



# Bursele Agora

Vă prezentăm în continuare enunțurile problemelor propuse spre rezolvare la primele patru etape ale ediției 2002/2003 a concursului de programare Bursele Agora.

## P070207: Găuri de vierme

În anul 2457 cercetătorii din *Federația Planetelor Unite* au descoperit secretul găurilor de vierme. Astfel ei au reușit să construiască un dispozitiv de creare a unor găuri de vierme artificiale. Din nefericire proiectul nu este încă bine pus la punct; din aceste motive nu se poate călători decât într-un singur sens printr-o gaură de vierme artificială. Astfel o gaură de vierme are o intrare într-un anumit sistem stelar și o ieșire aflată într-un alt sistem stelar (care se poate afla la milioane de ani lumină distanță).

Construirea găurilor de vierme artificiale necesită folosirea unei cantități uriașe de energie, motiv pentru care *Federația Planetelor Unite* poate construi doar  $N$  astfel de scurtături prin spațiu. De aceea s-au ales cele mai importante  $N$  sisteme stelare din *Federație* și s-a hotărât că în fiecare astfel de sistem va exista o intrare a unei găuri de vierme și o ieșire a alteia.

Spionii din *Imperiul Galactic*, cel mai important inamic al *Federației*, au aflat de acest proiect și care sunt cele  $N$  sisteme stelare alese pentru acest experiment. *Consiliul Imperial* și-a dat seama că este de o importanță vitală să se știe care este numărul de posibilități de creare a găurilor de vierme, motiv pentru care vă cere să determinați această valoare.

### Date de intrare

Fișierul de intrare **WORMHOLE.IN** conține numărul  $N$  al sistemelor stelare alese.

### Date de ieșire

Fișierul de ieșire **WORMHOLE.OUT** trebuie să conțină numărul posibilităților de amplasare a găurilor de vierme astfel încât în fiecare sistem stelar să existe exact o intrare și exact o ieșire a două găuri de vierme diferite.

### Restricții

- $2 \leq N \leq 255$ .

### Exemple

**WORMHOLE.IN**

3

**WORMHOLE.OUT**

2

**WORMHOLE.IN**

4

**WORMHOLE.OUT**

9

**Timp de execuție:** 1 secundă/test

## P070208: Matrice

Se consideră o matrice  $A$  de dimensiune  $M \times N$  ale cărei elemente au valoarea 0. Se aleg  $K$  cvadrupe de numere  $x_1, y_1, x_2, y_2$  cu  $x_1 \leq x_2$  și  $y_1 \leq y_2$ . Pentru fiecare astfel de cvadrupe valoarea tuturor elementelor  $A_{ij}$  ( $x_1 \leq j \leq x_2, y_1 \leq i \leq y_2$ ) crește cu 1.

Dându-se o matrice  $A$ , să se determine un număr cât mai mic de cvadrupe care să ducă la obținerea matricei  $A$ , pornind de la o matrice care are toate elementele nule.

### Date de intrare

Prima linie a fișierului de intrare **MATRIX.IN** conține numărul de linii  $M$  și numărul de coloane  $N$  ale matricei, separate printr-un singur spațiu.

Fiecare dintre următoarele  $M$  linii va conține  $N$  numere naturale, separate prin spații; al  $j$ -lea număr de pe a  $i$ -a dintre aceste  $N$  linii reprezintă valoarea elementului  $A_{ij}$ .

### Date de ieșire

Prima linie a fișierului de ieșire **MATRIX.OUT** va conține numărul  $K$  al cvadrupelelor folosite.

Fiecare dintre următoarele  $K$  linii va conține cele patru numere din componența unui cvadrupe, în ordinea  $x_1, y_1, x_2, y_2$ , separate prin spații.

### Restricții și precizări

- matricea  $A$  va putea fi obținută întotdeauna folosind cel mult 1000 de cvadrupe;
- $1 \leq M, N \leq 100$ ;
- $0 \leq A_{ij} \leq 255$ .

**Modalitatea de acordare a punctajului**

Pentru fiecare test, se vor putea obține cel mult  $X$  puncte.

Concurenții care vor găsi cea mai mică valoare  $K_{min}$  prin care se poate obține matricea  $A$ , vor primi  $X$  puncte pentru testul respectiv.

Ceilalți concurenți, care au reușit să obțină matricea  $A$  folosind  $K > K_{min}$  cvadrupele, vor obține  $X \cdot \frac{K_{min}}{K}$  puncte pentru testul respectiv. Această valoare va fi aproximată cu două zecimale exacte.

Punctajul final va fi obținut prin adunarea punctajelor de la fiecare test și rotunjirea acestuia la cel mai apropiat număr întreg.

