

BIG INTEGER

AND

BITWISE COMPLEMENT OPERATOR

Submitted by

Name : Yadhukrishna Mk

email : yadhu8824@gmail.com

BITWISE NOT(!) OPERATOR IN JAVA

In Java, the bitwise NOT operator is represented by the exclamation mark (!) symbol. However, it is important to note that the exclamation mark is not specifically a bitwise operator; it is primarily used as the logical negation operator. The logical negation operator reverses the logical state of its operand.

If you want to perform a bitwise NOT operation in Java, you should use the tilde (~) symbol. The tilde operator flips the bits of the operand, changing each 0 bit to 1 and each 1 bit to 0.

Here's an example:

```
int x = 10; // binary:00000000000000000000000000000001010  
int y = ~x; // binary: 1111111111111111111111111111110101
```

```
System.out.println(y);
```

Output: -11

BIG INTEGER IN JAVA

The BIGINT data type stores integers from $-(2^{63}-1)$ to $2^{63}-1$, which is $-9,223,372,036,854,775,807$ to $9,223,372,036,854,775,807$, in eight bytes.

BigInteger provides analogues to all of Java's primitive integer operators, and all relevant methods from java.lang. Math.BigInteger class is used for mathematical operation which involves very big integer calculations that are outside the limit of all available primitive data types.

In this way, BigInteger class is very handy to use because of its large method library and it is also used a lot in competitive programming.

EXAMPLE

```
import java.math.BigInteger;
public class BigIntegerExample
{
    public static void main(String[] args)
    {
        BigInteger num1 = new BigInteger("12345678901234567890");
        BigInteger num2 = new BigInteger("98765432109876543210");

        // Addition
        BigInteger sum = num1.add(num2);
        System.out.println("Sum: " + sum);
        // Subtraction
        BigInteger difference = num2.subtract(num1);
        System.out.println("Difference: " + difference);
    }
}
```

OUTPUT

```
sum: 111111110111111100
Difference: 86419753208641975320
```