



# Concursul INTERJUDE- ȚEAN de programare INFO-OLTENIA 2005

În perioada 18-20 februarie a avut loc la Târgu-Jiu ediția din acest an a concursului interjudețean de programare Info-Oltenia. În continuare vă vom prezenta enunțurile celor zece probleme propuse spre rezolvare la acest concurs.

## Clasa a IX-a

### P050215: Șir

Se dau  $n$  șiruri formate din cifre cuprinse între '1' și '8' (inclusiv), de lungime maximă 500.

Toate șirurile se repetă de  $k$  ori sau de multiplu de  $k$  ori, cu excepția unui singur șir care nu este repetat de  $k$  ori sau de multiplu de  $k$  ori.

Se cere să se afișeze șirul care nu se repetă de  $k$  ori sau de multiplu de  $k$  ori.

#### Date de intrare

Fișierul de intrare **sir.in** conține pe prima linie, valorile  $n$  și  $k$  separate printr-un spațiu.

Pe următoarele  $n$  linii se află șirurile, elementele șirurilor fiind despărțite printr-un spațiu.

#### Date de ieșire

Fișierul de ieșire **sir.out** trebuie să conțină șirul căutat.

#### Restricții și precizări

Numărul de șiruri  $n$  este întreg și este cuprins între 1 și 32.000.

Numărul  $k$  este întreg și este cuprins între 1 și 2005.

#### Exemplu

**sir.in**

```
7 2
1 2 2
1 2
2 1
1 2
1 2 2
2 1
1 2
```

**sir.out**

```
1 2
```

**Timp de execuție/test:** 1 secundă

### P050216: Biți

Ionel și Maria au devenit mari experți în ale criptologiei.

Ei au în față două șiruri de numere, dintre care unul reprezintă un șir criptat, iar al doilea o parolă și trebuie să determine pe baza acestora șirul inițial.

Cele două șiruri au aceeași lungime, iar metoda de decriptare a șirului

criptat este foarte simplă și constă în câțiva pași:

- se transformă toate numerele din șirul criptat în secvențe de opt biți fiecare (fiecare număr scris inițial în baza 10 se transformă în baza 2 reprezentându-l pe opt cifre); din aceste secvențe puse cap la cap se obține un șir de biți;
- se transformă toate numerele corespunzătoare parolei în secvențe de opt biți fiecare procedând ca la pasul anterior; din aceste secvențe puse cap la cap se obține un al doilea șir de biți;
- se parcurge cel de-al doilea șir de biți și, dacă pe o poziție se găsește un bit cu valoarea 0, se reține într-un alt șir de biți valoarea corespunzătoare din primul șir (un bit care poate avea valoarea 0 sau 1);
- șirul obținut la pasul anterior se împarte în secvențe de opt biți fiecare;
- fiecare secvență de opt biți se transformă într-un număr în baza 10.

Se cere ca pe baza unui șir criptat și a unei "parole" să se determine șirul inițial.



### Date de intrare

Fișierul de intrare **biti.in** conține pe prima linie un singur număr natural  $n$  care reprezintă lungimea șirului criptat, precum și lungimea parolei.

Următoarea linie conține  $n$  numere separate între ele prin spații, care reprezintă elementele șirului criptat.

Următoarea linie conține  $n$  numere separate între ele prin spații, care reprezintă elementele parolei.

### Date de ieșire

Fișierul de ieșire **biti.out** trebuie să conțină pe prima linie numerele din șirul rezultat, separate între ele prin spații.

### Restricții și precizări

Numărul  $n$  de numere dintr-un șir este întreg și este cuprins între 1 și 10.000.

Reprezentarea binară a numerelor din fișierul de intrare nu va depăși 8 biți (fiecare număr este cuprins între 0 și 255 inclusiv).

Numărul biților care se obțin în urma procedurii de decriptare va fi multiplu de 8.

### Exemplu

**biti.in**

```
3
144 255 167
131 91 255
```

**biti.out**

```
39
```

### Explicație

Primul șir se transformă în secvența de biți:

```
10010000 11111111 10100111
```

Al doilea șir se transformă în secvența de biți:

```
10000011 01011011 11111111
```

În urma decriptării rezultă șirul de biți: 00100111 care se transformă într-un șir format dintr-un singur element: 39.

