Rzeszów, iulie 2004

OLIMPIADA de Informatică a EUROPEI CENTRALE 2004

Vă prezentăm în continuare enunțurile celor șase probleme propuse spre rezolvare la cea de-a unsprezecea ediție a acestui concurs care este considerat, pe bună dreptate, ca fiind una dintre cele mai puternice competiții destinate elevilor informaticieni.

P040607: Nori

Pe cer sunt n nori care se deplasează în aceeași direcție, cu o viteză constantă $v = (v_x, v_y)$. Așadar, pentru orice număr real pozitiv t și orice punct al unui nor ale cărui coordonate inițiale sunt (x, y), poziția punctului respectiv la momentul t este $(x + t \cdot v_x, y + t \cdot v_y)$.

Pentru simplitate, vom presupune că fiecare nor este reprezentat de un poligon (norul este dat de interiorul și de frontiera poligonului), ale cărui vârfuri au coordonatele întregi.

Poligoanele nu trebuie să fie neapărat convexe, dar oricare două laturi nu se intersectează decât, eventual, extremitățile lor în vârfurile poligonului. Poligoanele care descriu norii pot avea puncte comune.

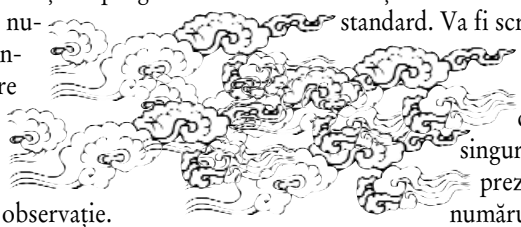
Pe pământ (în punctul de origine al axelor de coordonate) se află un centru de control al sateliților și exact deasupra centrului (și a norilor) se află un satelit.

De la centrul de control este trimisă o rază laser verticală pentru a se realiza comunicarea cu satelitul. În cazul în care raza laser intersectează un nor, comunicarea cu satelitul este întreruptă.

Se știe că la momentul 0, raza laser, nu intersectează nici un nor. În timpul transmisiilor, pot exista mai multe momente când raza laser intersectează norii (unul sau mai mulți) întrerupându-se comunicarea cu satelitul.

Comunicarea este întreruptă chiar dacă raza laser intersectează un vârf sau o latură a unui poligon care reprezintă un nod.

Trebuie să scrieți un program care determină numărul total al întreruperilor care au loc până în momentul în care norii părăsesc zona de observație.



Date de intrare

Datele se citesc de la intrarea standard.

Pe prima linie se vor afla trei numere întregi n, v_x și v_y, separate prin câte un spațiu, reprezentând numărul norilor, viteza norilor pe axa orizontală, respectiv viteza norilor pe axa verticală.

Următoarele n linii conțin date despre nori. Fiecare dintre aceste linii descrie câte un nor și conține un șir de numere întregi, separate prin câte un singur spațiu. Primul număr întreg k, de pe o astfel de linie reprezintă numărul vârfurilor care reprezintă norul corespunzător liniei. Urmează k perechi de numere întregi $(x_1, y_1, x_2, y_2, \ldots, x_k, y_k)$ care reprezintă coordonatele celor k vârfuri ale poligonului. Aceste vârfuri sunt descrise în ordinea dată de sensul arcelor de ceasornic.

Date de ieșire

Datele de ieșire vor fi scrise la ieșirea standard. Va fi scrisă o singură linie care va conține un singur număr, reprezentând numărul întreruperilor de comunicare.

Restricții

Numărul norilor este cuprins între 1 și 1000.

Valorile vitezelor pe cele două axe de coordonate sunt numere întregi cuprinse între -1.000.000.000 și 1.000.000.000 și nu pot fi ambele nule.

Numărul vârfurilor unui poligon care descrie un nor este cuprins între 3 și 1000.

