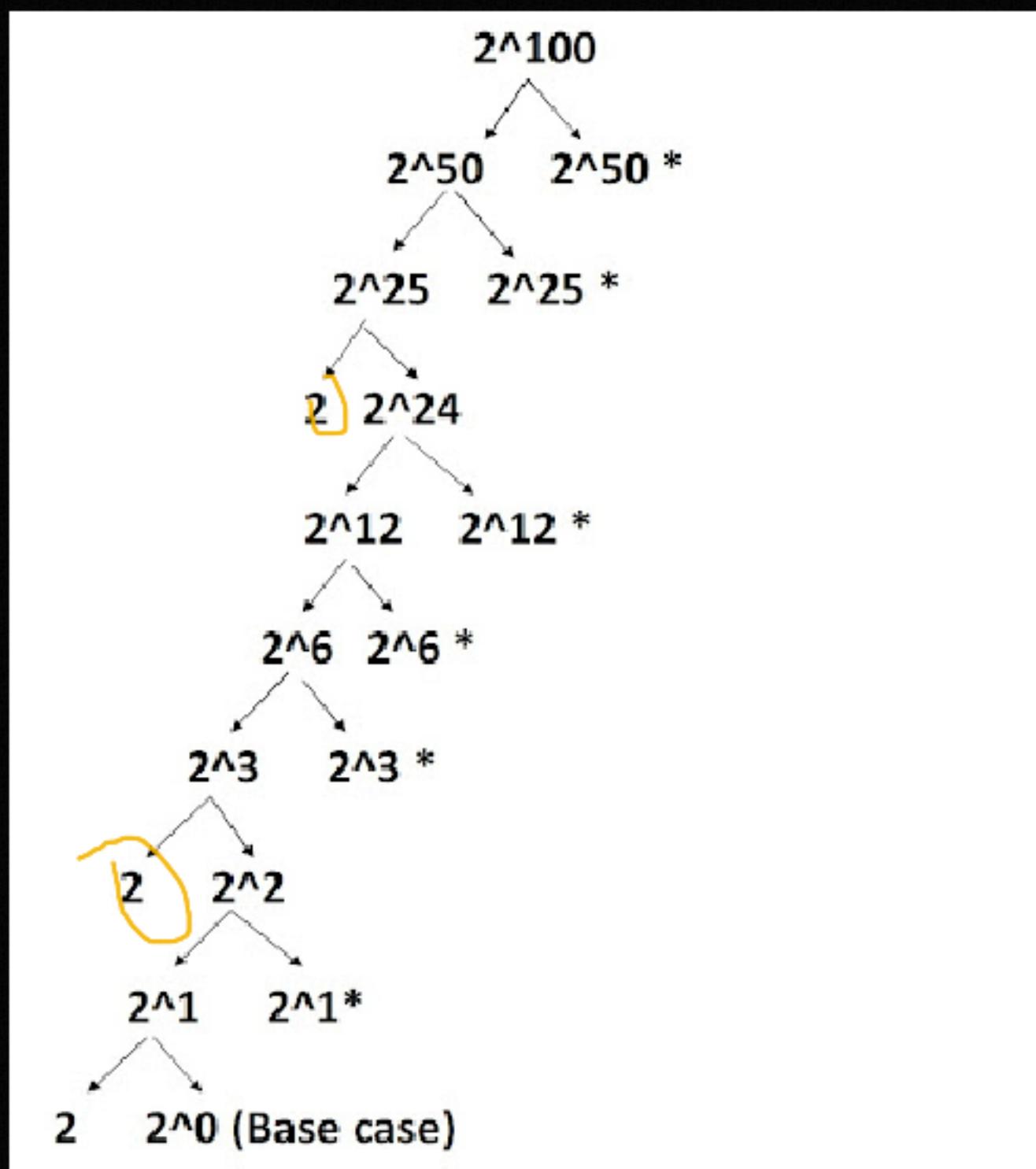
$$a^{p-1} \equiv 1 \pmod{p}$$

here, a & p are co-primes.

$$a^{p-2} \cdot a \equiv 1 \pmod{p} \rightarrow a^{p-2} \equiv a^{-1} \pmod{p}$$

Big Mod

Big Mod (Modular Exponentiation)



```
1 #define i64 long long
2 i64 M;
3 i64 F(i64 N,i64 P)
4 {
5     if(P==0) return 1;
6     if(P%2==0)
7     {
8         i64 ret=F(N,P/2);
9         return ((ret%M)*(ret%M))%M;
10    }
11    else return ((N%M)*(F(N,P-1)%M))%M;
12
13 }
```

Code of Modular Inverse



```
1 inline ll modInverse(ll a) { return modPow(a, mod - 2); }
2 inline ll modDiv(ll a, ll b) { return modMul(a, modInverse(b)); }
```

$$7 - \cancel{1} \cancel{1} \cancel{1}$$
$$6 = \cancel{1} \cancel{1} 0$$
$$\overline{6} \overline{0} 0$$

Code of Big Mod



```
1 inline ll modPow(ll b, ll p) { ll r = 1; while (p) { if (p & 1LL) r = modMul(r, b); b = modMul(b, b); p >>= 1LL; } return r; }
```

$$\left(\frac{6}{3} \right) \circ \sqrt{5} = 2$$

$$\frac{\left(15 \% \sqrt{5} \right)}{\left(3 \circ \sqrt{5} \right)} \circ \sqrt{5}$$

$$= \left(\frac{1}{3} \right) \cdot 15 = 5$$

