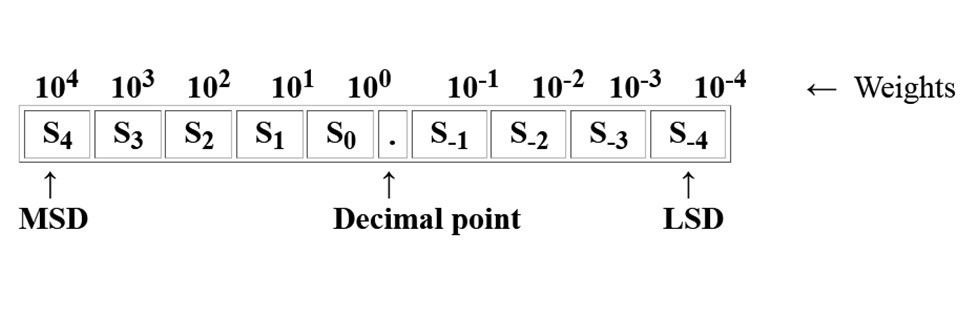
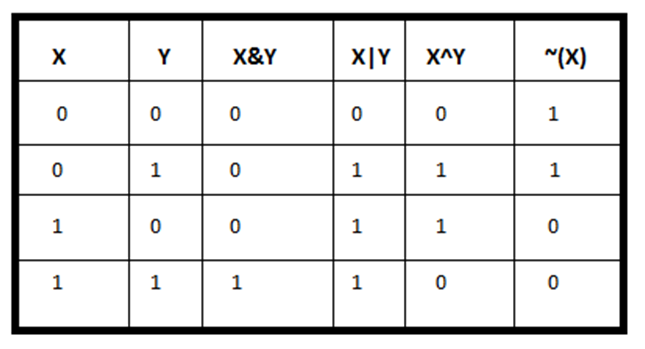
· We denote true/ON by 1 and false/OFF by 0

· binary is a base 2 number system.

· decimal is base 10 number system



**Bitwise operators**



4. **<< (left shift):** Left shifts the bits of the first operand, the second operand decides the number of places to shift.

Eg: 110111 << 1 = 1101110

0 0 1 1 1 0 1 << 2 = 1 1 1 0 1 0 0

5. **>> (right shift):** Right shifts the bits of the first operand, the second operand decides the number of places to shift.

Eg: 110111 >> 1 = 011011

0 0 1 1 1 0 1 >> 2 = 0 0 0 0 1 1 1

· **To swap (a,b) using XOR:** Do

a=a^b

b=a^b

a=a^b

·  **Find rightmost digit of a number:** x&1

**// Write a function that takes an integer and returns the number of 1 bits it has.**

import java.io.\*;

class Main {

static int countSetBits(int n) {

int count = 0;

while (n > 0) {

count += n & 1;

n >>= 1;

}

return count;

}

public static void main(String args[]) {

int i = 9;

System.out.println(countSetBits(i));

}

}

**//direct use method**

import java.io.\*;

class Main {

public static void main(String[] args)

{

System.out.println(Integer.bitCount(4));

System.out.println(Integer.bitCount(15));

}

}

**// Given an integer A. Write binary representation of the integer without leading zeros. Flip all bits then return the integer value of the binary number formed.**

import java.util.\*;

public class Main {

static void decToBin(int n) {

//calculate no of digits

int x = (int)(Math.log(n) /

Math.log(2)) + 1;

int num[]= new int[x];

for (int i = 0; i<x; i++) {

int k = n >> i;

if ((k & 1) > 0)

num[x-1-i]=1;

else

num[x-1-i]=0;

}

System.out.println(Arrays.toString(num));

flip(num);

}

static void flip(int []num) {

int dec=0;

int n=num.length;

for(int i=0, j=n-1; i<num.length; i++,j--)

if(num[i]==0) {

num[i]=1;

dec= dec + (int)Math.pow(2,j);

} else

num[i]=0;

System.out.println("Flipped binary number is : "+ Arrays.toString(num));

System.out.println("Flipped decimal number is : "+ dec);

}

public static void main(String[] args) {

int n = 10;

System.out.println("Decimal number is : " + n);

System.out.print("Binary number is : ");

decToBin(n);

}

}

**Integer. MAX\_VALUE** represents the maximum positive integer value that can be represented in 32 bits (i.e., 2147483647 ). This means that no number of type Integer that is greater than 2147483647 can exist in Java.

**READ UP:** Generate Pythagorean Triplets

https://www.geeksforgeeks.org/generate-pythagorean-triplets/

**READ UP:** GCD of array

https://www.geeksforgeeks.org/gcd-two-array-numbers/