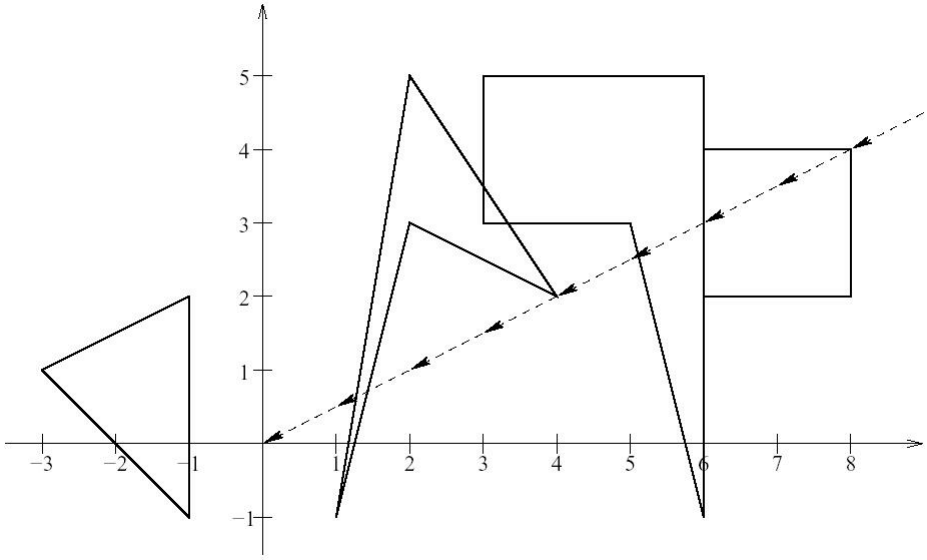


Figura 1

Coordonatele punctelor care reprezintă vârfurile poligoanelor corespunzătoare norilor sunt numere întregi cuprinse între -1.000.000.000 și 1.000.000.000.

Abscisele corespund direcției *Vest - Est*, iar ordonatele corespund direcției *Nord - Sud*.

Exemplu

Intrarea standard

```
4 -2 -1
4 6 2 6 4 8 4 8 2
4 2 3 1 -1 2 5 4 2
3 -3 1 -1 -2 -1 -1
5 5 3 3 3 3 5 6 5 6 -1
```

Ieșirea standard

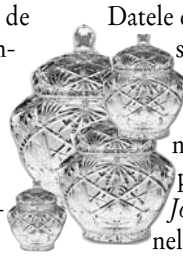
```
3
```

Configurația din acest exemplu este prezentată în figura 1.

Timp de execuție: 3 secunde/test
Memorie disponibilă: 64 MB

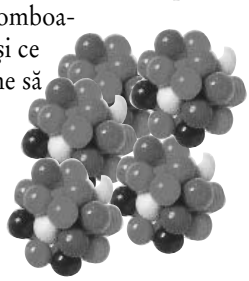
P040608: Bomboane

John are *n* bomboniere în care își ține bomboanele. Fiecare dintre bomboniere conține un anumit tip de bomboane, dar nici o altă bombonieră nu conține bomboane de acel tip). Se cunoaște numărul de bomboane din fiecare bombonieră.



John s-a hotărât să mănânce o parte dintre bomboanele sale. El ar vrea să mănânce cel puțin *a* bomboane, dar nu mai mult decât *b* bomboane.

Problema este că *John* nu se poate hotărî câte bomboane să mănânce și ce tip de bomboane să aleagă.



În câte moduri poate el să facă acest lucru?

Date de intrare

Datele se citesc de la intrarea standard.

Pe prima linie se vor afla trei numere întregi *n*, *a* și *b*, separate prin câte un spațiu, reprezentând numărul bomboniere lor, numărul minim de bomboane care trebuie alese, respectiv numărul maxim de bomboane care trebuie alese.

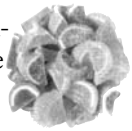
Fiecare din următoarele *n* linii conține un număr întreg care reprezintă numărul de bomboane dintr-o anumită bombonieră.

Date de ieșire

Datele de ieșire vor fi scrise la ieșirea standard.

Va fi scrisă o singură linie care va conține un singur număr, reprezentând numărul posibilităților pe care le are *John* pentru a alege bomboanele.

Restricții



Numărul bomboniere lor este cuprins între 1 și 10.

Numărul minim, precum și numărul maxim de bomboane pot varia între 0 și 10.000.000.

În fiecare bombonieră se pot afla între 0 și 1.000.000 de bomboane.

Exemplu

Intrarea standard

```
2 1 3
3
5
```

Ieșirea standard

```
9
```

Explicație

Cele nouă posibilități de a alege sunt:

Timp de execuție: 0.1 secunde/test
Memorie disponibilă: 16 MB

P040609: Excursii

În vacanța care urmează, o mulțime de oameni vor să plece în excursii de neuitat.

Pentru ca vacanța să fie cât mai plăcută, fiecare vrea să meargă însoțit de un grup de prieteni.

O agenție de turism oferă diferite excursii. Fiecare excursie se adresează unui grup de persoane, care tre-

buie să se încadreze între un număr minim și un număr maxim de participanti.

