

# 有序的计数

## Permutation

### 【问题描述】

矩阵博士近来从事研究排列的工作。他发现，给定一个大小为  $n$  的排列，比如我们把它记成  $P = (a_0, a_1, \dots, a_{n-1})$ （其中对每个  $i$ ，规定  $0 \leq a_i \leq n-1$ ）。我们马上就可以统计出  $P$  的“有序数”  $o(P)$ 。“有序数”是矩阵博士那些无聊发现中的又一个新名词，博士把它定义

成  $o(P) := \sum_{i=0}^{n-1} \delta_i$ ， $\delta_i = \begin{cases} 1 & \text{if } a_i = i \\ 0 & \text{if } a_i \neq i \end{cases}$ 。为了把有序数研究到底，矩阵博士挑出了所有  $n$  长

的排列中和  $P$  有相同有序数的排列，构成一个“有序同类集”，并且仔细地把它们以字典序排成  $\{P_0, P_1, \dots, P_m\}$ 。这样，一个排列就可以根据他的有序数和所在的有序同类集中的位置来描述了。注意：由于博士的奇怪天性，有序同类集中的位置总是从 0 开始算起的。

有一天，博士家失火了（这其实是常有的事）。博士的研究手稿里有一些纸张被烧糊了，之后，他急需知道某些排列的有序数和所在的有序同类集中的位置，作为博士的惟一一名助手，你义不容辞地接受了任务。

### 【输入文件】

文件的第一行只有一个整数  $n$  ( $1 \leq n \leq 64$ )，代表给定排列的长度。文件的第二行开始有  $n$  个整数，顺次代表一个排列，注意是排列中的元素是从 0 开始到  $n-1$  的。

### 【输出文件】

只有一行，输出两个整数，第一个整数代表给定排列的有序数，第二个整数代表它所在的有序同类集中的位置。

### 【样例】

| permutation.in | permutation.out |
|----------------|-----------------|
| 4<br>2 0 3 1   | 0 3             |
| 4<br>0 1 3 2   | 2 0             |

### 【数据规模】

对于 40% 的数据， $n \leq 10$ 。

对于 100% 的数据， $n \leq 64$ 。