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

## Clasa a X-a

### P050217: Soldați

Se consideră  $n$  soldați numerotați de la 1 la  $n$  (soldații sunt numerotați în ordinea înălțimilor). Căpitanul acestora este indignat deoarece în fiecare zi soldații se așează în altă ordine decât cea a înălțimilor. Căpitanul are o mare pasiune pentru numere și a reușit să învețe numerele de ordine ale soldaților, astfel încât, în loc să pronunțe numele, poate foarte bine să spună numărul de ordine al oricărui dintre ei și are și o memorie foarte bună. Acesta, la sfârșitul zilei poate să reconstituie ordinea în care soldații au fost aranjați în ziua respectivă.

Căpitanul și-a propus să abordeze elemente noi ale algebrei, astfel încât el a descoperit că soldații săi se pot aranja în  $n!$  moduri și că toate aceste aranjări pot fi sortate lexicografic. Deoarece sunt foarte multe posibilități de aranjare a soldaților, acesta dorește să afle dacă într-o anumită zi soldații s-au aranjat la fel cum s-au aranjat într-o zi precedentă și a descoperit că ar fi bine să determine numărul aranjării din ziua respectivă pentru a-i fi mai ușor la căutat.

Ajutați-l pe căpitan să determine acest număr de ordine.

### Date de intrare

Fișierul de intrare **soldati.in** conține pe prima linie un singur număr natural  $n$  care reprezintă numărul de soldați pe care-i are în subordine căpitanul. Următoarea linie conține  $n$  numere distincte din mulțimea  $\{1, 2, \dots, n\}$  separate între ele prin spații, care reprezintă numerele de ordine ale soldaților.

### Date de ieșire

Fișierul de ieșire **soldati.out** trebuie să conțină pe prima linie numărul de ordine al aranjării soldaților din ordonarea lexicografică.

### Restricții și precizări

Numărul soldaților este cuprins între 1 și 100. Aranjarea  $(1, 2, \dots, n)$  are numărul de ordine 1, iar aranjarea  $(n, \dots, 2, 1)$  are numărul de ordine  $n!$ .

### Exemplu

**soldati.in**

```
4
4 1 3 2
```

**soldati.out**

```
20
```

**Timp de execuție/test:** 1 secundă

### P050218: Frații

Fie un șir de fracții  $a_1/b_1, a_2/b_2, \dots, a_n/b_n$ . Aveți la dispoziție un număr nelimitat de perechi de paranteze și simboluri  $+$  și  $*$ .

Determinați valoarea maximă a expresiei matematice corecte care poate fi obținută adăugând simbolurile de mai sus între fracțiile date. Frațiile trebuie să apară în expresia finală în ordinea în care sunt date în fișierul de intrare. De exemplu, pentru fracțiile:  $3/2, 5/2, 20/1$  și  $1/2$  expresia de valoare maximă este:

$$(3/2 + 5/2) * (20/1 + 1/2) = 82$$

### Date de intrare

Fișierul de intrare **fractii.in** conține pe prima linie numărul  $n$ .

Pe a doua linie a fișierului se află  $n$  numere naturale separate prin câte un spațiu, reprezentând șirul numărătorilor  $a_i$ , iar pe a treia linie a fișierului se află alte  $n$  numere naturale separate prin câte un spațiu, reprezentând șirul numitorilor  $b_i$ .

### Date de ieșire

Fișierul de ieșire **fractii.out** conține rezultatul cu cinci zecimale exacte.

### Restricții și precizări

Numărul de fracții este cuprins între 1 și 50. Numitorii și numărătorii fracțiilor sunt numere întregi cuprinse între 1 și 20.

### Exemplu

**fractii.in**

```
4
3 5 20 1
2 2 1 2
```

**fractii.out**

```
82.00000
```

**Timp de execuție/test:** 1 secundă

### P050219: Ecuație

Fie  $A$  și  $B$  două matrice pătratică de ordin  $n$  conținând numere reale. Fie ecuația  $A \times X = B$ , unde  $X$  este o matrice pătratică de ordin  $n$ .

Să se calculeze determinantul matricei  $X$ .

#### Date de intrare

Fișierul de intrare **ecuatie.in** conține pe prima linie ordinul celor trei matrici  $n$ .

Pe următoarele  $n$  linii se află elementele matricei  $A$ , iar pe următoarele  $n$  linii se află elementele matricei  $B$ .

#### Date de ieșire

Fișierul de ieșire **ecuatie.out** va conține determinantul matricei  $X$  cu 2 zecimale exacte, dacă avem soluție unică, caracterul  $R$  dacă avem o infinitate de soluții sau caracterul  $E$  dacă ecuația nu are sens.

#### Restricții și precizări

Ordinul matricelor  $n$  este număr întreg și cuprins între 1 și 30.

