Faza JUDEȚEANĂ a olimpiadei de INFORMATICĂ pentru LICEE

În perioada 28 - 29 februarie a avut loc faza județeană a olimpiadei de informatică. Subiectele propuse spre rezolvare au fost aceleași în întreaga țară. Vă prezentăm în continuare enunțurile celor șase probleme propuse spre rezolvare elevilor de liceu.

Clasa a IX-a

P030419: Expresie

Se consideră un șir format din n numere naturale nenule $x_1, x_2, \ldots, x_n$ și un număr natural m.

Să se verifice dacă valoarea expresiei $\sqrt[m]{x_1 x_2 \ldots x_n}$ este un număr natural. În caz afirmativ să se afișeze acest număr descompus în factori primi.

Date de intrare

Fișierul de intrare **exp.in** conține pe prima linie un număr natural care reprezintă valoarea lui m, iar pe cea de-a doua linie conține un număr natural care reprezintă valoarea lui n.

Cea de-a treia linie a fișierului de intrare conține n numere naturale $x_1, x_2, \ldots, x_n$ separate între ele prin câte un spațiu.

Date de ieșire

Fișierul de ieșire **exp.out** trebuie să conțină pe prima linie cifra 0, dacă valoarea expresiei $\sqrt[m]{x_1 x_2 \ldots x_n}$ nu este un număr natural, respectiv 1 dacă este un număr natural.

Dacă valoarea expresiei este un număr natural, pe fiecare dintre următoarele linii se va scrie câte o pereche formată din două numere naturale p și e, separate între ele printr-un singur spațiu, cu semnificația: *p este factor prim care apare în descompunerea expresiei din enunț la puterea e* ($e \geq 1$).

Perechile se vor scrie în fișierul de ieșire în ordinea crescătoare a factorilor primi.

Restricții

- $0 < n < 5000$;
- $0 < x_i < 30000$, $i \in \{1, 2, \ldots, n\}$;
- $2 \leq m \leq 4$.

Exemple

exp.in	exp.out
2 4 32 81 100 19	0

exp.in	exp.out
2 4 32 81 100 18	1 2 4 3 3 5 1

Timp de execuție: 1 secundă/test

P030420: Reactivi

Într-un laborator de analize chimice se utilizează N reactivi. Se știe că, pentru a evita accidentele sau deprecierea reactivilor, aceștia trebuie să fie stocați în condiții de mediu speciale. Mai exact, pentru fiecare reactiv x, se precizează intervalul de temperatură $[min_x, max_x]$ în care trebuie să se încadreze temperatura de stocare a acestuia.

Reactivii vor fi plasați în frigidere. Orice frigider are un dispozitiv cu ajutorul căruia putem stabili temperatura (reprezentată de un număr întreg de grade *Celsius*) care va fi in interiorul acelui frigider .

Scrieți un program care să determine numărul minim de frigidere necesare pentru stocarea reactivilor chimici.

Date de intrare

Fișierul de intrare **react.in** conține pe prima linie numărul natural N, care reprezintă numărul de reactivi. Pe fiecare dintre următoarele N linii se află două numere întregi separate printr-un singur spațiu.

Numerele de pe linia $x + 1$ a fișierului de intrare reprezintă temperatura minimă min_x, respectiv temperatura maximă max_x de stocare a reactivului x.

Date de ieșire

Fișierul de ieșire **react.out** va conține o singură linie pe care este scris numărul minim de frigidere necesare.

Restricții și precizări

- $1 \le N \le 8000$;
- min_x și max_x sunt numere întregi cuprinse între -100 și 100, reprezentând grade *Celsius*, $\forall 1 \le x \le N$;
- $min_x \le max_x$;
- un frigider poate conține un număr nelimitat de reactivi.

Exemple

react.in
```
3
-10 10
-2 5
20 50
```
react.out
```
2
```

react.in
```
4
2 5
5 7
10 20
30 40
```
react.out
```
3
```

react.in
```
5
-10 10
10 12
-20 10
7 10
7 8
```
react.out
```
2
```

Timp de execuție: 1 secundă/test

Clasa a X-a

P030421: Perle

Granița nu se trece ușor. Asta pentru că *Balaurul Arhirel* (mare pasionat de informatică) nu lasă pe nimeni să treacă decât după ce răspunde la niște întrebări…

În acea țară există trei tipuri de perle normale (le vom nota cu 1, 2 și 3) și trei tipuri de perle magice (le vom nota cu *A*, *B* și *C*). Perlele magice sunt deosebite prin faptul că se pot transforma în alte perle (una sau mai multe, normale sau magice).

