

CLASA a IX-a

PROBLEMA 1

HANSEL

100 puncte

Celebrele personaje ale fraților Grimm, Hansel și Gretel au aflat, după cum știți, de intenția mamei lor de a-i părăsi în pădure. Hansel, pasionat de algoritmică, o asigură pe Gretel că va marca drumul cu N pietricele fluorescente pentru a se putea întoarce în siguranță acasă. Hansel s-a gândit să eticheteze cele N pietricele fluorescente în funcție de distanțele dintre ele, folosind pentru a măsura distanțele un *rigletcount*.

Hansel s-a gândit ca eticheta ETI_k să reprezinte distanța de la pietricica cu eticheta ETI_k până acasă (locul de plecare). Etichetele trebuie să fie în ordine strict crescătoare și să aibă proprietatea că distanța dintre pietricele cu etichetele ETI_k și ETI_{k-1} este $ETI_k - ETI_{k-1}$, distinctă pentru orice k , ($1 < k \leq N$) și diferită de orice etichetă.

Întrucât nu are suficiente pietricele, Hansel vrea ca la alegerea etichetelor $ETI_1, ETI_2, \dots, ETI_N$ să folosească cele mai mici numere naturale.

Problema este că tocmai aseară lui Hansel i s-a stricat calculatorul și vă roagă pe voi să-l ajutați să găsească numărul ETI_N .

Date de intrare:

Fișierul de intrare **hansel.in** conține un singur număr N , reprezentând numărul de pietricele fluorescente.

Date de ieșire:

În fișierul de ieșire **hansel.out** se va tipări eticheta ETI_N de pe pietricica N .

Restricții și precizări:

- $1 \leq N \leq 10.000.000$

Exemplu:

hansel.in	hansel.out	Explicație:
8	45	ETI: 1 3 7 12 18 26 35 45 Distanțele între pietricele: 2 4 5 6 8 9 10

Timp maxim de execuție: 0,05 secunde / test

Memorie: 2 MB

ENTERP

100 puncte

Nava spațială Enterprise pe care se află echipajul Star Trek, este atacată cu o armă care emite radiații de intensitate N. Radiațiile pot fi contracarate de plasarea a două barje antiradiații, pe axa de deplasare a navei, în direcția ei de deplasare. Barjele detectează arma și generează unde de ecranare a navei. Punctele de emisie ale undelor de ecranare sunt generate de cele două barje tot pe axa de deplasare. Ca nava să fie total ecranată, trebuie generate un număr maxim de puncte de emisie a undelor de ecranare, funcție de coordonatele x și y ale barjelor pe axa de deplasare (originea axei de deplasare se află într-un punct din coada navei). Coordonatele barjelor trebuie să îndeplinească următoarele condiții: să fie numere naturale mai mici sau egale cu N, care generează cel mai lung sir de numere naturale după regula $x = y*[x/y] + p_k$, $p_k \leq y$; dacă $p_k \neq 0$, $y = p_k * [y/p_k] + p_{k-1}$, $p_{k-1} \leq p_k$; dacă $p_{k-1} \neq 0$, $p_k = p_{k-1} * [p_k/p_{k-1}] + p_{k-2}$, $p_{k-2} \leq p_{k-1}$; ...; dacă $p_2 \neq 0$, $p_3 = p_2 * [p_3/p_2] + p_1$, $p_1 \leq p_2$, ultimul punct p_0 fiind chiar originea axei ($p_0=0$); prin $[a/b]$ s-a notat cîntul împărțirii numărului a la numărul b ≠ 0. Perechea (x, y) cu această proprietate trebuie să fie o pereche “minimală”, adică dacă există o altă pereche (a, b) care generează un asemenea sir tot de aceeași lungime, atunci fie $x < a$, fie $x=a$ și $y < b$.

Ajuatați-l pe locotenentul DATA de pe nava Enterprise să găsească rapid numărul de puncte de emisie și coordonatele punctelor unde trebuie plasate celor două barje pentru a asigura ecranarea navei.

