



PROBLEME propuse pentru REZOLVARE

Vă prezentăm enunțurile problemelor de la faza regională pentru sud-estul Europei a concursului ACM. Echipile universităților participante s-au confruntat cu opt probleme ale căror enunțuri le puteți citi în paginile următoare.

P010201: Polinom optim

Se consideră un număr întreg pozitiv K și un șir S care este format din cifre. Șirul S poate fi partiționat în mai multe șiruri nevide S_0, S_1, S_2, \dots astfel încât $S_0 + S_1 + S_2 + \dots = S$, unde $+$ reprezintă operația de concatenare.

Fiecare partiție a lui S care este formată din M subșiruri definește un polinom $P(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + \dots + a_M \cdot x^M$, unde a_i este numărul reprezentat de subșirul S_i .

De exemplu, șirul $S = 1204$ poate fi partiționat în $S = 1 + 204$ (polinomul $P(x) = 1 + 204 \cdot x$), în $S = 1 + 2 + 04$ (polinomul $P(x) = 1 + 2 \cdot x + 4 \cdot x^2$) etc.

Trebuie să scrieți un program care alege, dintre toate polinoamele care pot fi construite pe baza șirului S , polinomul pentru care valoarea $P(K)$ este minimă.

Se știe că valoarea maximă a rezultatului nu depășește $12 \cdot 10^{14}$.

Date de intrare

Fișierul de intrare **A.DAT** conține pe prima linie numărul n de perechi (K, S) pentru care se va construi polinomul optim.

Pe fiecare din următoarele n linii se află două numere separate printr-un singur spațiu; primul număr este K , iar al doilea este șirul de cifre S .

Date de ieșire

La ieșirea standard se vor scrie n linii. Pe fiecare linie se va scrie polinomul optim corespunzător testului din fișierul de intrare.

Exemplu

A.DAT

```
3
1 1234
3 123
1 1001
```

Ecran

```
1 + 2 * X^1 + 3 * X^2 + 4 * X^3
12 + 3 * X^1
1 + 1 * X^1
```

P010202: Soldați

Soldații armatei selenare sunt aliniați; unii privesc spre capătul din dreapta al liniei, iar ceilalți privesc spre capătul din stânga.

Așadar, anumite perechi de soldați aflați în linie pot sta *față în față*, anumite perechi pot sta *spate în spate*, iar altele *față în spate* (echivalent cu *spate în față*). Se dorește să nu existe perechi de soldați care stau *față în față*.

Pentru aceasta aveți la dispoziție un singur tip de comandă prin care puteți cere unei perechi de soldați care stau *față în față* să se întoarcă. După întoarcere, cei doi soldați vor sta *spate în spate*.

Pentru o aliniere dată a soldaților, trebuie să verificați dacă există o succesiune de comenzi în urma căreia să nu existe nici o pereche de soldați care să stea *față în față*. Dacă acest lucru este posibil, va trebui să determinați numărul minim de comenzi necesare.

Date de intrare

Fișierul de intrare **B.DAT** conține pe prima linie numărul t al seturilor de date ($0 < t < 10$). Următoarele linii conțin datele de test. Prima linie corespunzătoare unui test conține un număr natural k ($0 < k < 1000$) care indică numărul liniilor următoare corespunzătoare testului curent. Aceste linii descriu pozițiile soldaților; pe fiecare linie se află cel mult 50 de caractere. Pozițiile soldaților sunt indicate de simbolurile ' $<$ ' (soldatul privește spre stânga) și ' $>$ ' (soldatul privește spre dreapta).

Date de ieșire

La ieșirea standard se vor scrie t linii corespunzătoare celor t seturi de date. Dacă este posibilă aranjarea soldaților



astfel încât să nu existe nici o pereche de soldați care să stea față în față, pe linia corespunzătoare se va scrie numărul minim de întoarceri necesare. Dacă aranjarea nu este posibilă, pe această linie se va scrie valoarea -1.

Exemplu

B.DAT

```
3
2
><>
<
1
><<<<
3
<
<
>
```

Ecran

```
3
4
0
```

P010203: Numere secrete

Se aleg două numere naturale a și b astfel încât $1 < a < b$. Unei persoane P i se comunică produsul celor două numere ($a \cdot b$), iar unei persoane S i se comunică suma celor două numere ($a + b$). Are loc următoarea discuție între P și S :

P : Nu știu care sunt numerele a și b !

