BITWISE OPERATORS

BITWISE OPERATORS:

Bitwise operators are used to performing the manipulation of individual bits of a number.

TYPES OF BITWISE OPERATORS:

- Bitwise OR
- Bitwise AND
- Bitwise XOR
- Bitwise Left Shift
- Bitwise Right Shift
- Ones Complement

BITWISE OR:

It returns bit by bit OR of input values, i.e., if either of the bits is 1, it gives 1, else it shows 0.

For Example: “0 and 0” is given as input then the output is “0”
“0 and 1” is given as input then the output is “1”

It is represented as **“|”**

BITWISE AND:

It returns bit by bit AND of input values, i.e., if both bits are 1, it gives 1, else it shows 0.

For Example: “0 and 0” is given as input then the output is “0”
“0 and 1” is given as input then the output is “0”
“1 and 1” is given as input then the output is “1”

It is represented as **“&”**

BITWISE XOR:

It returns bit by bit XOR of input values, i.e., if corresponding bits are different, it gives 1, else it shows 0.

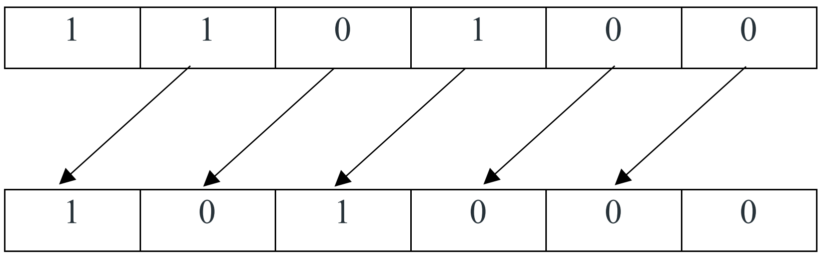
For Example: “0 and 0” is given as input then the output is “0”
“0 and 1” is given as input then the output is “1”
“1 and 1” is given as input then the output is “0”

It is represented as **“^”**

BITWISE OPERATORS

BITWISE LEFT SHIFT:

The left shift means that shift each of the bits is in binary representation toward the left

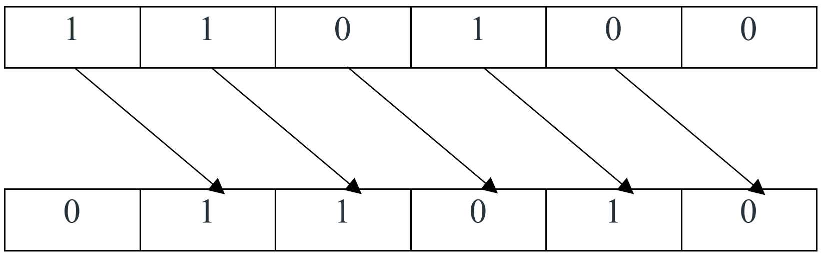


Here we are inserting 0 at the end.

It is Represented as “<<”

BITWISE RIGHT SHIFT:

The right shift operator in Java moves the bits of a value towards the right by the specified number of bits.



Here we are inserting 0 at the end (Beginning).

It is Represented as “>>”

BITWISE COMPLIMENT:

It returns the one’s complement representation of the input value, i.e., with all bits inverted, which means it makes every 0 to 1, and every 1 to 0.

For Example: “0” is given as input then the output is “1”

“1” is given as input then the output is “0”

It is represented as **“~”**

BITWISE OPERATORS

JAVA PROGRAM

```
class Bitwise
{
public static void main(String[] args)
{
int a = 5;
int b = 7;
System.out.println("BITWISE AND:" +(a&b));
System.out.println("BITWISE OR:" +(a|b));
System.out.println("BITWISE XOR:" +(a^b));
System.out.println("BITWISE COMPLIMENT:" +(~a));
System.out.println("BITWISE LEFT SHIFT:" +(a>>b));
System.out.println("BITWISE RIGHT SHIFT:" +(a<<b));
}
}
```