



# Rundele #41 - #50

În continuare vă vom prezenta enunțurile celor 10 probleme propuse spre rezolvare la semifinala ediției din acest an a concursului de programare Bursele Agora care a avut loc pe data de 21 martie.

## P040429: Minele de aur

Elmer Fud și Porky Pig sunt prieteni de foarte mult timp și au decis să meargă împreună în căutare de aur.

Ei au reușit să sape mai multe mine din care pot extrage aur în cantități foarte mari și s-au hotărât să se asocieze și să intre în afaceri cu aur. La un momet dat, când trebuirile mergeau bine, ei s-au certat și au decis să împartă minele și afacerile în două. A fost ușor să împartă afacerile, dar împărțitul minelor a devenit o problemă serioasă deoarece doresc să construiască un gard liniar și de fiecare parte a gardului să se găsească un număr egal de mine.

Elmer și Porky apelează la ajutorul vostru pentru a le spune unde să construiască gardul.

### Date de intrare

Fișierul de intrare **GOLDMINE.IN** conține pe prima linie un singur număr  $n$ , care reprezintă numărul de mine care trebuie să revină fiecăruia dintre cei doi.

Fiecare a  $i$ -a linie dintre următoarele  $2 \cdot n$  linii conține câte 2 numere, separate între ele printr-un singur spațiu, care reprezintă coordonatele unei mine.

### Date de ieșire

Fișierul de ieșire **GOLDMINE.OUT** trebuie să conțină pe o singură linie trei numere  $a$ ,  $b$  și  $c$  separate între ele prin spații, care reprezintă coeficienții drepte de ecuație  $a \cdot x + b \cdot y + c = 0$  care are de o parte sau pe ea  $n$  mine și de cealaltă parte sau pe ea celelalte  $n$  mine.

### Restricții și precizări

- $0 \leq n \leq 30.000$ ;
- coeficienții  $a$ ,  $b$  și  $c$  vor fi scriși cu opt zecimale exacte;
- pot exista două sau mai multe mine aflate la aceleași coordonate;
- coordonatele minelor sunt numere întregi cuprinse între 1 și 1000.

### Exemplu

**GOLDMINE.IN**

2

1 2

2 2

3 1

3 2

**GOLDMINE.OUT**

3.000000000 -2.000000000 -3.000000000

**Timp maxim de execuție/test:** 0,5 secunde

## P040430: Rivali

Două companii distribuie reviste într-un județ în care există un număr de  $n$  orașe și  $m$  șosele pe care se poate circula în ambele sensuri. Pentru fiecare șosea se cunoaște costul unui drum pe șoseaua respectivă. Fiecare companie dorește să pună la punct o rețea de distribuție care să-i permită să poată transporta reviste dintr-un oraș în oricare altul. Evident, fiecare companie va dori să obțină o rețea de distribuție al cărei cost total (suma costurilor șoselelor din rețea) să fie minim. Din nefericire, cele două companii rivale nu își pot permite să aibă rețele de distribuție identice. Ca urmare, ele vor alege rețele de distribuție puțin diferite. Va trebui să determinați diferența dintre costurile totale ale celor două rețele de distribuție știind că aceste două costuri sunt cele mai mici posibile și că rețelele nu pot fi complet identice.

### Date de intrare

Prima linie a fișierului de intrare **RIVALS.IN** conține numărul  $n$  al orașelor și numărul  $m$  al șoselelor. Fiecare dintre următoarele  $m$  linii conține trei numere întregi  $x$ ,  $y$  și  $c$  cu semnificația: există o șosea între orașele  $x$  și  $y$ , al cărei cost este  $c$ . Numerele de pe o linie vor fi separate prin spații.

### Date de ieșire

Fișierul de ieșire **RIVALS.OUT** trebuie să conțină o singură linie pe care se va afla diferența dintre costurile totale ale celor două rețele de distribuție.

### Restricții și precizări

- $2 \leq n \leq 500$ ,  $1 \leq m \leq 5000$ ;
- orașele sunt identificate prin numere cuprinse între 1 și  $n$ ;
- există posibilitatea ca diferența să fie 0 dacă cele două rețele de distribuție sunt diferite, dar au același cost total;

- costurile șoselelor sunt numere întregi cuprinse între 1 și 1000.

