**Day26**

**Bit Manipulation**

**What is bit manipulation?**

[Bit manipulation](https://www.educative.io/blog/bit-manipulation-algorithm) is the process of applying logical operations on a sequence of bits, the smallest form of data in a computer, to achieve a required result. Bit manipulation has constant time complexity and process in parallel, meaning it is very efficient on all systems.

Most programming languages will have you work with abstractions, like [objects](https://www.educative.io/blog/object-oriented-programming-concepts-java) or variables, rather than the bits they represent. However, direct bit manipulation is needed to improve performance and reduce error in certain situations.

Bit manipulation requires a strong knowledge of [binary and binary conversion](https://www.educative.io/blog/computer-number-systems-binary-hexadecimal-conversions).

**Here’s a few examples of tasks that require bit manipulation:**

* Low-level device control
* Error detection and correction algorithms
* Data compression
* Encryption algorithms
* Optimization

For example, take a look at the difference between an arithmetic and bit manipulation approach to finding the green portion of an RGB value:

**Get Bit**

**import java.util.\*;**

**public class Bits {**

**public static void main(String args[]) {**

**int n = 5; //0101**

**int pos = 3;**

**int bitMask = 1<<pos;**

**if((bitMask & n) == 0) {**

**System.out.println("bit was zero");**

**} else {**

**System.out.println("bit was one");**

**}**

**}**

**}**

**Set Bit**

**import java.util.\*;**

**public class Bits {**

**public static void main(String args[]) {**

**int n = 5; //0101**

**int pos = 1;**

**int bitMask = 1<<pos;**

**int newNumber = bitMask | n;**

**System.out.println(newNumber);**

**}**

**}**

**Clear Bit**

**import java.util.\*;**

**public class Bits {**

**public static void main(String args[]) {**

**int n = 5; //0101**

**int pos = 2;**

**int bitMask = 1<<pos;**

**int newBitMask = ~(bitMask);**

**int newNumber = newBitMask & n;**

**System.out.println(newNumber);**

**}**

**}**

**Update Bit**

import java.util.\*;

public class Bits {

   public static void main(String args[]) {

       Scanner sc = new Scanner(System.in);

       int oper = sc.nextInt();

       // oper=1 -> set; oper=0 -> clear

      int n = 5;

      int pos = 1;

      int bitMask = 1<<pos;

      if(oper == 1) {

          //set

          int newNumber = bitMask | n;

          System.out.println(newNumber);

      } else {

       //clear

       int newBitMask = ~(bitMask);

       int newNumber = newBitMask & n;

       System.out.println(newNumber);

      }

   }

}