



Sursa : risc.pas, risc.cpp, risc.c

## Problema 2 - risc

100 puncte

Pentru a participa la un concert,  $n$  persoane s-au așezat la coadă pe un singur rând în așteptarea deschiderii casei de bilete. Înălțimile celor  $n$  persoane sunt toate distincte. Pe baza acestei informații cruciale, agenții de securitate au decis ca din motive de ... securitate, ordinea persoanelor care așteaptă la coadă trebuie schimbată în funcție de înălțimile lor.

Astfel, agenții de pază vor alege, pe rând, câte o persoană și o vor trimite la sfârșitul rândului. După fiecare operație de tipul acesta, să-i spunem “de mutare”, rândul se restrânge, ocupându-se poziția rămasă liberă. Strategia agenților de pază este aceasta: la terminarea tuturor operațiilor de mutare, *riscul minim de securitate* se obține dacă toate persoanele aflate în dreapta persoanei celei mai înalte vor fi mai înalte decât cele aflate în stânga persoanei celei mai înalte. În plus, înălțimile persoanelor vor fi crescătoare până la poziția  $k$  a persoanei celei mai înalte și descrescătoare după poziția  $k$ .

Mai exact: dacă  $h_1, h_2, \dots, h_n$  sunt înălțimile persoanelor după finalizarea operațiilor de mutare, atunci: există o poziție  $k$ , cu  $1 \leq k \leq n$  astfel încât  $h_1 < h_2 < \dots < h_{k-1} < h_k > h_{k+1} > \dots > h_{n-1} > h_n$  și în plus  $h_i < h_j$  pentru oricare  $i < k$  și  $k < j$ .

Deoarece o asemenea logică este greu de combătut, iar agenții nu au aerul că vor să glumească, persoanele care așteaptă la coadă vor accepta toate mutările impuse de către aceștia.

### Cerințe

Cunoscând numărul de persoane  $n$  și înălțimile  $h_1, h_2, \dots, h_n$  ale acestora să se scrie un program care determină :

1. Poziția persoanei celei mai înalte în rândul inițial, în cazul în care nu sunt necesare operații de mutare.
2. Numărul **minim** de mutări necesare pentru ca rândul de persoane să prezinte un *risc minim de securitate*.

### Date de intrare

Pe prima linie a fișierului de intrare **risc.in** se găsește numărul natural  $p$  a cărui valoare poate fi doar **1** sau **2**.

Pe a doua linie a fișierului de intrare se află numărul natural  $n$  cu semnificația din enunț.

Pe a treia linie se găsesc  $n$  numere naturale, distincte:  $h_1, h_2, \dots, h_n$ , separate prin câte un singur spațiu, reprezentând înălțimile persoanelor.

### Date de ieșire

Fișierul de ieșire este **risc.out**.

- Dacă valoarea lui  $p$  este **1** atunci **se va rezolva numai cerința 1**. În acest caz, fișierul de ieșire va conține pe prima linie un număr natural **poz** ce reprezintă poziția persoanei celei mai înalte în rândul inițial. Dacă rândul inițial nu respectă condițiile de *risc minim de securitate*, atunci **poz** este **-1**.
- Dacă valoarea lui  $p$  este **2** **se va rezolva numai cerința 2**. În acest caz fișierul de ieșire va conține pe prima linie **un număr** natural  $m$ , reprezentând numărul minim de mutări necesare pentru a obține *risc minim de securitate*.



Sursa : risc.pas, risc.cpp, risc.c

### Restricții și precizări

- $1 \leq n \leq 100\,000$
- $1 \leq h_1, h_2, \dots, h_n \leq 100\,000$
- Persoana cea mai înaltă într-o configurație cu *risc minim de securitate* poate fi plasată pe oricare dintre pozițiile  $1 \dots n$
- Pentru 50% din teste,  $n \leq 2000$ , iar pentru alte 40% din teste,  $n \leq 10\,000$
- Pentru rezolvarea corectă a primei cerințe se acordă 20 de puncte, iar pentru cerința a doua se acordă 80 de puncte.

### Exemple

risc.in	risc.out	Explicații
1 4 35 20 10 1	1	$p = 1$ , deci se rezolvă <b>numai</b> cerința 1. Rândul îndeplinește condițiile de <i>risc minim de securitate</i> . Persoana cea mai înaltă se găsește pe poziția $poz = 1$ .
1 6 1 8 12 20 15 10	-1	$p = 1$ , deci se rezolvă <b>numai</b> cerința 1. Rândul <b>NU</b> îndeplinește condițiile de <i>risc minim de securitate</i> . Șirul înălțimilor nu respectă condiția ca toate valorile înălțimilor din dreapta poziției persoanei celei mai înalte să fie mai mari decât toate valorile înălțimilor din stânga acesteia. Deci $poz = -1$ .
2 5 2 8 4 20 16	1	$p = 2$ , deci se rezolvă <b>numai</b> cerința 2. Se mută persoana de înălțime 8 la sfârșitul rândului. Deci $m = 1$
2 6 3 19 7 30 10 25	2	$p = 2$ , deci se rezolvă <b>numai</b> cerința 2. Prima operație: se mută persoana de înălțime 19 la sfârșitul rândului. Rândul devine: 3 7 30 10 25 19 A doua operație: se mută persoana de înălțime 10 la sfârșitul rândului. Rândul devine: 3 7 30 25 19 10 Deci $m = 2$

Timp maxim de execuție: Linux: 0,02 secunde/test;

Windows: 0,2 secunde/test;

Memorie totală disponibilă 4 MB

Dimensiunea maximă a sursei: 10 KB.