#### Exemplu

RIVALS . IN

4 5  
1 2 3  
1 3 5  
2 3 3  
2 4 5  
3 4 1

RIVALS . OUT

2

**Tim maxim de execuție/test:** 2 secunde

#### P040431: Cerc

*Pluto* se plictisea și a început să se joace. El are o foaie de hârtie cu pătrățele pe care o stropește cu cerneală și după aceea își notează pe o altă hârtie coordonatele centrelor petelor. *Pluto* dorește să traseze un cerc pe foaia pătată astfel încât în interiorul cercului să fie cel mult  $k$  dintre punctele ale căror coordonate le-a notat, iar în interior și pe frontieră (circumferință) să existe un număr de puncte mai mare sau egal cu  $k$ . Deoarece pe hârtie sunt foarte multe pete și *Pluto* dorește să rezolve astăzi problema determinării cercului, vă roagă să scrieți un program care, pe baza coordonatelor centrelor petelor să determine coordonatele centrului unui asemenea cerc și raza acestuia.

#### Date de intrare

Fișierul de intrare **CIRCLE . IN** conține pe prima linie două numere întregi  $n$  și  $k$ , separate între ele printr-un singur spațiu, care reprezintă numărul de pete, respectiv numărul de centre ale petelor care trebuie să fie luat în considerare la găsirea cercului. Pe fiecare dintre următoarele  $n$  linii se află câte două numere, separate printr-un singur spațiu, care reprezintă coordonatele centrului unei pete.

#### Date de ieșire

Fișierul de ieșire **CIRCLE . OUT** trebuie să conțină trei numere  $x$ ,  $y$  și  $r$ , separate prin spații, care reprezintă coordonatele centrului și raza cercului care are proprietatea din enunț.

#### Restricții și precizări

- $1 \leq k \leq n \leq 100.000$ ;
- coordonatele centrelor petelor sunt numere întregi cuprinse între 1 și 10.000;
- numerele din fișierul de ieșire vor fi scrise cu opt zecimale exacte.

#### Exemplu

CIRCLE . IN

4 3  
0 10  
20 10  
10 5  
10 15

CIRCLE . OUT

10.00000000 10.00000000 10.00000000

**Tim maxim de execuție/test:** 1 secundă

#### P040432: Parola

Într-un seif se află niște documente pe care trebuie să le extrageți. Problema este că seiful este prevăzut cu un terminal care necesită introducerea unei parole pentru a putea deschide seiful. La accesarea seifului, pe ecranul terminalului este afișat un cuvânt cheie format din litere mici ale alfabetului englezesc.

Parola este dată de cea mai mică rotație la stânga (în ordine lexicografică) a cuvântului cheie.

#### Date de intrare

Fișierul de intrare **PASSWORD . IN** conține pe prima linie un șir de caractere format din litere mici ale alfabetului englezesc.

#### Date de ieșire

Fișierul de ieșire **PASSWORD . OUT** trebuie să conțină un singur număr care reprezintă numărul de deplasări circulare la stânga ale șirului din fișierul de intrare necesare pentru a obține parola de acces cerută.

#### Restricții și precizări

- lungimea șirului din fișierul de intrare este un număr întreg cuprins între 1 și 100.000;
- dacă există mai multe soluții va fi aleasă cea care necesită un număr minim de deplasări circulare la stânga.

#### Exemplu

PASSWORD . IN

mississippi

PASSWORD . OUT

10

**Tim maxim de execuție/test:** 1 secundă

#### P040433: Coridorul

Se consideră un coridor dreptunghiular de dimensiuni  $m \cdot n$ . În acest coridor există mai multe coloane drepte de dimensiune neglijabilă.

Se cere să se determine raza celei mai mari mingi sferice care poate străbate coridorul din capătul de vest până în capătul de est. Pentru minge se poate alege orice punct de pornire aflat în capătul din vest și orice punct final aflat în capătul de est. Înălțimea coridorului este întotdeauna suficient de mare.

#### Date de intrare

Fișierul de intrare **HALLWAY . IN** conține două numere întregi  $m$  și  $n$ , separate între ele printr-un singur spațiu, care reprezintă dimensiunile coridorului pe direcțiile *est-vest*, respectiv *nord-sud*. Cea de-a doua linie conține un singur număr  $k$ , reprezentând numărul de coloane care se află în acest coridor.





Fiecare dintre următoarele  $k$  linii conține două numere, separate între ele printr-un singur spațiu, care reprezintă coordonatele la care se află o coloană.

