

GCD & HCF :-

$$\begin{array}{c|cc} 2 & 18 \\ \hline & b \\ a & \end{array}$$

$$\begin{array}{l} a = 2 \\ b = 18 \end{array}$$

$$\begin{array}{c|cc} & a & b \\ \hline & 18 & 2 \\ & 9 & 18 \\ & 9 & 18 \\ \hline & 0 & 9 \end{array}$$

while($a \% b \neq 0$) {

$$r = a \% b;$$

$$\begin{array}{l} a = b \\ b = r \end{array}$$

}

return b;

}

$$\begin{array}{l} a = 10 \\ b = 15 \end{array}$$

$$10 \% 15 \geq 10$$

$$\begin{array}{l} a = b \\ a = 15 \end{array}$$

$$b = n \quad b = 10$$

```
public static int gcdRec(int a, int b) {
    1 int r = a % b;
    2 if (r == 0) return b;
    3 return gcdRec(b, r);
}
```

```
public static int gcd(int a, int b) {
    1 while (a % b != 0) {
    2     int r = a % b;
    3     a = b;
    4     b = r;
    5 }
    6 return b;
}
```

$$\begin{array}{c|cc} a & b & r \\ \hline 8 & 4 & 0 \\ 12 & 8 & 4 \\ 56 & 12 & 8 \\ 68 & 56 & 12 \\ 124 & 68 & 56 \end{array}$$

$$\begin{array}{c|cc} & a & b \\ \hline & 8 & 8 \\ & 0 & 0 \end{array}$$

$$\begin{array}{l} a = 2 \\ b = 18 \\ n = 9 \end{array}$$

Power :-

$$n^n = 2^5 = 32$$

$$n^1 = n$$

$$n^n = n \times n^{(n-1)}$$

$$2^5 = 2$$

		1
2	1	2
5	2	2
4	3	2
3	2	4

```
pow(n, n) { if (n == 1) return 1;
1. if (n == 1) return 1;
2. return 1 * pow(n, n-1);
3 }
```

4	5	2	3	2
8	3	2	4	22
16	5	2	5	22
32				

?

$$\underline{2}^{10} = 1024$$

$$\underline{2^5 \times 2^5} = 2^{10}$$

$$\underline{\underline{2^2 \times 2^2 \times 2^2}} = 2^5$$

$$\underline{2^2 \times 2^2} = 2^4$$

```

    PowO(n, n)
    if (n == 1) return n;
    res = PowO(n, n/2);
    res = res * res;
    if (n%2 == 1) {
        res = res * n;
    }
    return res;
}

```

```

public static int pow0(int n, int x) {
    1 if (x == 1) return n;
    2 int res = pow0(n, x / 2);
    3 res = res * res;
    4 if (x % 2 == 1) {
        res = res * n;
    }
    5 return res;
}

```

$$\underline{2/2 = 1}$$

$$\underline{2^5 = 2 \times 2 \times 2}$$

n	n	res
2	1	
2	2	2
2	5	2 \times 2 \times 2
2	10	2 \times 2 \times 2 \times 2 \times 2
32		1024

n	n	res
3	1	
3	2	3
3	5	3 \times 3 \times 3 \times 3 \times 3
9		243

```

public static int pow0(int n, int x) {
    1 if (x == 1) return n;
    2 int res = pow0(n, x / 2);
    3 res = res * res;
    4 if (x % 2 == 1) {
        res = res * n;
    }
    5 return res;
}

```

$$\underline{n \times n^{(n-1)}}$$

$$\text{unit } a = 2.5 = 2$$

$$\log_2(10) = 3 \dots$$

pow
 $\underline{2^{10}}$
 $\downarrow 2^4$
 2^8
 2^7
 2^6
 2^5
 2^4

$$\begin{bmatrix} 2^{10} \\ 2^5 \\ 2^2 \\ 2^1 \end{bmatrix} = 4$$

$$\frac{n}{2} = \frac{1}{2}$$

$$\underline{4 \approx 3}$$

$$\log_2(n)$$

$$\underline{\underline{2^{10} = 1024}} \quad \text{on int} = 2^0$$

$$\begin{array}{c}
 \begin{array}{l}
 2^5 \\
 2^4 \\
 2^3 \\
 2^2 \\
 2^1 \\
 n = 10 \\
 \hline
 2^3 = 8 \\
 2^4 = 16
 \end{array}
 &
 \begin{array}{c}
 \frac{\log(n)}{2} \\
 \log(10) \\
 \hline
 \end{array}
 &
 \begin{array}{c}
 \frac{n^k}{n} = n \\
 \log(n) = k
 \end{array}
 &
 \begin{array}{c}
 \approx \\
 \log(1024) \approx 10
 \end{array}
 &
 \left\lfloor \frac{\log(n)}{k} \right\rfloor
 \end{array}$$

$\Theta(\log(n))$

Time Complexity :-

- => Time taken by your code to get the result.
- => Big O notation ('O') :-

$$\begin{array}{c}
 [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, \dots, n] \\
 \hline
 \text{and} \\
 \underbrace{n + n}_{n+1} = 2n = \Theta(n)
 \end{array}$$

$$n^2 + n^2 = \Theta(n^2)$$

$$[\dots] \Rightarrow \text{size} = n + 1 \neq$$

$$\begin{array}{c}
 n^k \\
 2^2 \quad | \quad n \\
 \hline
 n^2 \quad | \quad 12
 \end{array}
 \quad \checkmark$$

$$n = \text{you } (\text{or } k) \\ \text{you } (0 - n)$$

$$\text{constant} \Rightarrow \Theta(1)$$

$$\begin{array}{l}
 2^5 = 2 \times 2^{(n-1)} \\
 2^4 = 2 \times 2^{(n-2)} \\
 2^3 = 2 \times 2^{(n-3)}
 \end{array}$$

$$\begin{aligned} \underline{\underline{2}}^4 &= 2 \times 2^{\underline{\underline{n-3}}} \\ \underline{\underline{2}}^3 &= 2 \times 2^{\underline{\underline{n-4}}} \\ \underline{\underline{2}}^2 &= 2 \times 2^{\underline{\underline{n-5}}} \\ \underline{\underline{2}}^1 &= 2 \times 2^{\underline{\underline{n-6}}} \end{aligned}$$

$$2^{(\underline{\underline{n-6}})} = \underline{\underline{2}}^1$$