Rundele #16 - #25

În continuare vă vom prezenta enunțurile celor 10 probleme propuse spre rezolvare la rundele 16-25 ale ediției din acest an a concursului de programare Bursele Agora.

P020424: Tăiere

Pe o tablă se află n cartonașe, așezate unul lângă altul pe o linie, de la stânga la dreapta. Pe fiecare dintre cartonașe este scrisă o cifră diferită de 0. Trebuie eliminate k grupuri a câte x cartonașe aflate pe poziții consecutive. În total vor fi eliminate $k \cdot x$ cartonașe. După eliminare, numărul obținut prin citirea cifrelor de pe cartonașe (de la stânga spre dreapta) trebuie să fie cât mai mic posibil.

Date de intrare

Prima linie a fișierului de intrare **CUT.IN** conține numărul n al cartonașelor, numărul k al grupurilor care trebuie eliminate și numărul x al cartonașelor dintr-un grup. Aceste numere vor fi separate prin spații. Cea de-a doua linie a fișierului conține cele n cifre scrise pe cartonașe, neseparate prin spații.

Date de ieșire

Fișierul de ieșire **CUT.OUT** va conține cel mai mic număr care poate fi obținut după eliminare.

Restricții

- $5 \le n \le 1000$;
- $1 \le k \le 100$;
- $1 \le x \le 9$;
- $k \cdot x < n$.

Exemplu

CUT.IN	CUT.OUT
7 2 3 5132789	2

Timp maxim de execuție/test: 2 secunde

P020425: Sume

Pe o masă se află n cartonașe; pe fiecare dintre cartonașe este scris un număr natural. De pe masă pot fi alese oricâte cartonașe. După alegere se calculează suma numerelor de pe cartonașele alese. Va trebui să determinați numărul sumelor distincte care pot fi obținute prin astfel de alegeri.

Date de intrare

Prima linie a fișierului de intrare **SUM.IN** conține numărul n al cartonașelor. Cea de-a doua linie a fișierului conține cele n numere scrise pe cartonașe, separate prin spații.

Date de ieșire

Fișierul de ieșire **SUM.OUT** va conține numărul sumelor distincte care pot fi obținute.

Restricții și precizări

- $1 \le n \le 100$;
- numerele de pe cartonașe sunt cuprinse între 0 și 10000;
- pot fi alese între 1 și n cartonașe.

Exemplu

SUM.IN	SUM.OUT
5 0 1 2 3 4	11

Timp maxim de execuție/test: 2 secunde

P020426: Warp

Imperiul Galactic a decis stabilirea unor rute galactice folosind găuri de vierme. Pentru a testa acest sistem au fost alese n planete care vor participa la testare. O gaură de vierme unește două planete distincte, iar în interiorul său se poate circula în ambele sensuri.

Pentru fiecare gaură de vierme se cunoaște cantitatea de energie *warp* (w) necesară pentru a o traversa. Dacă o navă intră în gaura de vierme având o cantitate de energie disponibilă e, atunci la ieșire va rămâne disponibilă o cantitate de energie e / w deoarece, indiferent de cantitatea inițială, prin trecerea prin gaura de vierme, energia se reduce de w ori. Se spune că este necesară o cantitate de energie w, deoarece o navă care ajunge să aibă o cantitate de energie mai mică decât 1 este distrusă instantaneu.

Folosind sistemul de găuri de vierme se poate ajunge de la o planetă la oricare alta. Totuși, în multe cazuri nu este posibilă călătoria directă între două planete. În aceste situații vor fi alese rute care conțin două sau mai multe găuri

de vierme. Va trebui să determinați cantitatea minimă de energie care este suficientă pentru a călători între oricare două planete.