Fiecare grup poate să aleagă o singură excursie. În plus, fiecare excursie poate fi aleasă de un singur grup.

Agenția de turism dorește să organizeze cât mai multe excursii.

Sarcina voastră este să alegeți câte o excursie pentru fiecare grup, astfel încât să se organizeze un număr cât mai mare de excursii.

Date de intrare

Datele se citesc de la intrarea standard.

Prima linie conține două numere întregi *n* și *m* separate printr-un spațiu, unde *n* este numărul de grupuri și *m* este numărul de excursii. Grupurile sunt numerotate de la 1 la *n*, iar excursiile de la 1 la *m*.

Următoarele *n* linii conțin dimensiunile grupurilor. Fiecare linie conține un singur număr întreg, iar ordinea liniilor respectă numerele de ordine ale grupurilor.

Următoarele *m* linii conțin descrierile excursiilor. Fiecare linie conține două numere întregi *l* și *u*, separate printr-un singur spațiu, unde *l* reprezintă dimensiunea minimă a unui grup care poate merge în excursia respectivă, iar *u* reprezintă dimensiunea maximă a unui astfel de grup. Ordinea liniilor respectă numerele de ordine ale excursiilor.

Date de ieșire

Datele de ieșire vor fi scrise la ieșirea standard.

Prima linie va conține un număr întreg *k* care reprezintă numărul maxim al excursiilor care pot fi organizate.

Fiecare dintre următoarele *k* linii vor conține câte o pereche de numere întregi, separate printr-un singur spațiu. Primul dintre ele va fi numărul de ordine al unui grup, iar al doilea va fi numărul de ordine al unei excursii. Membrii grupului specificat vor alege excursia aleasă.

În cazul în care există mai multe soluții, poate fi aleasă oricare dintre ele.

Restricții

Numărul grupurilor și numărul excursiilor sunt cuprinse între 1 și 400.000.

Numărul minim, precum și numărul maxim de membri ai unui grup pot varia între 1 și 1.000.000.000.

Din fiecare grup pot face parte între 1 și 1.000.000.000 de persoane.

Exemplu

Intrarea standard

```
5 4
54
6
9
42
15
6 6
20 50
2 8
7 20
```

Ieșirea standard

```
3
2 1
3 4
4 2
```

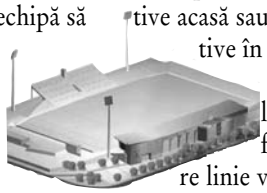
Timp de execuție: 3 secunde/test
Memorie disponibilă: 16 MB

P040610: Fotbal

Într-un campionat de fotbal sunt înscrise un număr de *n* echipe (*n* este număr par). Echipele sunt identificate prin numere întregi cuprinse între 1 și *n*.

Pe parcursul unui sezon fiecare echipă joacă cu fiecare dintre celelalte echipe, exact o dată. Așadar, sezonul constă din *n* - 1 etape.

Este de dorit ca fiecare echipă să joace meciurile consecutive pe stadioane diferite, adică un meci pe stadionul propriu, meciul următor în deplasare, apoi din nou pe stadionul propriu și așa mai departe.



Din nefericire nu este posibilă o planificare, astfel încât să nu existe echipe care joacă două meciuri consecutive acasă sau două meciuri consecutive în deplasare. Când se face o planificare a meciurilor dintr-un sezon, numărul acestor situații trebuie minimizat.

De exemplu, dacă o echipă joacă în deplasare, apoi de patru ori acasă, apoi în deplasare, se consideră că a jucat de trei ori două meciuri consecutive numai acasă sau numai în deplasare).

Programul vostru trebuie să determine numărul minim al situațiilor în care o echipă joacă două meciuri consecutive acasă sau două meciuri consecutive în deplasare și să construiască o planificare a meciurilor sezonului astfel încât să fie respectat acest număr minim.

Sezonul constă în *n* - 1 etape. O etapă constă în *n* / 2 partide, fiecare echipă jucând exact un meci într-o etapă.

Pe parcursul întregului sezon se vor desfășura *n* · (*n* - 1) / 2 partide. Fiecare dintre acestea se joacă pe unul dintre stadioanele celor două echipe participante (nu se joacă pe stadioane neutre).

Date de intrare

Datele se citesc de la intrarea standard.

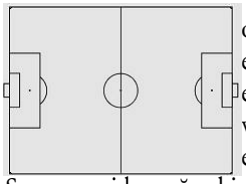
Intrarea constă într-o singură linie pe care se va afla un singur număr, reprezentând numărul echipelor participante.

