Rundele #26 - #40

În continuare vă vom prezenta enunțurile celor 15 probleme propuse spre rezolvare la rundele 26-40 ale ediției din acest an a concursului de programare Bursele Agora.

P030425: Perechi

Se consideră o pereche de numere naturale *a* și *b*. Asupra perechii pot fi efectuate următoarele transformări:
- (*a*, *b*) devine (*a* - *b*, *b*);
- (*a*, *b*) devine (*a* + *b*, *b*);
- (*a*, *b*) devine (*b*, *a*).

Să se determine numărul minim de transformări succesive care duc la apariția perechii (*c*, *d*).

Date de intrare

Fișierul de intrare **PAIRS.IN** conține două linii. Pe prima linie se află numerele *a* și *b* care formează perechea dată, iar pe cea de-a doua linie se află numerele *c* și *d* care formează perechea care trebuie obținută. Numerele de pe o linie sunt separate prin spații.

Date de ieșire

Fișierul de ieșire **PAIRS.OUT** trebuie să conțină o singură linie pe care se va afla numărul minim de transformări succesive necesare. În cazul în care perechea (*c*, *d*) nu poate fi obținută, valoarea scrisă trebuie să fie -1.

Restricție

- $0 \le a, b, c, d \le 1000$.

Exemplu

PAIRS.IN
```
2 2
2 4
```

PAIRS.OUT
```
2
```

Timp maxim de execuție/test: 1 secundă

P030426: Triunghi

Se consideră un număr natural *a*.

Să se determine numărul de triunghiuri dreptunghice distincte pentru care lungimea celei mai mici laturi este *a* și lungimile celorlalte laturi sunt numere întregi. Se consideră că un triunghi ale cărui laturi sunt *a*, *b* și *c* este identic cu triunghiul ale cărui laturi sunt *a*, *c* și *b*.

Date de intrare

Fișierul de intrare **TRIANGLE.IN** conține pe o singură linie numărul *a*.

Date de ieșire

Fișierul de ieșire **TRIANGLE.OUT** trebuie să conțină o singură linie pe care se va afla numărul de triunghiuri dreptunghice distincte pentru care lungimea celei mai mici laturi este *a* și lungimile celorlalte laturi sunt numere întregi.

Restricție

- $3 \le a \le 2.000.000.000$.

Exemplu

TRIANGLE.IN
```
9
```

TRIANGLE.OUT
```
2
```

Timp maxim de execuție/test: 0,5 secunde

P030427: Fotbal

Ciobănașul *Ion* se plictisește păzind oile și s-a gândit să diversifice activitățile. El a început să învețe un grup de oi să joace fotbal și vrea să împartă aceste oi în două echipe și să organizeze un meci între ele, dar pentru ca meciul să fie cât mai interesant el a impus restricția ca echipele să aibă valori cât mai apropiate posibil (deoarece meciul de fotbal este între oi, nu trebuie ca numărul de oi dintr-o echipă să fie egal cu numărul de oi din cealaltă echipă, dar fiecare oaie care știe fotbal trebuie repartizată în una dintre echipe).

Ciobănașul vrea să știe câte astfel de repartizări ale oilor în două echipe există astfel încât diferența de valoare dintre echipe să fie minimă și care este acea valoare.

Date de intrare

Fișierul de intrare **SOCCER.IN** conține pe prima linie numărul n al oilor care știu să joace fotbal, iar pe a doua linie conține n numere întregi, separate între ele prin spații, care reprezintă valorile fotbalistice pentru fiecare oaie.

Date de ieșire

Fișierul de ieșire **SOCCER.OUT** trebuie să conțină o singură linie pe care se vor afla două numere întregi separate între ele printr-un singur spațiu.

Primul număr reprezintă diferența minimă care se poate obține între valorile celor două echipe, iar cel de-al doilea număr reprezintă numărul de repartizări în două echipe ale oilor astfel încât diferența dintre valorile celor două echipe să fie minimă.

Restricții și precizări

- $2 \le n \le 32$;
- valoarea fotbalistică a unei oi este un număr întreg cuprins între 1 și 1000000.

