

# Java方向编程题答案

####

## day11

[编程题]24516-个位数统计

链接: <https://www.nowcoder.com/questionTerminal/a2063993dd424f9cba8246a3cf8ef445>

### 【题目解析】：

大家看到这道题的时候千万记住不要被前面那一串看起来复杂的数学公式给唬住了。实际上读完题大家就会发现，就是将给定的任意正整数拆分为每个数字字符，统计出现次数罢了。

### 【解题思路】：

将给定的数字按照每位拆分为单个数字然后保存在list中，每碰到一个相同的数字就值加一，扫描完整个字符串后遍历输出list集合即可。

### 【示例代码】：

```
import java.util.*;
public class Main {
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        String N = sc.next();
        char[] chars = N.toCharArray();
        /**
         * list**有10个item表示0-9，每个item的值表示数字的个数
        */
        List<Integer> list = new ArrayList<>();
        for (int i = 0; i < 10; i++) {
            list.add(0); //初始化每个item都是0
        }
        for (int i = 0; i < chars.length; i++) {
            // 每遇到一个数字，就把值+1
            list.set(chars[i]-'0',list.get(chars[i]-'0')+1);
        }
        for (int i = 0; i < list.size(); i++) {
            if (list.get(i)>0){
                System.out.println(i+":"+list.get(i));
            }
        }
    }
}
```

[编程题]23445-构建乘积数组

链接: <https://www.nowcoder.com/questionTerminal/94a4d381a68b47b7a8bed86f2975db46>

### 【题目解析】：

本题是剑指offer上的原题，构建一个数组，此数组的每个位置上的元素是由原数组相同位置i之前的所有元素相乘得来的。

### 【解题思路】：

$B[i]$ 的值可以看作下图的矩阵中每行的乘积。

下三角用连乘可以很容求得，上三角，从下向上也是连乘。

因此我们的思路就很清晰了，先算下三角中的连乘，即我们先算出 $B[i]$ 中的一部分，然后倒过来按上三角中的分布规律，把另一部分也乘进去。

$B_0$	1	$A_1$	$A_2$	...	$A_{n-2}$	$A_{n-1}$
$B_1$	$A_0$	1	$A_2$	...	$A_{n-2}$	$A_{n-1}$
$B_2$	$A_0$	$A_1$	1	...	$A_{n-2}$	$A_{n-1}$
...	$A_0$	$A_1$	...	1	$A_{n-2}$	$A_{n-1}$
$B_{n-2}$	$A_0$	$A_1$	...	$A_{n-3}$	1	$A_{n-1}$
$B_{n-1}$	$A_0$	$A_1$	...	$A_{n-3}$	$A_{n-2}$	1

【示例代码】：

```
public class Solution {
    public int[] multiply(int[] A) {
        int length = A.length;
        int[] B = new int[length];
        if(length != 0){
            B[0] = 1;
            //计算下三角连乘
            for(int i = 1; i < length; i++){
                B[i] = B[i-1] * A[i-1];
            }
            int temp = 1;
            //计算上三角
            for(int j = length-2; j >= 0; j--){
                temp *= A[j+1];
                B[j] *= temp;
            }
        }
        return B;
    }
}
```

}  
}

#####

比特科技制作