Date de intrare: Fișierul **enterp.in** conține un singur număr natural N.

Date de ieșire: Fișierul **enterp.out** va conține pe prima linie numărul maxim de puncte de emisie. A doua linie va conține un număr natural x reprezentând primul număr al perechii minime identificate, iar pe a treia linie se va scrie numărul y reprezentând al doilea număr din pereche.

Restricții și precizări

$$4 < N < 10^{400}$$

Exemple

enterp.in	enterp.out	Explicație
8	5	Numărul maxim de puncte de emisie generate este 5.
	5	Perechea de coordonate ale barjelor 5 și 8 este „minimală” cu această proprietate.
12345678910	48	Numărul maxim de puncte de emisie generate de oricare pereche de numere mai mici sau egale cu 12345678910 este 48.
	4807526976	Perechea (4807526976, 7778742049) este „minimală” cu această proprietate.
	7778742049	

Timp maxim de execuție: 0,05 secunde / test

Memorie: 1 MB

R2D2

100 puncte

Celebrul roboțel R2D2 din filmul Star Wars, a primit misiunea de a ajunge într-un punct din spațiu în care Luke este prins într-o capcană. După cum știți, R2D2 nu se poate deplasa decât pe direcțiile nord-sud, est-vest sau sus-jos, adică înainte, înapoi, la dreapta, la stânga, sus sau jos. Cu GPS-ul său, R2D2 a reușit să determine coordonatele în care se află $(lat_0, long_0, alt_0)$ și coordonatele în care se află Luke $(lat_L, long_L, alt_L)$. R2D2 are la dispoziție **n** acumulatori de energie, de trei tipuri: o parte din acumulatori pot fi utilizati la deplasarea pe direcția nord-sud, o parte pe direcția est-vest și alții pe direcția sus-jos (utilizând un rucsac zburător). De asemenea, pe eticheta acumulatorilor este înscrisă și cantitatea de energie pe care o poate degaja fiecare. O unitate de energie permite parcurgerea unei unități de deplasare în direcția lat, long sau alt. R2D2 poate folosi un acumulator doar pe direcția indicată pe etichetă și numai dacă are de parcurs în acea direcție o porțiune de drum pe care acumulatorul să se descarcă integral.

Noul roboțel BB8, pus pe șotii, a răvășit baza de date, astfel încât R2D2, trebuie să-și aleagă din cei **n** acumulatori pe aceia pentru care consumul de energie pe drumul până la Luke este minim, utilizând acumulatorii în condițiile descrise mai sus. În baza de date pot fi acumulatori ce pot fi folosiți pe aceeași direcție și pot degaja și aceeași cantitate de energie, dar fiecare acumulator va fi utilizat o singură dată.

R2D2 vă roagă să-i scrieți un program care să determine cantitatea de energie minimă necesară pentru a ajunge la Luke.

Date de intrare: Fișierul **r2d2.in** conține :

- pe prima linie numărul **n** de acumulatori;
- pe următoarea linie coordonatele în care se află R2D2: **lat₀, long₀ și alt₀**, numere naturale separate printr-un spațiu;
- pe următoarea linie coordonatele punctului unde se află Luke: **lat_L, long_L și alt_L**, numere naturale separate printr-un spațiu;
- pe următoarele **n** linii descrierea fiecărui acumulator realizată astfel: un caracter (E pentru direcția de deplasare est sau vest, N pentru direcția de deplasare nord sau sud, respectiv J pentru deplasările pe direcția sus sau jos), urmat de spațiu și apoi de un număr natural reprezentând cantitatea de energie pe care o degajă la deplasare.

Date de ieșire: Fișierul **r2d2.out** va conține pe prima linie un număr reprezentând cantitatea de energie minimă determinată sau cuvântul **imposibil** dacă nu se poate realiza drumul cu acumulatorii disponibili.