Aranjarea  $(1, 2, \dots, n)$  are numărul de ordine 1, iar aranjarea  $(n, \dots, 2, 1)$  are numărul de ordine  $n!$ .

#### Exemplu

**ecuatie.in**

```
3
1 0 0
0 2 0
0 0 2
2 0 0
0 4 0
0 0 1
```

**ecuatie.out**

```
2.00
```

**Timp de execuție/test:** 1 secundă

### P050220: Livada

În livada lui *Vasile*, există  $N$  pomi și  $M$  cărări între pomi. El poate ajunge de la orice pom la orice alt pom. Într-o zi, din lipsă de activitate, *Vasile* s-a hotărât să planteze iarbă peste unele cărări.

El vrea ca după acoperirea cărărilor să poată ajunge de la orice pom la orice alt pom, dar lungimea totală a celor rămase să fie minimă.

Ajutați-l pe *Vasile* să realizeze acest lucru, calculând cele mai mici două valori posibile ale lungimii totale a cărărilor neacoperite.

#### Date de intrare

Fișierul de intrare **livada.in** conține pe prima linie numerele înregi  $N$  și  $M$ .

Pe fiecare dintre următoarele  $M$  linii se află câte trei valori, primele două reprezentând pomii aflați la capetele unei cărări, iar a treia lungimea acesteia.

#### Date de ieșire

Fișierul de ieșire **livada.out** va conține o singură linie pe care se vor afla cele două valori care trebuie determinate, separate între ele printr-un singur spațiu.

#### Restricții și precizări

Numărul de pomi din livadă este cuprins între 1 și 150.

Numărul total al cărărilor dintre pomi este cuprins între valorile 1 și 10.000.

Lungimea unei cărări între doi pomi este un număr întreg cuprins între valorile 1 și 16.000.

Pentru fiecare test se va acorda jumătate din punctaj pentru prima valoare care trebuie determinată și jumătate din punctaj pentru cea de-a doua valoare.

#### Exemplu

**livada.in**

```
5 8
1 2 8
2 3 2
1 4 2
4 5 12
3 5 7
5 1 2
2 4 9
```

**livada.out**

```
13 14
```

**Timp de execuție/test:** 1 secundă

### P050221: Tablou

Fie două tablouri bidimensionale  $A$  și  $B$  cu  $N$  linii și  $N$  coloane fiecare și operația  $\bullet$  astfel încât fiecare element  $C_{ij}$  al tabloului  $C = A \bullet B$  să fie calculat prin formula:

$$C_{ij} = (A_{i,1} \cdot B_{1,j} + A_{i,2} \cdot B_{2,j} + \dots + A_{i,n} \cdot B_{n,j}) \mod 9901,$$

unde  $x \mod y$  înseamnă restul împărțirii lui  $x$  la  $y$ , iar  $i$  și  $j$  iau valori de la 1 la  $N$  ( $C$  va avea, de asemenea,  $N$  linii și  $N$  coloane).

Fie  $A^K = A \bullet A \bullet \dots \bullet A$  (de  $K$  ori), unde  $A$  este un tablou cu  $N$  linii și  $N$  coloane. Pentru  $N, K$  și  $A$  date, se cere să se determine tabloul  $A^K$ .

#### Date de intrare

Fișierul de intrare **tablou.in** conține pe prima linie, numerele  $N$  și  $K$ , separate între ele printr-un singur spațiu. Pe fiecare dintre următoarele  $N$  linii se află câte  $N$  numere naturale, separate între ele prin câte un spațiu, reprezentând câte o linie a tabloului  $A$ .

#### Date de ieșire

Fișierul de ieșire **tablou.out** trebuie să conțină  $N$  linii, fiecare formată din câte  $N$  numere separate prin câte un spațiu, reprezentând câte o linie a tabloului  $A^K$ .

#### Restricții și precizări

Numărul  $N$  este cuprins între 1 și 75.

Numărul  $K$  este cuprins între 1 și 100.000.

Elementele tabloului  $A$  sunt numere întregi cuprinse între 1 și 16.000.

#### Exemplu

**tablou.in**

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

**tablou.out**

```
14 20 26
20 29 38
26 38 50
```

**Timp de execuție/test:** 1 secundă





## P050222: Joc

Ștefan a cumpărat un joc de curând. Prima etapă a jocului constă în realizarea unui contur cu ajutorul unui fir, care trebuie să treacă prin toate punctele marcate pe tabla de joc, în fiecare punct marcat aflându-se un singur steguleț.

Se știe că printr-un steguleț firul nu poate ajunge decât o singură dată.

Întrucât în interiorul conturului se vor plasa mai multe piese, firul nu trebuie să se suprapună.

