

Java方向编程题答案

####

day16

[编程题]46126-洗牌

链接: <https://www.nowcoder.com/questionTerminal/5a0a2c7e431e4fbbbb1ff32ac6e8dfa0>

【题目解析】

无

【解题思路】

每次读取一个数之后, 算出他经过k次洗牌后的位置, 只用一个长度为2n数组用来输出 根据当前数的位置, 可以算出经过一次洗牌后的位置

- 如果当前数小于等于n (即在左手), 则它下次出现的位置是 $2 * \text{当前位置} - 1$
- 如果当前位置大于n (即在右手), 则它下次出现的位置是 $2 * (\text{当前位置} - n)$

【示例代码】

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int groups = sc.nextInt(); // 数组组数
        while (groups-- > 0) {
            int n = sc.nextInt(); // 数字个数的一半
            int k = sc.nextInt(); // 洗牌次数
            int[] res = new int[2 * n]; // 数据总个数
            for (int i = 0; i < 2 * n; i++) { // 遍历数据数组
                int tmp = i + 1; // 此处修正内层循环更加temp计算位置
                for (int j = 0; j < k; j++) {
                    if (tmp <= n) {
                        tmp = 2 * tmp - 1;
                    } else {
                        tmp = 2 * (tmp - n);
                    }
                }
                res[tmp - 1] = sc.nextInt(); // 数组中temp-1的位置的数据, 即就是洗牌后的结果
            }
            // 输出
            if (res.length > 0) {
                System.out.print(res[0]);
            }
            for (int i = 1; i < 2 * n; i++) {
                System.out.print(" " + res[i]);
            }
            System.out.println();
        }
    }
}
```

```
}  
}
```

[编程题]23535-统计同成绩学生人数

链接: <https://www.nowcoder.com/questionTerminal/987123efea5f43709f31ad79a318ca69>

【题目解析】

无

【解题思路】

输入和计算过程:

- 输入人数
- 循环变量, 输入分数
- 输入目标分数
- 循环变量分数数组, 计算和目标分数相等的分数出现的频次
- 打印输出目标分数出现的频次

【示例代码】

```
import java.util.Scanner;  
public class Main {  
    public static void main(String[] args) {  
        Scanner scan = new Scanner(System.in);  
        while (scan.hasNext()) {  
            int N = scan.nextInt(); // 输入的人数  
            if (N == 0) { // 如果检测到输入的人数为0 则结束  
                return;  
            }  
            int[] arr = new int[N]; // 保存N个人的分数  
            for (int i = 0; i < arr.length; i++) {  
                arr[i] = scan.nextInt();  
            }  
            int t = scan.nextInt(); // 目标分数 输出击中目标分数的个数  
            int count = 0;  
            for (int i = 0; i < arr.length; i++) {  
                if (t == arr[i]) {  
                    count++;  
                }  
            }  
            System.out.println(count);  
        }  
    }  
}
```