Exemplu

SOCCER.IN

```
4
1 2 4 6
```

SOCCER.OUT

```
1 2
```

Timp maxim de execuție/test: 2 secunde

P030428: Dispozitive de comunicare

Agora GSM deține n dispozitive pentru comunicație identificate prin numere cuprinse între 1 și n.

Din nefericire nu este posibilă comunicația între oricare două dispozitive. Mai mult, dacă un mesaj poate fi transmis de la un dispozitiv x la un dispozitiv y, nu este sigur că poate fi transmis și un mesaj de la dispozitivul y la dispozitivul x.

Compania dorește să determine numărul minim al grupurilor care pot fi formate astfel încât să existe posibilitatea comunicării directe sau indirecte între oricare două dispozitive care fac parte din același grup.

Date de intrare

Prima linie a fișierului de intrare **DEVICES.IN** conține numărul n al dispozitivelor de comunicare și numărul m al legăturilor care pot fi stabilite între dispozitive. Fiecare dintre următoarele m linii va conține două numere întregi x și y cu semnificația: *există posibilitatea de a transmite un mesaj de la dispozitivul x la dispozitivul y*. Numerele de pe o linie vor fi separate prin spații.

Date de ieșire

Fișierul de ieșire **DEVICES.OUT** trebuie să conțină o singură linie pe care se va afla numărul minim al grupurilor care pot fi formate astfel încât să existe posibilitatea comunicării directe sau indirecte între oricare două dispozitive care fac parte din același grup.

Restricții și precizări

- $1 \le n \le 500$;
- $1 \le m \le 5000$;
- există posibilitatea ca un grup să fie format dintr-un singur dispozitiv;
- va exista întotdeauna cel puțin o gaură.

Exemplu

DEVICES.IN

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

DEVICES.OUT

```
2
```

Timp maxim de execuție/test: 2 secunde

P030429: Media aritmetică

Se consideră un șir a format din n numere întregi. Să se determine numărul k al valorilor a_i cu proprietatea că a_i este media aritmetică a celorlalte n - 1 numere din șir.

Date de intrare

Prima linie a fișierului de intrare **AVERAGE.IN** conține valoarea n.

Cea de-a doua linie conține n numere întregi, separate prin spații, care reprezintă valorile elementelor din șirul a.

Date de ieșire

Fișierul de ieșire **AVERAGE.OUT** trebuie să conțină pe prima linie numărul k determinat.

Cea de-a doua linie trebuie să conțină cele k poziții ale numerelor a_i care au proprietatea cerută. Acestea vor fi separate prin spații.

În cazul în care avem $k = 0$, cea de-a doua linie a fișierului de ieșire va fi vidă.

Restricții și precizări

- $1 \le n \le 1000000$;
- $1 \le a_i \le 1000$, $\forall 1 \le i \le n$;
- primul element al șirului a se află pe poziția 1.

Exemplu

AVERAGE.IN

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

AVERAGE.OUT

```
1
3
```

Timp maxim de execuție/test: 1 secundă

P030430: Keops

Doi hoți au reușit să intre în *Marea Piramidă* a lui *Keops*. Nici unul dintre ei nu știe de prezența celuilalt. Piramida poate fi privită ca un caroiaj cu m linii și n coloane.

Se cunosc coordonatele la care se află inițial hoții. În fiecare minut un hoț se poate deplasa din poziția sa într-o poziție vecină pe verticală sau pe orizontală dacă aceasta este accesibilă.

Va trebui să determinați numărul minim de minute necesare pentru ca cei doi hoți să se întâlnească.

Date de intrare

Prima linie a fișierului de intrare **KEOPS.IN** conține valorile m și n care reprezintă dimensiunile piramidei. Cea de-a doua linie conține coordonatele l_1 și c_1 (linia și coloana) ale primului hoț, iar cea de-a treia linie conține coordonatele l_2 și c_2 (linia și coloana) ale celui de-al doilea hoț.

Numerele de pe aceste linii sunt separate prin spații.

Pe fiecare dintre următoarele m linii se află un șir de n numere neseparate prin spații. Valorile acestor numere pot fi 0 sau 1.

Valoarea 0 indică faptul că celula corespunzătoare este accesibilă, iar valoarea 1 indică faptul că celula este inaccesibilă.

