

题目描述：

小扇和小船今天又玩起来了数字游戏，小船给小扇一个正整数 n ($1 \leq n \leq 1e9$)，小扇需要找到一个比 n 大的数字 m ，使得 m 和 n 对应的二进制中 1 的个数要相同（如 4 对应二进制 100 , 8 对应二进制 1000 , 1 的个数都为 1 ），现在求 m 的最小值。

输入描述：

输入：第一行输入一个正整数 n ($1 \leq n \leq 1e9$)。

输出描述：

输出：输出一个正整数 m 。

补充说明：

示例 1

输入：

2

输出：

4

说明：

2 的二进制 10 , 4 的二进制位 100 , 1 的个数相同，且 4 是满足条件的最小数

示例 2

输入：

7

输出：

11

说明：

7 的二进制 **111**, 11 的二进制位 **1011**, 1 的个数相同, 且 **11** 是满足条件的最小数

```
import java.util.Scanner;
```

```
// 注意类名必须为 Main, 不要有任何 package xxx 信息
```

```
public class Main {
```

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

```
        Scanner in = new Scanner(System.in);
```

```
        // 注意 hasNext 和 hasNextLine 的区别
```

```
        while (in.hasNextInt()) { // 注意 while 处理多个 case
```

```
            int a = in.nextInt();
```

```
            System.out.println(minNumber(a));
```

```
        }
```

```
    }
```

```
    public static final int minNumber(int number){
```

```
        // 转成二进制
```

```
        String biNumber = Integer.toBinaryString(number);
```

```
        int oneNumber = biNumber.replaceAll("0", "").length();
```

```
        for(int start = number + 1;; start ++ ){
```

```
            String startNumber = Integer.toBinaryString(start);
```

```
            if(startNumber.replaceAll("0", "").length() == oneNumber){
```

```
                return start;
```

```
            }
```

```
        }
```

```
        // 比当前数大
```

```
    }
```

```
}
```