



# Bursele Agora

Vă prezentăm în continuare enunțurile problemelor propuse spre rezolvare la etapele #13-#16 ale celei de-a patra ediții a concursului de programare Bursele Agora.

## P030311: Grupuri

Se consideră  $N$  numere întregi care trebuie repartizate în  $p$  grupuri. Grupurile sunt identificate prin numere naturale cuprinse între 1 și  $p$ . Repartizarea în grupuri trebuie să se realizeze astfel încât suma numerelor din oricare grup  $i$  să fie divizibilă cu numărul total de numere care fac parte din grupurile identificate prin valori cuprinse între  $i$  și  $p$ .

### Date de intrare

Prima linie a fișierului de intrare **GROUPS.IN** conține numărul  $N$  al valorilor care trebuie repartizate în grupuri. Cea de-a doua linie va conține cele  $N$  numere, separate prin câte un spațiu.

### Date de ieșire

Fișierul de ieșire **GROUPS.OUT** trebuie să conțină un număr de linii egal cu numărul grupurilor. Prima linie corespunde primului grup, a doua linie celui de-al doilea grup etc. Linia corespunzătoare unui grup va conține numerele care fac parte din grupul respectiv, separate prin câte un spațiu.

### Restricții și precizări

- $1 \leq N \leq 1000$ ;
- cele  $N$  numere sunt cuprinse între 1 și 1000;
- numărul grupurilor nu este cunoscut; el va trebui determinat de program;
- fiecare grup va conține cel puțin un număr;
- dacă există mai multe soluții va fi generată doar una dintre ele;
- va exista întotdeauna cel puțin o soluție.

### Exemplu

GROUPS.IN	GROUPS.OUT
6	2 4
4 10 3 9 2 3	3 9
	10
	3

### Explicație

Sunt formate patru grupuri dintre care primele două conțin câte două numere, iar ultimele două conțin unul singur.

Suma numerelor din primul grup este  $2 + 4 = 6$  care este divizibilă cu 6 (numărul elementelor din grupurile identificate prin 1, 2, 3 și 4). Suma numerelor din al doilea grup este  $3 + 9 = 12$  care este divizibilă cu 4 (numărul elementelor din grupurile identificate prin 2, 3 și 4). Al treilea grup este format doar din numărul 10; acesta este divizibil cu 2 (numărul elementelor din grupurile identificate prin 3 și 4). Al patrulea grup este format doar din numărul 3; acesta este divizibil cu 1 (numărul elementelor din grupul identificat prin 4). Există mai multe alte soluții alternative; una dintre ele este:

```
2 3 9 10
4
3
```

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

## P030312: Războiul stelelor

Cu mult timp în urmă, într-o galaxie foarte îndepărtată, s-a desfășurat un *Consiliu* al *Cavalerilor Jedi* în cadrul căruia s-a discutat problema noii amenințări reprezentate de partea întunecată a *Forței*. În ultima perioadă au fost semnalate numeroase apariții ale *Lorzilor Sith*, motiv pentru care *Consiliul* a trebuit să ia o decizie importantă. După luarea deciziei *Yoda* a ieșit din *Sala de Consiliu* și s-a îndreptat direct spre *Obi Wan* care aștepta cu nerăbdare să afle decizia luată.

*Yoda: Consiliul decis este!*

*Obi Wan: Pari cam nemulțumit, Maestre Yoda.*

*Yoda: În dezacord cu decizia eu sunt!*

*Obi Wan: Dar ce s-a hotărât până la urmă?*

*Yoda: Mulți Cavaleri pe planete îndepărtate trimiși trebuie!*

*Obi Wan: De ce?*

*Yoda: Găuri de vierme descoperite au fost! Pentru deplasare rapidă între planete găurile folosite pot fi!*

*Obi Wan: Și care este problema?*

*Yoda: Lorzii Sith găuri folosesc pentru scopurile lor mârșave!*

*Obi Wan: Se cunosc planetele legate prin găuri de vierme?*



**Yoda:** Amplasamentul tuturor intrărilor și ieșirilor găurilor Consiliul îl cunoaște!

**Obi Wan:** Și ce e de făcut acum?

**Yoda:** Cavaleri Jedi pe fiecare planetă lângă care se află găuri trimiși vor fi!

**Obi Wan:** Deci va fi trimis câte un Jedi pe fiecare dintre aceste planete?

**Yoda:** Mult mai complicată problema este! Numărul Cavalerilor Jedi de pe o planetă diferit să fie trebuie de numărul Cavalerilor de pe oricare planetă legată printr-o gaură de planeta aceea!