Perla magică de tipul *A* se poate transforma în orice perlă normală (una singură).

Perla magică de tipul *B* se poate transforma într-o perlă normală de tipul 2 și una magică de tipul *B*, sau într-o perlă normală de tipul 1, una magică de tipul *A*, una normală de tipul 3, una magică de tipul *A* și una magică de tipul *C*.

Perla magică de tipul *C* se poate transforma într-o perlă normală de tipul 2 sau într-o perlă normală de tipul 3, una magică de tipul *B* și una magică de tipul *C* sau într-o perlă normală de tipul 1, una normală de tipul 2 și una magică de tipul *A*.

Ca să rezumăm cele de mai sus putem scrie:

- A -> 1 | 2 | 3
- B -> $2B$ | $1A3AC$
- C -> 2 | $3BC$ | $12A$

Balaurul Arhirel ne lasă la început să ne alegem o perlă magică (una singură), iar apoi folosind numai transformările de mai sus trebuie să obținem un anumit șir de perle normale. Când o perlă magică se transformă, perlele din stânga și din dreapta ei rămân la fel (și în aceeași ordine). De asemenea, ordinea perlelor rezultate din transformare este chiar cea prezentată mai sus.

De exemplu, dacă balaurul ne cere să obținem șirul de perle 21132123, putem alege o perlă magică de tipul *B* și următorul șir de transformări: B -> $2B$ -> $21A3AC$ -> $21A3A12A$ -> 21132123.

Întrucât *Balaurul* nu are prea multă răbdare, el nu ne cere decât să spunem dacă se poate sau nu obține șirul respectiv de perle.

Să se determine pentru fiecare șir din fișierul de intrare dacă se poate obține prin transformările de mai sus sau nu (alegând orice primă perlă magică, la fiecare șir).

Date de intrare

Fișierul de intrare **perle.in** conține pe prima linie un număr natural *N*, reprezentând numărul de șiruri din fișierul de intrare. Fiecare a *i*-a linie dintre următoarele *N* linii descrie șirul *i*.

Un șir este dat printr-o succesiune de numere naturale despărțite prin câte un spațiu. Primul număr reprezintă lungimea șirului (L_i), iar următoarele L_i numere sunt tipurile de perle normale, în ordine, de la stânga la dreapta.

Date de ieșire

Fișierul **perle.out** trebuie să conțină *N* linii. Pe linia *i* se va scrie un singur număr 1 sau 0, 1 dacă se poate obține șirul respectiv (al *i*-lea) și 0 dacă nu se poate.

Restricții și precizări

- $0 < N < 11$;
- $0 < L_i < 10001$, $\forall 1 \le i \le N$.

Exemplu

perle.in
```
3
8 2 1 1 3 2 1 2 3
2 2 2
1 3
```
perle.out
```
1
0
1
```

Timp de execuție: 1 secundă/test

P030422: Romeo și Julieta

În ultima ecranizare a celebrei piese shakespeariene, *Romeo* și *Julieta* trăiesc într-un oraș modern, comunică prin *e-mail* și chiar învață să programeze. Într-o secvență tulburătoare sunt prezentate frământările interioare ale celor doi eroi încercând fără succes să scrie un program care să determine un punct optim de întâlnire.

Ei au analizat harta orașului și au reprezentat-o sub forma unei matrice cu *n* linii și *m* coloane, în matrice fiind marcate cu spațiu zonele prin care se poate trece (străzi lipsite de pericole) și cu *X* zonele prin care nu se poate trece.

De asemenea, în matrice au marcat cu *R* locul în care se află locuința lui *Romeo*, iar cu *J* locul în care se află locuința *Julietei*.

Ei se pot deplasa numai prin zonele care sunt marcate cu spațiu, din poziția curentă în oricare dintre cele 8 poziții învecinate (pe orizontală, verticală sau diagonale).

Cum lui *Romeo* nu îi place să aștepte și nici să se lase așteptat n-ar fi tocmai bine, ei au hotărât că trebuie să aleagă un punct de întâlnire în care atât *Romeo*, cât și *Julieta* să poată ajunge în același timp, plecând de acasă. Fiindcă la întâlniri amândoi vin într-un suflet, ei estimează timpul necesar pentru a ajunge la întâlnire prin numărul de elemente din matrice care constituie drumul cel mai scurt de acasă până la punctul de întâlnire. Cum probabil există mai multe puncte de întâlnire posibile, ei vor să îl aleagă pe cel în care timpul necesar pentru a ajunge la punctul de întâlnire este minim.