Restricții și precizări:

$$1 \leq n \leq 1.000$$

$$0 \leq \text{cantitatea de energie a fiecărui acumulator} \leq 10.000$$

$$0 \leq lat_0, long_0, alt_0, lat_L, long_L, alt_L \leq 1.000.000$$

Exemplu

r2d2.in	r2d2.out	Descriere exemplu
12 3 2 7 8 6 13 E 9 N 3 J 5 N 13 N 8 J 3 E 2 N 4 J 7 J 2 E 15 N 12	47	O posibilă deplasare cu consum minim de energie este:

Timp maxim de execuție: 0,4 secunde / test**Memorie:** 10 MB

BRONTO

100 puncte

Pe planeta Apatosaurus din galaxia Aicad trăiesc brontozaurii. Aceștia sunt organizați în mai multe ecosisteme de producere a energiei ecologice, necesare comunității planetei. Într-un astfel de ecosistem, există N brontozauri. Fiecare dintre ei își rezervă inițial un teritoriu unde trăiește o perioadă de timp T , aceeași pentru toți brontozaurii: brontozaurul k trăiește în teritoriul k o unitate de timp T . După o lege bine stabilită de Consiliul Galaxiei, brontozaurul președinte al ecosistemului asigură schimbarea teritoriilor la fiecare perioadă T . Ecosistemul produce energie până când toți brontozaurii din ecosistem ajung fiecare în teritoriul său inițial. Sistemul este stabil și comunitatea este mulțumită, pentru că legea unui ecosistem nu se schimbă, decât după expirarea timpului de funcționare. Preocuparea Consiliului Galaxiei este de a stabili care este perioada maximă de funcționare a unui astfel de ecosistem, pentru ca apoi să transmită brontozaurilor președinții legea de mișcare pentru fiecare ecosistem, valabilă după fiecare unitate de timp T .

Date de intrare: fișierul **bronto.in** conține pe prima linie un număr natural N , care reprezintă numărul brontozaurilor din ecosistem.

Date de ieșire: fișierul **bronto.out** conține pe prima linie numărul natural M , care reprezintă numărul maxim de unități T , după care ecosistemul nu mai funcționează.

Restricții și precizări: $1 < N \leq 5000$;

Exemplu:

bronto.in	bronto.out	Explicație
8	15	<p>Pentru un M maxim, o lege este următoarea (și nu este unică cu această proprietate):</p> <p>Configurațiile după fiecare T sunt, în ordine:</p> <pre> T0: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 T1: 6 - 4 - 1 - 5 - 7 - 3 - 8 - 2 T2: 3 - 5 - 6 - 7 - 8 - 1 - 2 - 4 T3: 1 - 7 - 3 - 8 - 2 - 6 - 4 - 5 T4: 6 - 8 - 1 - 2 - 4 - 3 - 5 - 7 T5: 3 - 2 - 6 - 4 - 5 - 1 - 7 - 8 T6: 1 - 4 - 3 - 5 - 7 - 6 - 8 - 2 T7: 6 - 5 - 1 - 7 - 8 - 3 - 2 - 4 T8: 3 - 7 - 6 - 8 - 2 - 1 - 4 - 5 T9: 1 - 8 - 3 - 2 - 4 - 6 - 5 - 7 T10: 6 - 2 - 1 - 4 - 5 - 3 - 7 - 8 T11: 3 - 4 - 6 - 5 - 7 - 1 - 8 - 2 T12: 1 - 5 - 3 - 7 - 8 - 6 - 2 - 4 T13: 6 - 7 - 1 - 8 - 2 - 3 - 4 - 5 T14: 3 - 8 - 6 - 2 - 4 - 1 - 5 - 7 T15: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 M = 15 </pre>

Timp de execuție maxim: 0,2 secunde / test.

Memorie: 3 MB

CLASA a XI-a

PROBLEMA 1

GARLAIL

100 puncte

Pe planeta Garlail locuiesc N rebeli planetari, care vor să atace întreg sistemul solar. Rebelii au un sistem de comunicare foarte bun, cu care pot comunica unul cu altul direct sau prin intermediul altui rebel. Sistemul de comunicare are totuși niște puncte vulnerabile. Rebelii sunt puternici atâtă timp cât pot comunica toți între ei. Sistemul de bruiaj de pe nava Enterprise, aflată întâmplător în explorarea sistemului, nu poate bruia decât comunicațiile unui singur rebel. Locotenentul DATA caută o soluție care să găsească rebelul ale cărui comunicații trebuie să fie bruiate, astfel încât sistemul de comunicare al rebelilor să aibă cât mai mult de suferit, prin disipaerea rebelilor în cât mai multe grupuri izolate.

