



# PROBLEME propuse pentru REZOLVARE

În luna martie a avut loc faza județeană a Olimpiadei de Informatică. În cadrul acestui număr vă prezentăm enunțurile problemelor propuse spre rezolvare în cinci județe ale țării.

## P040101: Baza

Arad, clasa a IX-a

Se dau două numere întregi  $X$  și  $Y$  și se cere să se determine cea mai mică bază pentru  $X$  și o bază pentru  $Y$  astfel încât numerele  $X$  și  $Y$  reprezentate în cele două baze să reprezinte aceeași valoare dacă le citim în baza 10. Dacă există mai multe valori ale bazei lui  $Y$  care satisfac condiția, se va alege valoarea minimă. Să considerăm numerele 12 și 5. Evident, considerate în baza 10, aceste numere nu sunt egale. Dar, dacă 12 este considerat scris în baza 3 iar 5 ca un număr în baza 6, avem  $12_{(3)} = 5_{(10)}$ ,  $5_{(6)} = 5_{(10)}$ ; deci cele două numere sunt egale. Cu alte cuvinte 12 și 5 sunt egale dacă selectăm corect bazele în care sunt scrise.

### Intrare

Datele de intrare se vor citi din fișierul **INPUT.IN**. Pe fiecare linie a fișierului de intrare se află o pereche de numere întregi  $X, Y$  separate prin cel puțin un spațiu și care conțin cel mult 80 de cifre. Bazele asociate numerelor  $X$  și  $Y$  sunt cuprinse între 2 și 36 (inclusiv) și nu sunt neapărat egale. În reprezentarea numerelor, cifrele 0, 1, ..., 9 au interpretarea uzuală, iar literele A, B, ..., Z reprezintă cifrele cuprinse între 10 și 35. Toate datele sunt corecte.

### Ieșire

Pentru fiecare pereche de numere întregi de la intrare, se va scrie în fișierul **OUTPUT.OUT** o linie care va conține în ordine: bazax, bazay, separate prin cel puțin un spațiu. Dacă nu există soluție, linia va conține două valori zero (0 0).

**Timp de execuție:** 1 secundă/test

### Exemplu

| BAZA . IN |     | BAZA . OUT |
|-----------|-----|------------|
| 12        | 5   | 3 6        |
| 10        | A   | 10 11      |
| 12        | 34  | 17 5       |
| 123       | 456 | 0 0        |
| 1         | 2   | 0 0        |
| 10        | 2   | 2 3        |

## P040102: Numărul care lipsește

Arad, clasa a IX-a

Se dă un număr  $n$  și un șir de  $n - 1$  numere naturale distincte cuprinse între 1 la  $n$ . Să se afișeze numărul care lipsește.

### Intrare

Datele de intrare se citesc din fișierul **NUM.IN**. Pe prima linie a acestui fișier se află numărul  $n$  ( $1 \leq n \leq 10000000$ ), iar pe următoarele  $n - 1$  linii se vor afla cele  $n - 1$  numere.

### Ieșire

Fișierul de ieșire **NUM.OUT** va conține numărul care lipsește.

**Timp de execuție:** 60 secunde/test

### Exemplu

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

## P040103: Numere puternice

Arad, clasa a IX-a

Numerele puternice sunt acele numere naturale care au numărul divizorilor mai mare decât numărul divizorilor oricărui număr natural mai mic decât numărul respectiv. Cel mai mic număr puternic este 1; urmează 2, 4, 6 etc. Scrieți un program care determină al  $n$ -lea număr puternic ( $n < 70$ ). Numărul  $n$  se citește din fișierul **PUTERNIC.IN**, iar ieșirea va apărea în fișierul **PUTERNIC.OUT**.

**Timp de execuție:** 1 secundă/test

### Exemplu

| PUTERNIC . IN | PUTERNIC . OUT |
|---------------|----------------|
| 3             | 4              |





## P040104: Raport

Arad, clasa a IX-a

Se dau două numere naturale  $M$  și  $N$  ( $0 < M < N \leq 100000$ ). Se cere să se tipărească numărul  $M/N$  în formă zecimală completă (cu perioadă, dacă este cazul).

Datele de intrare se citesc din fișierul **RAPORT.IN** care conține pe o singură linie numerele  $M$  și  $N$ , despărțite printr-un spațiu.

Rezultatul se va tipări în fișierul **RAPORT.OUT** pe o singură linie sub forma

0.parte\_neperiodicr

sau

0.(parte\_periodicr)

sau

0.parte\_neperiodicr(parte\_periodicr)

după cum este cazul.

**Timp de execuție:** 1 secundă/test

### Exemple

|                  |                   |
|------------------|-------------------|
| <b>RAPORT.IN</b> | <b>RAPORT.OUT</b> |
| 13 25            | 0.52              |

|                  |                   |
|------------------|-------------------|
| <b>RAPORT.IN</b> | <b>RAPORT.OUT</b> |
| 73 505           | 0.1(4455)         |

|                  |                   |
|------------------|-------------------|
| <b>RAPORT.IN</b> | <b>RAPORT.OUT</b> |
| 1 3              | 0.(3)             |

## P040105: Printre puncte

Olt, clasa a IX-a

Se dă o rețea pătratică de  $n \times n$  puncte, astfel încât distanța dintre două puncte vecine pe orizontală sau verticală este identică.

În fiecare punct al rețelei se află un cartonaș pe care este scrisă o literă. Nu se face distincție între literele mari și literele mici.

- În rețea se formează dreptunghiuri având laturile paralele cu marginile rețelei, dreptunghiuri care au ca vârfuri punctele care conțin același caracter în colțuri. Un punct se consideră că face parte din dreptunghiul format respectând cerințele problemei, dacă se află pe marginile acestuia, în interiorul lui sau în colțuri. Să se afișeze coordonatele colțurilor stânga jos și dreapta sus pentru acele dreptunghiuri care conțin cel mai mare număr de caractere care se repetă (de exemplu, dacă există în dreptunghi patru caractere 'a', șapte caractere 'c' și un caracter 'd', atunci numărul caracterelor care se repetă este  $4 + 7 = 11$  caractere).
- Unele puncte din rețea nu pot fi incluse în dreptunghiuri, altele fac parte din mai multe dreptunghiuri. Să se precizeze numărul punctelor din rețea care nu fac parte din astfel de dreptunghiuri.
- Să se determine coordonatele acelor pătrate care se pot forma, nu neapărat cu laturile paralele cu marginile matricei și care nu conțin neapărat același caracter în col-

țuri, pentru care suma codurilor ASCII ale caracterelor din colțuri să fie maximă. Se va afișa suma codurilor maximă pentru pătratele găsite, precum și coordonatele acestora. (Nu contează ordinea de afișare a coordonatelor pătratului.)

Datele de intrare se citesc de la tastatură în ordinea:

- $n$  reprezentând dimensiunea rețelei ( $n \leq 10$ );
- caracterele rețelei - se va citi de fiecare dată câte o linie întreagă de caractere și între caractere nu există nici un spațiu.

Rezultatele se vor afișa pe ecran.

**Timp de execuție:** 1 secundă/test

### Exemplu

|                |                             |
|----------------|-----------------------------|
| <b>Intrare</b> | <b>Ieșire</b>               |
| n=4            | a) Nu exista dreptunghiuri. |
| abcd           | b) 16                       |
| efgh           | c) 438                      |
| ijkl           | (3,3), (3,4), (4,3), (4,4)  |
| mnop           |                             |

## P040106: Expresie

Olt, clasa a IX-a

Se dă un șir de caractere, cu maxim 255 de caractere. Să se determine cea mai lungă expresie corectă matematic din punct de vedere sintactic, obținută prin ștergere de caractere din șirul inițial.

Expresia nu conține decât operatorii binari: +, -, \*, /. Un operand din expresie este o literă mică sau o literă mare. Expresia nu conține paranteze.

Dacă există mai multe expresii corecte matematic de aceeași lungime maximă, se va afișa numai una singură.

Șirul de caractere se citește de la tastatură, iar expresia corectă matematic va fi afișată pe ecran.