Scrieți un program care să determine o poziție pe hartă la care *Romeo* și *Julieta* pot să ajungă în același timp. Dacă există mai multe soluții, programul trebuie să determine o soluție pentru care timpul este minim.

Date de intrare

Fișierul de intrare **rj.in** conține pe prima linie două numere naturale, *N* și *M*, separate printr-un singur spațiu, care reprezintă numărul de linii și respectiv de coloane ale matricei, separate prin spațiu.

Pe fiecare dintre următoarele *N* linii se află *M* caractere (care pot fi doar *R*, *J*, *X* sau spațiu) reprezentând matricea.

Date de ieșire

Fișierul de ieșire **rj.out** va conține o singură linie pe care sunt scrise trei numere naturale separate prin câte un spațiu *tmin x y*, având semnificația:

- *tmin* reprezintă timpul minim în care *Romeo*, respectiv *Julieta* ajung la punctul de întâlnire;
- *x* reprezintă numărul liniei punctului de întâlnire și *y* reprezintă numărul coloanei.

Restricții și precizări

- $1 < N, M < 101$;
- liniile și coloanele matricei sunt numerotate începând cu 1;
- pentru datele de test există întotdeauna soluție.

Exemplu

rj.in

```
5 8
XXR  XXX
 X  X  X
J X X  X
       XX
XXX XXXX
```

rj.out

```
4 4 4
```

Explicație

Traseul lui *Romeo* poate fi: (1,3), (2,4), (3,4), (4,4). Deci timpul necesar lui *Romeo* pentru a ajunge de acasă la punctul de întâlnire este 4. Traseul *Julietei* poate fi: (3,1), (4,2), (4,3), (4,4). Timpul necesar *Julietei* pentru a ajunge de acasă la punctul de întâlnire este de asemenea 4.

În plus (4,4), este punctul cel mai apropiat de ei cu această proprietate.

Timp de execuție: 1 secundă/test

Clasele a XI-a și a XII-a

P030423: Moșia lui Păcală

Păcală a primit, așa cum era învoiala, un petec de teren de pe moșia boierului. Terenul este împrejmuit complet cu segmente drepte de gard care se sprijină la ambele capete de câte un par zdravăn.

La o nouă prinsoare, *Păcală* iese iar în câștig și primește dreptul să strămute niște pari, unul câte unul, cum i-o fi voia, astfel încât să-și extindă suprafața de teren. Dar învoiala prevede că fiecare par poate fi mutat în orice direcție, dar nu pe o distanță mai mare decât o valoare dată (scrisă pe fiecare par) și fiecare segment de gard, fiind cam șubred, poate fi rotit și prelungit de la un singur capăt, celălalt rămânând nemișcat.

Cunoscând pozițiile inițiale ale parilor și valoarea înscrisă pe fiecare par, se cere suprafața maximă cu care poate să-și extindă *Păcală* proprietatea. Se știe că parii sunt dați într-o ordine oarecare, pozițiile lor inițiale sunt date prin numere întregi de cel mult 3 cifre, distanțele pe care fiecare par poate fi deplasat sunt numere naturale strict pozitive și figura formată de terenul inițial este un poligon neconcav.

Date de intrare

Fișierul **MOSIA.IN** conține pe prima linie un număr natural *n* care reprezintă numărul de pari. Fiecare a *i*-a linie dintre următoarele *n* linii conține trei numere naturale, separate printr-un singur spațiu, x_i, y_i și d_i, care au semnificația: *parul i are coordonatele inițiale* (x_i, y_i) *și poate fi deplasat pe distanța* d_i.

Date de ieșire

În fișierul **MOSIA.OUT** se scrie un singur număr real cu 4 zecimale care reprezintă suprafața maximă cu care se poate mări moșia.

Restricții și precizări

- $3 < N \le 200$;
- x_i și y_i sunt numere întregi cuprinse între -1000 și 1000, $\forall 1 \le i \le N$;
- d_i este număr întreg cuprins între 0, exclusiv, și 20, $\forall 1 \le i \le N$;
- poligonul neconcav se definește ca un poligon convex cu unele vârfuri coliniare;
- pozițiile parilor sunt date într-o ordine oarecare;
- poligonul obținut după mutarea parilor poate fi concav;
- pozițiile finale ale parilor nu sunt în mod obligatoriu numere naturale.

