## Invsort — 100 puncte

**Fişier sursă: invsort.pas, invsort.c sau invsort.cpp**

Se dă un şir de `N` numere naturale, care trebuie ordonat crescător. Singura operație permisă este să considerați elementele de pe pozițiile `i`**,** `i+1`**, ...,** `j` (pentru `i` şi `j` arbitrare, `i<j`), şi să inversați ordinea acestor elemente (adică elementul de pe poziția `i` ajunge pe poziția `j`, `i+1` ajunge pe poziția `j-1`, ..., `j` ajunge pe poziția `i`). Costul unei astfel de operații este numărul de elemente din subşirul inversat, şi anume `j-i+1`.

### Cerință
Scrieți un program care să determine o secvență de operații care ordonează crescător şirul dat şi are un cost total cât mai mic (dar nu obligatoriu minim).

### Date de intrare
Fişierul de intrare `invsort.in` conține pe prima linie numărul `N`, şi apoi `N` linii cu elementele şirului dat (nu neapărat distincte).

### Date de ieşire
Fişierul de ieşire `invsort.out` va conține pe fiecare linie descrierea unei operații. O operație este descrisă prin numerele `i` şi `j`, separate printr-un spațiu. Aceste numere satisfac `1 <= i < j <= N`.

### Restricții

- `2 <= N <= 32000`
- valorile şirului care trebuie ordonat sunt între `0` şi `31999`

### Punctaj
- dacă şirul de operații (executate în ordinea din fişierul de ieşire) nu ordonează intrarea, primiți 0 puncte
- în cazul în care costul total este cel mult `4000000` (patru milioane) primiți punctaj maxim
- în cazul în care costul total este cel mult 40000000 (patruzeci de milioane) primiți 40% din punctajul pe test
- în 50% din teste şirul de intrare conține numai elemente de `0` şi `1`
- pentru toate testele folosite la corectare, `N=32000`

### Exemplu

| invsort.in | invsort.out | *Explicație* |
|---|---|---|
| 5<br>1<br>0<br>1<br>1<br>0 | 3 5<br>1 3 | • prima operație are efectul: 1 0 [1 1 0] → 1 0 0 1 1<br>• a doua operație are efectul: [1 0 0] 1 1 → 0 0 1 1 1<br>• costul total este 3 + 3 = 6 |

**Timp maxim de execuție/test: 0.5 secunde pentru Linux şi 2 secunde pentru Windows.**

# Peri

**100 puncte**

**Fişier sursă: peri.pas, peri.c sau peri.cpp**

Se consideră o matrice dreptunghiulară `A` cu `m` linii şi `n` coloane cu valori `0` sau `1`, liniile şi coloanele fiind numerotate de la `1` la `m`, respectiv de la `1` la `n`. Numim dreptunghi de colțuri (`x1,y1`) (`x2,y2`) cu `x1<x2` şi `y1<y2` mulțimea elementelor $A_{ij}$ cu `x1≤i≤x2` si `y1≤j≤y2`. Numim perimetru al dreptunghiului de colțuri (`x1,y1`) (`x2,y2`) mulțimea elementelor $A_{ij}$ pentru care (`i=x1` şi `y1≤j≤y2`) sau (`i=x2` şi `y1≤j≤y2`) sau (`j=y1` şi `x1≤i≤x2`) sau (`j=y2` şi `x1≤i≤x2`).

## Cerință

Determinați diferența maximă dintre numărul de elemente egale cu `1` şi numărul de elemente egale cu `0` aflate pe perimetrul aceluiaşi dreptunghi, precum şi numărul de dreptunghiuri pentru care se obține această diferență.

## Date de intrare

Pe prima linie a fişierului de intrare **`peri.in`** sunt scrise numerele `m` şi `n`, separate printr-un singur spațiu. Pe următoarele `m` linii este dată matricea `A`, numerele de pe aceeaşi linie fiind separate de câte un spațiu.

## Date de ieşire

Fişierul de ieşire **`peri.out`** va conține o singură linie pe care se află două numere întregi separate printr-un spațiu. Primul număr este diferența maximă dintre numărul de elemente `1` şi numărul de elemente `0` de pe perimetrul unui dreptunghi. Al doilea întreg este numărul de dreptunghiuri pentru care diferența dintre numărul de elemente `1` şi numărul de elemente `0` de pe perimetru este maximă.