### Date de ieșire

Fișierul de ieșire **HALLWAY.OUT** trebuie să conțină un singur număr care reprezintă raza celei mai mari mingi sferice care poate străbate coridorul din capătul de vest până în capătul de est.

### Restricții și precizări

- $1 \leq m, n, k \leq 1.000$ ;
- direcția *est-vest* este reprezentată de prima coordonată;
- raza determinată a mingii trebuie scrisă cu opt zecimale exacte.

### Exemplu

| HALLWAY . IN | HALLWAY . OUT |
|--------------|---------------|
| 5 2          | 0.50000000    |
| 1            |               |
| 1 1          |               |

**Timp maxim de execuție/test:** 1 secundă

### P040434: Ordinea

Se consideră  $n$  copii așezați în cerc și numerotați de la 1 la  $n$  în sens trigonometric. Copiii joacă următorul joc: jocul începe de la primul copil (cel al cărui număr de ordine este 1); la fiecare al  $i$ -lea pas al jocului se numără  $i$  copii în sens trigonometric și este eliminat copilul la care se ajunge; la pasul următor număratoarea începe de la copilul care urmează după cel eliminat.

Așadar, dacă numărul copiilor este suficient de mare, la primul pas este eliminat al doilea copil, la al doilea pas al patrulea, la al treilea pas al șaptelea, apoi la al patrulea pas al unsprezecelea și așa mai departe. Va trebui să determinai ordinea în care vor fi eliminați copiii.

### Date de intrare

Fișierul de intrare **ORDER.IN** conține pe prima linie un număr întreg  $n$ , care reprezintă numărul de copii.

### Date de ieșire

Fișierul de ieșire **ORDER.OUT** trebuie să conțină o singură linie pe care se vor afla  $n$  numere distincte cuprinse între 1 și  $n$  care reprezintă numerele de ordine ale copiilor în ordinea în care au fost eliminați.

### Restricție

- $2 \leq n \leq 30.000$ .

### Exemplu

| ORDER . IN | ORDER . OUT |
|------------|-------------|
| 6          | 2 4 1 3 5 6 |

**Timp maxim de execuție/test:** 1 secundă

### P040435: Pioni

Se consideră o tablă împărțită în mai multe celule care conțin pioni. Celulele sunt conectate între ele prin săgeți, și mergând de la orice celulă în direcția în care indică săgețile nu vom ajunge niciodată în locul din care am plecat. Deci, există celule din care nu pleacă săgeți și celule în care nu ajung săgeți.

Considerăm doi jucători. Fiecare jucător poate muta, când îi vine rândul, un pion din celula în care se află într-o celulă spre care pleacă o săgeată.

Cel care nu mai poate muta atunci când îi vine rândul pierde partida.

La începutul unui joc, prima mutare o face jucătorul 1.

Să se determine dacă jucătorul 1 are strategie sigură de câștig.

### Date de intrare

Fișierul de intrare **PAWNS.IN** conține pe prima linie două numere întregi  $n$  și  $m$ , separate printr-un singur spațiu, care reprezintă numărul de celule de pe tablă, respectiv numărul de săgeți.

Pe fiecare dintre următoarele  $m$  linii se află câte două numere întregi  $x$  și  $y$ , separate între ele printr-un singur spațiu, cu semnificația că *există o săgeată care pleacă din celula cu numărul de ordine  $x$  și ajunge în celula cu numărul de ordine  $y$ .*

Pe linia următoare se află un număr  $t$  care reprezintă numărul de partide care se joacă.

Pe fiecare dintre următoarele  $t$  linii se află  $n$  numere întregi, separate între ele prin spații, care reprezintă numărul de pioni din fiecare celulă pentru o anumită partidă.

### Date de ieșire

Fișierul de ieșire **PAWNS.OUT** trebuie să conțină  $t$  linii. Fiecare dintre cele  $t$  linii va conține valoarea 0, dacă pentru o configurație jucătorul 1 nu are strategie sigură de câștig și valoarea 1 în caz contrar.

### Restricții și precizări

- $1 \leq n, m \leq 500$ ;
- $1 \leq t \leq 15$ ;
- numărul de pioni dintr-o celulă nu va depăși valoarea 500;
- celulele sunt numerotate de la 1 la  $n$ .