Dacă cvadrupele din fișierul de ieșire nu duc la obținerea matricei  $A$  sau programul nu respectă limita de timp admisă, concurenții nu vor primi nici un punct pentru testul respectiv.

De exemplu, dacă pentru un test se pot obține cel mult 5 puncte, cel mai bun rezultat obținut de un concurent constă în 7 cvadrupele, iar dacă un alt concurent obține o soluție cu 12 cvadrupele, atunci, pentru testul respectiv, punctajul concurentului va fi de  $5 \cdot \frac{7}{12} \approx 2.92$  puncte.

**Exemplu****MATRIX.IN**

```
5 6
1 2 2 1 1 0
1 3 3 2 2 0
1 3 3 2 2 0
0 2 2 3 3 1
0 1 1 2 2 1
```

**MATRIX.OUT**

```
4
2 2 5 4
1 1 3 3
2 1 5 5
4 4 6 5
```

**Timp de execuție:** 1 secundă/test

**P070209: Ecuații**

Se consideră  $N$  variabile care pot avea valori numere reale și  $M$  ecuații în care pot apărea cele  $N$  variabile, operatorii de adunare ('+'), scădere ('-') și înmulțire ('\*'), constante întregi cuprinse între -10000 și 10000 și, o singură dată, semnul de egalitate ('=').

Va trebui să determinați valori pentru cele  $N$  variabile astfel încât toate cele  $M$  egalități (ecuații) să fie respectate.

De exemplu, pentru variabilele  $a, b, c$  și  $d$  și ecuațiile:

```
a+b+c+d=18
a*b+c*d=43
a+b=41-c*d
a+b=-1219+c*d*d*c
```

valorile pot fi  $a=2, b=4, c=5$  și  $d=7$ . Soluția nu este unică; o altă posibilitate este  $a=4, b=2, c=7$  și  $d=5$ ; mai există și alte soluții.

Se garantează faptul că există întotdeauna cel puțin o posibilitate de a atribui valori celor  $N$  variabile astfel încât toate cele  $M$  egalități să fie respectate.

În cele  $M$  ecuații nu apar paranteze, iar operatorii de înmulțire au prioritate față de cei de scădere și adunare. Aceștia din urmă au aceeași prioritate, deci operațiile de adunare și scădere sunt efectuate de la stânga la dreapta.

**Date de intrare**

Aceasta este o problemă cu *fișierele de intrare deschise*. În pagina <http://www.ginfo.ro/concurs/runda03/index.shtml>, pagină dedicată problemei sunt disponibile *link*-uri spre zece fișiere de intrare. Numele acestora au forma **EQXX.IN**, unde **XX** ia valori între **01** și **10** și reprezintă numărul testului.

Fiecare dintre cele 10 fișiere de intrare va conține, pe prima linie, numărul  $N$  al variabilelor. Pe fiecare dintre următoarele  $N$  linii se află identificatorul uneia dintre variabile. Identificatorii sunt formați din cel puțin una și cel mult opt litere ale alfabetului englezesc. Următoarea linie conține numărul  $M$  al ecuațiilor, iar pe fiecare dintre următoarele  $M$  linii apare una dintre ecuații, în formatul specificat anterior.

**Date de ieșire**

Pentru această problemă nu va trebui să trimiteți un program care să o rezolve, ci doar cele 10 fișiere de ieșire corecte. Acestea vor fi denumite **EQXX.OUT**, unde **XX** ia valori între **01** și **10** și reprezintă numărul testului.

Prima linie a fișierului de ieșire va conține textul **EQUATIONS - TEST #XX**, unde **XX** reprezintă numărul testului. Fiecare dintre următoarele  $N$  linii va conține un șir de tipul *variabilă = valoare*, unde *variabilă* este una dintre cele  $N$  variabile din fișierul de intrare, iar *valoare* este valoarea atribuită acesteia. Valorile vor fi scrise cu opt zecimale exacte. Variabilele vor apărea în fișierul de ieșire în aceeași ordine în care apar în fișierul de intrare corespunzător. Fiecare variabilă va apărea exact o dată.

**Exemplu**

Vom considera că acest exemplu reprezintă testul 00.

**EQ00.IN**

```
4
b
c
a
d
4
a*b+c*d=43
a+b=41-c*d
a+b=-1219+c*c*d*d
a+b+c+d=18
```

**EQ00.OUT**

EQUATIONS - TEST #00

b=2.00000000

c=5.00000000

a=4.00000000

d=7.00000000

**Modalitatea de acordare a punctajului**

Pentru fiecare fișier de ieșire corect veți obține 10 puncte. Punctajul maxim care poate fi obținut (dacă toate cele 10 fișiere de ieșire sunt corecte) este de 100 de puncte.

