Rundele #06 - #15

În continuare vă vom prezenta enunțurile celor 10 probleme propuse spre rezolvare la rundele 06-15 ale ediției din acest an a concursului de programare Bursele Agora.

P010416: Arena din Erathia

O mulțime de creaturi din *Erathia* se află în arenă. Ele se vor lupta pentru a demonstra care este cea mai puternică rasă.

Vrăjitorii din *Bracada* au teleportat creaturile în diverse poziții ale *Arenei*. Pentru a putea colabora, două creaturi din aceeași rasă pot comunica folosind tehnici ale *Școlii de Magie a Spiritului*. Totuși, calitatea comunicării telepatice depinde de distanță. Așadar, o rasă va fi dezavantajată dacă două creaturi ale rasei respective se află la distanță mare.

Se cunosc coordonatele la care au fost teleportate creaturile și trebuie determinată cea mai defavorizată rasă. Cu alte cuvinte, va trebui să determinați, pentru fiecare rasă, distanța maximă dintre doi reprezentanți ai rasei respective și apoi, pe baza distanțelor determinate, să alegeți cea mai defavorizată rasă.

Date de intrare

Fișierul de intrare **ARENA.IN** conține pe prima linie un singur număr natural *N*, reprezentând numărul de creaturi.

Pe fiecare dintre următoarele *N* linii se află trei valori *x*, *y* și *r*, separate prin spații. Primele două valori sunt numere reale și reprezintă coordonatele la care se află creatura, iar cea de-a treia este un șir de caractere care reprezintă rasa creaturii.

Date de ieșire

Fișierul de ieșire **ARENA.OUT** trebuie să conțină o singură linie pe care se va afla un șir de caractere reprezentând denumirea rasei celei mai defavorizate.

Restricții

- $2 \le N \le 5000$;
- coordonatele creaturilor sunt numere reale cuprinse între 0 și 1000 și conțin cel mult două zecimale exacte;
- numele creaturilor sunt formate din cel mult 25 de caractere; acestea pot fi litere mici ale alfabetului englezesc sau spații;
- vor exista cel mult 1000 de rase;
- pentru fiecare rasă există cel puțin doi reprezentanți.

Exemplu

ARENA.IN

```
7
1 1 dragon
1 2 minotaur
1 3 wood elf
1 4 wood elf
1 5 wood elf
2 5 minotaur
2 6 dragon
```

ARENA.OUT

```
dragon
```

Explicație

Distanța dintre cei doi *dragoni* este aproximativ 5.099, distanța dintre cei doi *minotauri* este aproximativ 3.162, iar distanțele dintre *elfi* sunt 1 sau 2. Așadar, *dragonii* sunt cei mai defavorizați (doar din punct de vedere al capacității de comunicare!).

Timp maxim de execuție/test: 1 secundă

P010417: Merlin

Celebrul vrăjitor *Merlin* trebuie să îndeplinească un nou ritual magic care îl va ajuta să îi înfrângă pe inamicii regelui *Arthur*. Ritualul constă în amplasarea a *M* pietricele într-o zonă pătratică cu latura *N*. Pietricelele trebuie amplasate în așa fel încât oricare patru dintre ele să nu se afle în colțurile unui pătrat cu laturile paralele cu cele ale zonei pătratice.

Date de intrare

Fișierul de intrare **MERLIN.IN** conține pe prima linie numărul natural *N*, reprezentând latura zonei pătratice. Cea de-a doua linie conține numărul natural *M*, reprezentând numărul de pietricele.

Date de ieșire

Fișierul de ieșire **MERLIN.OUT** trebuie să conțină configurația zonei pătratice. Aceasta va fi reprezentată printr-o matrice pătratică ale cărei elemente pot fi 0 sau 1. Valoarea 0 indică faptul că în locația respectivă nu se află nici o pietricică, iar valoarea 1 indică faptul că locația conține o pietricică.

Așadar, fișierul va conține N linii, pe fiecare dintre ele aflându-se câte N valori 0 sau 1. Aceste valori nu vor fi separate prin spații.

Restricții și precizări

- $3 \le N \le 100$;
- $1 \le M \le 5000$;
- într-o locație poate fi amplasată cel mult o pietricică;
- dacă există mai multe soluții, poate fi aleasă oricare dintre ele;
- va exista întotdeauna cel puțin o soluție.