**Timp de execuție:** 1 secundă/test

### Exemplu

|                |               |
|----------------|---------------|
| <b>Intrare</b> | <b>Ieșire</b> |
| a+b-c*d        | a+b-c*d       |

## P040107: Stație

Satu Mare, clasa a IX-a

Se dă o autostradă circulară cu  $n$  stații de benzină, numerotate de la 1 la  $n$ . O mașină parcurge tot traseul într-un singur sens, pornind de la o stație dată.

Să se afișeze de câte ori poate să parcurgă mașina traseul dat, dacă la fiecare stație de benzină poate procura combustibil. Dacă traseul dat nu poate fi parcurs nici măcar o dată se va afișa mesajul "Traseul nu poate fi parcurs".

Datele de intrare se citesc din fișierul **STATIE.TXT** care are următoarea structură:

- pe prima linie a fișierului se află un număr care reprezintă numărul de stații și unul care indică numărul de kilometri care pot fi parcurși cu un litru de benzină;





- pe linia a doua se află un număr întreg care reprezintă capacitatea tancului de benzină și un număr care reprezintă stația din care se pleacă;
- pe următoarele linii, pentru fiecare stație de benzină, sunt indicați numărul de litri de benzină care sunt depozitați în stația respectivă și distanța, în kilometri, până la stația următoare.

Pe ecran se va afișa numărul de parcurgeri ale traseului.

### Exemplu

**STATIE . TXT**      *Ieșire (ecran)*

3 10      2

40 2

10 55

10 50

13 45

### P040108: Factorial

*Satu Mare, clasa a IX-a*

Din fișierul **FACT . IN** se citește un număr natural  $N$  ( $1000 \leq N \leq 2000000000$ ) despre care se știe că reprezintă factorialul unui număr  $k$  ( $k! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot k$ ). Scrieți un program care determină și afișează în fișierul **FACT . OUT** numărul  $k$ .

### Exemple

**FACT . IN**      **FACT . OUT**

40320      8

**FACT . IN**      **FACT . OUT**

479001600      12

### P040109: Numere subprime

*Suceava, clasa a IX-a*

Se dă un număr natural  $n$  nenul format din cel mult nouă cifre. Să se determine toate numerele prime distincte care se pot forma cu cifre aflate pe poziții consecutive ale numărului dat.

Fișierul de intrare **NUMAR . IN** conține numărul dat  $n$ , iar fișierul de ieșire **NUMAR . OUT** va conține toate numerele prime distincte care pot fi formate folosind cifre ale numărului  $n$  aflate pe poziții consecutive.

**Timp de execuție:** 2 secunde/test

### Exemplu

**NUMAR . IN**      **NUMAR . OUT**

131      3 13 31 131

### P040110: Cazemate

*Suceava, clasa a IX-a*

Un dispozitiv de apărare este format din  $n$  ( $1 \leq n \leq 10000$ ) cazemate așezate în linie și numerotate de la stânga la dreapta cu 1, 2, 3, ...,  $n$ . Pe un aliniament paralel cu acesta se deplasează două tancuri care încearcă să distrugă cazematele.

Primul tanc pleacă din dreptul cazematei 1 și se deplasează până în dreptul cazematei  $n$ , apoi se întoarce și așa mai departe.

Al doilea tanc pleacă din dreptul cazematei  $n$  și se deplasează până la cazemata 1, apoi se întoarce și așa mai departe.

Fiecare tanc parcurge spațiul dintre două cazemate consecutive într-o secundă.

Primul tanc trage o dată la  $p$  ( $1 \leq p \leq 500$ ) secunde, iar al doilea tanc trage o dată la  $q$  ( $1 \leq q \leq 500$ ) secunde, fiecare în cazemata în dreptul căreia se află.

Tancurile pot trage simultan în aceeași cazemată sau într-o cazemată deja distrusă.

Se cere să se determine timpul în care cele două tancuri distrug toate cazematele. Dacă tancurile nu pot distrugă cazematele, în fișierul de ieșire se va scrie 0.

Prima linie a fișierului de intrare **CAZEMATE . IN** conține numărul  $n$  al cazematelor, iar a doua linie conține numerele  $p$  și  $q$  care reprezintă intervalul de timp care trece între două tiruri ale celor două tancuri.

Fișierul de ieșire **CAZEMATE . OUT** va conține intervalul de timp care va trece până în momentul în care toate cazematele vor fi distruse sau valoarea 0 dacă nu este posibilă distrugerea lor.

**Timp de execuție:** 1 secundă/test

### Exemplu

**CAZEMATE . IN**      **CAZEMATE . OUT**

5      9

2 3

### P040111: Iceberg

*Timiș, clasa a IX-a*

Descriem un iceberg cu ajutorul unei matrice. Punctele marcate cu gri reprezintă pozițiile aparținând iceberg-ului. Dacă asupra lui suflă un vânt cald, el începe să se topească de pe margini spre interior.

Regula topirii este următoarea: într-un interval de timp se topește acea porțiune de gheață care are cel puțin în două vecinătăți aer (notate cu 1). Astfel se produc alte astfel de câmpuri (notate cu 2), care se vor topi în al doilea interval de timp ș.a.m.d.

Să generalizăm această problemă pentru trei dimensiuni, considerând iceberg-ul format prin suprapunerea a  $K$  astfel de matrice (în loc de pătrate lucrăm cu cuburi). În acest caz se va topi acea bucată de gheață, care din cele șase vecinătăți are în cel puțin trei aer.

Scrieți un program care pentru un iceberg dat, returnează în câte intervale de timp se topește întreg iceberg-ul, respectiv pentru fiecare interval câte câmpuri de gheață mai are iceberg-ul.

### Intrare

Numele fișierului de intrare se citește de la tastatură. Pe prima linie apar trei numere întregi, separate printr-un spa-





țiu, care reprezintă numărul de linii ( $1 \leq N \leq 40$ ), numărul de coloane ( $1 \leq M \leq 40$ ) și numărul de plane ( $1 \leq K \leq 40$ ) ale iceberg-ului. Urmează descrierea celor  $K$  plane. Fiecărui plan îi corespund  $N$  linii și pe fiecare linie sunt  $M$  cifre egale cu valoarea 0 dacă este aer sau valoarea 1 dacă este gheață pe acea poziție. Pe margini este sigur aer.

### Ieșire

Numele fișierului de ieșire se citește de la tastatură. Acesta va conține pe prima linie numărul de unități de timp în care se va topi toată gheața ( $T$ ). Următoarele  $T$  linii vor conține pe fiecare linie  $i$  numărul de unități de gheață existente la începutul intervalului de timp  $i$ . (La ultimul interval de timp vor exista 0 bucăți de gheață; această informație nu va mai fi scrisă în fișier.)

### Exemplu

| ICEBERG.IN | ICEBERG.OUT |
|------------|-------------|
| 4 5 4      | 2           |
| 00000      | 12          |
| 00000      | 4           |
| 00000      |             |
| 00000      |             |
| 00000      |             |
| 00000      |             |
| 01110      |             |
| 01110      |             |
| 00000      |             |
| 00000      |             |
| 01110      |             |
| 01110      |             |
| 00000      |             |
| 00000      |             |
| 00000      |             |
| 00000      |             |
| 00000      |             |
| 00000      |             |

### P040112: Concurs de dans

Timiș, clasa a IX-a

Un ansamblu are  $n$  fete și  $m$  băieți ( $1 \leq n \leq m \leq 25$ ), unii profilați pe dans modern, alții pe dans clasic.

În cadrul unui concurs, fiecărui component al ansamblului, în funcție de performanțele acestuia, i se atribuie un punctaj (diferit de al celorlalți) din intervalul  $(0, 25]$ .

Determinați  $n$  perechi (fată, băiat), astfel încât echipa să obțină punctaj maxim știind că pentru o pereche la care ambii concurenți sunt profilați pe același tip de dans, punctajul echipei va crește cu produsul punctajelor celor doi concurenți, iar dacă perechea este formată din concurenți profilați pe dansuri diferite, punctajul echipei va scădea cu produsul punctajelor celor doi concurenți. Să se afișeze punctajul maxim al echipei și perechile care se prezintă în concurs.