**Trimiterea soluțiilor**

Arhiva pe care o veți trimite se va numi **YYYYYR03.ZIP**, **YYYYYR03.RAR** sau **YYYYYR03.ACE** (în funcție de programul de arhivare pe care îl folosiți), unde **YYYYY** este codul dumneavoastră de identificare (**ID**).

Ea nu va conține fișiere **batch** (**YYYYYR03.BAT**) sau fișiere sursă (**YYYYYR03.PAS**, **YYYYYR03.CPP** sau **YYYYYR03.C**). În arhivă se vor afla cele 10 fișiere de ieșire corecte, denumirea lor fiind **EQXX.OUT**, unde **XX** reprezintă numărul testului (cuprins între **01** și **10**).

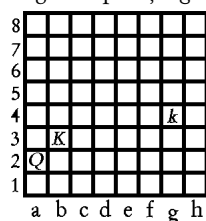
**P070210: Șah**

Considerăm poziția finală a jocului de șah în care pe tablă au rămas regina albă, regele alb și regele negru.

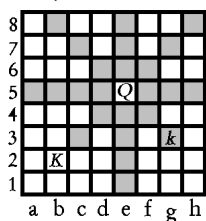
Vom identifica regina albă prin litera **Q**, regele alb prin **K** și regele negru prin **k**.

Tabla de șah este formată din 64 de pătrățele dispuse într-o matrice cu opt linii și opt coloane. Coloanele matricei sunt identificate prin primele opt litere ale alfabetului (**a - h**), iar liniile prin primele opt numere naturale strict pozitive (**1 - 8**).

Poziția unei piese este dată de litera corespunzătoare coloanei și numărul corespunzător liniei. De exemplu, în figura 1 regina albă se află în poziția **a2**, regele alb în poziția **b3**, iar regele negru în poziția **g4**.

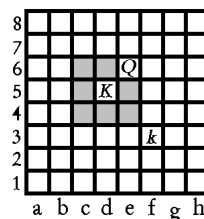
**Figura 1**

Definim aria de acțiune a reginei albe ca fiind totalitatea pătrățelelor de pe aceeași linie, coloană sau diagonală în care nu se află piese albe și între un astfel de pătrățel și

**Figura 2**

regină nu se află nici o altă piesă. Pătrățelul în care se află regina nu face parte din aria sa de acțiune. În figura 2 este ilustrată aria de acțiune a reginei albe la un moment dat.

Definim aria de acțiune a unui rege ca fiind totalitatea pătrățelelor alăturate (pe linie, coloană sau diagonală) care nu sunt ocupate de piese de aceeași culoare. În figura 3 este ilustrată aria de acțiune a regelui alb într-o anumită situație.

**Figura 3**

Jucătorii vor muta alternativ; considerăm că, la început jucătorul cu piesele albe este la mutare.

La un moment dat jucătorul, cu piesele albe poate muta fie regele, fie regina. Regele alb poate fi mutat în oricare dintre pătrățelele din aria sa de acțiune dacă pătrățelul respectiv nu se află în aria de acțiune a regelui negru.

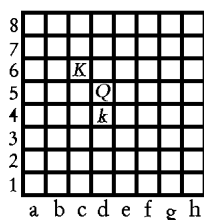
Regina albă poate fi mutată în oricare dintre pătrățelele din aria sa de acțiune, chiar dacă intră în aria de acțiune a regelui negru.

Regele negru poate fi mutat în oricare dintre pătrățelele din aria sa de acțiune dacă, după mutare, nu ajunge în aria de acțiune a reginei albe sau a regelui alb.

Dacă o piesă ajunge în poziția ocupată de o altă piesă, atunci această a doua piesă este capturată și eliminată de pe tablă.

Se observă că, datorită regulilor anterioare, cei doi regi nu pot ajunge în poziții alăturate și singura piesă care poate fi capturată este regina albă.

Trebuie observat faptul că regele negru nu poate ajunge lângă regele alb prin capturarea reginei deoarece, chiar dacă momentan poziția respectivă nu este în aria de acțiune a regelui alb (respectând definiția anterioară), după mutare această poziție intră în aria de acțiune datorită eliminării reginei. De exemplu, în situația din figura 4 regina nu poate fi capturată.

**Figura 4**

Se știe că într-o situație finală de acest tip jucătorul alb nu poate pierde; mai mult, el are șanse foarte mari de câștig.

Jucătorul negru pierde partida dacă se află în aria de acțiune a reginei albe și nu poate muta în nici unul dintre

pătrățelele din aria sa de acțiune. Această situație se numește **mat**.

