

BIT MANIPULATION SOLUTIONS

Solution 1: The value of $x \wedge x = 0$.

Think about it, xor gives 0 when the bits are the same. If we compare the same number to itself, the bits will always be the same. So, the answer of $x \wedge x$ will always be 0.

Solution 2: The idea is to use XOR operators to swap two numbers by their property

$$x \wedge x = 0$$

```
public class Solution {  
    public static void main(String[] args) {  
        int x = 3, y = 4;  
        System.out.println("Before swap: x = " + x + " and y = " + y);  
        //swap using xor  
        x = x ^ y;  
        y = x ^ y;  
        x = x ^ y;  
        System.out.println("After swap: x = " + x + " and y = " + y);  
    }  
}
```

Solution 3 : The expression $\sim x$ will add 1 to an integer x. We know that to get negative of a number, invert its bits and add 1 to it (Remember negative numbers are stored in 2's complement form), i.e.,

$$\begin{aligned} -x &= \sim x + 1; \\ -\sim x &= x + 1 \text{ (by replacing } x \text{ by } \sim x) \end{aligned}$$

```
public class Solution {  
    public static void main(String[] args) {  
        int x = 6;  
        System.out.println(x + " + " + 1 + " is " + -~x);  
        x = -4;  
        System.out.println(x + " + " + 1 + " is " + -~x);  
        x = 0;  
        System.out.println(x + " + " + 1 + " is " + -~x);  
    }  
}
```

Solution 4 :

```
public class Solution {  
    public static void main(String[] args) {  
        // Convert uppercase character to lowercase  
        for (char ch = 'A'; ch <= 'Z'; ch++) {  
            System.out.println((char)(ch | ' '));  
            // prints abcdefghijklmnopqrstuvwxyz  
        }  
    }  
}
```

APNA
COLLEGE

er.dm.zer03@gmail.com