Date de intrare:

Fișierul de intrare ***garlail.in*** conține, pe prima linie, numărul N de rebeli și numărul M de relații de comunicare directă între aceștia. Pe următoarele M linii sunt indicate căile de comunicare, sub forma câte unei perechi X Y cu semnificația că rebelul X comunică direct cu rebelul Y , și invers.

Date de ieșire:

În fișierul de ieșire ***garlail.out*** se vor scrie, pe prima linie, $R =$ numărul de grupuri izolate în care se poate disipa sistemul de comunicare și $V =$ numărul de variante de a bruia un rebel din sistem care să producă acest număr maxim de grupuri izolate. Pe a doua linie se vor scrie cele V numere de ordine ale acestor rebeli, în ordine crescătoare și despărțite prin spații.

Restricții și precizări:

- $1 \leq N \leq 100.000$
- $1 \leq M \leq 300.000$
- $1 \leq X, Y \leq N$
- DATA garantează că există cel puțin un rebel care, scos din sistem, produce fragmentarea sistemului de comunicare.

Exemplu:

<i>garlail.in</i>	<i>garlail.out</i>	Explicații
9 10	3 2	Prin bruierea comunicațiilor rebelului 3, sistemul se fragmentează în 3 grupuri izolate: {1, 2}, {4, 5} și {6, 7, 8, 9}.
1 2	3 6	Prin bruierea comunicațiilor rebelului 6, sistemul se fragmentează în 3 grupuri izolate: {1, 2, 3, 4, 5}, {7} și {8, 9}.
3 1		Bruierea comunicațiilor rebelului 8 fragmentează sistemul în numai două grupuri izolate, iar bruierea comunicațiilor altor rebeli nu încurcă cu nimic sistemul lor de comunicății.
2 3		
6 7		
8 6		
6 3		
3 4		
5 3		
8 9		
4 5		

Timp maxim de execuție: 0,2 secunde / test**Memorie:** 12 MB

BRONTO

100 puncte

Pe planeta Apatosaurus din galaxia Aicad trăiesc brontozaurii. Aceștia sunt organizați în mai multe ecosisteme de producere a energiei ecologice, necesare comunității planetei. Într-un astfel de ecosistem, există N brontozauri. Fiecare dintre ei își rezervă inițial un teritoriu unde trăiește o perioadă de timp T , aceeași pentru toți brontozaurii: brontozaurul k trăiește în teritoriul k o unitate de timp T . După o lege bine stabilită de Consiliul Galaxiei, brontozaurul președinte al ecosistemului asigură schimbarea teritoriilor la fiecare perioadă T . Ecosistemul produce energie până când toți brontozaurii din ecosistem ajung fiecare în teritoriul său inițial. Sistemul este stabil și comunitatea este mulțumită, pentru că legea unui ecosistem nu se schimbă, decât după expirarea timpului de funcționare. Preocuparea Consiliului Galaxiei este de a stabili care este perioada maximă de funcționare a unui astfel de ecosistem, pentru ca apoi să transmită brontozaurilor președinți legea de mișcare pentru fiecare ecosistem, valabilă după fiecare unitate de timp T .

Date de intrare: fișierul **bronto.in** conține pe prima linie un număr natural N , care reprezintă numărul brontozaurilor din ecosistem.

Date de ieșire: fișierul **bronto.out** conține pe prima linie numărul natural M , care reprezintă numărul maxim de unități T , după care ecosistemul nu mai funcționează.