Date de ieșire

Fișierul de ieșire **KEOPS.OUT** trebuie să conțină o singură linie pe care se va afla numărul minim de minute necesare pentru ca cei doi hoți să se întâlnească.

Dacă nu există posibilitatea ca cei doi hoți să se întâlnească, valoarea scrisă va fi -1.

Restricții și precizări

- $2 \le m, n \le 250$;
- liniile sunt numerotate de la 1 la m;
- coloanele sunt numerotate de la 1 la n;
- hoții nu se pot afla în aceeași celulă la început;
- ambii hoți se află inițial în celule accesibile;
- hoții nu vor părăsi piramida.

Exemplu

KEOPS.IN

```
5 5
5 5
1 1
00000
11110
00000
01111
00000
```

KEOPS.OUT

```
8
```

Timp maxim de execuție/test: 1 secundă

P030431: Colectare

Anual, regele *Arthur* testează capacitatea intelectuală a subalternilor săi. Anul acesta vrăjitorul *Merlin* va fi așezat într-un labirint a cărui hartă se cunoaște.

Labirintul conține o mulțime de obiecte pe care *Merlin* trebuie să le colecteze și să le scoată din labirint.

Merlin este pus în dificultate deoarece vrăjile sale nu au efect nici asupra zidurilor și nici asupra obiectelor și trebuie să părăsească labirintul cât mai repede posibili.

Chiar dacă labirintul conține multe obiecte, acestea sunt mici și ușoare și transportul lor nu constituie o dificultate pentru *Merlin*.

Labirintul poate fi privit ca un caroiaj de dimensiune $m \times n$ împărțit în $m \cdot n$ celule.

În fiecare minut, *Merlin* se poate deplasa din celula în care se află într-o alta care poate fi liberă sau poate conține unul dintre obiecte, vecină pe orizontală sau pe verticală. *Merlin* poate colecta din mers obiectele, deci timpul necesar colectării obiectelor este 0.

Fiecare celulă a labirintului poate conține cel mult un obiect. Celula în care se află inițial *Merlin*, este goală.

Merlin va putea părăsi labirintul dacă ajunge într-o celulă care se află pe marginea acestuia și a colectat toate obiectele.

În cazul în care *Merlin* nu reușește să iasă din labirint în cel mai scurt timp posibil, regele *Arthur* îl va concedia.

Pentru a se asigura că nu va da greș, *Merlin* vă roagă să realizați un program care, pe baza labirintului și a unei celule inițiale, să determine timpul minim în care el poate colecta obiectele și poate ieși din labirint.

Date de intrare

Prima linie a fișierului de intrare **COLLECT.IN** conține valorile m și n care reprezintă dimensiunile labirintului.

Fiecare dintre următoarele m linii conține un șir format din n caractere care pot avea valorile *W*, *O*, *E* și *M*.

Litera *W* indică faptul că în labirint, în celula respectivă este zid. Litera *O* indică faptul că în labirint, în celula respectivă se află un obiect. Litera *E* indică faptul că în labirint celula respectivă este goală. Litera *M* indică faptul că în labirint celula respectivă reprezintă poziția inițială a lui *Merlin*.

În fișierul de intrare va exista un singur caracter cu valoarea *M*.

Date de ieșire

Fișierul de ieșire **COLLECT.OUT** trebuie să conțină o singură linie pe care se va afla numărul minim de minute necesare pentru ca *Merlin* să adune toate obiectele și să părăsească labirintul.

Restricții și precizări

- $2 \le m, n \le 50$;
- *Merlin* nu va putea distruge zidurile;
- labirintul conține cel mult cinci obiecte;
- se garantează faptul că *Merlin* va putea ieși din labirint.

Exemplu

COLLECT.IN

```
5 5
EMEEE
WWWOW
EEOEW
EOEEE
WWWWW
```

COLLECT.OUT

```
9
```

Timp maxim de execuție/test: 1 secundă

P030432: Lego

Se consideră o tablă triunghiulară ale cărei laturi au lungimea n (număr întreg), împărțită în triunghiuri ca în figura alăturată și se consideră m triunghiuri care au dimensiunile laturilor egale cu 1. Pe fiecare latură a unui triunghi mic este înscris un număr.