S : Nici eu nu le știu, dar știam că tu nu le poți ști!

P : Acum știu numerele!

S : Și eu le știu acum!

Date de intrare

Fișierul de intrare **C.DAT** conține perechi de numere x și y ($2 \leq x < y \leq 550$), câte una pe o linie. Se garantează corectitudinea intrării.

Date de ieșire

Pentru fiecare pereche furnizată la intrare, trebuie găsite toate perechile de numere a și b ($x \leq a < b \leq y$) pentru care discuția dată este posibilă.

Perechile corespunzătoare unui interval dat la intrare vor fi scrise pe aceeași linie. Dacă există astfel de perechi, linia se va încheia cu textul "no more pairs.". Dacă nu există nici o astfel de pereche, linia va conține doar textul "no pairs.".

Numerele care formează o pereche vor fi separate prin caracterul ' ', iar perechile vor fi separate între ele prin caracterul ' ', urmat de un spațiu.

Exemplu

C.DAT

```
2 10
2 20
```

Ecran

no pairs.

4,13; no more pairs.

P010204: Tipografie

Directorii unei tipografii foarte mari (cu resurse nelimitate) doresc să știe care este timpul minim necesar pentru a satisface cerințele unui anumit client. O comandă constă în cererea de tipărire a unui anumit număr de reviste, pliante, cărți etc. Pentru fiecare produs care este tipărit, trebuie parcurs un șir de pași. De exemplu, prima mașină preia o foaie de hârtie dintr-un teanc, a doua taie hârtia pentru ca aceasta să aibă dimensiunile dorite, alta tipărește pagina, alta pune paginile tipărite într-un alt teanc, iar ultima lipește paginile din acest teanc. Acest flux de tipărire poate fi descris dându-se cele N operații necesare, timpul consumat de fiecare dintre acestea și dependențele care există între operații (de exemplu, operațiile 0, 1 și 2 pot începe imediat, operația 3 poate începe doar după terminarea operației 2, operația 4 poate începe după terminarea operațiilor 1 și 3, operația 5 poate începe după terminarea operației 1, iar operația 6 poate începe după terminarea operațiilor 0, 4 și 5). Trebuie să scrieți un program care determină timpul minim necesar pentru a satisface cerințele clientului.

Date de intrare

Fișierul de intrare **D.DAT** conține mai multe seturi de date, separate printr-o linie vidă. Fiecare set de date reprezintă comanda unui client și are următoarea structură:

- pe prima linie se află numărul N (cel mult 100) al operațiilor necesare pentru a satisface comanda;
- pe fiecare dintre următoarele N linii se află informații referitoare la o operație; primul număr de pe linie este codul de identificare al operației (cuprins între 0 și $N - 1$), al doilea (un întreg strict pozitiv) reprezintă timpul necesar pentru îndeplinirea operației, iar următoarele reprezintă codurile de identificare ale operațiilor care trebuie terminate înainte ca operația curentă să poată începe; ultimul număr al liniei este un terminator cu valoarea -1.

Date de ieșire

Pentru fiecare set de date se va tipări la ieșirea standard câte o linie care va conține timpul minim necesar pentru satisfacerea comenzii. În cazul în care comanda nu poate fi satisfăcută, se va scrie valoarea -1.

Exemplu

D.DAT

```
3
0 2 1 2 -1
1 3 -1
2 2 -1

2
0 1 1 -1
1 1 0 -1
```

Ecran

5
-1

P010205: Permutare

Pentru o permutare $A = \{a_1, a_2, \dots, a_n\}$ a mulțimii $\{1, 2, \dots, n\}$ definim o secvență $P(A) = (p_1, p_2, \dots, p_{n-1})$ unde $p_i = 0$ dacă $a_i > a_{i+1}$ și $p_i = 1$ dacă $a_i < a_{i+1}$. Se dă o permutare B și se cere determinarea numărului permutărilor C pentru care $P(B) = P(C)$, incluzând permutarea B .