Restricții și precizări: $1 < N \leq 5000$;

Exemplu:

bronto.in	bronto.out	Explicație
8	15	<p>Pentru un M maxim, o lege este următoarea (și nu este unică cu această proprietate):</p> <p>Configurațiile după fiecare T sunt, în ordine:</p> <p> T0: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 T1: 6 - 4 - 1 - 5 - 7 - 3 - 8 - 2 T2: 3 - 5 - 6 - 7 - 8 - 1 - 2 - 4 T3: 1 - 7 - 3 - 8 - 2 - 6 - 4 - 5 T4: 6 - 8 - 1 - 2 - 4 - 3 - 5 - 7 T5: 3 - 2 - 6 - 4 - 5 - 1 - 7 - 8 T6: 1 - 4 - 3 - 5 - 7 - 6 - 8 - 2 T7: 6 - 5 - 1 - 7 - 8 - 3 - 2 - 4 T8: 3 - 7 - 6 - 8 - 2 - 1 - 4 - 5 T9: 1 - 8 - 3 - 2 - 4 - 6 - 5 - 7 T10: 6 - 2 - 1 - 4 - 5 - 3 - 7 - 8 T11: 3 - 4 - 6 - 5 - 7 - 1 - 8 - 2 T12: 1 - 5 - 3 - 7 - 8 - 6 - 2 - 4 T13: 6 - 7 - 1 - 8 - 2 - 3 - 4 - 5 T14: 3 - 8 - 6 - 2 - 4 - 1 - 5 - 7 T15: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 $M = 15$ </p>

Timp de execuție maxim: 0,2 secunde / test.

Memorie: 3 MB

ICARSU

100 puncte

Cercetătorii de la ICARSU (Institutul de Cercetări Aero spațiale pentru Studiul Universului), au descoperit un nou sistem planetar din constelația ROBAR. S-a observat că sistemul are N planete, între care există M rute de transport bidirecționale; transportul pe fiecare rută necesită un consum specific de energie. ICARSU a decis să asigure un sistem de transport cu navete care să permită cercetarea tuturor planetelor din ROBAR.

Având în vedere că energia costă, cercetătorii vor să știe care sunt rutele prin care se poate ajunge la oricare planetă, cu consum total minim de energie.

Energia potențială a sistemului planetar este folosită pentru a deschide, în fiecare an galactic, o nouă rută de transport între două dintre planetele sistemului. Cercetătorii trebuie să recalculeze consumul total minim de energie necesară operării rutelor de transport între cele N planete, pentru fiecare din cei K ani galactici, cât se estimează că va dura cercetarea sistemului ROBAR.

Scrieți un program care să îndeplinească cerințele cercetătorilor.

Date de intrare

Fișierul de intrare **icarsu.in** conține

- pe prima linie numerele N, M și K , separate prin spații;
- pe următoarele M linii tripletele $a \ b \ c$ cu semnificația „între planetele a și b există o rută de transport, pentru parcurgerea căreia consumul de energie este c ”;
- pe ultimele K linii tripletele $a_i \ b_i \ c_i$ cu semnificația „în al i -lea an galactic apare o rută de transport între planetele a_i și b_i , pentru parcurgerea căreia consumul de energie este c_i ”.

Date de ieșire

În fișierul de ieșire **icarsu.out** se vor tipări $K + 1$ linii conținând, în ordine, valorile C_0, C_1, \dots, C_K , reprezentând consumul energetic total minim în starea inițială și în fiecare din cei K ani galactici.

Restricții și precizări:

- $1 \leq N \leq 10.000$
- $1 \leq M \leq 100.000$
- $1 \leq K \leq 1.000$
- $1 \leq c_0, c_i \leq 100.000$
- Între oricare două planete există (sau apare) cel mult o rută de transport.
- Se garantează că între oricare două planete există cel puțin o rută, directă sau indirectă.

