********** OUTPUT **********

SCENARIO AS PER QUESTION:
UNFORTUNATELY DUE TO THE DESIGN IT WILL SHOW ALL BINARY CONVERSIONS WHICH ARE NOT REELVANT. HOWEVER IT WILL STOP AT THE EXPECTED OUTCOME WITH CLEAR ON SCREEN INFORMATION.

```
Decimal number is: 6
*** This program will convert decimal 6 into binary
The binary version is: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0]
Number bits set: 2


***Program will execute until next number found with: 2  bits set***


Decimal number is: 7
*** This program will convert decimal 7 into binary
The binary version is: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1]
Number bits set: 3


Decimal number is: 8
*** This program will convert decimal 8 into binary
The binary version is: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
Number bits set: 1


Decimal number is: 9
*** This program will convert decimal 9 into binary
The binary version is: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1]
Number bits set: 2
6  has 2  bits set.
9  has 2  bits set.


** Process exited - Return Code: 0 **
```

AN EXAMPLE WITH 1 BIT SET.. THE NEXT EXPECTED WOULD BE 32

```
Decimal number is: 16
*** This program will convert decimal 16 into binary
The binary version is: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
Number bits set: 1


***Program will execute until next number found with: 1  bits set***
```

IT RUNS TO COMPLETION AND HALTS AS BELOW:

```
Decimal number is: 32
*** This program will convert decimal 32 into binary
The binary version is: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
Number bits set: 1
16  has 1  bits set.
32  has 1  bits set.
```

CODE

```
//***************CODE*******************
import java.util.Scanner;
import java.util.Arrays;
import java.lang.Math;

public class Main
{
   public static void main(String[] args)
   {
      int num;
      int count;
      test t;
      test eo;
      Scanner reader = new Scanner(System.in); // Reading from System.in
      System.out.println("Enter a number no larger than 2,000,000,000: ");
      num = reader.nextInt(); // Scans the next token of the input as an int.
      reader.close();

      System.out.println("\n");
      // THE FUNCTION HAS TO START here
      t= new test(num);

      System.out.println("\n\n***Program will execute until next number found with: " +
t.countOnes() + "  bits set***");

      for (int i=1; i<2*num; i++)
      {
         eo=new test(num+i);
         if (eo.countOnes()==t.countOnes() && (num+i)>num)
      {

         System.out.println(num + "  has " + t.countOnes() + "  bits set." );
         System.out.println(num+i + "  has " + eo.countOnes() + "  bits set." );
         break;
       }
      }
   }
}
```

```
class test
{
   int binary[] = new int[31]; //This is to define 31 bit array to hold binary value. Maximum value
is over 2 billion.
// This is Java's limitation.
   int j; //counter
   int i; //counter
   String conversion; // this will be used to output binary conversion on screen without having to
iterate through loop again
   int num;
   int numberOnes;
   int divisor;

public test(int num)
{
   System.out.println("\n");
   System.out.println("Decimal number is: " + num); // This is tricky part... Unsure of how to
store array of binary conversions.
   int length = binary.length;
   System.out.println("*** This program will convert decimal " + num+ " into binary");
//Execute a for loop to check modulus (i.e no remainder)

for (j=length-1; j>=0; j--)
{
   divisor = (int)(Math.pow(2,j));

   if (num - divisor >=0)
   {
      binary[(length-1)-j] = 1;
      num=num-divisor;
      numberOnes++;
   }
}
conversion = Arrays.toString(binary);
System.out.println("The binary version is: " + conversion );
System.out.println("Number bits set: " + countOnes());

}

int countOnes()
{
   return numberOnes;

}

String binaryConversion()
{
   return conversion;

}

int valueDivisor()
{
   return j;
```

```
}

}
```