Date de intrare

Fișierul de intrare **E.DAT** conține, pe câte o linie, mai multe seturi de date. Primul număr de pe linie reprezintă valoarea n și este urmat de alte n numere care reprezintă permutarea B . Toate numerele de pe o linie sunt separate prin câte un spațiu. Ultima linie a fișierului conține doar valoarea 0 și nu trebuie procesată.

Date de ieșire

La ieșirea standard se va scrie câte o linie pentru fiecare set de date. Fiecare dintre acestea va conține numărul permutărilor C corespunzătoare unei permutări date la intrare.

Exemplu

E.DAT

2 1 2
4 1 3 2 4
0

Ecran

1
5

P010206: Găsirea numărului

Se dă o mulțime S de numere întregi nenegative în care un anumit număr întreg poate apărea în S de mai multe ori și un număr întreg pozitiv k . Fie m cel mai mare număr întreg mai mic sau egal cu k pentru care există o submulțime a lui S ale cărei elemente au suma m .

Problema determinării numărului m dacă sunt cunoscute S și k , este foarte dificilă.

Sarcina voastră va fi puțin mai simplă. Va trebui să determinați un număr întreg p a cărui valoare este cu cel mult 10% mai mică decât m . Mai exact, numărul p trebuie să respecte inegalitatea $m \cdot 0.9 < p \leq m$.

Date de intrare

Prima linie a fișierului de intrare **F.DAT** conține numărul seturilor de date. Fiecare set de date este descris prin trei linii. Pe prima dintre ele se va afla numărul n ($1 \leq n \leq 500$) al elementelor din mulțimea S , pe a doua linie se vor afla cele n elemente (numere întregi nenegative) ale mulțimii, separate prin câte un spațiu, iar pe a treia linie se va afla numărul k ($1 \leq k \leq 2000000000$). Nu există linii vide între două seturi de date consecutive.

Date de ieșire

La ieșirea standard se va scrie câte o linie pentru fiecare set de date. Fiecare dintre acestea va conține un singur număr întreg p care satisface inegalitatea $m \cdot 0.9 < p \leq m$. Nu trebuie să apară nici un spațiu la începutul liniilor.

Exemplu

F.DAT

2
5
10 4 5 6 4
24
4
3 8 2 11
10

Ecran

22
10

P010207: Monede contrafăcute

Banca Populară "Marile Speranțe" verifică monedele pe care le are în seif. Monedele ar trebui să aibă aceeași greutate și sunt așezate în grămezi. Serviciul Secret a furnizat o informație potrivit căreia **exact o monedă din fiecare grămadă a fost înlocuită cu o monedă contrafăcută care este mai grea sau mai ușoară decât celelalte monede**. Conștientizați această informație ca fiind veritabilă și scrieți un program care verifică grămezile.

Pentru fiecare grămadă și o serie de măsurători, programul trebuie să încerce să identifice moneda contrafăcută și greutatea sa relativă.

Date de intrare

Fiecare linie din fișierul **G.DAT** conține informații referitoare la o grămadă și la măsurătorile efectuate asupra monedelor din grămadă.

Formatul unei linii este:

N M măsurătoare₁ ... măsurătoare _{M_P}

unde N ($3 \leq N \leq 1000$) este numărul monedelor din grămadă, iar M (număr pozitiv) este numărul măsurătorilor efectuate.

O măsurătoare are forma:

$monedă_1 \dots monedă_{w/2} indicator monedă_{w/2+1} \dots monedă_w$, unde w (număr par strict pozitiv) este numărul monedelor plasate pe talerele balanței, $monedă_i$ reprezintă o monedă amplasată pe taler ($1 \leq monedă_i \leq 1000$, $i = 1..w$, $monedă_i \neq monedă_j$ pentru $i \neq j$), iar indicator este unul din caracterele \backslash , $|$ sau $/$ și are următoarea semnificație:

- \backslash - balanța se înclină spre stânga;
- $|$ - balanța este echilibrată;
- $/$ - balanța se înclină spre dreapta.

Elementele din fișierul de intrare sunt separate prin cel puțin un spațiu. Datele din fișier respectă în totalitate aceste specificații, dar pot fi inconsistente sau incomplete pentru a putea determina soluția.