Exemplu

icarsu.in	icarsu.out	Explicații
6 9 3	12	Inițial, consumul minim se obține, de exemplu, folosind următoarele rute de transport: (1, 2), (1, 3), (1, 6), (3, 4), (5, 6).
1 2 3	11	După primul an galactic, consumul minim se obține, de exemplu, folosind următoarele rute de transport: (1, 2), (1, 3), (1, 5), (3, 4), (5, 6).
1 3 1	11	
1 4 8	9	
1 6 3		După al doilea an galactic, consumul minim se

2 3 4		obține, de exemplu, folosind următoarele rute de transport: (1, 2), (1, 3), (1, 5), (3, 4), (5, 6). După al treilea an galactic, consumul minim se obține, de exemplu, folosind următoarele rute de transport: (1, 3), (1, 5), (2, 4), (3, 4), (5, 6).
3 4 3		
4 5 4		
4 6 6		
5 6 2		
1 5 2		
2 5 4		
2 4 1		

Timp maxim de execuție: 0,1 secunde / test

Memorie: 1 MB

AR

100 puncte

Făt - Frumos află că Ileana Cosânzeana a fost răpită de Zmeul Zmeilor. În încercarea lui de a o salva, ajunge pe tărâmul Zmeului, dar nu poate pătrunde în castel, pentru că Zmeul Zmeilor are un sistem de pază format din N roboți. Făt Frumos studiază sistemul de pază și observă că fiecare din cei N roboți se mișcă liniar, cu o viteză constantă, pe aceeași axă paralelă cu zidul castelului.

Roboții pot fi neutralizați printr-un sistem de radiații, iar Făt Frumos are în posesia sa AR-uri (aruncătoare de radiații). Un AR poate anula acțiunea tuturor roboților aflați la o distanță mai mică sau egală cu raza de acțiune a radiației emisă de AR. Numai că Făt Frumos trebuie să utilizeze un număr minim de AR-uri și nu le poate lansa decât o singură dată, pe toate în același timp. Timpul îl presează, pentru că Zmeul se poate întoarce la castel și atunci lucrurile se complică. Ajutați-l pe Făt Frumos să calculeze câte AR-uri poate utiliza pentru neutralizarea acțiunii celor N roboți și care este timpul minim la care le poate lansa pentru a realiza acest lucru.

Date de intrare

Fișierul de intrare **ar.in** conține pe prima linie un număr natural N , reprezentând numărul de roboți, urmat de un număr real R , reprezentând raza de acțiune a AR-urilor. Pe următoarele N linii sunt descriși cei N roboți, câte unul pe linie. Pe fiecare linie vor fi scrise două numere separate prin spațiu x v , unde x reprezintă un număr natural ce indică distanța robotului (măsurată în metri față de intrarea în castel, considerată originea axei de pază) la momentul de timp 0, iar v este un număr real indicând viteza (măsurată în metri / secundă) cu care se deplasează robotul. Dacă viteza este pozitivă, deplasarea robotului se realizează în sensul pozitiv al axei de pază, iar dacă viteza este negativă, deplasarea robotului se realizează în sens invers.

Date de ieșire

Fișierul de ieșire **ar.out** va conține o singură linie pe care vor fi scrise două numere separate prin spațiu nr_{min} t_{min} , unde nr_{min} reprezintă numărul minim de AR-uri necesare pentru neutralizarea tuturor roboților, iar t_{min} este timpul minim (exprimat în secunde, cu 3 zecimale exacte) la care pot fi lansate AR-urile.

Restricții și precizări:

$$1 \leq N \leq 300$$

$$0 < R \leq 5$$

$$x_i \leq 1.000.000 \text{ m, pentru orice } 1 \leq i \leq N$$

$$|v_i| \leq 100 \text{ m/s, pentru orice } 1 \leq i \leq N$$

Este posibil ca la un moment dat doi roboți să se afle în același punct, dar niciodată nu vor exista 3 roboți în același punct.

Timpul minim afișat este considerat corect dacă diferă prin cel mult 0.001 secunde față de timpul minim corect.

Exemplu

ar.in	ar.out
6 0.25	4 0.333
2 0.5	
3 -1	
5 -1	
5 0.333	
8 0.5	
10 -0.2	

Timp maxim de execuție: 0.5 secunde/test

Memorie: 1 MB