

**CLASA a X a****Problema 1.****PROBLEMA PRIMARULUI (100 puncte)**

Primarul din județul X își propune să strângă o sumă de bani pentru ajutorarea a 2 case de copii din județ (C1 și C2). Pentru această acțiune, directorul societății trimite scrisori la toate societățile comerciale din județ, solicitându-le ajutorul. Ca urmare a acestei cereri, societățile care sunt de acord, vor trimite, în funcție de posibilități, într-un plic sigilat un număr ( $\leq 5$ ) de bancnote având o valoare V.

Pe fiecare plic va fi înscris numele directorului societății care l-a trimis. Fiecare plic va fi trimis fie casei C1, fie casei C2, fără însă a fi desigilat. Primarul ar vrea să trimită plicurile celor 2 case în așa fel încât diferența în valoare absolută dintre numărul total de bancnote ce va fi trimis casei C1 (N1) și numărul total de bancnote ce va fi trimis casei C2 (N2) să fie minim. Dar marea problemă a primarului este următoarea: el nu știe de la început câte bancnote se găsesc în fiecare plic. Știe în schimb că fiecare plic are o etichetă pe care este înscris numărul de bancnote ce-l conține comparativ față de alte plicuri. De exemplu plicul trimis de 'ION' ar putea fi etichetat cu 'POP - 1', cu semnificația că numărul de bancnote din plicul trimis de ION este cu 1 mai mic decât numărul de bancnote din plicul trimis de POP. Câteva plicuri vor fi marcate cu numărul exact de bancnote ce-l conțin.

Ajutați-l pe primar să soluționeze problema, dacă este posibil, furnizându-i valorile N1 și N2.

**Intrare**

Datele de intrare se citesc din fișierul '**bani.in**', care conține o succesiune de linii de forma

**EXEMPLUL 1**      **bani.in**

Ion = Pop - 3

Marin = 4

Pop = Marin + 1

În legătură cu datele de intrare se mai presupun următoarele:

toate numele sunt formate din maxim 5 litere

există cel mult 1000 de plicuri

liniile de intrare fie dau direct numărul de bancnote dintr-un plic, fie îl consideră egal cu numărul de bancnote dintr-un alt plic +/- un număr

toate liniile de intrare sunt corecte din punct de vedere sintactic

fiecare nume apare o singură dată în stanga semnului '='.

**Iesire**

Se vor scrie în fișierul '**bani.out**' cele 2 valori întregi N1 și N2 separate printr-un spațiu, respectiv 'NU' dacă nu s-a putut deduce exact din fișierul de intrare numărul de bancnote din fiecare plic.

Pentru exemplul anterior ieșirea va fi **bani.out**

6 5      sau      5 6

**EXEMPLUL 2****bani.in**

⇒

**bani.out**

Ion = Pop - 3

NU

Dan = 4

Pop = Dan + 1

Marin = Mihai

**CLASA a X a****Problema 2.  
SECVENȚĂ FAREY (100 puncte)**

Secvența Farey de ordinul  $n$  este  $(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots$ . Unde  $x_i, y_i$  sunt numere naturale, și :

$$0 < y_i \leq n \quad \text{pentru orice } i \geq 1$$

$$\frac{x_i}{y_i} < \frac{x_{i+1}}{y_{i+1}} \quad \text{pentru orice } i \geq 1$$

$$0 \leq \frac{x_i}{y_i} \leq 1 \quad \text{pentru orice } i \leq n$$

De exemplu, secvența Farey de ordinul 7 este

$(0,1), (1,7), (1,6), (1,5), (1,4), (2,7), (1,3), (2,5), (3,7), (1,2), (4,7), (3,5), (2,3), (5,7), (3,4), (4,5), (5,6), (6,7), (1,1)$

Să se scrie un program care, pentru un număr natural  $N$  dat, generează secvența Farey de ordinul  $N$ .

—

## OLIMPIADA JUDETEANA DE INFORMATICA

CLASA a XI a si a XII a

### Problema 1 (100 puncte) Drum cu semafoare

Un sofer amator trebuie să ajungă la o oră prestabilită într-un anumit punct al orasului. Circulația fiind reglementată de recent instalatele semafoare, al căror ciclu de funcționare este cunoscut soferului nostru, acesta dorește să determine la ce oră trebuie să plece de acasă pentru a ajunge la timp la destinație.