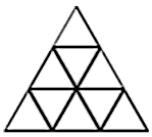

Figura 1: $n = 3$

Se cere să se așeze triunghiuri mici pe tablă astfel încât:

- tabla să fie acoperită complet;
- două laturi alăturate care sigur aparțin la două triunghiuri diferite să aibă înscrise același număr;
- suma tuturor numerelor înscrise pe laturile triunghiurilor așezate pe tablă să fie maximă.

Date de intrare

Prima linie a fișierului de intrare **LEGO.IN** conține valorile n și m, separate printr-un singur spațiu, care reprezintă lungimile celor trei laturi ale tablei, respectiv numărul de triunghiuri care au lungimile laturilor egale cu 1.

Fiecare dintre următoarele m linii conține câte trei numere, separate între ele prin spațiu, care reprezintă numerele înscrise pe laturile unui triunghi.

Date de ieșire

Fișierul de ieșire **LEGO.OUT** trebuie să conțină o singură linie pe care se va afla suma tuturor numerelor înscrise pe laturile triunghiurilor așezate pe tablă, astfel încât să fie respectate condițiile din enunț.

Restricții

- $2 \le n \le 5$;
- $3 \le m \le 30$;
- numerele înscrise pe laturile triunghiurilor sunt numere întregi cuprinse între 1 și 100;
- triunghiurile pot fi rotite în orice direcție și oglindite;
- se garantează existența unei soluții.

Exemplu

LEGO.IN

```
2 6
3 7 8
1 2 3
1 2 7
5 2 1
2 4 4
4 1 2
```

LEGO.OUT

```
46
```

Timp maxim de execuție/test: 1 secundă

P030433: Cele două (tipuri de) turnuri

De foarte mult timp, cele două specii de *elfi*, *Elfii Pădurii* și *Elfii Nopții*, trăiesc în pace pe același teritoriu.

Datorită amenințărilor din partea *orcilor*, *elfii* și-au construit turnuri pentru a-și apăra teritoriul. Bineînțeles că cele două specii de *elfi* și-au construit turnurile în stil propriu. (Adică, turnurile construite de *Elfii Pădurii* sunt diferite de cele ale *Elfilor Nopții*.)

Atunci când *orcii* nu mai reprezentau o amenințare pentru *elfi*, cele două specii au hotărât de comun acord că ar fi bine să stabilească, în funcție de turnurile pe care le-au construit, teritoriul fiecărei specii.

Ei au decis că teritoriul fiecărei specii să fie determinat pe baza turnurilor, astfel încât oricare ar fi două turnuri aparținând aceleiași specii, drumul cel mai scurt dintre ele să aparțină teritoriului și teritoriul trebuie să fie cât mai mic posibil pentru a fi mai ușor de apărat în cazul apariției unui pericol. Teritoriul unei specii trebuie să conțină toate turnurile aparținând speciei respective.

Datorită faptului că acest lucru depășește inteligența *elfilor*, ei apelează la voi pentru a-i ajuta.

Sarcina voastră este de a determina numărul de turnuri ale *Elfilor Pădurii* care se află pe teritoriul *Elfilor Nopții* și numărul de turnuri ale *Elfilor Nopții* care se află pe teritoriul *Elfilor Pădurii*.

Date de intrare

Prima linie a fișierului de intrare **TOWERS.IN** conține valorile n și m, separate printr-un singur spațiu, care reprezintă numărul de turnuri ale *Elfilor Pădurii*, respectiv numărul de turnuri ale *Elfilor Nopții*.

Fiecare dintre următoarele n linii conține câte două numere, separate între ele printr-un singur spațiu, care reprezintă coordonatele la care se află un turn al *Elfilor Pădurii*.

Fiecare dintre următoarele m linii conține câte două numere, separate între ele printr-un singur spațiu, care reprezintă coordonatele la care se află un turn al *Elfilor Nopții*.

Date de ieșire

Fișierul de ieșire **TOWERS.OUT** trebuie să conțină o singură linie pe care se vor afla două numere, separate între ele printr-un singur spațiu, care reprezintă numărul de turnuri ale *Elfilor Pădurii* care se află pe teritoriul *Elfilor Nopții*, respectiv numărul de turnuri ale *Elfilor Nopții* care se află pe teritoriul *Elfilor Pădurii*.

