Bitwise operation:

<https://www.geeksforgeeks.org/bitwise-operators-in-c-cpp/>

see above Given a set of numbers where all elements occur even a number of times except one number, find the odd occurring number” This problem can be efficiently solved by just doing XOR of all numbers.

**2. Brian Kernighan’s Algorithm:**   
Subtracting 1 from a decimal number flips all the bits after the rightmost set bit(which is 1) including the rightmost set bit.   
for example :   
10 in binary is 00001010   
9 in binary is 00001001   
8 in binary is 00001000   
7 in binary is 00000111   
So if we subtract a number by 1 and do it bitwise & with itself (n & (n-1)), we unset the rightmost set bit. If we do n & (n-1) in a loop and count the number of times the loop executes, we get the set bit count.   
The beauty of this solution is the number of times it loops is equal to the number of set bits in a given integer. 