Exemplu

MERLIN.IN	MERLIN.OUT
10	1111111111
54	1001010101
	1001000011
	1110000001
	1000001011
	1100000101
	1000100011
	1100110001
	1010101001
	1111111110

Timp maxim de execuție/test: 1 secundă

P010418: Distribuție

Revista *GInfo* este tipărită în două tipografii și este distribuită în N orașe. În fiecare lună jumătate dintre orașele în care este difuzată *GInfo* primesc reviste de la una dintre tipografii, iar cealaltă jumătate de la cealaltă tipografie.

De la fiecare tipografie pornesc câte $N / 2$ camioane încărcate cu reviste. Fiecare dintre cele N camioane trebuie să ajungă la unul dintre cele N orașe.

Departamentul de distribuție al *GInfo* dorește să determine care orașe vor fi deservite de prima tipografie și care de cea de-a doua, astfel încât distanța totală pe care o vor parcurge camioanele să fie minimă. Un camion va ajunge la orașul destinație pe cel mai scurt drum posibil.

Între orașe se află un număr total de M străzi pe care se poate circula în ambele sensuri.

Pentru fiecare stradă sunt cunoscute orașele pe care le unește, precum și lungimea străzii.

Orașele sunt identificate prin numere cuprinse între 1 și N, iar tipografiile se află în orașele identificate prin 1 și 2. Evident, distanța dintre tipografia din orașul 1 și orașul 1 este 0, iar distanța dintre tipografia din orașul 2 și orașul 2 este tot 0.

Date de intrare

Fișierul de intrare **GINFO.IN** conține pe prima linie numărul natural N, reprezentând numărul orașelor, precum și numărul natural M, reprezentând numărul străzilor. Cele două numere sunt separate printr-un spațiu. Fiecare dintre următoarele M linii va conține câte trei numere naturale x, y și L, cu semnificația: *există o stradă de lungime L între orașele x și y*.

Date de ieșire

Fișierul de ieșire **GINFO.OUT** trebuie să conțină două linii, pe fiecare aflându-se câte $N / 2$ numere naturale, separate prin câte un spațiu. Numerele de pe prima linie reprezintă orașele deservite de prima tipografie, iar cele de pe a doua linie reprezintă orașele deservite de cea de-a doua tipografie.

Restricții și precizări

- $1 \le N \le 1000$;
- $1 \le M \le 10000$;
- numărul orașelor este întotdeauna par;
- între oricare două orașe există cel mult o stradă;
- străzile nu se intersectează;
- dacă există mai multe soluții, poate fi aleasă oricare dintre ele;
- va exista cel puțin un drum de la fiecare dintre cele două tipografii la toate orașele;
- lungimile străzilor sunt numere naturale cuprinse între 1 și 1000.

Exemplu

GINFO.IN	GINFO.OUT
6 8	1 4 5
1 2 5	2 3 6
1 4 8	
1 6 7	
2 3 6	
3 4 5	
3 6 2	
4 5 3	
5 6 2	

Timp maxim de execuție/test: 1 secundă

P010419: Cerc

Galul *Obelix* vrea să îl convingă pe druidul *Panoramix* să îi dea puțină poțiune magică. Nemairezistând insistențelor lui *Obelix*, *Panoramix* îi promite că îi va da o picătură dacă reușește să găsească soluția unei probleme.

Panoramix așează pe un cerc N papirusuri pe care se află numerele de la 1 la N, în această ordine. Așadar, lângă numărul 1 se află, în stânga, numărul N, iar în dreapta numărul 2, lângă numărul 2 se află, în stânga, numărul 1, iar în dreapta numărul 3 etc. La fiecare pas, *Obelix* are voie să interschimbe două numere vecine. *Obelix* va trebui să efectueze un număr minim de interschimbări astfel încât numărul 1 să ajungă în locul numărului N, numărul 2 să ajungă în locul numărului N - 1 etc.

Datorită faptului că *Obelix* nu stă prea bine la capitolul "inteligență", el cere ajutorul prietenului său *Asterix* care, după multe proteste, acceptă să îl ajute. Cum magia este la ordinea zilei, tocmai ați intrat în pielea unui spion imperial care trebuie să știe câte interschimbări va face *Asterix*.