**Obi Wan:** De ce?

**Yoda:** Forța în acest fel ne-a sfătuit!

**Obi Wan:** Atunci cred că ar trebui să facem în așa fel încât să trimitem cât mai puțini Jedi!

**Yoda:** Sfatul tău bun este! Așa vom face!

Așadar, Consiliul Cavalerilor Jedi va trebui să trimită Jedi pe fiecare planetă lângă care a fost descoperită o gaură de vierme, iar numărul Cavalerilor Jedi de pe oricare dintre planete trebuie să fie diferit de cel al Cavalerilor trimiși pe oricare dintre planetele învecinate (legate de planeta respectivă printr-o gaură de vierme). În plus, numărul total al Cavalerilor Jedi trimiși va trebui să fie cât mai mic posibil.

#### Date de intrare

Prima linie a fișierului de intrare **STARWARS.IN** conține numărul  $N$  al planetelor lângă care au fost descoperite găuri de vierme și numărul total  $M$  al găurilor descoperite.

Fiecare dintre următoarele  $M$  linii va conține câte două numere  $x$  și  $y$ , separate printr-un spațiu, cu semnificația: există o gaură de vierme între planeta identificată prin numărul  $x$  și planeta identificată prin numărul  $y$ .

#### Date de ieșire

Fișierul de ieșire **STARWARS.OUT** va conține  $N$  linii. Cea de-a  $i$ -a linie va conține numărul Cavalerilor Jedi trimiși pe planeta identificată prin numărul  $i$ .

#### Restricții și precizări

- $1 \leq N \leq 100$ ;
- $1 \leq M \leq 1000$ ;
- planetele sunt identificate prin numere întregi cuprinse între 1 și  $N$ ;
- printr-o gaură de vierme se poate circula în ambele sensuri;
- între două planete poate exista cel mult o gaură de vierme.

#### Exemplu

STARWARS.IN	STARWARS.OUT
5 4	2
1 2	1
1 3	1
1 4	1
1 5	1

#### Modalitatea de acordare a punctajului

Vom considera că pentru fiecare test, se vor putea obține cel mult  $X$  puncte. Concurenții care vor obține cea mai mică valoare  $NrMin$  pentru numărul total al Cavalerilor Jedi vor primi  $X$  puncte pentru testul respectiv.

Cealalți concurenți, care au stabilit o repartizare corectă a Cavalerilor Jedi, dar numărul total al acestora este  $Nr > NrMin$ , vor obține  $X \cdot NrMin / Nr$  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 repartizarea nu respectă condițiile impuse, 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 într-o repartizare în care sunt folosiți 24 de Cavaleri Jedi, iar un alt concurent obține o soluție în care este nevoie de 57 de Cavaleri Jedi, atunci, pentru testul respectiv, punctajul concurentului va fi de  $5 \cdot 24 / 57 = 2.10$  puncte.

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

#### P030313: Galaxie

Confederația Galactică deține controlul asupra a  $N$  planete între care există un număr total de  $M$  rute comerciale directe. Planetele vor fi identificate prin numere cuprinse între 1 și  $N$ , iar rutele comerciale prin numere cuprinse între 1 și  $M$ .

Fiecare rută comercială directă este folosită pentru transporturi între perechi distincte de planete. Un circuit comercial este format din  $k$  planete distincte  $a_1, a_2, \dots, a_k$  astfel încât există o rută comercială directă între planetele  $a_i$  și  $a_{i+1}$  pentru  $i$  cuprins între 1 și  $k - 1$  și există o rută comercială directă între planetele  $a_k$  și  $a_1$ .

Două circuite comerciale sunt distincte dacă și numai dacă există cel puțin o rută comercială care apare în unul dintre ele și nu apare în celălalt. Pentru fiecare rută comercială, va trebui să stabiliți din câte circuite comerciale distincte face parte.

#### Date de intrare

Fișierele de intrare sunt deschise și pot fi descărcate folosind *link*-urile de pe pagina dedicată problemei; adresa acestuia este <http://www.ginfo.ro/concurs/runda15/index.shtml>. Numele fișierelor au forma **GALAXYXX.IN**, unde **XX** ia valori între 01 și 20 și reprezintă numărul testului.

Fiecare dintre cele 20 de fișiere de intrare conține pe prima linie numărul  $N$  al planetelor și numărul  $M$  al rutelor comerciale directe. Următoarele  $M$  linii vor conține câte două numere întregi distincte cuprinse între 1 și  $N$  care sunt numerele de identificare ale planetelor legate între ele prin ruta comercială corespunzătoare liniei. Numerele de pe o linie sunt separate prin câte un spațiu.

**Date de ieșire**

Pentru această problemă nu va trebui să scrieți un program, ci să generați doar cele 20 de fișiere de ieșire corecte. Acestea vor fi denumite **GALAXYXX.OUT**, unde **XX** ia valori între 01 și 20 și reprezintă numărul testului.