### Intrare

Numele fișierului de intrare se citește de la tastatură. Pe prima linie se află numărul  $n$  al fetelor, iar pe următoarele

$n$  linii se află informații despre punctaj, numărul de concurs și tipul de dans practicat de fiecare fată.

Pe următoarea linie se află numărul  $m$  al băieților, iar pe următoarele  $m$  linii se află informații despre punctaj, numărul de concurs și tipul de dans practicat de fiecare băiat.

Numărul de concurs este un număr natural format din maxim două cifre, iar tipul de dans este codificat cu un caracter: "m" pentru dans modern și "c" pentru dans clasic.

### Ieșire

Datele de ieșire se vor scrie într-un fișier text al cărui nume se va citi de la tastatură. Fișierul va conține, pe câte o linie, perechi de forma  $(nr\_concurs\_fată, nr\_concurs\_băiat)$ . Pe ultima linie se va scrie punctajul echipei.

**Timp de execuție:** 40 secunde/test

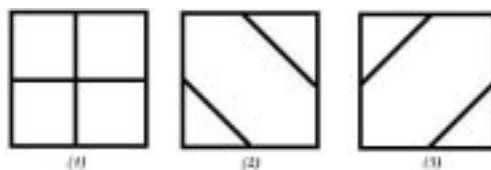
### Exemplu

| DANS.IN | DANS.OUT |
|---------|----------|
| 6       | (16,16)  |
| 2 21 c  | (21,29)  |
| 6 31 c  | (31,25)  |
| 10 64 c | (15,33)  |
| 8 62 c  | (62,21)  |
| 7 15 c  | (64,18)  |
| 1 16 c  | 287      |
| 9       |          |
| 3 17 c  |          |
| 9 21 c  |          |
| 8 33 c  |          |
| 7 25 c  |          |
| 4 24 c  |          |
| 1 62 c  |          |
| 5 16 c  |          |
| 10 18 c |          |
| 6 29 c  |          |

### P040113: Black

Arad, clasa a X-a

Se consideră o tablă de  $5 \times 5$  pătrățele și 25 de piese pătrate, fiecare piesă putând avea una din formele:



Se observă că piesa (1) are conectate marginile N-S și E-V, piesa (2) are conectate marginile N-E și S-V, iar piesa (3) are conectate marginile N-V și S-E. Cele două linii din piesa (1) nu se intersectează, ci trec "una pe sub cealaltă".

Se cere să se așeze cele 25 de piese pe tablă în așa fel încât să se obțină un drum care:

- să treacă prin fiecare pătrat exact o dată;
- să nu se autointersecteze;





- să pornească din colțul de NV al tablei (linia 1, coloana 1), începând de la exteriorul tablei (fie dinspre nord, fie dinspre vest);
- să se termine în colțul de SE al tablei (linia 5, coloana 5) și să părăsească tabla.

Fișierul de intrare **BLACK.IN** conține, pe o singură linie, trei numere, separate printr-un singur spațiu, reprezentând numărul de piese din tipurile (1), (2) și (3). Suma lor va fi întotdeauna 25.

În fișierul de ieșire **BLACK.OUT** se va tipări o matrice cu  $5 \times 5$  numere, separate prin spații, reprezentând tipul piesei plasate în fiecare pătrățel. Dacă există mai multe soluții, se va tipări una la alegere. Dacă nu există nici o soluție, fișierul **BLACK.OUT** va conține numai mesajul *Imposibil*.

**Timp de execuție:** 1 secundă/test

### Exemple

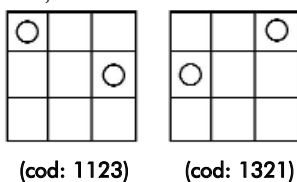
| BLACK.IN | BLACK.OUT |
|----------|-----------|
| 6 9 10   | 1 2 3 1 2 |
|          | 3 3 1 3 3 |
|          | 1 3 3 2 2 |
|          | 1 2 2 3 3 |
|          | 2 1 3 2 2 |

| BLACK.IN | BLACK.OUT |
|----------|-----------|
| 25 0 0   | Imposibil |

### P040114: Blatistul

Arad, clasa a X-a

În orașul B. biletele de autobuz au o grilă de  $N \times N$  pătrățele, dintre care aparatele de taxat compozitează (găuresc)  $K$  pătrățele. Biletul poate fi introdus în aparat numai cu un capăt (celălalt capăt este prins în cotor), dar poate fi introdus fie pe față, fie pe dos. În acest fel, unele din configurațiile posibile de  $K$  găuri în grila de  $N \times N$  pătrățele sunt oglindiri stânga-dreapta ale altor configurații. Iată un exemplu pentru  $N = 3$  și  $K = 2$ :



Un blatist veteran face colecție de bilete perforate și dorește să catalogheze toate configurațiile posibile de găuri, ignorând însă oglindirile (deoarece controlorii nu se supăra dacă un bilet este taxat pe dos). În acest scop, el codifică fiecare configurație printr-un șir de forma  $l_1c_1l_2c_2...l_kc_k$  unde  $(l_i, c_i)$  sunt coordonatele găurii  $i$  relativ la colțul din stânga-sus al biletului. Găurile sunt enumerate de la stânga la dreapta și de sus în jos. De exemplu, cele două configurații de mai sus sunt codificate prin "1123" și "1321". Acum blatistul este capabil să elimine oglindirile după următorul criteriu: dacă o configurație este oglindirea alteia, dintre cele două se va alege numai cea a cărei codificare are

o poziție lexicografică mai mică, cealaltă ignorându-se. Dintre cele două configurații de mai sus, "1123" are o poziție lexicografică minimă.

Fișierul de intrare **BILET.IN** conține numerele  $N$  ( $1 \leq N \leq 8$ ) și  $K$  ( $1 \leq K \leq 3$ ), separate printr-un singur spațiu.

Fișierul de ieșire **BILET.OUT** va conține, pe fiecare linie, codificarea unei configurații. Configurațiile vor apărea în ordine lexicografică deoarece blatistul trebuie să poată găsi cât mai repede orice bilet.

**Timp de execuție:** 1 secundă/test

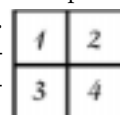
### Exemplu

| BILET.IN | BILET.OUT |
|----------|-----------|
| 3 2      | 1112      |
|          | 1113      |
|          | 1121      |
|          | 1122      |
|          | 1123      |
|          | 1131      |
|          | 1132      |
|          | 1133      |
|          | 1221      |
|          | 1222      |
|          | 1231      |
|          | 1232      |
|          | 2122      |
|          | 2123      |
|          | 2131      |
|          | 2132      |
|          | 2133      |
|          | 2231      |
|          | 2232      |
|          | 3132      |
|          | 3133      |

### P040115: Fractali

Arad, clasa a X-a

Se consideră pătratul de latură 1 care are colțurile în punctele de coordonate  $(0, 0)$ ,  $(1, 0)$ ,  $(1, 1)$  și  $(0, 1)$ . Împărțim acest pătrat în patru părți egale și numerotăm sferturile în ordinea indicată de figura alăturată.



Acest procedeu se reia pentru fiecare dintre pătrățele obținute; rezultă 16 pătrate, fiecare având atașat câte un cod format din două cifre.

De exemplu, pătratul care are codul 24 are colțurile în punctele de coordonate  $(0.75, 0.5)$ ,  $(1, 0.5)$ ,  $(1, 0.75)$  și  $(0.75, 0.75)$ . Configurația obținută este următoarea:

|    |    |    |    |
|----|----|----|----|
| 11 | 12 | 21 | 22 |
| 13 | 14 | 23 | 24 |
| 31 | 32 | 41 | 42 |
| 33 | 34 | 43 | 44 |





Conform aceluiași procedeu fiecare pătrățel se reîmparte, obținând pătrățele care au coduri formate din trei cifre ș.a.m.d. Problema cere ca, indicându-se două pătrate  $P_1$  și  $P_2$  prin codurile lor, să se determine dacă pătratele sunt:

- identice;
- $P_1$  inclus în  $P_2$ , fără ca cele două pătrate să aibă un punct comun;
- $P_2$  inclus în  $P_1$ , fără ca cele două pătrate să aibă un punct comun;
- $P_1$  inclus în  $P_2$ , astfel încât cele două pătrate să aibă puncte comune;
- $P_2$  inclus în  $P_1$ , astfel încât cele două pătrate să aibă puncte comune;
- exterioare unul altuia, dar cu cel puțin două puncte comune;
- exterioare unul altuia, dar cu un singur punct (vârf) comun;
- exterioare unul altuia, dar fără să aibă nici un punct comun.