Date de intrare
Fișierul de intrare **CIRCLE.IN** conține o singură linie pe care se va afla numărul natural N.

Date de ieșire
Fișierul de ieșire **CIRCLE.OUT** trebuie să conțină o singură linie pe care se va afla numărul minim de interschimbări care trebuie efectuate.

Restricție
• $3 \le N \le 30000$.

Exemplu

CIRCLE.IN	CIRCLE.OUT
4	2

Explicație
La primul pas vor fi interschimbate numerele 1 și 4, iar la al doilea vor fi interschimbate numerele 2 și 3.

Timp maxim de execuție/test: 1 secundă

P010420: Explozie
Flota romulană asediază planeta *Vulcan* cu scopul declarat de a alipi spațiul vulcanian celui controlat de *Imperiul Romulan*. Un număr inimaginabil de navete cu soldați romulani au intrat în sistemul stelar vulcanian. Vulcanienii au o singură armă la dispoziție, care poate fi detonată în orice punct al spațiului (nu prea contează unde, deoarece romulanii ocupă, practic întregul sistem).

Vom considera că navetele romulane ocupă fiecare punct de coordonate întregi a spațiului. Arma vulcaniană va produce o explozie care va distruge toate navele aflate în interiorul razei de acțiune a armei. Așadar, va fi afectată o zonă sferică a spațiului. Vor fi distruse toate navele, inclusiv cele aflate pe suprafața acestei sfere.

Cunoscând raza de acțiune R, va trebui să determinați numărul navelor romulane distruse.

Date de intrare
Fișierul de intrare **WEAPON.IN** conține o singură linie pe care se va afla numărul natural R, reprezentând raza de acțiune a armei.

Date de ieșire
Fișierul de ieșire **WEAPON.OUT** trebuie să conțină pe o singură linie numărul navelor romulane care vor fi distruse.

Restricție
• $1 \le N \le 10000$.

Exemplu

WEAPON.IN	WEAPON.OUT
1	7

Timp maxim de execuție/test: 1 secundă

P010421: Recunoașterea caracterelor
Se consideră o matrice pătratică cu 16 linii și 16 coloane. Elementele acestei matrice pot fi 0 sau 1. Această matrice reprezintă, de fapt, o literă a alfabetului englezesc (litera este formată din elementele a căror valoare este 1). Va trebui să determinați care este litera reprezentată de această matrice.

În acest scop veți avea la dispoziție un număr de N matrice pătratice cu 16 linii și 16 coloane care conțin reprezentări standard ale literelor. Litera corespunzătoare matricei date este "*cea mai apropiată matrice dintre cele N*".

Va trebui să definim distanța dintre două matrice. Pentru a determina o astfel de distanță, fiecărui element cu valoarea 1 din prima matrice va trebui să îi corespundă un element cu valoarea 1 din cea de-a doua matrice și fiecărui element cu valoarea 1 din cea de-a doua matrice să îi corespundă un punct din prima matrice. Așadar se va realiza o corespondență între elementele celor două matrice.

Presupunem, fără a restrânge generalitatea, că prima matrice conține mai multe elemente cu valoarea 1 decât a doua. Numărul elementelor cu valoarea 1 din prima matrice va fi notat cu p, iar numărul elementelor cu valoarea 1 din cea de-a doua va fi notat cu q. În aceste condiții, fiecărui element cu valoarea 1 din cea de-a doua matrice îi vor corespunde $[p / q]$ elemente din prima matrice. Dacă q nu este divizor al lui p, vor rămâne elemente în prima matrice care nu au corespondent. Acestora le vor corespunde elemente distincte din cea de-a doua matrice.

Cu alte cuvinte, corespondențele vor fi "*echilibrate*", în sensul că, dacă unui element dintr-o matrice îi corespund k elemente (k fiind minimul numărului de elemente corespunzătoare elementelor din matrice), atunci tuturor celorlalte elemente le vor corespunde fie k, fie $k + 1$ elemente.

Distanța dintre două matrice va fi dată de suma distanțelor dintre elementele puse în corespondență. Dacă unui element de pe poziția (i_1, j_1) în prima matrice îi corespunde un element (i_2, j_2) în cea de-a doua, atunci distanța dintre cele două elemente va fi $|i_1 - i_2| + |j_1 - j_2|$.

