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<=n<=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<=n<=1e9)。
输出描述：
输出：输出一个正整数 m。
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

```
const rl = require("readline").createInterface({ input: process.stdin });
var iter = rl[Symbol.asyncIterator]();
const readline = async () => (await iter.next()).value;

void (async function () {
    const source = Number(await readline());
    function findNum(n) {
        const toCheck = n + 1;
        const sourceString = Number(source).toString(2);
        const targetString = Number(toCheck).toString(2);
        const originalOneCount = sourceString
            .split("")
            .filter((s) => Number(s) === 1)?.length;
        const targetOneCount = targetString
            .split("")
            .filter((s) => Number(s) === 1)?.length;

        if (targetOneCount === originalOneCount) {
            console.log(toCheck);
        } else {
```

```
                findNum(toCheck);
            }
        }

        findNum(source);
})();
```