În continuare vă prezentăm câteva exemple pentru fiecare tip de pereche:

- perechi de tipul a): 31 și 31;
- perechi de tipul b): 141 și 1;
- perechi de tipul c): 2 și 41;
- perechi de tipul d): 124 și 1;
- perechi de tipul e): 4 și 44;
- perechi de tipul f): 1 și 2;
- perechi de tipul g): 1 și 4;
- perechi de tipul h): 11 și 44.

Fișierul de intrare **PATRAT.IN** conține două linii, pe fiecare dintre ele aflându-se codul unui pătrat. Aceste coduri conțin cel mult 10000 de cifre.

Fișierul de ieșire **PATRAT.OUT** va conține tipul perechii formate de cele două pătrate.

Tipul se va codifica printr-o literă mare cuprinsă între A și H.

**Timp de execuție:** 1 secundă/test

#### Exemplu

**PATRAT.IN**  
1444444444  
3222222

**PATRAT.OUT:**  
F

#### P040116: Numărare de funcții

Arad, clasa a X-a

Să se determine numărul de funcții surjective  $f: \{a_1, a_2, \dots, a_n\} \rightarrow \{-1, 0, 1\}$  cu proprietatea  $\sum_{i=1}^n |f(a_i)| = s$ .

Prima linie a fișierului de intrare **FUNCȚIE.IN** conține valorile  $n$  și  $s$  ( $1 \leq n, s \leq 50$ ), separate printr-un singur spațiu.

Fișierul de ieșire **FUNCȚIE.OUT** va conține o singură linie pe care se va afla numărul funcțiilor care au proprietatea cerută.

**Timp de execuție:** 1 secundă/test

#### Exemplu

**FUNCȚIE.IN**  
5 3

**FUNCȚIE.OUT:**  
60

#### P040117: Speologi

Olt, clasa a X-a

Un grup de speologi a descoperit o peșteră în Munții Apuseni, formată din cel puțin două încăperi care nu comunică între ele. Unul dintre speologi își notează date despre peșteră.

Pentru aceasta, folosește echivalența cu o zonă dreptunghiulară cu  $m$  linii și  $n$  coloane, cu celule identice și pereți între celule (colțul din stânga sus al zonei are coordonatele 1, 1).

În fiecare celulă speologul înscrie un număr care se obține astfel: se pornește de la valoarea 0 și se adaugă 1 dacă există perete spre vest, apoi se adaugă 2 dacă există perete spre nord, 4 dacă există perete spre est și 8 dacă există perete spre sud.

La sfârșitul explorării, speologii vor să determine câte încăperi au descoperit, din câte celule este formată cea mai mare încăpere și ce perete trebuie să spargă, astfel încât să se creeze o încăpere de arie maximă, propice vizitatorilor.

Aria unei încăperi este dată de numărul de celule componente.

Datele de intrare se citesc din fișierul **PESTERA.IN** astfel:

- prima linie conține valorile  $m$  și  $n$  ( $1 \leq m, n \leq 10$ ) reprezentând dimensiunile zonei dreptunghiulare; ele sunt separate printr-un spațiu;
- pe următoarele  $m$  linii se află  $n$  valori cuprinse între 0 și 15 reprezentând codificările celulelor; ele sunt separate prin spații.

Datele de ieșire se scriu în fișierul **PESTERA.OUT** astfel:

- pe prima linie numărul de încăperi;
- pe a doua linie numărul de celule din încăperea cea mai mare;
- pe a treia linie patru numere  $11 \ c1 \ 12 \ c2$  reprezentând coordonatele celulelor între care se sparge peretele.

**Timp de execuție:** 2 secunde/test

#### Exemplu

**PESTERA.IN**  
3 3  
15 15 15  
15 15 15  
15 15 15

**PESTERA.OUT**  
9  
1  
1 1 1 2

#### P040118: Timbre

Olt, clasa a X-a

Gigel este angajat la Oficiul Poștal din cartier. Zilnic, el trebuie să timbreze plicurile care urmează a fi expediate în străinătate.





În funcție de destinație, pentru fiecare plic în parte, el trebuie să pună timbre de o anumită valoare  $S$  ( $1 \leq S \leq 5000$ ).

Gigel și-a propus să folosească, pentru fiecare plic, cât mai puține timbre. Timbre de aceeași valoare pot fi lipite de mai multe ori pe același plic, respectând condiția ca suma valorilor timbrelor lipite pe plic să fie exact  $S$ , iar numărul timbrelor folosite să fie minim.

Dacă nu găsește o combinație de timbre ale căror valori însumate să fie  $S$ , plicul respectiv este dat șefului.

La dispoziția lui Gigel există cantități suficiente pentru orice fel de timbru.

Datele de intrare se citesc din fișierul **TIMBRE . IN** astfel:

- prima linie conține numărul  $n$ , reprezentând numărul de categorii de timbre disponibile ( $n \leq 50$ ); prin categorie se înțelege timbre de aceeași valoare;
- pe următoarea linie se află  $n$  valori întregi separate prin spații care reprezintă valorile timbrelor din fiecare categorie disponibilă. Valoarea maximă a unui timbru este 1000.

Datele de ieșire se vor scrie în fișierul **TIMBRE . OUT** astfel:

- pe prima linie numărul  $m$  de timbre lipite de Gigel;
- pe următoarea linie  $m$  valori, separate prin spații, reprezentând valorile timbrelor alese.

În cazul în care plicul a fost trimis șefului, fișierul va conține o singură linie, cu valoarea 0.

**Timp de execuție:** 2 secunde/test

### Exemplu

| TIMBRE . IN | TIMBRE . OUT |
|-------------|--------------|
| 5           | 4            |
| 2 3 5 6 8   | 2 5 8 8      |
| 23          |              |

### P040119: Frazе

*Satu Mare, clasa a X-a*

În fișierul **CUVINTE . IN** se află un număr natural  $n$  (pe prima linie), iar pe următoarele  $n$  linii se găsesc cuvinte (pe fiecare linie se află un cuvânt despărțit în silabe cu ajutorul caracterului '-').

Cuvintele conțin numai majuscule. Spunem că două cuvinte rimează dacă au cel puțin două litere de la sfârșit identice.

Se cere să se distribuie cuvintele în grupe care rimează și să se afișeze frazele ce conțin număr maxim de silabe și pentru care cuvântul aflat pe poziția  $i$  rimează cu cel aflat pe poziția  $i + 2$ , iar cuvintele care rimează vor fi ordonate alfabetic. Fiecare frază trebuie să conțină cel puțin trei cuvinte.

Rezultatele vor fi afișate în fișierul **FRAZE . OUT**. Pe fiecare dintre primele linii se vor scrie cuvintele dintr-o grupă separate prin spații (fiecare grupă pe o linie). Urmează o linie vidă și pe următoarele linii se află câte o frază.

### Exemplu

| CUVINTE . IN | FRAZE . OUT                     |
|--------------|---------------------------------|
| 15           | TALISMAN GERMAN UMAN            |
| TA-LIS-MAN   | NICI                            |
| NICI         | GENIUL                          |
| GE-NIUL      | MACAR DOAR                      |
| MA-CAR       | LACAT                           |
| GER-MAN      | UN                              |
| LA-CAT       | TREI                            |
| U-MAN        | INFORMATI                       |
| DOAR         | ALGORITM                        |
| UN           | CARMEN                          |
| TREI         | CLARA                           |
| IN-FOR-MATI  |                                 |
| AL-GO-RITM   | GERMAN DOAR TALISMAN MACAR UMAN |
| U-NIC        |                                 |
| CAR-MEN      |                                 |
| CLA-RA       |                                 |

### P040120: Teren

*Satu Mare, clasa a X-a*

Locuitorii unui oraș caută un teren pentru construirea unui aeroport. Ei au la dispoziție harta regiunii.