Date de ieșire

Datele de ieșire vor fi scrise la ieșirea standard.

Pe prima linie va fi scris un singur număr întreg care va reprezenta numărul minim al situațiilor în care o echipă joacă două meciuri consecutive acasă sau două meciuri consecutive în deplasare.

Următoarele *n* - 1 linii vor conține o planificare a jocurilor. Fiecare linie va corespunde unei etape, iar liniile vor fi scrise în ordinea desfășurării etapelor. O astfel de linie va conține toate cele *n* numere de



ordine ale echipelor (fiecare număr va apărea exact o dată). Se va considera că echipele aflate pe o poziție impară în cadrul liniei vor juca acasă cu echipa de pe poziția imediat următoare.

Restricție
Numărul echipelor care participă în campionat este cuprins între 2 și 1000.

Exemplu
Intrarea standard
```
4
```

Ieșirea standard
```
2
1 2 3 4
4 1 2 3
1 3 4 2
```

Timp de execuție: 3 secunde/test
Memorie disponibilă: 32 MB

P040611: Puzzle

Regele din *Byteland* a primit cadou un *puzzle*. Acesta are forma unui pătrat de latură n și conține n^2 celule pătratice, dispuse pe n linii și n coloane, fiecare celulă având latura egală cu unitatea.

Fiecare celulă conține o piesă cu un număr cuprins între 1 și n^2. Numerele pieselor din fiecare celulă a *puzzle*-ului sunt distincte.

Liniile și coloanele *puzzle*-ului sunt numerotate de la 1 la n; fiecare celulă va fi identificată printr-o pereche de numere întregi i și j, iar numărul din celula respectivă va fi notat prin $p(i, j)$.

O reprezentare schematică a *puzzle*-ului este prezentată în figura 2.

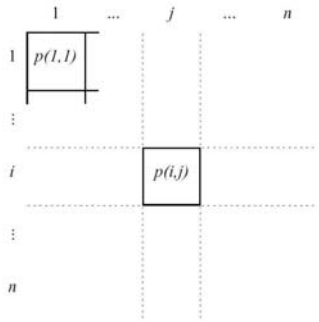


Figura 2

Pentru a rezolva *puzzle*-ul trebuie așezate piesele în ordine astfel încât pentru toate celulele *puzzle*-ului să avem:

$$p(i, j) = i \cdot n + j.$$

Mutările permise pentru rezolvarea *puzzle*-ului sunt:
- o permutare circulară a celulelor unei linii la dreapta cu un anumit număr de celule;
- O permutare circulară a celulelor unei coloane în jos cu un anumit număr de celule.

Regele din *Byteland* a reușit să rezolve *puzzle*-ul, dar el nu este sigur dacă l-ar putea rezolva pornind de la o configurație inițială diferită.

Ajutați regele să rezolve *puzzle*-ul pentru o configurație inițială specificată.

Date de intrare
Datele se citesc de la intrarea standard.

Pe prima linie se află un număr întreg n care reprezintă dimensiunea *puzzle*-ului.

Următoarele linii conțin descrierea configurației inițiale.

Fiecare dintre aceste linii conține câte n numere întregi, separate prin câte un spațiu, care reprezintă numerele de pe piesele de pe linia respectivă a configurației inițiale a *puzzle*-ului pe care l-a primit regele.

Date de ieșire
Datele de ieșire vor fi scrise la ieșirea standard.

Dacă nu există nici o soluție, se va scrie o singură linie conținând numai cuvântul `NO`.

Dacă există cel puțin o soluție, prima linie va conține un număr întreg m care reprezintă numărul de mutări care conduc la rezolvarea *puzzle*-ului.

Următoarele m linii vor conține descrierea mutărilor, câte o mutare pe o linie. Fiecare astfel de linie conține o literă `R` (pentru permutarea circulară a unei linii la dreapta) sau litera `C` (pentru permutarea circulară a unei coloane în jos) un spațiu și două numere întregi k și l, unde k reprezintă numărul de ordine al liniei sau coloanei care va fi permutată, iar l reprezintă numărul de poziții cu care este permutată aceasta (liniile sunt permutate spre dreapta, iar coloanele sunt permutate în jos).

Restricții
Dimensiunea *puzzle*-ului este un număr întreg cuprins între 2 și 200.

Numărul maxim de mutări permise este 400.000 indiferent de configurație.

