小熊的果篮（fruit）

【题目描述】

小熊的水果店里摆放着一排 n 个水果。每个水果只可能是苹果或桔子，从左到右依次用正整数 1、2、3、……、n 编号。连续排在一起的同一种水果称为一个“块”。小熊要把这一排水果挑到若干个果篮里，具体方法是：每次都把每一个“块”中最左边的水果同时挑出，组成一个果篮。重复这一操作，直至水果用完。注意，每次挑完一个果篮后，“块”可能会发生变化。比如两个苹果“块”之间的唯一桔子被挑走后，两个苹果“块”就变成了一个“块”。请帮小熊计算每个果篮里包含的水果。

【输入格式】

从文件 ***fruit.in*** 中读入数据。

输入的第一行包含一个正整数 n，表示水果的数量。

输入的第二行包含 n 个空格分隔的整数，其中第 i 个数表示编号为 i 的水果的种类，1 代表苹果，0 代表桔子。

【输出格式】

输出到文件 ***fruit.out*** 中。

输出若干行。

第 i 行表示第 i 次挑出的水果组成的果篮。从小到大排序输出该果篮中所有水果的编号，每两个编号之间用一个空格分隔。

【样例 1 输入】

```
12
1 1 0 0 1 1 1 0 1 1 0 0
```

【样例 1 输出】

```
1 3 5 8 9 11
2 4 6 12
7
10
```

【样例 1 解释】

这是第一组数据的样例说明。
所有水果一开始的情况是 `1 1 0 0 1 1 1 0 1 1 0 0`，一共有 6 个块。
在第一次挑水果组成果篮的过程中，编号为 1 3 5 8 9 11 的水果被挑了出来。
之后剩下的水果是 `1 0 1 1 1 0`，一共 4 个块。
在第二次挑水果组成果篮的过程中，编号为 2 4 6 12 的水果被挑了出来。
之后剩下的水果是 `1 1`，只有 1 个块。
在第三次挑水果组成果篮的过程中，编号为 7 的水果被挑了出来。
最后剩下的水果是 `1`，只有 1 个块。
在第四次挑水果组成果篮的过程中，编号为 10 的水果被挑了出来。

【样例 2 输入】

```
20
1 1 1 1 0 0 0 1 1 1 0 0 1 0 1 1 0 0 0 0
```

【样例 2 输出】

```
1 5 8 11 13 14 15 17
2 6 9 12 16 18
3 7 10 19
4 20
```

【样例 3】

见选手目录下的 ***fruit/fruit3.in*** 与 ***fruit/fruit3.ans***。

【数据范围】

对于 10% 的数据，$n \leq 5$。
对于 30% 的数据，$n \leq 1000$。
对于 70% 的数据，$n \leq 50000$。
对于 100% 的数据，$n \leq 2 \times 10^5$。

【提示】

由于数据规模较大，建议 C/C++ 选手使用 scanf 和 printf 语句输入、输出。