Să se determine numerele de ordine ale stegulețelor care determină un contur.

### Date de intrare

Fișierul de intrare **joc.in** conține pe prima linie numărul  $n$ , care reprezintă numărul de stegulețe.

Pe fiecare dintre următoarele  $n$  linii se vor afla coordonatele punctelor. Pe linia  $(i + 1)$  a fișierului de intrare se află coordonatele punctului cu numărul de ordine  $i$ , separate printr-un spațiu.

### Date de ieșire

Fișierul de ieșire **joc.out** va conține pe prima linie numerele de ordine ale stegulețelor în ordinea în care determină conturul, despărțite prin câte un spațiu.

### Restricții și precizări

Numărul de stegulețe este cuprins între 1 și 10.000.

Coordonatele stegulețelor sunt numere întregi cuprinse între -100 și 100.

### Exemplu

**joc.in**

```
6
1 1
2 2
2 4
1 3
1 2
-1 1
```

**joc.out**

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

**Timp de execuție/test:** 1 secundă

## Echipe (avansați)

### P050223: Party

Directorul unei mari companii a hotărât să dea o petrecere. Fiecare angajat al companiei are asociat un coeficient de activitate socială. Nici un angajat nu își poate suferi șeful direct, deci președintele nu poate invita un angajat și un subaltern direct al său.

De asemenea, deoarece el organizează petrecerea, președintele trebuie să fie prezent.

Ajutați-l să organizeze petrecerea, astfel încât produsul coeficienților tuturor oaspeților de la petrecere să fie maxim.

### Date de intrare

Fișierul de intrare **party.in** conține pe prima linie numărul  $N$ .

Pe a doua linie a fișierului se află un șir notat cu  $P$ , având  $N$  elemente separate printr-un spațiu, cu semnificația că elementele  $P_i$  reprezintă numărul de identificare al șefului angajatului cu numărul  $i$ . Președintele va avea întotdeauna numărul de identificare 1, iar  $P_1$  va fi întotdeauna 0.

Pe a treia linie a fișierului se află un alt șir  $A$  cu  $N$  numere separate prin câte un spațiu,  $A_i$  reprezentând coeficientul celui de-al  $i$ -lea angajat.

### Date de ieșire

Fișierul de ieșire **party.out** va conține pe o singură linie produsul maxim al coeficienților invitaților modulo 9901.

### Restricții și precizări

Numărul total de angajați este cuprins între 1 și 1000.

Coeficienții angajaților sunt numere întregi cuprinse între 1 și 250.

### Exemplu

**party.in**

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

**party.out**

```
8
```

**Timp de execuție/test:** 1 secundă

## P050224: Gali

*Galii* dețin  $n$  sate legate între ele prin  $m$  cărări. Satele sunt identificate prin numere întregi cuprinse între 1 și  $n$ . Există  $m$  cărări care sunt folosite de *gali* pentru a transmite mesaje între sate despre pericolele care există în regiune. Inițial pot fi transmise mesaje între oricare două sate.

*Galii* doresc să afle care sunt satele, care dacă ar fi ocupate de romani, ar duce la imposibilitatea de a transmite un mesaj între două sate libere.

Romanii vor ataca o singură dată și vor ocupa un singur sat, cel mai vulnerabil. Sarcina voastră este de a-i ajuta pe *gali* să determine satele care, dacă ar fi ocupate ar duce la imposibilitatea comunicării dintre oricare două sate neocupate.

### Date de intrare

Fișierul de intrare **gali.in** conține pe prima linie două numere întregi,  $n$  și  $m$ , separate între ele printr-un singur spațiu, care reprezintă numărul de sate deținute de *gali* și legăturile dintre acestea.

Fiecare dintre următoarele  $m$  linii conține două numere care reprezintă două sate unite de o cărare.

### Date de ieșire

Fișierul de ieșire **gali.out** va conține pe prima linie un singur număr  $k$ , care reprezintă numărul de sate care, dacă ar fi ocupate, ar duce la imposibilitatea comunicării dintre alte sate.

Următoarea linie trebuie să conțină  $k$  numere, separate între ele prin spații, care reprezintă numerele de ordine ale satelor cu proprietatea din enunț.

În cazul în care  $k$  este 0, cea de-a doua linie va fi vidă.

### Restricții și precizări

Numărul de sate deținute de *gali* este cuprins între 1 și 1.000.

Numărul de cărări dintre sate este cuprins între 1 și 10.000.

Între două sate poate exista cel mult o cărare.

**Timp de execuție/test:** 1 secundă