Orasul are  $N$  intersecții, numerotate de la 1 la  $N$ . Fiecare stradă este identificată prin numerele intersecțiilor pe care le leagă. În fiecare intersecție sunt instalate mai multe semafoare, câte unul pentru fiecare conexiune posibilă. Astfel, fiecare semafor este identificat printr-un triplet: semaforul  $(i, j, k)$  reglementând accesul de pe strada  $(i, j)$  către strada  $(j, k)$ . Punctul de plecare este intersecția 1, iar punctul de sosire este intersecția  $N$ . Din intersecția 1 se poate pleca în orice moment pe oricare din străzile ce părăsesc intersecția. Dacă soferul ajunge într-o intersecție și semaforul corespunzător direcției în care vrea să plece este roșu, atunci el va aștepta trecerea pe verde. În intersecția  $N$  se poate ajunge oricând, dar cel târziu la momentul prestabilit.

#### Intrarea

Datele se citesc din fisierul **semafor.in** având următorul format:

- Prima linie conține patru numere naturale (separate prin spații):  $N, M, P$  și  $T$  reprezentând numărul de intersecții, numărul de străzi (considerate în sens unic), numărul de semafoare și respectiv momentul sosirii la destinație (în secunde).
- Următoarele  $M$  linii descriu străzile și conțin câte trei numere naturale (separate prin spații):  $i, j$  și  $t$ , unde  $t$  reprezintă timpul (în secunde) necesar deplasării pe strada directă de la intersecția  $i$  la intersecția  $j$ .
- Următoarele  $P$  linii descriu semafoarele și conțin câte 6 numere naturale:  $i, j, k, r, v, t$  cu următoarea semnificație: semaforul este instalat în intersecția  $j$  și reglementează plecarea spre  $k$  a vehiculelor venite dinspre  $i$ ; el este roșu timp de  $r$  secunde, apoi verde timp de  $v$  secunde (după care funcționarea se repetă periodic); prima trecere de pe roșu pe verde are loc la momentul  $t$ .

*Limite:*  $N \leq 500$ ,  $M \leq 5000$ ,  $P \leq 10000$ ; restul numerelor sunt mai mici sau egale cu 32000.

#### Iesirea

În fisierul **semafor.out** se vor scrie:

- Pe prima linie, momentul cel mai târziu al plecării
- Pe a doua linie, sirul intersecțiilor parcurse (1 și  $N$  inclusiv)

**CLASA a XI a si a XII a**

**Problema 2. (100 puncte)**

**PROBLEMA CUBURILOR (100 puncte)**

La un examen psihologic, un concurent este supus următoarei probe. Are la dispoziție două rânduri de cuburi. Pe fiecare rând e dispus un anumit număr de cuburi ( $\leq 100$ ), fiecare cub având fețele colorate cu o anumită culoare (toate fețele fiind colorate cu aceeași culoare). La un moment dat concurentul poate face următoarea operație: din oricare rând de cuburi, poate elimina un cub, fără însă a modifica ordinea acestora. Pentru a trece proba, concurentului i se cere, dacă e posibil, să elimine un număr minim de cuburi astfel încât cele două rânduri de cuburi să rămână identice (ca număr de cuburi, si ca ordine a culorilor cuburilor)

**Intrare**

Datele se citesc din fișierul text '**cuburi.in**' având următoarea structură:

- pe prima linie șirul culorilor de pe primul rând de cuburi separate printr-un spațiu
- pe a doua linie șirul culorilor de pe al doilea rând de cuburi separate printr-un spațiu
- datele de intrare se presupun a fi corecte
- lungimea unei linii din fișier nu depășește 256 de caractere

**Ieșire**

Rezultatul se va tipări în fișierul text '**cuburi.out**' astfel

- dacă se găsește soluție,
  - pe prima linie se va tipări numărul de cuburi rămase
  - pe a doua linie se va tipări șirul culorilor cuburilor rămase pe cele două rânduri, separate printr-un spațiu
- dacă nu se găsește soluție, se va tipări 'NU'

**EXEMPLUL 1**

**cuburi.in**

⇒

**cuburi.out**

rosu galben verde negru  
violet rosu galben alb negru

3  
rosu galben negru

**EXEMPLUL 2**

**cuburi.in**

⇒

**cuburi.out**

rosu galben alb  
verde negru

NU

**Timp de execuție: 1 secundă/test**

**CLASA a IX a**

**Problema 1. (100 puncte)**  
**Numere nemarcate**

Numerele de la 1 la N sunt așezate în ordine crescătoare pe circumferința unui cerc astfel că N ajunge lângă 1. Începând cu numărul S se marchează numerele din K în K, în ordinea crescătoare a lor, până când un număr este marcat de două ori.