Date de intrare
Prima linie a fișierului de intrare **WARP.IN** conține numărul n al planetelor, precum și numărul m al găurilor de vierme. Cele două numere sunt separate printr-un spațiu. Fiecare dintre următoarele m linii va conține câte trei numere întregi (x, y și w), separate prin spații. Aceste numere au semnificația: există o gaură de vierme între planetele identificate prin x și y, iar cantitatea de energie necesară supraviețuirii în interiorul găurii este w.

Date de ieșire
Fișierul de ieșire **WARP.OUT** va conține cantitatea minimă de energie care este suficientă pentru a călători între oricare două planete.

Restricții și precizări
- $2 \le n \le 100$;
- $1 \le m \le 1000$;
- cantitățile de energie, pentru a supraviețui în interiorul găurilor de vierme, sunt numere întregi cuprinse între 2 și 9;
- planetele sunt identificate prin numere cuprinse între 1 și n;
- între două planete poate exista cel mult o gaură de vierme.

Exemplu

WARP.IN
```
4 5
1 2 4
1 3 6
1 4 8
2 3 5
3 4 7
```

WARP.OUT
```
32
```

Timp maxim de execuție/test: 0,5 secunde

P020427: Găuri
Se consideră o matrice cu m linii și n coloane, ale cărei elemente pot avea fie valoarea 0, fie valoarea 1. O zonă este o mulțime de elemente care au valoarea 0.

Două elemente învecinate (pe verticală sau orizontală), care au ambele valoarea 0, vor face parte din aceeași zonă. O gaură este o zonă care nu conține elemente aflate pe prima linie, pe ultima linie, pe prima coloană sau pe ultima coloană a matricei. În elementele din găurile matricei trebuie scrise numere. Toate elementele din aceeași gaură trebuie să aibă aceeași valoare. Va trebuie să determinați cel mai mic număr care poate reprezenta suma elementelor din oricare dintre găuri.

Date de intrare
Fișierul de intrare **HOLES.IN** conține pe prima linie două numere naturale m și n, separate printr-un spațiu, care reprezintă dimensiunile matricei.

Pe fiecare dintre următoarele m linii se află un șir de n numere care pot fi 0 sau 1 și care nu sunt separate prin spații. Aceste numere reprezintă elementele matricei.

Date de ieșire
Fișierul de ieșire **HOLES.OUT** va conține cel mai mic număr care poate reprezenta suma elementelor din oricare dintre găurile matricei.

Restricții și precizări
- $3 \le m, n \le 100$;
- $1 \le M \le 10000$;
- va exista întotdeauna cel puțin o gaură.

Exemplu

HOLES.IN
```
10 10
1110010101
1010111111
1010110001
1111110011
1000010111
1011111111
1011111111
1000000000
1111111111
```

HOLES.OUT
```
6
```

Timp maxim de execuție/test: 0,5 secunde

P020428: Daneel
R. Daneel Olivaw trebuie să ajungă de pe *Trantor* pe *Terminus*. În acest scop el va trebui să folosească sistemul galactic de găuri de vierme. O gaură de vierme unește două planete distincte, iar în interiorul său se poate circula în ambele sensuri. Pentru fiecare gaură de vierme se cunoaște durata deplasării prin intermediul ei, precum și costul închirierii unei nave spațiale care trece prin acea gaură.

Daneel are la dispoziție o sumă limitată, dar dorește să ajungă cât mai repede pe *Terminus*. Așadar, el va trebui să facă o călătorie pe care și-o poate permite, și care să dureze cât mai puțin.

Date de intrare
Prima linie a fișierului de intrare **DANEEL.IN** conține numărul n al planetelor, precum și numărul m al găurilor de vierme. Cele două numere sunt separate printr-un spațiu. Cea de-a doua linie a fișierului va conține suma s pe care o are la dispoziție *Daneel*.

Fiecare dintre următoarele m linii va conține câte patru numere întregi (x, y, t și c), separate prin spații. Aceste numere au semnificația: există o gaură de vierme între planetele identificate prin x și y, durata călătoriei prin gaura de vierme este t, iar costul călătoriei este c.