Exemplu

MOSIA.IN

```
4
-3 0 2
3 0 3
0 6 2
0 -6 6
```

MOSIA.OUT

```
30.0000
```

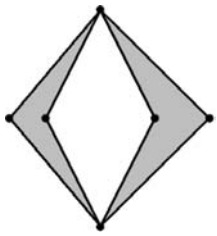

Figura 1

Explicație

Prin mutarea parilor 1 și 2 cu câte 2, respectiv 3 unități, se obține un teren având suprafața cu 30 de unități mai mare decât terenul inițial.

Timp de execuție: 1 secundă/test

P030424: Lanterna

Un agent secret are o hartă pe care sunt marcate N obiective militare. El se află, inițial, lângă obiectivul numerotat cu 1 (baza militară proprie) și trebuie să ajungă la obiectivul numerotat cu N (baza militară inamică). În acest scop, el va folosi drumurile existente, fiecare drum legând două obiective distincte.

Fiind o misiune secretă, deplasarea agentului va avea loc noaptea; de aceea, el are nevoie de o lanternă. Pentru aceasta, el are de ales între K tipuri de lanterne - o lanternă de tipul W ($1 \le W \le K$) are baterii care permit consumul a W wați, după consumul acestor wați, lanterna nu mai luminează.

Din fericire, unele dintre obiective sunt baze militare prietene, astfel că, o dată ajuns acolo, el își poate reîncărca bateriile complet. Agentul trebuie sa aibă grijă ca, înainte de a merge pe un drum între două obiective, cantitatea de wați pe care o mai poate consuma să fie mai mare sau egală cu cantitatea de wați pe care o va consuma pe drumul respectiv.

Cunoscând drumurile dintre obiective și, pentru fiecare drum, durata necesară parcurgerii drumului și numărul de wați consumați de lanternă, determinați tipul de lanternă cu numărul cel mai mic, astfel încât durata deplasării să fie minimă (dintre toate tipurile de lanternă cu care se poate ajunge în cel mai scurt timp la destinație, interesează lanterna cu consumul cel mai mic).

Date de intrare

Pe prima linie a fișierului **lanterna.in** se află numerele întregi N și K, separate printr-un spațiu. Pe următoarea linie se află N numere întregi din mulțimea {0, 1}. Dacă al i-lea număr este 1, aceasta înseamnă că obiectivul cu numărul i este o bază militară prietenă (adică agentul își poate reîncărca bateriile lanternei dacă ajunge la acest obiectiv); dacă numărul este 0, agentul nu își va putea reîncărca bateriile. Primul număr din linie este 1, iar ultimul este 0. Pe cea de-a treia linie a fișierului se află numărul M de drumuri dintre obiective. Fiecare din următoarele M linii conține câte patru numere întregi separate prin spații: a b T W, având semnificația că există un drum bidirecțional între obiectivele a și b ($a \ne b$), care poate fi parcurs într-un timp T și cu un consum de W wați.

Date de ieșire

În fișierul **lanterna.out** se vor afișa două numere întregi, separate printr-un spațiu: *Tmin* și *Wmin*. *Tmin* reprezintă durata minimă posibilă a deplasării de la obiectivul 1 la obiectivul N, iar *Wmin* reprezintă tipul de lanternă cu numărul cel mai mic pentru care se obține acest timp.

Restricții și precizări

- $2 \le N \le 50$:
- $1 \le K \le 1000$;
- $1 \le M \le N \cdot (N-1)/2$;
- între două orașe diferite poate exista maximum un drum direct;
- pentru fiecare drum, durata parcurgerii este un număr întreg cuprins între 1 și 100, iar numărul de wați consumați este un număr întreg între 0 și 1000;
- se garantează că există cel puțin un tip de lanternă pentru care deplasarea să fie posibilă;
- punctajul pentru un test se va acorda în felul următor:
  - 30%: dacă este determinat corect *Tmin*;
  - 100%: dacă sunt determinate corect atât *Tmin*, cât și *Wmin.*

Exemplu

lanterna.in

```
7 10
1 0 1 0 0 0 0
7
1 2 10 3
1 4 5 5
2 3 10 3
4 3 15 1
3 6 4 3
6 5 2 2
5 7 1 0
```

lanterna.out

```
27 6
```

Timp de execuție: 1 secundă/test