**Problema anagrame 100 puncte**

Se consideră două numere naturale nenule **N** și **K** **(K** ≤ **N).** Un șir de **N** numere naturale conține numere de la **1** la **K.** Fiecare valoarede la **1** la **K** apare în șir cel puțin o dată. Se cunosc elementele șirului, dar nu și ordinea acestora.

**Cerință**

Se cere să reconstituiți șirul folosind anumite informații despre acesta.

Va trebui să implementați o funcție

**void Solve(int NrTest, int N, int \*S)**

Graderul comisiei va apela funcția o singură dată și va transmite prin intermediul parametrilor următoarele informații (vezi tabelul de restricții de mai jos):

**NrTest** = numărul testului

**N** = numărul de elemente din șir

**S** = un șir conținând cele **N** elemente ale șirului căutat în ordine crescătoare.

În funcția **Solve()** veți apela următoarele funcții puse la dispoziție de graderul comisiei.

**int Ask(int \*X)**

transmite prin parametrul **X** un șir conținând elementele șirului căutat într-o ordine aleasă de voi și returnează numărul minim de **swap**-uri între elemente de pe poziții consecutive care pot fi aplicate pe șirul **X** pentru a-l transforma în șirul cerut.

Această funcție poate fi apelată de un număr limitat de ori (conform tabelului de restricții).

**int GiveSolution(int \*X)**

transmite prin parametrul **X** șirul căutat.

Această funcție va fi apelată o singură dată la final, după toate apelurile funcției **Ask()**.

**Exemplu**

|  |  |
| --- | --- |
| **Acțiune** | **Explicaţie** |
| Solve(0,4,{1,2,3,4}) | Șirul care trebuie reconstituit este {3,2,1,4} |
| Ask({1,2,3,4}) | Returnează 3   * swap (2,3) – {1,3,2,4} * swap (1,3) – {3,1,2,4} * swap (1,2) – {3,2,1,4} |
| Ask({2,4,3,1}) | Returnează 3   * swap (4,3) – {2,3,4,1} * swap (2,3) – {3,2,4,1} * swap (4,1) – {3,2,1,4} |
| GiveSolution ({3,2,1,4}) | Șirul reconstituit este {3,2,1,4} |

**Structura testelor**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NrTest | N | Nr. max. apeluri | Observații | Punctaj |
| 0 | 22 | 24 | exemplu | 0 |
| 1 | 24 | 28 | șirul e o permutare | 7 |
| 2 | 25 | 210 |
| 3 | 25 | 25 \* 5 \* 2 | șirul e o permutare | 6 |
| 4 | 26 | 26 \* 6 \* 2 |
| 5 | 26 | 26 \* 6 | șirul e o permutare | 14 |
| 6 | 27 | 27 \* 7 |
| 7 | 29 | 29 \* 2 | șirul conține doar elementele 1 și 2 | 11 |
| 8 | 210 | 210 \* 2 |
| 9 | 29 | 29 \* 2 | șirul e o permutare | 19 |
| 10 | 210 | 210 \* 2 |
| 11 | 29 | 29 \* 2 |  | 18 |
| 12 | 210 | 210 \* 2 |
| 13 | 29 | 29 | șirul e o permutare | 11 |
| 14 | 210 | 210 |
| 15 | 29 | 29 |  | 14 |
| 16 | 210 | 210 |

**Observații referitoare la verdictele de evaluare**

Dacă numărul de apeluri depășește valoarea maximă alocată pe test, puteți primi unul dintre următoarele verdicte:

* 400 (max 100) apeluri ale funcției Ask
* > 500 (max 100) apeluri ale funcției Ask

**Verdictele de mai sus nu garantează corectitudinea răspunsului găsit**

**Limită de timp: 0,5 secunde**

**Memorie totală disponibilă: 512 MB**