Evident, corespondențele trebuie alese în așa fel încât distanța dintre cele două matrice să fie cât mai mică posibil.

Date de intrare
Fișierul de intrare **OCR.IN** conține pe primele 16 linii elementele matricei care reprezintă litera care trebuie recunoscută. Pe următoarea linie se va afla numărul N al literelor date. Va urma descrierea literelor. Pentru fiecare literă vom avea 17 linii. Prima dintre ele va identifica litera (un caracter). Următoarele 16 linii vor conține descrierea literei. Valorile de pe o linie nu vor fi separate prin spații.

Date de ieșire
Fișierul de ieșire **OCR.OUT** trebuie să conțină pe o singură linie un singur caracter care reprezintă litera recunoscută.

Restricții și precizări
• $1 \le N \le 26$;
• literele descrise nu vor fi neapărat distincte;

- fiecare matrice va conține cel puțin unul și cel mult 100 de elemente cu valoarea 1;
- va exista întotdeauna o singură soluție.

Exemplu

OCR.IN

```
0000000000000000
0000000000000000
0000000000000000
0000000100000000
0000001010000000
0000010001000000
0000100000100000
0001000000010000
0001000000010000
0001111111110000
0001000000010000
0001000000010000
0001000000010000
0000000000000000
0000000000000000
0000000000000000
2
A
0000000000000000
0000000000000000
0000000100000000
0000001110000000
0000010001000000
0000100000100000
0001000000010000
0001100000110000
0001111111110000
0001100000110000
0001000000010000
0001000000010000
0001000000010000
0000000000000000
0000000000000000
0000000000000000
C
0000000000000000
0000000000000000
0000000000000000
0000111111111100
0001111000000010
0001110000000100
0001100000000000
0001100000000000
0001100000000000
0001100000000000
0001100000000000
0001100000000000
0001110000000100
0001111000000010
0000111111111100
0000000000000000
```

OCR.OUT

```
A
```

Timp maxim de execuție/test: 10 secunde

P010422: Ciobănașul

Ciobănașul *Ion* s-a decis să construiască un țarc nou pentru oile sale.

Terenul pe care ciobănașul *Ion* vrea să construiască țarcul este de formă dreptunghiulară, parcelat în $n \cdot m$ parcele și este puțin denivelat. Despre fiecare parcelă știm dacă se află la nivelul normal sau este denivelată. Țarcul trebuie să aibă o forma dreptunghiulară, laturile paralele cu laturile terenului și este susținut de patru țăruși situați în cele patru colțuri. Cei patru țăruși pot fi înfipți în pământ numai în parcele nivelate.

Pentru ciobănaș ar fi foarte important să știe câte amplasări posibile există pentru țarc, dar el este sătul de numărat oi, așa că vă cere ajutorul vostru pentru a număra dreptunghiuri.

Date de intrare

Fișierul de intrare **FOLD.IN** conține pe prima linie două numere întregi n și m separate printr-un singur spațiu care reprezintă dimensiunile terenului.

Fiecare dintre următoarele n linii conțin câte m numere, separate prin spațiu, care pot avea valorile `0` pentru teren denivelat sau `1` pentru nivel normal.

Date de ieșire

Fișierul de ieșire **FOLD.OUT** trebuie să conțină o singură linie pe care se va afla un singur număr reprezentând numărul de posibilități de amplasare.

Restricții

- $1 \le n \le 250$;
- $1 \le m \le 2000$.

Exemplu

FOLD.IN	FOLD.OUT
3 3 0 1 1 1 1 1 1 1 1	5

Timp maxim de execuție/test: 0,5 secunde

P010423: Joc

Andrei a învățat un joc matematic interesant pe care vrea să îl arate lui *Mihai*. Jocul începe cu n grămezi de pietre. Fiecare jucător aflat la mutare poate lua pietre dintr-o singură grămadă. Numărul pietrelor luate poate fi 1 sau un număr prim. Jucătorii mută alternativ, iar câștigătorul este cel care ia ultima piatră. *Andrei* se oferă să îl lase pe *Mihai* să facă prima mutare, dar *Mihai* știe că *Andrei* a învățat deja strategia jocului, așa că dorește să afle dacă are sau nu

o strategie sigură de câștig. Băieții joacă 10 jocuri consecutiv. Sarcina voastră este să îl ajutați pe *Mihai* cu răspunsul exact pentru fiecare dintre cele 10 configurații.