Restricții și precizări

- turnurile de pe marginea unui teritoriu aparțin acestuia;
- nu există două turnuri la aceleași coordonate;
- numărul total al turnurilor ($m + n$) este un întreg cuprins între 2 și 1000;
- coordonatele turnurilor sunt numere întregi cuprinse între 0 și 1000.

Exemplu

TOWERS.IN

```
4 4
0 0
4 4
0 4
4 0
3 1
5 3
3 3
5 1
```

TOWERS.OUT

```
0 2
```

Timp maxim de execuție/test: 1 secundă

P030434: 008 - James Bomb

James Bomb are o nouă misiune de îndeplinit. El se află într-o clădire cu m etaje. Fiecare etaj poate fi privit ca un caroiaj având $n \times p$ celule.

Într-un anumit loc din clădire se află o bombă care explodează după t minute. *Agentul* 008 poate să dezamorseze bomba în w minute, dar mai întâi trebuie să ajungă la ea.

În fiecare minut *James Bomb* se poate deplasa cu o celulă pe orizontală (stânga/dreapta sau înainte/înapoi) și verticală (sus/jos) într-o celulă goală.

Sarcina voastră este ca, pe baza hărții clădirii, a poziției lui *James Bomb* și a poziției bombei, să stabiliți dacă *James Bomb* are timp sau nu să dezamorseze bomba.

În cazul în care *James Bomb* are timp să dezamorseze bomba, determinați timpul minim necesar în care el ajunge la bombă. În caz contrar, determinați timpul minim necesar în care *James Bomb* poate părăsi clădirea.

Date de intrare

Fișierul de intrare **JBOMB.IN** conține pe prima linie trei numere m, n și p, separate între ele prin spațiu, care reprezintă dimensiunile clădirii. A doua linie conține două numere t și w, separate printr-un singur spațiu, care reprezintă timpul în care bomba explodează, respectiv timpul în care *James Bomb* poate să dezamorseze bomba.

Cea de-a treia linie conține trei numere, separate între ele prin spații, care reprezintă coordonatele bombei.

Cea de-a patra linie conține trei numere, separate între ele prin spații, care reprezintă coordonatele lui *James Bomb*.

În continuare urmează harta celor m etaje. Fiecare etaj este descris de n linii.

Pe fiecare dintre liniile corespunzătoare descrierii unui etaj se află p numere care nu sunt separate prin spații și care pot avea valorile 0 sau 1. Valoarea 0 indică faptul că celula respectivă este liberă, iar valoarea 1 indică faptul că celula respectivă este ocupată (conține zid).

Date de ieșire

Fișierul de ieșire **JBOMB.OUT** trebuie să conțină o singură linie pe care se vor afla două numere separate între ele printr-un singur spațiu.

Primul număr va avea valoarea 1 dacă *James Bomb* are timp să dezamorseze bomba, caz în care cel de-al doilea număr reprezintă timpul minim în care *Agentul* 008 poate ajunge la bombă și valoarea 0 dacă *James Bomb* nu are timp să dezamorseze bomba, caz în care cel de-al doilea număr reprezintă timpul minim în care *Agentul* 008 poate părăsi clădirea.

Restricții și precizări

- $3 \le m, n, p \le 20$;
- $1 \le w, t \le 100$;
- poziția inițială a lui *James Bomb* și poziția bombei sunt distincte;
- *James Bomb* poate părăsi clădirea dacă se află într-o celulă liberă de pe marginea clădirii și nu are timp să dezamorseze bomba;
- se garantează existența unei soluții.

Exemplu

JBOMB.IN

```
3 3 3
5 2
3 3 2
1 2 2
011
001
111
000
100
001
111
111
101
```

JBOMB.OUT

```
1 3
```

Timp maxim de execuție/test: 1 secundă

P030435: Raze LASER

În anul 2457, în provincia *Etram* de pe planeta *Marte* au avut mai multe explozii nucleare.

Locuitorii planetei dețin o tehnologie cu ajutorul căreia pot împiedica răspândirea particulelor radioactive care sunt capabile să distrugă toate formele de viață de pe planetă.