Date de ieșire
Fișierul de ieșire **DANEEL.OUT** va conține durata minimă a călătoriei lui *Daneel*, în cazul în care aceasta poate avea loc. În caz contrar, fișierul va conține valoarea -1.

Restricții și precizări
- $2 \le n \le 100$;
- $1 \le m \le 10.000$;
- durata traversării unei găuri de vierme este un număr întreg cuprins între 1 și 100;
- costul traversării unei găuri de vierme este un număr întreg cuprins între 1 și 100;
- planetele sunt identificate prin numere cuprinse între 1 și n;
- *Trantor* este planeta identificată prin 1;
- *Terminus* este planeta identificată prin n;
- există întotdeauna posibilitatea de a ajunge de pe *Trantor* pe *Terminus*, deși este posibil ca *Daneel* să nu își poată permite o astfel de călătorie;
- între două planete pot exista mai multe găuri de vierme.

Exemplu
DANEEL.IN
```
4 5
15
1 2 2 3
1 3 1 6
1 4 15 1
2 3 3 1
3 4 5 10
```

DANEEL.OUT
```
10
```

Timp maxim de execuție/test: 3 secunde

P020429: Munte
Un număr are aspect de munte dacă există o poziție k, astfel încât $c_1 \le c_2 \le c_k$ și $c_k \ge c_{k+1} \ge ... \ge c_n$, unde c_1, ..., c_n sunt cifrele numărului.

Dându-se un număr, să se determine numărul minim de cifre care trebuie eliminate, astfel încât numărul rămas să aibă aspect de munte.

Date de intrare
Fișierul de intrare **MOUNTAIN.IN** conține o singură linie pe care se află numărul dat.

Date de ieșire
Fișierul de ieșire **MOUNTAIN.OUT** va conține o singură linie pe care se va afla numărul minim al cifrelor care trebuie eliminate.

Restricții și precizări
- numărul va conține cel mult 10000 de cifre;
- cifrele numărului sunt nenule.

Exemplu
MOUNTAIN.IN
```
248793
```

MOUNTAIN.OUT
```
1
```

Timp maxim de execuție/test: 2 secunde

P020430: Puncte
Se consideră n puncte în plan. Să se determine numărul posibilităților de a alege trei dintre aceste puncte, astfel încât aria triunghiului determinat de acestea să fie un număr întreg.

Date de intrare
Fișierul de intrare **POINTS.IN** conține pe prima linie numărul n al punctelor din plan. Fiecare dintre următoarele n linii va conține câte două numere, separate prin spații, reprezentând coordonatele unui punct.

Date de ieșire
Fișierul de ieșire **POINTS.OUT** va conține o singură linie pe care se va afla numărul posibilităților de a alege trei dintre puncte, astfel încât aria triunghiului determinat de acestea să fie un număr întreg.

Restricții și precizări
- $3 \le n \le 10.000$;
- coordonatele punctelor sunt numere întregi cuprinse între 0 și 1000;
- aria triunghiului determinat de trei puncte coliniare este considerată a fi 0;
- nu există două puncte aflate la aceleași coordonate.

Exemplu
POINTS.IN
```
0 0
0 2
2 2
2 0
```

POINTS.OUT
```
4
```

Timp maxim de execuție/test: 2 secunde

P020431: Era imperiilor
Împăratul francez dorește să-și împrejmuiască palatul. Muncitorii săi pot construi un număr de n ziduri. Acestea pot fi amplasate în orice direcție și în orice poziție, dar fiecare zid trebuie să aibă forma liniară.

El dorește ca palatul să fie împrejmuit complet, așadar sistemul de ziduri va avea forma unui poligon. Pentru fiecare zid se cunoaște lungimea acestuia. Va trebui să determinați aria maximă a regiunii care poate fi împrejmuită.

Date de intrare
Fișierul de intrare **AOE.IN** conține pe prima linie numărul *n* al zidurilor. Fiecare dintre următoarele *n* linii va conține câte un număr care reprezintă lungimea unui zid.

