

Problem C. pm(pm.c/cpp/pas)

Input file: pm.in
 Output file: pm.out
 Time limit: 1 second
 Memory limit: 1 gigabyte

有一个 $\{1, 2 \dots n\}$ 的排列 $\{a_1, a_2 \dots a_n\}$ ，你需要把它变成 $\{1, 2 \dots n\}$ 。

你可以进行两个阶段的操作。第一阶段中，你可以重复若干次，每次任选两个相邻元素并进行交换。第二阶段中，你可以重复若干次，每次修改一个位置上的元素。第二阶段中，你可以任意修改元素，即使中间过程中序列不再是排列仍然可行。

你需要最小化两个阶段操作的总次数，并给出一种总操作次数最小的合法方案。为了简单起见，你只需要输出第一阶段的操作，SPJ 会自动计算第二阶段的最优操作，并将操作总次数和最优总次数比较。

Input

第一行一个正整数 n 。

第二行 n 个正整数 $a_1, a_2 \dots a_n$ 。

Output

输出第一行为第一阶段的操作数量 u 。

第二行输出 u 个以空格分隔的正整数 $p_1, p_2 \dots p_u$ ($1 \leq p_i < n$)，表示依次进行如下交换： $(a_{p_1}, a_{p_1 + 1}), (a_{p_2}, a_{p_2 + 1}) \dots (a_{p_u}, a_{p_u + 1})$ 。

Examples

pm.in	pm.out
4	1
4 3 2 1	2

第一阶段交换了中间的两个元素，第二阶段修改第一和第四个元素，总共需要 3 次操作。可以证明这是最优的。

Notes

Subtask 1 (10pts): $n \leq 5$ 。

Subtask 2 (10pts): $n \leq 9$ 。

Subtask 3 (10pts): $n \leq 50$ 。

Subtask 4 (20pts): $n \leq 500$ 。

Subtask 5 (10pts): $n \leq 2000$ 。

Subtask 6 (20pts): $n \leq 10^5$ 。

Subtask 7 (20pts): $n \leq 2 \times 10^5$ 。