Tehnologia lor se bazează pe două unde de tip *LASER* care, transmise din același punct pe direcții diferite distrug toate particulele radioactive care se află într ele sau le intersectează. Locuitorii planetei au ales un loc la marginea provinciei unde au instalat un dispozitiv bazat pe tehnologia prezentată anterior.

Datorită faptului că locuitorii planetei doresc ca cele două raze să fie cât mai apropiate posibil, astfel încât să nu existe pericolul răspândirii particulelor radioactive pe toată planeta, vă roagă să determinați unghiul minim pe care undele *LASER* îl pot forma.

Date de intrare

Fișierul de intrare **LASER.IN** conține numărul n al punctelor în care au avut loc explozii nucleare.

Pe fiecare dintre următoarele n linii se află câte două numere, separate printr-un singur spațiu, care reprezintă coordonatele punctelor în care au avut loc explozii nucleare relative la locul în care a fost montat dispozitivul.

Date de ieșire

Fișierul de ieșire **LASER.OUT** trebuie să conțină o singură linie pe care se va afla un singur număr care reprezintă valoarea unghiului minim care poate fi format de cele două raze laser ale dispozitivului. Unghiul va fi scris în grade cu două zecimale exacte.

Restricții și precizări

- $1 \le n \le 100000$;
- coordonatele punctelor sunt numere întregi cuprinse între 0 și 1000;
- există puncte în care au avut loc mai multe explozii;
- dispozitivul este montat în punctul de coordonate (0, 0).

Exemplu

LASER.IN

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

LASER.OUT

```
36.87
```

Timp maxim de execuție/test: 1 secundă

P030436: Sertare

Mickey și *Minnie* s-au plictisit să stea toată ziua în casă și datorită faptului că locuința lor era mare și conținea foarte multe sertare s-au gândit la următorul joc.

Se aleg $2 \cdot n$ sertare care se numerotează de la 1 la $2 \cdot n$, și n bile albe și n bile negre. *Minnie* pune câte o bilă în fiecare dintre cele $2 \cdot n$ sertare.

Mickey începe să adune bilele din sertare în ordinea în care au fost numerotate, și dacă, la un moment dat, are mai multe bile albe decât negre, atunci el câștigă runda. La fiecare rundă, bilele trebuie așezate în altă ordine.

Minnie vă roagă să determinați, pe baza numărului n, numărul total de runde pe care le poate câștiga.

Date de intrare

Fișierul de intrare **DRAWERS.IN** conține numărul n.

Date de ieșire

Fișierul de ieșire **DRAWERS.OUT** trebuie să conțină o singură linie pe care se va afla un singur număr care reprezintă numărul total de posibilități distincte de a așeza bilele în cele $2 \cdot n$ sertare astfel încât să fie respectată condiția de așezare a bilelor.

Restricție

- $1 \le n \le 500$.

Exemple

DRAWERS.IN

```
3
```

DRAWERS.OUT

```
5
```

DRAWERS.IN

```
2
```

DRAWERS.OUT

```
2
```

Timp maxim de execuție/test: 1 secundă

P030437: Rețea de calculatoare

O companie deține o rețea de calculatoare. Oricare două calculatoare pot comunica între ele în mod direct sau indirect.

Datorită faptului că rețeaua este veche și nu a fost întreținută corespunzător, conducerea dorește să știe cât de mare este riscul ca două calculatoare din rețea să nu mai poată comunica.

Compania v-a angajat să realizați un program care, pe baza legăturilor dintre calculatoare, să determine numărul minim de legături care ar trebui distruse astfel încât să existe două calculatoare care nu mai pot comunica.

Date de intrare

Fișierul de intrare **NETWORK.IN** conține pe prima linie două numere n și m, separate între ele printr-un singur spațiu, care reprezintă numărul de calculatoare din rețea, respectiv numărul de legături directe dintre ele.

Fiecare dintre următoarele m linii conține câte două numere, separate printr-un singur spațiu, care reprezintă numerele de ordine a două calculatoare între care există legătură directă.

Calculatoarele din rețea sunt numerotate de la 1 la n și între două calculatoare există cel mult o singură conexiune.

