1.xor of odd and even numbers will be odd

A B A xor B  
odd odd even  
odd even odd  
even even even

2.n&-n gives the first set bit

3.1+log2(n&-n) gives the position of first set bit

4 .invert bits without considering the leading zeroes

. ll invertBits(ll num)

{

// calculating number of bits

// in the number

ll x = log2(num) + 1;

// Inverting the bits one by one

for (ll i = 0; i < x; i++)

num = (num ^ (1 << i));

return num;

}