### Exemplu

| PAWNS . IN | PAWNS . OUT |
|------------|-------------|
| 3 3        | 1           |
| 1 2        | 0           |
| 1 3        |             |
| 2 3        |             |
| 2          |             |
| 1 0 0      |             |
| 2 2 2      |             |

**Timp maxim de execuție/test:** 1 secundă

### P040436: Matrice binară

Se consideră o matrice binară de dimensiune  $m \cdot n$  (elementele matricei sunt 0 sau 1).

Se cere să se determine aria maximă care poate fi acoperită de două dreptunghiuri care conțin numai elemente cu valoarea 0.

#### Date de intrare

Fișierul de intrare **BMATRIX.IN** conține pe prima linie două numere întregi  $m$  și  $n$ , separate printr-un singur spațiu, care reprezintă dimensiunile matricei.

Pe fiecare dintre următoarele  $m$  linii se află  $n$  numere care pot avea valorile 0 sau 1 și care nu sunt separate între ele prin spații.

#### Date de ieșire

Fișierul de ieșire **BMATRIX.OUT** trebuie să conțină un singur număr, care reprezintă aria maximă care poate fi acoperită de două dreptunghiuri care conțin numai elemente cu valoarea 0.

#### Restricții și precizări

- $1 \leq m, n \leq 200$ ;
- cele două dreptunghiuri nu se pot suprapune.

#### Exemplu

| BMATRIX.IN | BMATRIX.OUT |
|------------|-------------|
| 6 8        | 23          |
| 10000000   |             |
| 10000000   |             |
| 11100011   |             |
| 00100011   |             |
| 00100011   |             |
| 00111111   |             |

**Timp maxim de execuție/test:** 1 secundă

### P040437: Numere

Se consideră șirul obținut prin concatenarea numerelor naturale (în ordinea lor firească). Acest șir va avea forma:  
123456789101112...

Va trebui să determinați poziția primei apariții în șirul considerat a unui număr  $k$  dat.

#### Date de intrare

Fișierul de intrare **NUMBERS.IN** conține pe prima linie un număr întreg  $n$ , care reprezintă numărul de numere care trebuie căutate în șirul dat. Pe fiecare dintre următoarele  $n$  linii se află câte un număr care trebuie căutat.

#### Date de ieșire

Fișierul de ieșire **NUMBERS.OUT** trebuie să conțină  $n$  linii. Pe fiecare dintre cele  $n$  linii trebuie să se afle un număr care reprezintă poziția primei apariții în șirul considerat a numărului corespunzător din fișierul de intrare.

#### Restricții și precizări

- $1 \leq n \leq 20$ ;
- numerele din fișierul de intrare sunt întregi și sunt cuprinse între 1 și 1.000.000.000;

#### Exemplu

| NUMBERS.IN | NUMBERS.OUT |
|------------|-------------|
| 2          | 6           |
| 67         | 3           |
| 345        |             |

**Timp maxim de execuție/test:** 1 secundă

### P040438: Cercuri

Se consideră zece cercuri pentru care se cunosc coordonatele centrelor și razele lor.

Să se determine cel mai mic pătrat care conține în interior cele zece cercuri.

#### Date de intrare

Fișierul de intrare **CIRCLES.IN** conține zece linii. Pe fiecare dintre acestea se află câte trei numere  $x$ ,  $y$  și  $r$ , separate între ele prin spații, care reprezintă coordonatele centrului și raza unui cerc.

#### Date de ieșire

Fișierul de ieșire **CIRCLES.OUT** trebuie să conțină o singură linie pe care se va afla latura minimă a unui pătrat care conține în interior cele zece cercuri.

#### Restricții și precizări

- coordonatele centrelor cercurilor sunt numere întregi cuprinse între 1 și 999;
- razele cercurilor sunt numere întregi cuprinse între 1 și 100;
- coordonatele tuturor punctelor aflate pe circumferințele cercurilor sunt cuprinse între 0 și 1000;
- laturile pătratului nu sunt neapărat paralele cu axele de coordonate;
- valoarea din fișierul de ieșire va fi scrisă cu opt zecimale exacte.

#### Exemplu

| CIRCLES.IN | CIRCLES.OUT |
|------------|-------------|
| 1 1 1      | 12.00000000 |
| 3 3 1      |             |
| 6 2 2      |             |
| 10 2 2     |             |
| 2 6 2      |             |
| 6 6 2      |             |
| 10 6 2     |             |
| 2 10 2     |             |
| 6 10 2     |             |
| 10 10 2    |             |

**Timp maxim de execuție/test:** 1 secundă