Partida se termină la egalitate dacă regina albă este capturată sau se ajunge în situația de **pat**. Această situație apare atunci când regele negru nu se află în aria de acțiune a reginei albe, dar nu poate efectua nici o mutare validă. O astfel de situație este ilustrată în figura 5.

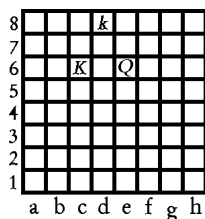


Figura 5

Veți juca cu piesele albe (regele și regina) și va trebui să câștigați partida folosind un număr limitat de mutări.

Mutările adversarului (regele negru) vor fi efectuate de o bibliotecă externă.

### Bibliotecile Pascal și C/C++

Pentru a putea folosi biblioteca va trebui să includeți în programul dumneavoastră instrucțiunea **uses chess**; pentru limbajul *Pascal* și **#include "chess.h"** pentru limbajul *C/C++*. În continuare sunt prezentate sintaxele funcțiilor și procedurilor incluse în bibliotecă:

- **procedure** Init;  
**void** Init()  
 - trebuie apelată la începutul jocului și este folosită de către bibliotecă pentru inițializarea tablei;  
 - întrerupe execuția programului dacă este apelată a doua oară.
- **function** MovesLeft: Integer;  
**int** MovesLeft()  
 - returnează un număr întreg care reprezintă numărul de mutări pe care le mai aveți la dispoziție pentru a câștiga.
- **function** GetWhiteQueenPosition: String;  
**char\*** GetWhiteQueenPosition()  
 - returnează un șir care conține două caractere, primul fiind o literă care reprezintă coloana pe care se află regina albă în acel moment, iar al doilea un număr care reprezintă linia.
- **function** GetWhiteKingPosition: String;  
**char\*** GetWhiteKingPosition()  
 - returnează poziția curentă a regelui alb.
- **function** GetBlackKingPosition: String;  
**char\*** GetBlackKingPosition()  
 - returnează poziția curentă a regelui negru.
- **procedure** MoveWhiteQueen(position: String);  
**void** MoveWhiteQueen(char\* position)  
 - mută regina albă în poziția indicată de șirul position;  
 - întrerupe execuția programului dacă:
  - ♦ șirul de caractere nu reprezintă o poziție validă;
  - ♦ mutarea este nepermisă;
  - ♦ este a doua mutare consecutivă a jucătorului alb;

- ♦ s-a epuizat numărul de mutări admis și partida s-a încheiat;
- ♦ se ajunge în situație de **pat**;
- ♦ se ajunge în situație de **mat** (singurul caz în care obțineți puncte pentru testul respectiv).

- **procedure** MoveWhiteKing(position: String);  
**void** MoveWhiteKing(char\* position)  
 - mută regele alb în poziția indicată de șirul position;  
 - întrerupe execuția programului dacă este verificată una dintre condițiile prezentate în cadrul procedurii anterioare.
  - **procedure** MoveBlackKing;  
**void** MoveBlackKing()  
 - indică bibliotecii că se așteaptă mutarea regelui negru;  
 - întrerupe execuția programului dacă:
    - ♦ este a doua mutare consecutivă a regelui negru;
    - ♦ prin mutare poate fi capturată regina albă;
- noua poziție a regelui negru trebuie determinată printr-un apel al funcției GetBlackKingPosition.
- Toate funcțiile și procedurile (cu excepția procedurii Init) întrerup execuția programului dacă sunt apelate înaintea inițializării tablei.

Programul vostru nu va citi datele din nici un fișier de intrare și nu va scrie date în nici un fișier de ieșire.

### Exemplu de utilizare corectă a bibliotecii

În figura 6 este ilustrată poziția inițială:

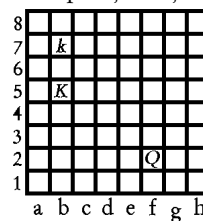


Figura 6

Subprogram apelat	Valoare returnată
Init	-
MovesLeft	12
GetWhiteQueenPosition	f2
GetWhiteKingPosition	b5
GetBlackKingPosition	b7
MoveWhiteQueen(c5)	-
MoveBlackKing	-
GetBlackKingPosition	a7
MoveWhiteQueen(c7)	-
MoveBlackKing	-
GetBlackKingPosition	a8
MoveWhiteQueen(d7)	-
MoveBlackKing	-
GetBlackKingPosition	b8
MoveWhiteKing(b6)	-
MoveBlackKing	-
GetBlackKingPosition	a8
MoveWhiteQueen(b7)	-

**Timp de execuție:** 1 secundă/test