Scrieți un program Pascal care să calculeze câte numere au rămas nemarcate.

Exemplu: Dacă  $N = 8$  și începând cu numărul 2 marcăm numerele din 5 în 5 atunci numerele marcate sunt în ordine 2, 7, 4, 1, 6, 3, 8, 5, 2, deci au rămas nemarcate 0 numere.

Dacă  $N = 8$  și începând cu numărul 2 marcăm numerele din 4 în 4 atunci numerele marcate sunt în ordine 2, 6, 2, deci au rămas nemarcate 6 numere.

**Problema 2. (100 puncte)**  
**Traseu cu stații de benzină**

Pe un traseu circular se găsesc  $n$  ( $n \leq 25000$ ) stații de benzină. Se dau distanțele dintre două stații consecutive (1->2, 2->3, 3->4, ...  $n-1 \rightarrow n$ ,  $n \rightarrow 1$ ) Fiecare dintre stații conține o anumită cantitate de combustibil (posibil zero). Suma cantităților de benzină din cele  $n$  stații ajunge unei mașini exact cât să acopere tot traseul o singură dată.

Se consideră că pentru parcurgerea unui km este necesară o cantitate de benzină egală cu unitatea.

Când ajunge la o stație și găsește benzină, mașina poate lua toată cantitatea stocată aici (rezervorul nu este limitat)

Scrieți un program Pascal care va determina de la ce stație va pleca o mașină și în ce direcție, astfel încât ea să poată revenii la locul de plecare.

exemplu: Pentru o intrare de forma

5	– numărul stațiilor
12, 3, 61, 10, 4	- distanțele dintre ele
0, 20, 40, 0, 30	- cantitățile de combustibil

Răspunsul va fi dat sub forma:

Se pleacă de la stația 3 spre stația 2

**CLASA a XI a si a XII a**

**Problema 2. (100 puncte)**

Role de banda

La un centru de calcul (ramas în urma cu vreo 30 de ani) exista  $N$  benzi magnetice, numerotate de la 1 la  $N$ . Fiecare banda este asezata pe o rola; nu exista doua benzi asezate pe aceeași rola. Rolele sunt în numar de  $N+1$ , numerotate de la 0 la  $N$ , existând o rola goala. Fiecare banda având un început si un sfârșit, ea poate fi înfasurata în doua moduri pe rola: cu începutul în interior sau cu începutul în exterior.

În urma utilizarii, benzile s-au amestecat pe role. Este necesar deci sa fie readuse la locul lor, adica banda  $i$  trebuie readusa pe rola  $i$  si înfasurata cu începutul în exterior. În acest scop, singura operatie permisa este transferul unei benzi de pe rola pe care se afla pe rola goala; în urma transferului sensul de înfasurare se inverseaza, adica daca banda era cu începutul în exterior ea ajunge cu începutul în interior si viceversa. Se cere, daca este posibil, gasirea unei succesiuni de transferuri astfel încât în final fiecare banda sa ajunga pe rola corespunzatoare.

**Intrarea:**

Datele de intrare se citesc din fisierul **role.in** având urmatorul format:

- pe prima linie se gaseste numarul  $N$  de benzi
- pe fiecare din urmatoarele  $N$  linii se gasesc câte doua numere naturale reprezentând asezarea câte unei benzi: a  $i$ -a linie (dintre cele  $N$ ) specifica asezarea benzii cu numarul  $i$ , primul numar fiind numarul rolei pe care este asezata, iar al doilea este 0 daca banda este cu începutul în exterior si 1 daca banda este cu începutul în interior.

*Limite:  $N \leq 20000$ .*

**Iesirea:**

În fisierul **role.out** se va scrie:

## OLIMPIADA JUDETEANA DE INFORMATICA

- pe prima linie numărul  $T$  de transferuri
- pe următoarele  $T$  linii se va scrie câte un număr reprezentând numărul rolei de pe care se transfera banda către rola goală.

Numărul  $T$  de transferuri nu va depăși 100000.

Dacă nu există soluție, atunci fișierul de ieșire va conține doar cuvântul IMPOSIBIL.

.