

Java-数组-喊 7 的次数重排

题目描述：喊 7 是一个传统的聚会游戏，N 个人围成一圈，按顺时针从 1 到 N 编号。编号为 1 的人从 1 开始喊数，下一个人喊的数字为上一个人的数字加 1，但是当将要喊出来的数字是 7 的倍数或者数字本身含有 7 的话，不能把这个数字直接喊出来，而是要喊“过”。假定玩这个游戏的 N 个人都没有失误地在正确的时机喊了“过”，当喊到数字 K 时，可以统计每个人喊“过”的次数。现给定一个长度为 N 的数组，存储了打乱顺序的每个人喊“过”的次数，请把它还原成正确的顺序，即数组的第 i 个元素存储编号 i 的人喊“过”的次数。

输入描述：输入为一行，为空格分隔的喊“过”的次数，注意 K 并不提供，K 不超过 200，而数字的个数即为 N。

输出描述：输出为一行，为顺序正确的喊“过”的次数，也由空格分隔。

示例

展开

示例1

输入：0 1 0

输出：1 0 0

说明：一共只有一次喊“过”，那只会发生在需要喊7时，按顺序，编号为1的人会遇到7，故输出1 0 0。注意，结束时的K不一定是7，也可以是8、9等，喊过的次数都是1 0 0。

示例2

输入：0 0 0 2 1

输出：0 2 0 1 0

说明：一共有三次喊“过”，发生在7 14 17，按顺序，编号为2的人会遇到7 17，编号为4的人会遇到14，故输出0 2 0 1 0。

```

1  import java.util.Arrays;
2  import java.util.Scanner;
3
4  public class Main {
5      public static void main(String[] args) {
6          Scanner scanner = new Scanner(System.in);
7
8          String s = scanner.nextLine();
9          String[] split = s.split(" ");
10         int sum = 0, pc = split.length, cur = 0;
11         int[] res = new int[pc];
12         for (String s1 : split) {
13             sum += Integer.parseInt(s1);
14         }
15         int n = 0, num = 1;
16         while (cur < sum) {
17             // System.out.println(n + " " + num);
18             if (has7(num)) {
19                 res[n]++;
20                 cur++;
21             }
22             n++;
23             num %= pc;
24             num++;
25         }
26         for (int i = 0; i < res.length; i++) {
27             System.out.print(res[i]);
28             if (i < res.length - 1) {
29                 System.out.print(" ");
30             }
31         }
32         // System.out.println(Arrays.toString(res));
33     }
34
35     private static boolean has7(int num) {
36         if (num % 7 == 0) {
37             return true;
38         }
39         String s = String.valueOf(num);
40         return s.contains("7");
41     }
42 }

```