Date de ieșire
Fișierul de ieșire **AOE.OUT** va conține o singură linie pe care se va afla aria maximă a regiunii care poate fi împrejmuită de cele *n* ziduri.

Restricții
- $3 \le n \le 1000$;
- lungimile zidurilor sunt numere întregi cuprinse între 1 și 10000;
- aria maximă va fi scrisă cu o precizie de două zecimale exacte;
- lățimile zidurilor sunt neglijabile.

Exemplu
AOE.IN
```
4
1
1
1
1
```

AOE.OUT
```
1
```

Timp maxim de execuție/test: 2 secunde

P020432: Aladin
Aladin s-a gândit la un moment dat să intre în afaceri cu covoare, deoarece crede că acestea sunt foarte profitabile. El a reușit să rezolve toate problemele tehnice și acum poate să realizeze covoare de dimensiune $m \times n$.

Aceste covoare sunt împărțite în $m \times n$ celule de dimensiune 1×1. Fiecare celulă va fi colorată cu alb sau cu negru.

Datorită faptului că vrea să obțină covoare cu anumite proprietăți magice este necesar ca orice pătrat, de dimensiune 2×2, să conțină două celule colorate cu alb și două celule colorate cu negru.

Aladin dorește să știe câte covoare diferite, de dimensiune $m \times n$, care să beneficieze de proprietățile magice se pot fabrica.

Date de intrare
Fișierul de intrare **ALADDIN.IN** conține pe prima linie două numere întregi *m* și *n*, separate printr-un singur spațiu, care reprezintă dimensiunile pentru un covor.

Date de ieșire
Fișierul de ieșire **ALADDIN.OUT** trebuie să conțină o singură linie pe care se va afla un singur număr care reprezintă numărul de covoare de dimensiune $m \times n$ care se pot construi astfel încât acestea să beneficieze de proprietățile magice.

Restricție
- $2 \le m, n \le 10000$.

Exemplu
ALADDIN.IN	**ALADDIN.OUT**
4 2	18

Timp maxim de execuție/test: 1 secundă

P020433: Santinele
În drumul spre *Zion* nava *Nabucodonosor* este urmărită de santinele. Nava dispune de o armă pe bază de puls electromagnetic care, dacă este activată, atunci va distruge toate santinelele pe o rază de *r* metri în jurul ei.

Datorită faptului că nava este foarte avariată, arma nu mai poate fi activată decât o singură dată.

Știind că în drumul spre *Zion* santinelele nu își schimbă formația, că nava se poate deplasa oriunde între santinele și că nava și santinelele se află tot timpul în același plan se cere să se determine numărul maxim de santinele care pot fi distruse prin activarea armei cu puls electromagnetic.

Date de intrare
Fișierul de intrare **SENTINEL.IN** conține pe prima linie numărul *n* al santinelelor și raza *r* a armei cu puls electromagnetic, separate printr-un singur spațiu.

Fiecare dintre următoarele *n* linii va conține câte două numere, separate prin spații, reprezentând coordonatele la care se află o santinelă relativ la un observator care se deplasează cu aceeași viteză cu cea a santinelelor.

Date de ieșire
Fișierul de ieșire **SENTINEL.OUT** trebuie să conțină o singură linie pe care se va afla un singur număr care reprezintă numărul maxim de santinele care pot fi distruse prin activarea armei cu puls electromagnetic.

Restricții și precizări
- $1 \le n \le 1000$;
- raza *r* a armei cu puls electromagnetic este un număr întreg cuprins între 1 și 1000;
- coordonatele santinelelor sunt numere întregi cuprinse între 0 și 1000;
- nu există două santinele la aceleași coordonate.

Exemplu
SENTINEL.IN
```
3 2
2 2
0 2
2 0
```

SENTINEL.OUT
```
2
```

Timp maxim de execuție/test: 1 secundă