→ BINARY EXPONENTIATION : ITERATIVE METHOD 48

NOTE: This approach is faster than the recursive one, so, we should follow this only.

$n = 3^{13}$

Suppose $a = 3$ and $b = 13$

So, what we have to do is to write b in the powers of 2 and consider ~~all~~ ~~ab~~ only those which have set bits.

e.g.:

$$3^{13} \longrightarrow 3^{(1101)_2}$$
$$\longrightarrow 3^{(8+4+0+1)}$$
$$\longrightarrow 3^8 \times 3^4 \times 3^0 \times 3^1$$
$$\longrightarrow 3^{13}$$

```
LL  BINEXPITER (INT A, INT B)
{
    LL ANS = 1;
    WHILE (B)
    {
        IF (B & 1)      || check for set bit
        {
            ANS = ANS * A;
        }
        A = A * A;
        B >>= 1;        || Right shifting B
    }
    RETURN ANS;
}
```

→ So, what is happening here :

$3^{13} \longrightarrow 3^{1101}$ // 13 in binary 1101

DRY RUN

Initially →	13	3	1	
	B	A	ANS	
	1101	3	3	→ Set bit
	110	3^2	3	→ Unset bit
	11	3^4	3^5	→ Set bit
	1	3^8	3^{13}	→ Set bit.

It's Time Complexity is O log (B).