Această hartă are formă dreptunghiulară și este împărțită în pătrate având lungimea laturii de o unitate.

Fiecare pătrat se identifică printr-o pereche de coordonate  $(x, y)$ , unde  $x$  este coordonata orizontală (vest-est), iar  $y$  este coordonata verticală (sud-nord). Pe hartă este indicată altitudinea fiecărui pătrat.

Găsiți o regiune dreptunghiulară de pătrate având aria cea mai mare (adică un teren dreptunghiular format din cel mai mare număr de pătrate) astfel încât:

- cea mai mare diferență dintre pătratul cu cea mai mare altitudine și pătratul cu cea mai mică altitudine de pe teren este mai mic sau egal cu o valoare dată  $C$  ( $0 \leq C \leq 10$ );
- lățimea (măsurată de-a lungul direcției vest-est) terenului este cel mult 100.

În cazul în care există mai multe terenuri ce satisfac cerințele problemei, se va afișa unul singur.

$U$  și  $V$  ( $1 \leq U, V \leq 700$ ) reprezintă dimensiunile hărții.  $U$  indică numărul pătratelor pe direcția vest-est, iar  $V$  indică numărul pătratelor pe direcția sud-nord.

Numărul întreg  $H_{xy}$  ( $-30000 \leq H_{xy} \leq 30000$ ) reprezintă altitudinea pătratului având coordonatele  $(x, y)$ ,  $1 \leq x \leq U$ ,  $1 \leq y \leq V$ ;

Numele fișierului de intrare este **LAND . INP**. Pe prima linie se află trei numere întregi:  $U$ ,  $V$  și  $C$ .

Fiecare dintre următoarele  $V$  linii conține câte  $U$  numere întregi reprezentând valorile  $H_{xy}$  pentru fiecare pătrățel.

Fișierul de ieșire **LAND . OUT** va conține o linie pe care se vor scrie patru numere întregi, localizând terenul determinat:  $Xmin$ ,  $Ymin$ ,  $Xmax$ ,  $Ymax$ , unde  $(Xmin, Ymin)$  sunt coordonatele pătratului din colțul sud-vest al terenului, iar  $(Xmax, Ymax)$  sunt coordonatele pătratului din colțul nord-est al terenului.





## Exemplu

LAND . INP

```
10 15 4
41 40 41 38 39 39 40 42 40 40
39 40 43 40 36 37 35 39 42 42
44 41 39 40 38 40 41 38 35 37
38 38 33 39 36 37 32 36 38 40
39 40 39 39 39 40 40 41 43 41
39 40 41 38 39 38 39 39 39 42
36 39 39 39 39 40 39 41 40 41
31 37 36 41 41 40 39 41 40 40
40 40 40 42 41 40 39 39 39 39
42 40 44 40 38 40 39 39 37 41
41 41 40 39 39 40 41 40 39 40
47 45 49 43 43 41 41 40 39 42
42 41 41 39 40 39 42 40 42 42
41 44 49 43 46 41 42 41 42 42
45 40 42 42 46 42 44 40 42 41
```

LAND . OUT

4 5 8 11

## P040121: Bile

Suceava, clasa a X-a

Într-un sistem ortogonal de coordonate  $xOy$  se dau  $p$  ( $1 \leq p \leq 1000$ ) bile prin coordonatele lor. Bilele se deplasează orizontal și vertical. Să se determine poziția  $k$  a unei linii orizontale pe care cele  $p$  bile să se alinieze una lângă alta, fără goluri între ele, astfel încât deplasarea bilelor până la linia de aliniere să fie minimă. Deplasarea unei bile este dată de numărul de poziții prin care trece aceasta. Două bile nu pot ocupa simultan aceeași poziție după aliniere.

Prima linie a fișierului de intrare **BILE . IN** conține numărul  $p$  al bilelor, iar următoarele  $p$  linii conțin coordonatele pozițiilor inițiale ale celor  $p$  bile. Coordonatele sunt numere întregi cuprinse între 0 și 255.

Fișierul de ieșire **BILE . OUT** va conține pe prima linie poziția  $k$  la care se face alinierea și pe a doua linie deplasarea minimă  $d$ .

**Timp de execuție:** 5 secunde/test

## Exemplu

BILE . IN

```
11
2 3
6 6
6 8
6 9
6 10
6 15
9 6
11 9
13 8
13 6
15 9
```

BILE . OUT

```
8
35
```

## P040122: Mărgele

Suceava, clasa a X-a

Se dau  $n$  ( $1 \leq n \leq 20$ ) mărgele de  $k$  ( $1 \leq k \leq 9$ ) culori;  $m_1$  mărgele de culoarea 1,  $m_2$  mărgele de culoarea 2, ...,  $m_k$  mărgele de culoarea  $k$ .

Să se determine toate modurile posibile de a se așeza cele  $n$  mărgele pe un șirag (circular), astfel încât să nu existe două mărgele alăturate de aceeași culoare. O mărgea este considerată prima în șirag, astfel permutările circulare sunt considerate soluții distincte.

De exemplu, pentru trei mărgele de trei culori diferite, 123 și 312 sau 231 sunt considerate soluții distincte și... mai sunt și altele!

Fișierul de intrare **MARGELE . IN** conține pe prima linie numărul  $n$  al mărgelilor, iar pe a doua linie valorile  $m_1, m_2, \dots, m_k$  reprezentând numărul de mărgele din cele  $k$  culori.

Fișierul de ieșire **MARGELE . OUT** va conține numărul  $h$  al soluțiilor.

**Timp de execuție:** 10 secunde/test

## Exemplu

MARGELE . IN

```
4
2 2
```

MARGELE . OUT

```
2
```

## Explicație

Cele două soluții din exemplul anterior sunt 1212 și 2121. Acestea nu se vor scrie în fișierul de ieșire.

## P040123: Audiență TV

Timiș, clasa a X-a

$N$  ( $1 \leq n \leq 20$ ) posturi de televiziune fac un sondaj pentru a afla perioadele de maximă audiență pe parcursul unei zile. În acest scop se prelucrează datele culese de la  $m$  ( $1 \leq m \leq 1000$ ) persoane.

Fiecare persoană furnizează numărul de intervale de timp pe care le va specifica și pentru fiecare interval un număr ce indică postul de televiziune și perioada de timp [ora de început, ora de sfârșit] în care s-a uitat la televizor. O persoană poate indica mai multe astfel de intervale în care a vizionat posturi diferite.

Se cere să se determine pentru fiecare post de televiziune perioadele de audiență maximă precum și postul sau posturile de televiziune cu audiență maximă.

Prima linie a fișierului de intrare **INP . TXT** conține numărul  $n$  al posturilor de televiziune și numărul  $m$  al telespectatorilor intervievați (cele două numere sunt separate printr-un singur spațiu). Fiecare dintre următoarele  $m$  linii conține răspunsurile unui spectator.

Primul număr de pe linie indică numărul  $p$  al intervalelor specificate, iar următoarele  $p$  grupuri de câte trei numere indică descrierea unui interval. Cele trei numere sunt  $t, i$  și  $s$ ; ele reprezintă postul de televiziune vizionat, ora de începere a vizionării, respectiv ora de terminare a vizionării.





Datele de ieșire se vor scrie în fișierul **OUT.TXT**. Primele  $n$  linii indică intervalele de audiență maximă pentru fiecare post.

Primul număr identifică postul, al doilea indică numărul telespectatorilor pe care i-a avut atunci când audiența sa a fost maximă și următoarele grupuri de două numere (separate prin virgulă) indică intervalele de timp în care postul a avut audiență maximă.

Ultima linie conține posturile care au avut cea mai mare audiență.

Numerele de pe aceeași linie a fișierului de ieșire sunt separate prin spații.

Posturile sunt numerotate de la 1 la  $n$ , intervalele de timp sunt specificate folosind numere naturale cuprinse între 0 și 24 și datele de intrare sunt corecte.

