

## 牛客网-华为机试练习题 53

### 题目描述

- 1
- 1 1 1
- 1 2 3 2 1
- 1 3 6 7 6 3 1
- 1 4 10 16 19 16 10 4 1

以上三角形的数阵，第一行只有一个数1，以下每行的每个数，是恰好是它上面的数，左上角数到右上角的数，3个数之和（如果不存在某个数，认为该数就是0）。

求第n行第一个偶数出现的位置。如果没有偶数，则输出-1。例如输入3,则输出2，输入4则输出3。

输入n(n <= 1000000000)

### 输入描述:

输入一个int整数

### 输出描述:

输出返回的int值

示例1

输入

4

输出

3

### 解决代码:

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner in = new Scanner(System.in);

        while (in.hasNext()) {
            int num = in.nextInt();
            int[][] arrys = new int[num][];
            for (int i = 0; i < num; i++) {
                arrys[i] = new int[2*i+1];
            }
            getyanghuiTriangle(num, arrys);
        }
    }
}
```

```

        int i;
        for ( i = 0; i < arrys[num-1].length; i++) {
            if (arrys[num-1][i] % 2 == 0) {
                System.out.println(i+1);
                break;
            }
        }

        if (i == arrys[num-1].length ) {
            System.out.println("-1");
        }
    }
    in.close();
}

public static void getyanghuiTriangle(int num, int[][] arrys){
    int num1;
    int num2;
    int num3;
    for (int i = 0; i < num; i++) {
        for (int j = 0; j < arrys[i].length; j++) {
            if (i==0||j==0||j==arrys[i].length-1) {
                arrys[i][j]=1;
            }else {
                if (j-2 <0) {
                    num1 = 0;
                }else {
                    num1 = arrys[i-1][j-2];
                }

                if (j-1 <0) {
                    num2 = 0;
                }else {
                    num2 = arrys[i-1][j-1];
                }

                if (j >= arrys[i-1].length) {
                    num3 = 0;
                }else {
                    num3=arrys[i-1][j];
                }
                arrys[i][j]=num1+num2+num3;
            }
        }
    }
}

```