

Sursa : perm.pas, perm.cpp, perm.c

Problema 2 - Perm

100 puncte

Considerând K un număr natural, vom numi *permutare de mărime K* o aranjare într-o ordine oarecare a elementelor mulțimii $\{1, 2, \dots, K\}$. Ana găsește scris pe o foaie de hârtie, un șir de numere naturale $\mathbf{v} = (v_1, v_2, v_3, \dots, v_N)$

Plecând de la acest șir, Ana numește *secvență* a lui \mathbf{v} , un subșir de numere care apar pe poziții consecutive în șirul inițial. De exemplu, șirul $5\ 7\ 8\ 9\ 1\ 6$ conține secvența $8\ 9\ 1$, secvența $7\ 8\ 9\ 1\ 6$, dar nu conține secvența $8\ 9\ 6$.

Anei îi vine ideea să încerce să verifice dacă șirul de numere poate fi împărțit în secvențe care reprezintă permutări de diferite mărimi.

De exemplu, dacă Ana găsește șirul $\mathbf{v} = (2, 1, 4, 1, 3, 2)$, atunci ea îl poate împărți în două secvențe de permutări astfel: secvența $\mathbf{v}[1], \mathbf{v}[2]$ respectiv secvența $\mathbf{v}[3], \mathbf{v}[4], \mathbf{v}[5], \mathbf{v}[6]$.

Dacă Ana găsește șirul $\mathbf{v} = (1, 2, 2, 3)$ atunci nu va putea obține o împărțire în secvențe de permutări.

Cerințe

Realizați un program care citește un șir de N numere naturale și rezolvă următoarea cerință:

- determină o împărțire a acestuia în secvențe de permutări. Dacă există mai multe soluții se va afișa soluția **minim** lexicografică.

Dacă șirul poate fi împărțit în secvențe de permutări, pentru fiecare număr din șir, în ordinea v_1, \dots, v_N , se va afișa numărul permutării din care face parte. Numerotarea permutărilor se va face consecutiv începând cu numărul 1.

Date de intrare:

Fișierul de intrare **perm.in** conține pe prima linie numărul natural N , iar pe a doua linie, un șir format din N numere naturale separate prin câte un spațiu.

Date de ieșire:

Fișierul de ieșire **perm.out** conține pe prima linie mesajul **NU** dacă șirul nu se poate împărți în secvențe de permutări. Dacă împărțirea se poate efectua atunci pe prima linie se va scrie numărul secvențelor de permutări obținut, iar pe a doua linie un șir de numere reprezentând împărțirea **minim** lexicografică determinată. Numerele pe linia a doua vor fi separate prin câte un spațiu.

Restricții și precizări

- $1 \leq N \leq 100\ 000$
- $1 \leq v_i \leq 20\ 000$
- Pentru două șiruri de N numere a_1, a_2, \dots, a_N și b_1, b_2, \dots, b_N spunem ca șirul \mathbf{a} este **mai mic lexicografic** decât șirul \mathbf{b} , dacă există un indice j , cu $1 \leq j < N$, astfel încât $a_1 = b_1, a_2 = b_2, \dots, a_j = b_j$ și $a_{j+1} < b_{j+1}$

Exemplu

perm.in	perm.out	Explicație
10 2 1 3 4 1 1 2 3 4 5	3 1 1 1 1 2 3 3 3 3 3	Prima secvență care reprezintă o permutare e formată din primele 4 elemente, a doua din elementul de pe poziția 5, iar a treia din ultimele 5 elemente
4 1 2 2 3	NU	

Timp maxim de execuție/test: 0.3 sec/test

Memorie totală: 4MB

Dimensiunea maximă a sursei: 15 KB