Date de ieșire

La ieșirea standard se va scrie câte o linie pentru fiecare set de date. Acestea pot avea unul dintre următoarele formate:

- Coin # is light - moneda cu numărul # este contrafăcută și este mai ușoară;
- Coin # is heavy - moneda cu numărul # este contrafăcută și este mai grea;
- Inconsistent data - informația furnizată de *Serviciul Secret* este contrazisă sau folosind anumite măsurători date la intrare, se ajunge la concluzii contradictorii;
- Incomplete data - datele de intrare nu furnizează informații suficiente pentru a găsi moneda contrafăcută și greutatea relativă a acesteia.

Exemplu

Cazurile din exemplul următor prezintă câteva situații de date de intrare corecte, incomplete sau inconsistente.

Pentru primul set de date avem trei monede și două măsurători. La măsurătoarea 1 | 2 se cântăresc monedele 1 și 2. Datorită faptului că balanța este echilibrată, ambele monede sunt veritabile. La măsurătoarea 1 \ 3 sunt cântărite monedele 1 și 3. Balanța se înclină spre stânga, deci moneda 3 este mai ușoară.

Pentru al doilea set de date (3 monede și două măsurători) este contrazisă informația furnizată de *Serviciul Secret* deoarece rezultă că toate cele trei monede sunt veritabile.

În cazul celui de-al treilea set de date măsurătorile duc la concluzii contradictorii:

- în urma primei măsurători tragem concluzia că monedele 1 și 2 sunt veritabile;
- în urma celei de-a doua măsurători tragem concluzia că una dintre monedele 1 și 2 este contrafăcută.

Pentru al patrulea set de date, putem trage concluzia că moneda 3 este contrafăcută, dar nu putem determina dacă ea este mai ușoară sau mai grea decât celelalte.

Pentru ultimul test, știm că una dintre monedele 3 și 4 este mai ușoară, dar nu știm care dintre ele.

G.DAT

```
3 2   1 | 2   1 \ 3
3 2   1 | 2   1 | 3
3 2   1 | 2   1 \ 2
3 1   1 | 2
4 2   1 | 2   1 2 \ 3 4
```

Ecran

```
Coin 3 is light
Inconsistent data
Inconsistent data
Incomplete data
Incomplete data
```

P010208: Joc

Brian este un entuziast al jocurilor pe calculator, în special al celor care simulează realitatea virtuală. Acum, el se află în fața *Porții Stelare*. Pentru a putea deschide poarta, el

trebuie să spargă protecția cât mai curând posibil. Pentru spargerea protecției, el are nevoie de un cod (o secvență de numere) care să se potrivească cu secvența de numere care reprezintă protecția *Porții Stelare*. Această secvență de numere este foarte lungă; poziția de start a primei apariții a codului în secvența de numere a protecției va deschide poarta; îl puteți ajuta?

Codul este o secvență de cel mult 60000 de numere întregi cuprinse între 0 și 255. Programul va trebui să găsească poziția primei potriviri (dacă există cel puțin o potrivire) sau să raporteze absența unei potriviri.

Date de intrare

Fișierului de intrare **H.DAT** conține mai multe seturi de date. Fiecare set de date este descris prin patru linii. Pe prima dintre ele se va afla numărul *code_dimension* care reprezintă lungimea codului. Pe cea de-a doua linie se află numerele întregi care reprezintă codul. A treia linie conține numărul *protection_dimension* care reprezintă dimensiunea protecției. Pe ultima linie corespunzătoare unui set de date se află numerele întregi care reprezintă protecția. Datele de pe aceeași linie a fișierului de intrare sunt separate prin unul sau mai multe spații. De asemenea, pot apărea spații la începutul și/sau la sfârșitul anumitor linii ale acestuia, sau linii vide sau care conțin doar spații. Acestea trebuie ignorate.

Date de ieșire

La ieșirea standard se va scrie câte o linie pentru fiecare set de date. În cazul în care codul se potrivește cu protecția, pe linia corespunzătoare se va scrie poziția primei apariții a codului în protecția porții. Numerotarea pozițiilor începe de la 0. Dacă nu există nici o potrivire, pe linia corespunzătoare se va scrie mesajul *no solution*.

Exemplu

H.DAT

```
3
0 1 2
7
2 3 4 0 1 2 5

2
1 4
6
4 1 4 1 4 4

3
1 2 3
7
3 2 1 3 2 255 7
```

Ecran

```
3
1
no solution
```