

JS-喊 7 的次数重排-喊 7 是一个传统的聚会游戏

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

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  const rl = require("readline").createInterface({ input: process.stdin });
2  var iter = rl[Symbol.asyncIterator]();
3  const readline = async () => (await iter.next()).value;
4
5  void async function () {
6    // Write your code here
7    line=await readline()
8    let tokens = line.split(' ');
9    let tokensLength=tokens.length
10   let count=0
11   let number
12   for(let i=0;i<tokensLength;i++){
13     count+=parseInt(tokens[i])
14     tokens[i]=0

```

```

15   }
16   i=7
17   while(count>0){
18     if(i%7==0){
19       if(i%tokensLength==0){
20         tokens[tokensLength-1]++
21       }else{
22         tokens[i%tokensLength-1]++
23       }
24       count--
25       i++
26       continue
27     }
28     number=(i+ "").split("")
29     if(number.indexOf('7')!=-1){
30       if(i%tokensLength==0){
31         tokens[tokensLength-1]++
32       }else{
33         tokens[i%tokensLength-1]++
34       }
35       count--
36       i++
37       continue
38     }
39     i++
40   }
41   console.log(tokens.join(" "))
42 }0

```

```

1  const rl = require("readline").createInterface({ input: process.stdin });
2  var iter = rl[Symbol.asyncIterator]();
3  const readline = async () => (await iter.next()).value;
4
5  void async function () {
6    // Write your code here
7    line=await readline()
8    let tokens = line.split(' ');
9    let tokensLength=tokens.length
10   let count=0
11   let number
12   for(let i=0;i<tokensLength;i++){
13     count+=parseInt(tokens[i])
14     tokens[i]=0
15   }
16   i=7
17   while(count>0){
18     if(i%7==0){
19       if(i%tokensLength==0){
20         tokens[tokensLength-1]++
21       }else{
22         tokens[i%tokensLength-1]++
23       }
24       count--
25       i++
26       continue
27     }
28     number=(i+"").split("")
29     if(number.indexOf("7")!=-1){
30       if(i%tokensLength==0){
31         tokens[tokensLength-1]++
32       }else{
33         tokens[i%tokensLength-1]++
34       }
35       count--
36       i++
37       continue
38     }
39     i++
40   }
41   console.log(tokens.join(" "))
42 }0

```