

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。

补充说明：

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。

```
const rl = require("readline").createInterface({ input: process.stdin });
```

```
var iter = rl[Symbol.asyncIterator]();
```

```
const readline = async () => (await iter.next()).value;
```

```
void async function () {  
    // Write your code here  
    while(line = await readline()){  
        let arr = line.split(' ').map(Number)  
        let len = arr.length  
        let resultArr = new Array(len).fill(0)  
        let total = arr.reduce((a,b)=>a+b)  
        let sum = 0 ,init = 1  
        let result = []  
        while(sum<total){  
            if(init%7==0 | String(init).includes('7')) {  
                sum++  
                result.push(true)  
            }else{  
                result.push(false)  
            }  
            init++  
        }  
        let resultInit = 0  
        for(let i=0;i<result.length;i++){  
            if(resultInit==len) resultInit=0  
            if(result[i]) resultArr[resultInit]=resultArr[resultInit]+1  
            resultInit++  
        }  
        console.log(resultArr.join(' '))  
    }  
}
```