Exemplu
Intrarea standard
```
4
4 6 2 3
5 10 7 8
9 14 11 12
13 1 15 16
```

Ieșirea standard
```
2
C 2 1
R 1 3
```

Explicație
În continuare este prezentată configurația inițială a *puzzle*-ului, precum și configurațiile obținute după fiecare

dintre mutările care trebuie efectuate:

4	6	2	3
5	10	7	8
9	14	11	12
13	1	15	16

4	1	2	3
5	6	7	8
9	10	11	12
13	14	15	16

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Timp de execuție: 0.3 secunde/test
Memorie disponibilă: 32 MB

P040612: Cherestea

De-a lungul unei șosele, care pornește din vârful unui deal și ajunge la poalele acestuia au fost plantați *n* copaci.

La un moment dat, administrația locală a decis să taie copacii de la marginea șoselei. Pentru a nu irosi lemn, s-a decis că, după tăiere, copacii vor fi prelucrați în fabrici de cherestea.

Pentru ca efortul necesar transportului să fie cât mai mic, copacii vor fi transportați doar de sus în jos (dinspre vârf spre poalele dealului).

La poalele dealului se află o fabrică de prelucrat cheresteaua. Pentru a micșora distanțele pe care ar trebui transportat lemnul, administrația locală a decis să se construiască două alte fabrici de cherestea de-a lungul șoselei.

Va trebui să determinați pozițiile celor două fabrici de cherestea astfel încât costul necesar pentru a transporta lemnul să fie cât mai mic posibil.

Evident, fiecare copac va fi transportat la cea mai apropiată fabrică de cherestea a cărei poziție este mai jos decât poziția copacului respectiv.

Costul transportului este de un eurocent pentru a transporta un kilogram de lemn pe o distanță de un metru.

Date de intrare

Datele se citesc de la intrarea standard.

Prima linie a fișierului de intrare conține un număr întreg *n* care reprezintă numărul copacilor plantați de-a lungul șoselei.

Cei *n* copaci vor fi identificați prin numere întregi cuprinse între 1 și *n*, începând cu copacul aflat cel mai aproape de vârful dealului.

Fiecare dintre următoarele *n* linii conține câte două numere întregi pozitive, separate prin câte un spațiu. Cele două numere indică, în această ordine, greutatea copacului exprimată în kilograme, respectiv distanța, exprimată în metri, dintre copacul respectiv și copacul următor (cel descris pe linia următoare).

Pentru ultimul copac (cel descris pe ultima linie), distanța va fi cea dintre copacul respectiv și fabrica de cherestea aflată la poalele dealului.

Date de ieșire

Datele de ieșire vor fi scrise la ieșirea standard.

Va fi scrisă o singură linie care va conține un singur număr, reprezentând costul total minim al transportului (exprimat în eurocenți) după ce vor fi construite cele două fabrici de cherestea.

Restricții

Numărul copacilor plantați la marginea șoselei este cuprins între 2 și 20.000.

Se garantează faptul că, în orice situație, costul total minim al transportului nu va depăși 2.000.000.000.

Exemple

Intrarea standard

```
9
1 2
2 1
3 3
1 1
3 2
1 6
2 1
1 2
1 1
```

Ieșirea standard

```
26
```

Explicație

În figura 3 sunt prezentate pozițiile optime ale fabricilor de cherestea pentru configurația specificată în exemplul anterior.

Copacii sunt reprezentați prin cercuri, iar greutățile lor sunt specificate sub cercurile respective.

Fabricile de cherestea sunt marcate cu negru și sunt poziționate în această figură în poziții în care se află copaci.

Rezultatul este:

$$1 \cdot (2 + 1) + 2 \cdot 1 + 1 \cdot (1 + 2) + 3 \cdot 2 + 2 \cdot (1 + 2 + 1) + 1 \cdot (2 + 1) + 1 \cdot 1 = 26$$

Timp de execuție: 1 secundă/test
Memorie disponibilă: 16 MB

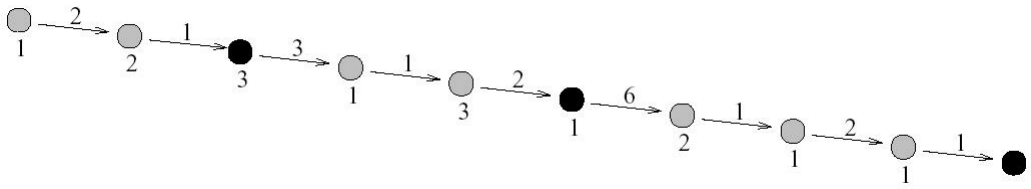


Figura 3