**Timp de execuție:** 1 secundă/test

### Exemplu

**INP.TXT**

```
3 4
3 1 11 15 2 15 20 1 21 24
3 3 10 14 2 15 18 3 20 22
3 1 10 12 2 14 18 1 20 24
2 3 10 15 1 18 20
```

**OUT.TXT**

```
1 2 11,12 20,20 21,24
2 3 15,18
3 2 10,14
2
```

### P040124: Puteri ale lui 2

*Timiș, clasa a X-a*

Se dă un număr  $n$  ( $1 \leq m \leq 1000$ ) și se cere să se scrie acest număr ca sumă (diferență) de oricâte puteri ale lui 2 și ca sumă (diferență) de număr minim de puteri ale lui 2. Expresia va avea forma:

$$n = 2^{k_0} \pm 2^{k_1} \pm \dots \pm 2^{k_m}$$

Valoarea  $n$  se va citi de la tastatură. Rezultatele se vor scrie în fișierele **A.TXT** și **B.TXT**.

Prima linie a acestor fișiere va conține numărul de termeni ai expresiei.

Următoarele  $m$  linii vor descrie câte un termen prin semnul său (+ sau -), urmat de puterea lui 2 corespunzătoare termenului respectiv.

**Timp de execuție:** 1 secundă/test

### Exemplu

**Intrare (tastatură)**

23

**A.TXT**

4  
+4  
+2  
+1  
+0

**B.TXT**

3  
+5  
-3  
-0

### P040125: Dreptunghi

*Arad, clasele a XI-a și a XII-a*

Se dă un dreptunghi ale cărui dimensiuni sunt numere naturale cuprinse între 1 și 100.

Scrieți un program care determină o descompunere a acestui dreptunghi în cât mai puține pătrate ale căror laturi să fie paralele cu laturile dreptunghiului. Într-un dreptunghi se poate executa o tăietură completă (de la o margine la alta) paralelă cu o latură. Tăieturile se pot executa și în dreptunghiurile formate, nu numai în dreptunghiul inițial.

Datele de intrare se citesc din fișierul **RECT.IN** care conține, pe o singură linie, dimensiunile  $n$  și  $m$  ale dreptunghiului, separate printr-un singur spațiu.

În fișierul de ieșire **RECT.OUT** se va scrie numărul minim de pătrate în care se poate descompune dreptunghiul inițial.

**Timp de execuție:** 1 secundă/test

### Exemple

**RECT.IN**

2 4

**RECT.OUT**

2

**RECT.IN**

3 2

**RECT.OUT**

3

### P040126: Parcurgere Euler

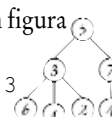
*Arad, clasele a XI-a și a XII-a*

Fie un arbore general cu rădăcina (un nod poate avea oricâți fii). Arborele are  $N$  noduri numerotate de la 1 la  $N$ . O parcurgere *Euler* a acestui arbore este realizată astfel:

- se tipărește rădăcina arborelui;
- pentru fiecare dintre fiii rădăcinii se parcurge după aceeași metodă subarborele care are drept rădăcină fiul respectiv și apoi se tipărește rădăcina.

Să considerăm arborele cu șapte noduri din figura alăturată.

Parcurgerea *Euler* a acestui arbore este 5 3 6 3 1 3 2 3 5 7 4 7 5.



Se dă un număr  $N$  și o succesiune de numere și se cere să se verifice dacă succesiunea reprezintă o parcurgere *Euler* corectă a unui arbore cu  $N$  noduri și, în caz afirmativ, să se reconstituie arborele.

Fișierul de intrare **EULER.IN** conține pe prima linie valoarea  $N$  și pe a doua linie o succesiune de numere (cuprinse între 1 și  $N$ ), separate prin spații.

Fișierul de ieșire **EULER.OUT** va conține pe prima linie unul dintre cuvintele DA (dacă succesiunea este o parcurgere *Euler*) și NU (dacă succesiunea nu este o parcurgere *Euler*).

În cazul în care succesiunea este corectă, următoarele  $N - 1$  linii vor conține câte două numere, separate prin spații, care vor reprezenta câte o muchie a arborelui. Ordinea de afișare a muchiilor nu contează.

**Timp de execuție:** 1 secundă/test





## Exemple

| EULER . IN                | EULER . OUT |
|---------------------------|-------------|
| 7                         | DA          |
| 5 3 6 3 1 3 2 3 5 7 4 7 5 | 5 3         |
|                           | 3 6         |
|                           | 3 1         |
|                           | 3 2         |
|                           | 5 7         |
|                           | 7 4         |

| EULER . IN              | EULER . OUT |
|-------------------------|-------------|
| 3                       | NU          |
| 1 2 3 1 2 3 1 2 3 1 2 3 |             |

## P040127: Modulo

Arad, clasele a XI-a și a XII-a

Se dau numerele  $A$ ,  $B$  și  $C$  ( $1 \leq A, B, C \leq 2000000000$ ) și se cere calcularea valorii  $(A^B) \bmod C$  (restul împărțirii lui  $A^B$  la  $C$ ).

Cele trei numere se citesc din fișierul **MOD . IN**;  $A$  se află pe prima linie,  $B$  pe a doua și  $C$  pe a treia.

Rezultatul (restul împărțirii lui  $A^B$  la  $C$ ) va fi scris în fișierul **MOD . OUT**.

**Timp de execuție:** 1 secundă/test

## Exemplu

| RECT . IN | RECT . OUT |
|-----------|------------|
| 2 4       | 2          |

## P040128: Scherzo

Arad, clasele a XI-a și a XII-a

O partitură muzicală este scrisă pentru  $V$  voci, fiecare voce având câte  $N$  note. Spunem că un solfegiu *se brodește* peste această partitură dacă solfegiul are lungimea de  $N$  note și fiecare din notele solfegiului coincide cu nota de pe poziția corespunzătoare a partituri, pe cel puțin una dintre voci.

De exemplu, solfegiul MI RE LA DO SI FA SOL LA se brodește următoarei partituri pe două voci care conține opt note:

Vocea I: MI RE DO DO SI FA FA SI

Vocea II: SOL SI LA DO SOL MI SOL LA

Notele 1, 2, 5 și 6 din solfegiu corespund primei voci, iar notele 3, 7, 8 corespund celei de-a doua voci. A patra notă (DO) corespunde ambelor voci.

Spunem că un solfegiu de o lungime oarecare *falsează* în  $K$  locuri dacă este nevoie de cel puțin  $K$  modificări asupra lui pentru a-l face să se brodească peste partitură. Modificările admise sunt:

- ștergerea unei note din solfegiu;
- adăugarea unei note în solfegiu;
- înlocuirea unei note în solfegiu.

De exemplu, solfegiul MI RE SOL DO SI FA SI falsează în două locuri, pentru că trebuie să modificăm nota SOL în LA și să inserăm încă un FA între SI și FA pentru a-l

aduce la forma MI RE LA DO SI FA FA SI care se brodește peste partitură.

Dumneavoastră sunteți profesor la conservator și, la examen, puneți studenții să reproducă o partitură la prima audiție, după care îi notați în funcție de cât de fidelă este reproducerea.

Evident, pentru a vă ușura munca, aveți nevoie de un program care să compare solfegiul (reproducerea) cu partitura și să spună dacă solfegiul se brodește sau în câte locuri falsează. În lumea prozaică a calculatoarelor, notele sunt reprezentate prin numere cuprinse între 1 și 100.

Fișierul de intrare **SOLFEGIU . IN** conține pe prima linie numărul  $V$  ( $1 \leq V \leq 100$ ) al vocilor și numărul  $N$  ( $1 \leq N \leq 100$ ) al notelor, separate printr-un spațiu. Următoarele  $V$  linii conțin câte  $N$  numere care reprezintă notele fiecărei voci. Următoarea linie conține numărul  $M$  ( $1 \leq M \leq 100$ ) al notelor din solfegiu, iar următoarea conține  $M$  numere întregi care reprezintă notele solfegiului.