Prima linie a fișierului de ieșire va conține textul **GALAXY - TEST #XX**, unde **XX** reprezintă numărul testului. Următoarele **M** linii vor conține câte un număr întreg care reprezintă numărul de circuite comerciale distincte din care face parte ruta corespunzătoare liniei. Informațiile referitoare la rute vor apărea în ordinea în care sunt descrise rutele în fișierul de intrare corespunzător.

**Exemplu**

Vom considera că acest exemplu reprezintă testul **00**.

<b>GALAXY00.IN</b>	<b>GALAXY00.OUT</b>
4 5	GALAXY - TEST #00
1 2	2
1 3	2
2 3	2
2 4	2
3 4	2

**P030314: Săgeți**

*Mickey* și *Minnie* au ajuns în fața unui perete pe care erau desenate mai multe cercuri și mai multe săgeți, fiecare săgeată pornind dintr-un cerc și ajungând la un alt cerc.

Desenul era foarte complicat, dar *Minnie* l-a întrebat pe *Mickey* dacă există vreun cerc în care intră săgeți care pornesc de la toate celelalte cercuri, dar din care nu pleacă nici o săgeată.

Pentru a o impresiona pe *Minnie*, *Mickey* a spus că va studia problema, dar se pare că nu se prea descurcă, motiv pentru care vă cere ajutorul.

Din nefericire, nu puteți comunica decât prin intermediul unui telefon. *Mickey* vă spune la început numărul **N** al cercurilor, dar nu are răbdare să vă descrie toate săgețile.

Datorită nerăbdării lui *Mickey*, îl veți putea întreba doar de  $2 \cdot N$  ori dacă există o săgeată care pleacă de la un anumit cerc și ajunge la un anumit cerc. Din fericire, ați auzit-o la telefon pe *Minnie* strigând "*L-am găsit*", motiv pentru care știți cu siguranță că există un astfel de cerc.

Pentru a simula convorbirea cu *Mickey* aveți la dispoziție o bibliotecă externă care vă pune la dispoziție rutine pentru:

- inițializare;
- determinarea numărului de cercuri;
- verificarea existenței unei săgeți care pornește de la un anumit cerc *x* și ajunge la un alt cerc *y*;
- furnizarea cercului în care intră săgeți de la toate celelalte cercuri și din care nu pleacă nici o săgeată.

**Bibliotecile Pascal și C/C++**

Pentru a putea folosi biblioteca va trebui să includeți în programul dumneavoastră instrucțiunea **uses arrows**; pentru limbajul *Pascal* și **#include "arrows.h"** pentru

limbajul *C/C++*. În continuare sunt prezentate sintaxele funcțiilor și procedurilor incluse în bibliotecă:

**procedure** Init;

**void** Init();

- trebuie apelată la începutul programului și este folosită de către bibliotecă pentru inițializare;
- întrerupe execuția programului dacă este apelată a doua oară.

**function** GetN: Integer;

**int** GetN();

- returnează numărul de cercuri de pe perete.

**function** IsArrow(x,y: Integer): Boolean;

**int** IsArrow(int x, int z)

- returnează **true** (valoare nenulă) dacă există o săgeată care pornește de la cercul identificat prin *x* și ajunge la cercul identificat prin *y* și **false** (0) în caz contrar;
- întrerupe execuția programului dacă cel puțin una dintre valorile *i* și *j* nu este un număr întreg cuprins între 1 și *N* sau dacă este apelată a  $2 \cdot N + 1$ -a oară.

**procedure** Result(circle: Integer);

**void** Result(int circle)

- acceptă ca rezultat faptul că *circle* este cercul căutat;
- validează rezultatul;
- întrerupe execuția programului.

Toate funcțiile și procedurile (excepție: **Init**) întrerup execuția programului dacă sunt apelate înaintea inițializării.

**Restricții și precizări**

- $1 \leq N \leq 100$ ;
- programul vostru nu va citi date din nici un fișier de intrare și nu va scrie date în nici un fișier de ieșire;
- cercurile sunt identificate prin numere cuprinse între 1 și *N*;
- există cel mult o săgeată de la un cerc *x* la un cerc *y*;
- nu există săgeți care pleacă de la un cerc *x* și ajung la același cerc *x*;
- dacă există o săgeată de la un cerc *x* la un cerc *y*, atunci este posibil să existe și o săgeată de la cercul *y* la cercul *x*.

**Exemple de utilizare a bibliotecii**

<i>Subprogram apelat</i>	<i>Valoare returnată</i>
Init	-
GetN	3
IsArrow(1,2)	true (1)
IsArrow(1,3)	true (1)
IsArrow(2,3)	true (1)
IsArrow(3,1)	false (0)
IsArrow(2,1)	true (1)
IsArrow(3,2)	false (0)
Result(3)	-

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