Date de ieșire

Fișierul de ieșire **NETWORK.OUT** trebuie să conțină o singură linie pe care se va afla un singur număr care reprezintă numărul minim de legături care ar trebui distruse astfel încât să existe două calculatoare care nu mai pot comunica în mod direct sau indirect.

Restricție

- $1 \le n \le 100$ și $1 \le m \le 4950$.

Exemplu

NETWORK.IN

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

NETWORK.OUT

```
2
```

Timp maxim de execuție/test: 1 secundă

P030438: Extratereștri

Două grupuri de extratereștri comunică între ele prin intermediul unor mesaje. Cercetătorii de la *Institutul de Cercetări Spațiale* au reușit să intercepteze mesajele și să le decodifice sub forma unor șiruri de caractere formate din litere mici ale alfabetului englezesc.

Cercetătorii au descoperit că fiecare grup transmite două mesaje unul după celălalt și că *mesajul original* este dat de cel mai lung subșir comun al celor două mesaje.

Datorită faptului că mesajele transmise sunt foarte lungi, cercetătorii vă roagă ca, pe baza a două mesaje să determinați *mesajul original*.

Date de intrare

Fișierul de intrare **ALIENS.IN** conține pe prima linie primul mesaj, iar pe cea de-a doua linie cel de-al doilea mesaj.

Date de ieșire

Fișierul de ieșire **ALIENS.OUT** trebuie să conțină pe prima linie numărul de caractere ale mesajului original, iar pe cea de-a doua linie mesajul original.

Restricții și precizări

- lungimile celor două mesaje din fișierul de intrare este un număr întreg cuprins între 1 și 2000;
- în cazul existenței mai multor soluții se va afișa doar una.

Exemplu

ALIENS.IN	**ALIENS.OUT**
abracadabra	7
arbadacarba	abaaaba

Timp maxim de execuție/test: 2 secunde

P030439: Matrice magică

Se consideră o matrice A de dimensiune $n \times m$ și un număr natural p. Elementele matricei A sunt numere întregi. Matricea A are proprietatea că toate sumele elementelor care se află pe o linie sau o coloană a matricei este divizibilă cu p.

O matrice B este *asemănătoare* cu matricea A dacă are următoarele proprietăți:

- matricea B are aceeași dimensiune cu matricea A;
- $|a_{ij} - b_{ij}| < p$, $\forall 1 \le i \le n$ și $\forall 1 \le j \le m$;
- suma elementelor de pe linia i, $1 \le i \le n$, a matricei B este egală cu suma elementelor de pe linia i a matricei A;
- suma elementelor de pe coloana j, $1 \le j \le m$, a matricei B este egală cu suma elementelor de pe linia j a matricei A.

Spunem că matricea B este *magică* dacă b_{ij}, $\forall 1 \le i \le n$ și $\forall 1 \le j \le m$, este divizibil cu p și $b_{ij} \ge 0$.

Sarcina voastră este să determinați o matrice B *magică*, dacă există, *asemănătoare* cu matricea A.

Date de intrare

Fișierul de intrare **MAGIC.IN** conține pe prima linie două numere n și m, separate între ele printr-un singur spațiu, care reprezintă dimensiunea matricei A. Cea de-a doua linie conține numărul p.

Fiecare a i-a linie dintre următoarele n linii conține câte m numere, separate între ele prin spațiu, care reprezintă elementele de pe a i-a linie din matricea A.

Date de ieșire

Fișierul de ieșire **MAGIC.OUT** trebuie să conțină n linii. Fiecare a i-a dintre cele n linii trebuie să conțină m numere care reprezintă elementele de pe a i-a linie a matricei B determinate pe baza condițiilor impuse de enunțul problemei.

Dacă nu există o matrice B care să respecte condițiile din enunț, atunci în fișierul de ieșire se va scrie pe o singură linie valoarea `-1`.

Restricții

- $1 \le n, m \le 30$;
- $1 \le p \le 100$;
- $1 \le a_{ij} \le 100$, $\forall 1 \le i \le n$ și $\forall 1 \le j \le m$.

Exemplu

MAGIC.IN

```
3 3
3
4 5 6
7 8 9
10 11 12
```

MAGIC.OUT

```
3 6 6
6 9 9
12 9 12
```

Timp maxim de execuție/test: 5 secunde