Datele de ieșire se vor scrie în fișierul **SOLFEGIU . OUT** care va conține numărul minim de locuri în care falsează solfegiul. În cazul în care solfegiul se brodește peste partitură, în fișier se va scrie valoarea 0.

**Timp de execuție:** 1 secundă/test

## Exemplu

| SOLFEGIU . IN   | SOLFEGIU . OUT |
|-----------------|----------------|
| 2 8             | 2              |
| 3 2 1 1 7 4 4 7 |                |
| 5 7 6 1 5 3 5 6 |                |
| 7               |                |
| 3 2 5 1 7 4 7   |                |

## P040129: Foaie de calcul

Satu Mare, clasele a XI-a și a XII-a

O foaie de calcul (*spreadsheet*) este un tablou dreptunghiular de celule rectangulare. Fiecare celulă poate conține date sau expresii, care pot fi evaluate obținându-se date. Liniile din foaia de calcul sunt numerotate, începând cu 0, iar coloanele se numesc cu litere majuscule ale alfabetului englez (primele 26 de linii), apoi folosind combinații de litere (AA, AB etc). O celulă poate fi referită specificând coloana și linia corespunzătoare (de exemplu, prima celulă este A0).

O foaie de calcul de jucărie conține cel mult 26 de coloane (denumite de la A la Z) și cel mult 10 linii (numerotate de la 0 la 9). Fiecare celulă poate conține date de tip întreg sau expresii aritmetice în care se folosesc doar operatorii binari +, -, \*, /, cu semnificația de adunare, scădere, înmulțire, respectiv împărțire întreagă. Operanzii pot fi valori întregi sau referințe către celule.

Problema constă în a evalua, dacă este posibil, o foaie de calcul de jucărie dată. Prin evaluarea unei foi de calcul înțelegem înlocuirea tuturor expresiilor cu valorile lor.

Pe prima linie a fișierului de intrare **EXCEL . IN** sunt scrise două numere întregi  $n$  (reprezentând numărul de





linii din foaia de calcul) și  $m$  (reprezentând numărul de coloane), separate prin spațiu.

Fișierul conține în continuare  $n \times m$  linii de date, câte una pentru fiecare celulă. Conținutul celulelor (valoarea sau expresia) este specificat în ordinea crescătoare a liniilor, iar pe fiecare linie în ordinea alfabetică a coloanelor.

Fișierul de ieșire **EXCEL.OUT** va conține mesajul **EVALUARE IMPOSIBILA** sau foaia de calcul evaluată ( $n$  linii în ordinea dată, fiecare linie conținând cele  $m$  valori ale celulelor componente, în ordinea dată a coloanelor, separate prin spațiu).

### Observații

- Expresiile pot conține paranteze rotunde.
- Expresiile nu conțin spații.
- Prioritatea operatorilor este cea uzuală.
- Datele și valorile absolute ale expresiilor din foaia de calcul de jucărie sunt numere naturale mai mici decât două miliarde.
- Fiecare expresie conține cel mult 100 de caractere.
- Expresiile date în foaia de calcul de jucărie sunt sintactic corecte (din punctul de vedere al sintaxei *Pascal/C*).

**Timp de execuție:** 0.5 secunde/test

### Exemple

| EXCEL.IN | EXCEL.OUT           |
|----------|---------------------|
| 1 2      | EVALUARE IMPOSIBILA |
| 1+B0     |                     |
| 2+A0     |                     |

| EXCEL.IN | EXCEL.OUT  |
|----------|------------|
| 2 4      | 1 6 1 5    |
| 1        | 18 19 5 18 |
| 6        |            |
| 1        |            |
| 5        |            |
| 18       |            |
| A1+A0    |            |
| 5        |            |
| B0*3     |            |

### P040130: Plimbare

*Satu Mare, clasele a XI-a și a XII-a*

Doi îndrăgostiți se află într-un oraș  $k$  și au la dispoziție  $p$  ore ( $1 \leq p \leq 150$ ), timp în care să se plimbe cu mașina și apoi să ajungă fiecare la el acasă, fata în orașul  $i$ , iar băiatul în orașul  $j$ .

Ei doresc să se plimbe cât mai mult timp împreună, dar să ajungă în cel mult  $p$  ore, fiecare în orașul său. Mașina se deplasează cu viteză constantă (cea legal admisă).

Cei doi au la dispoziție o hartă cu  $n$  orașe ( $3 \leq V \leq 200$ ) printre care se află, evident, orașul de plecare  $k$  și orașele destinație  $i$  și  $j$ . Harta indică orașele legate prin șosele și durata exprimată în ore a parcurgerii cu viteză constantă (legal admisă) a distanței dintre două orașe. Să se determi-

ne traseul comun pe care îl vor parcurge cei doi îndrăgostiți astfel încât timpul petrecut împreună să fie maxim și să ajungă amândoi la propria destinație în cel mult  $p$  ore de la plecare.

### Observații

- Cronometrarea duratei plimbării se face începând cu momentul 0 din orașul de plecare  $k$ .
- În orice oraș unul dintre ei poate continua drumul separat, cu o altă mașină; (despărțirea nu necesită timp suplimentar).
- Pe fiecare șosea se circulă în ambele sensuri.
- După plecarea din orașul  $k$  nu se staționează nici un moment.
- Nu sunt permise întoarceri pe șosea sau parcurgeri incomplete ale unei șosele.

Pe prima linie a fișierului de intrare **MASINA.IN** se află două numere naturale  $n$  și  $m$  despărțite printr-un spațiu, reprezentând numărul de orașe de pe hartă și numărul de șosele ce leagă aceste orașe. Pe cea de a doua linie se află două numere naturale  $k$  și  $p$ , despărțite printr-un spațiu, reprezentând numărul de ordine al orașului de plecare și numărul de ore pe care îl au la dispoziție cei doi îndrăgostiți. Pe cea de a treia linie sunt scrise două numere naturale  $i$  și  $j$  despărțite printr-un spațiu, reprezentând numerele de ordine ale orașelor de destinație pentru cei doi. Pe următoarele  $m$  linii sunt scrise triplete de numere naturale  $a$ ,  $b$  și  $d$ , despărțite prin câte un spațiu, reprezentând faptul că între orașele  $a$  și  $b$  există o șosea directă cu durata  $d$  de parcurgere a acesteia.

Rezultatele se vor scrie în fișierul **MASINA.OUT** structurate pe două linii. Pe prima linie este scris numărul natural  $d_{max}$ , reprezentând durata maximă (în ore) în care cei doi se vor plimba împreună. Pe linia a doua sunt scrise numerele naturale:  $k, x_1, x_2, \dots$  reprezentând orașele pe care le vor vizita împreună, în ordine, începând cu orașul  $k$  de plecare (datele vor fi despărțite prin câte un spațiu). În cazul în care soluția optimă nu este unică se va afișa o singură soluție.

**Timp de execuție:** 4 secunde/test

### Exemplu

| MASINA.IN | MASINA.OUT |
|-----------|------------|
| 8 9       | 6          |
| 7 8       | 7 6 5 4 3  |
| 1 2       |            |
| 1 3 1     |            |
| 3 4 1     |            |
| 4 2 1     |            |
| 4 5 1     |            |
| 4 6 2     |            |
| 5 6 3     |            |
| 6 8 1     |            |
| 7 8 1     |            |
| 7 6 1     |            |





## P040131: Baterii

Suceava, clasele a XI-a și a XII-a

Direcția de informații a aflat prin agenții săi punctele de amplasare ale celor  $p$  ( $1 \leq p \leq 10000$ ) baterii de apărare antiaeriană inamice, dispuse într-un sistem ortogonal de coordonate  $xOy$ .

Un atac fulger poate distruge tot ce se află într-un dreptunghi de dimensiuni  $m \times n$ .

Să se determine poziția unui dreptunghi cu laturile paralele cu axele de coordonate, de dimensiuni  $m \times n$  ( $1 \leq m, n \leq 10000$ ) date, care conține în interior sau pe frontieră un număr maxim de baterii inamice.