## Restricții

`1 < m,n ≤ 250`
Prin diferență nu se înțelege diferență în valoare absolută!

## Exemplu

| peri.in | peri.out |
|---|---|
| 4 5<br>1 0 0 1 0<br>0 1 1 0 0<br>0 1 0 1 0<br>1 1 1 0 1 | 4 2 |

**Timp maxim de execuție/test: 0.2 secunde pentru Linux şi 1 secundă pentru Windows.**

# Trans — 100 puncte

**Fişier sursă: trans.pas, trans.c sau trans.cpp**

În depozitul unei companii de construcții se află N blocuri de piatră, de culoare albă sau neagră. Ele sunt aşezate în ordinea 1,2,..,N, de la intrarea în depozit către interior.
Blocurile de piatră trebuie să fie transportate pe un şantier de construcții, în ordinea în care ele sunt depozitate, iar pentru aceasta va trebui închiriat un camion de la o companie de transport. Aceasta deține Q tipuri de camioane. Camionul de tipul i (1 <= i <= Q) poate transporta maxim $K_i$ blocuri de piatră la un moment dat şi pentru un transport se percepe taxa $T_i$.
Compania de transport este de parere că, pentru a-şi păstra clientela, trebuie să impună anumite standarde, indiferent de cât de absurde ar fi, deci impune condiția ca toate blocurile de piatră plasate în camion la un transport să aibă aceeaşi culoare. Aşadar, pentru a fi transportate toate blocurile pe şantier, compania de construcții va alege un camion de un anumit tip, iar camionul va efectua unul sau mai multe transporturi.
Pentru a micşora suma totală plătită, compania de construcții are posibilitatea de a schimba culoarea oricărui bloc de piatră (din alb în negru sau din negru în alb); pentru fiecare bloc i (1 <= i <= N) se cunoaşte suma $S_i$ ce trebuie plătită de a-i schimba culoarea.

**Cerință**

Pentru fiecare dintre cele Q tipuri de camioane deținute de compania de transport, determinați suma minimă pe care o va plăti compania de construcții pentru a transporta toate cele N blocuri pe şantier.

**Date de intrare**

Fişierul de intrare trans.in conține pe prima linie numărul întreg N, reprezentând numărul de blocuri de piatră din depozit. Pe fiecare dintre următoarele N linii se află informații referitoare la câte un bloc de piatra. Pe a i-a dintre aceste N linii se găsesc două numere întregi separate printr-un spațiu: Ci Si, reprezentând culoarea celui de-al i-lea bloc (Ci este 0 pentru alb şi 1 pentru negru) şi respectiv suma ce trebuie plătită pentru a-i schimba culoarea (dacă este necesar). Pe următoarea linie se află numărul natural Q, reprezentând numărul de tipuri de camioane deținute de compania de transport. Pe fiecare dintre următoarele Q linii se află informații referitoare la câte un camion. Pe cea de a i-a dintre aceste Q linii sunt scrise două numere naturale separate printr-un spațiu Ki Ti, reprezentând numărul maxim de blocuri ce pot fi transportate simultan de către un camion de tipul i şi respectiv taxa ce trebuie plătită pentru fiecare transport efectuat.

**Date de ieşire**

Fişierul de ieşire trans.out va conține Q linii. Pe cea de a i-a dintre aceste linii va fi afişată suma totală minimă plătită de compania de construcții pentru a transporta toate cele N blocuri de piatră, în cazul în care ar închiria un camion de tipul i.

**Restricții şi precizări**

- **1 <= N <= 16000**
- **1 <= Si <= 10000**
- **1 <= Q <= 100**
- **1 <= Ki <= N**
- **1 <= Ti <= 100000**
- Cel putin **40%** dintre teste vor avea **Q <= 10** şi **Ki <= min(N, 100)**

**Exemplu**

| trans.in | trans.out |
|---|---|
| 4<br>0 2<br>1 3<br>0 10<br>1 2<br>3<br>4 1000<br>4 1<br>2 5 | 1005<br>4<br>14 |

**Timp maxim de execuție/test: 0.3 secunde pentru Linux şi 1 secundă pentru Windows.**