Date de intrare
Fișierul de intrare **GAME.IN** conține zece linii. Primul număr de pe o astfel de linie reprezintă numărul n al grămezilor, iar în continuare se află n numere care reprezintă numărul pietrelor din fiecare grămadă. Numerele de pe o linie vor fi separate prin spații.

Date de ieșire
Fișierul de ieșire **GAME.OUT** va conține câte o linie pentru fiecare dintre configurațiile din fișierul de intrare (în aceeași ordine). Valoarea de pe o linie va fi 1 dacă *Mihai* are o strategie sigură de câștig pentru configurația corespunzătoare și 0 în caz contrar.

Restricții
- $1 \le n \le 1000$;
- numărul total al pietrelor nu va depăși 2000000000 pentru o configurație.

Exemplu

GAME.IN	GAME.OUT
2 2 1	1
3 7 7 7	1
1 1000	0
2 5 5	0
4 3 3 3 3	0
2 2 1	1
3 7 7 7	1
1 1000	0
2 5 5	0
4 3 3 3 3	0

Explicație
Pentru prima configurație *Mihai* va lua o piatră din prima grămadă. Apoi, *Andrei* va lua o piatră din una dintre cele două grămezi, iar *Mihai* va lua piatra din cealaltă grămadă și va câștiga.

Pentru cea de-a doua configurație *Mihai* va lua toate cele șapte pietre din una dintre grămezi. Apoi, *Andrei* va lua un număr de pietre dintr-una dintre grămezile rămase, iar *Mihai* va lua același număr de pietre din cealaltă grămadă rămasă. Procesul va continua până în momentul în care nu va mai rămâne nici o piatră. Ultima mutare va fi efectuată cu siguranță de către *Mihai*.

Timp maxim de execuție/test: 0,5 secunde

P010424: Cifre
Se consideră un număr natural n și k cifre distincte între care se află, cu siguranță, și cifra 0. Să se determine un multiplu al lui n care conține numai cifrele date, fiecare dintre ele fiind prezentă cel puțin o dată în acest multiplu.

Date de intrare
Prima linie a fișierului de intrare **DIGITS.IN** conține valoarea n. Cea de-a doua linie a fișierului conține cifrele, neseparate prin spații.

Date de ieșire
Fișierul de ieșire **DIGITS.OUT** va conține o singură linie pe care se va afla multiplul determinat.

Restricție
- $1 \le n \le 1000$.

Exemplu

DIGITS.IN	DIGITS.OUT
11	1210
012	

Timp maxim de execuție/test: 1 secundă

P010425: Stația spațială
La stația spațială 92 a ajuns flota *Federației Planetelor Unite* care a participat la bătălia cu navale *Imperiului Cardasian*. Prima navă ajunsă este *USS Enterprise*. Celelalte nave se află pe aceeași linie, la diferite distanțe unele de altele. Pentru fiecare navă se cunoaște distanța față de nava anterioară. Din nefericire, pentru ca o navă să poată fi reparată, distanța față de navele de lângă ea trebuie să fie cel puțin egală cu o valoare x.

Așadar, anumite nave trebuie să părăsească zona. Va trebui să determinați numărul navelor care trebuie să plece, astfel încât să poată fi reparate cât mai multe nave, dar este obligatoriu ca nava *USS Enterprise* să fie reparată.

Date de intrare
Prima linie a fișierului de intrare **STATION.IN** conține numărul n al navelor, precum și valoarea x a distanței minime dintre două nave care pot fi reparate. Cea de-a doua linie a fișierului conține n - 1 numere, separate prin spații, reprezentând distanțele dintre nave.

Date de ieșire
Fișierul de ieșire **STATION.OUT** va conține pe o singură linie numărul navelor care trebuie să părăsească zona.

Restricții și precizări
- $2 \le n \le 1000$;
- $1 \le x \le 1000$;
- distanțele dintre nave sunt numere întregi cuprinse între 1 și 1000.

Exemplu

STATION.IN	STATION.OUT
5 3	2
2 3 3 2	

Timp maxim de execuție/test: 1 secundă