Fișierul de intrare **BATERII.IN** conține pe prima linie dimensiunile  $m$  și  $n$  ale dreptunghiului, pe a doua linie numărul total  $p$  al bateriilor inamice și pe următoarele  $p$  linii coordonatele celor  $p$  baterii inamice.

Coordonatele bateriilor sunt numere întregi cuprinse între 0 și 255.

Fișierul de ieșire **BATERII.OUT** va conține pe prima linie coordonatele  $u$  și  $v$  ale colțului din stânga-jos al dreptunghiului, pe a doua linie numărul  $k$  al bateriilor inamice distruse și pe următoarele  $k$  linii coordonatele celor  $k$  baterii inamice distruse.

Dacă există mai multe soluții în fișierul de ieșire se va scrie una singură.

**Timp de execuție:** 4 secunde/test

### Exemplu

| BATERII.IN | BATERII.OUT |
|------------|-------------|
| 3 2        | 2 2         |
| 5          | 4           |
| 3 3        | 2 3         |
| 4 2        | 3 3         |
| 2 3        | 4 2         |
| 5 4        | 5 4         |
| 1 5        |             |

## P040132: Segmente disjuncte

Suceava, clasele a XI-a și a XII-a

Se dau  $n$  ( $1 \leq n \leq 1000$ ) segmente situate pe o dreaptă. Să se determine cardinalul maxim al unei mulțimi de segmente disjuncte de pe acea dreaptă.

Fișierul de intrare **SEGMENTE.IN** conține pe prima linie numărul  $n$  al segmentelor și pe următoarele  $n$  linii coordonatele celor  $n$  segmente, ordonate în funcție de coordonatele capătului din stânga.

Coordonatele segmentelor sunt numere întregi cuprinse între 0 și 255.

Fișierul de ieșire **SEGMENTE.OUT** va conține pe prima linie numărul maxim  $k$  al segmentelor disjuncte și, pe următoarea linie  $k$  numere întregi reprezentând numerele de ordine ale segmentelor rămase în mulțimea de cardinal maxim.

**Timp de execuție:** 1 secundă/test

### Exemplu

| SEGMENTE.IN | SEGMENTE.OUT |
|-------------|--------------|
| 4           | 3            |
| 1 3         | 1 3 4        |
| 2 7         |              |
| 4 5         |              |
| 6 8         |              |

## P040133: Poștașul

Timiș, clasele a XI-a și a XII-a

Se dă un cartier ale cărui blocuri sunt poziționate sub forma unei matrice de patru linii și  $N$  coloane.

Pentru un  $N$  dat să se determine numărul variantelor de traseu pe care le poate parcurge poștașul zonal, astfel încât să treacă pe la toate blocurile, o singură dată pe la fiecare.

Poștașul pleacă de la blocul de pe poziția (1, 1) spre blocul de pe poziția (1, 2) și trebuie să se întoarcă la blocul (1, 1).

Valoarea  $N$  ( $3 \leq N \leq 25$ ) se citește de la tastatură și rezultatul se afișează pe ecran.

### Exemple

Intrare (tastatură)

3

Ieșire (ecran)

2

Intrare (tastatură)

17

Ieșire (ecran)

1029864

## P040134: Multiplu

Timiș, clasele a XI-a și a XII-a

Scrieți un program care pentru un număr natural  $N$  ( $3 < N \leq 4999$ ) și pentru  $M$  cifre date  $X_1, X_2, \dots, X_M$  ( $0 < M \leq 9$ ,  $0 \leq X_i \leq 9$ ) să găsească cel mai mic multiplu strict pozitiv al lui  $N$  care conține doar cifrele  $X_1, X_2, \dots, X_M$  (cifrele  $X_1, X_2, \dots, X_M$  pot apărea în multiplu niciodată, o dată sau de mai multe ori).

Numele fișierului de intrare și al celui de ieșire se vor citi de la tastatură.

Fișierul de intrare conține mai multe seturi de date separate de câte o linie vidă. Fiecare set de date are următoarea formă:

- pe prima linie numărul  $N$ ;
- pe a doua linie numărul  $M$ ;
- pe următoarele  $M$  linii cifrele  $X_1, X_2, \dots, X_M$ .

Pentru fiecare set de date programul trebuie să tipărească în fișierul de ieșire, pe câte o linie, multiplul găsit, în cazul în care acesta există.

Dacă nu există soluție în fișierul de ieșire se va afișa doar valoarea 0.



## Exemple

| Intrare | Ieșire   |
|---------|----------|
| 3       | 110      |
| 22      | 0        |
| 3       | 20000999 |
| 7       |          |
| 0       |          |
| 1       |          |
| 2       |          |
| 1       |          |
| 1       |          |
| 4999    |          |
| 6       |          |
| 5       |          |
| 7       |          |
| 6       |          |
| 2       |          |
| 9       |          |
| 0       |          |

## P040135: Printre isteți

Olt, clasele a XI-a și a XII-a Verde Împărat a decis să-și căsătorească fiica. La curtea palatului sosesc mai mulți cavaleri care doresc să ceară mâna fetei.

Împăratul dorește să-l aleagă ca ginere pe cel mai isteț dintre ei. Pentru aceasta le cere să găsească soluția unei probleme pentru a cărei rezolvare se chinuie de mai mult timp. Se dă o matrice cu  $m$  linii și  $n$  coloane și se cere să se precizeze în câte moduri putem așeza valori  $-1$  și  $1$  astfel încât să se obțină pe fiecare linie și coloană produsul  $-1$ , dacă acest lucru este posibil. Cavalerii vă cer să-i ajutați la rezolvarea problemei.

Datele de intrare se citesc din fișierul **ISTET.IN** care conține dimensiunile  $m$  și  $n$  ( $1 \leq m, n \leq 250$ ) ale rețelei, separate printr-un singur spațiu.

Datele de ieșire se vor scrie în fișierul **ISTET.OUT** care va conține numărul de posibilități de amplasare a valorilor  $1$  și  $-1$  astfel încât să se obțină pe fiecare linie și coloană produsul  $-1$ . Dacă nu există nici un astfel de amplasament se va scrie valoarea  $0$ .

**Timp de execuție:** 2 secunde/test

### Exemplu

| ISTET.IN | ISTET.OUT |
|----------|-----------|
| 3 3      | 16        |

## P040136: Cărți

Olt, clasele a XI-a și a XII-a

$N$  elevi au decis să schimbe cărți între ei, respectând algoritmul următor: fiecare elev dă cărți la jumătate dintre elevii cunoscuți de el și primește cărți de la cealaltă jumătate de elevi care-l cunosc.

Se știe că fiecare elev are un număr par de cunoștințe și, un elev care a primit o carte de la un prieten nu poate să dea o carte la același elev.

Fiecare elev cunoaște cel puțin doi alți elevi.

Dacă elevul  $i$  îl cunoaște pe elevul  $j$ , atunci elevul  $j$  îl cunoaște și el pe elevul  $i$ .

Să se stabilească, pentru fiecare elev, care sunt elevii cărora trebuie să le dea cărți.

Datele de intrare se citesc din fișierul **CARTI.IN**. Pe prima linie se află numărul  $n$  al elevilor ( $3 \leq n \leq 50$ ).

Pe următoarele  $n$  linii care conțin valori, separate între ele prin spații, reprezentând elevii cunoscuți de fiecare elev.

Pe prima linie se află elevii cunoscuți de primul elev, pe a doua elevii cunoscuți de al doilea ș.a.m.d.

Datele de ieșire vor fi scrise în fișierul **CARTI.OUT**. Pentru fiecare elev vor fi scrise câte două linii. Prima dintre ele va conține numărul  $c$  al elevilor care primesc cărți de la elevul respectiv, iar pe a doua vor fi scrise  $c$  valori, separate prin spații, care identifică cei  $c$  elevi care primesc cărți de la elevul respectiv.

**Timp de execuție:** 2 secunde/test

### Exemplu

**CARTI.IN**

2

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

**CARTI.OUT**

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



probleme

Gințio nr. 4 - aprilie 2001

## Notă

Anumite enunțuri și exemple au fost modificate pentru a elimina unele neclarități.

## ONI 2001

Mult succes tuturor participanților la Olimpiada Națională de Informatică!!!