



Olimpiada **BALCANICĂ** de informatică

În perioada 14-20 iulie 2003 a avut loc cea de-a XI-a ediție a Olimpiadei Balcanice de Informatică. Acest concurs a fost inițiat de țara noastră în anul 1993. După zece ani Balcaniada s-a întors în România, ediția din acest an a concursului desfășurându-se la Iași.

Beneficiind de avantajele țării organizatoare, România a avut dreptul de a avea două echipe în concurs. Cei opt concurenți din România care au participat la această competiție sunt *Leonard Crestez, Dan Ghinea, Emilian Miron, Marius Nicolae, Alexandru Moșoi, Matei Gruber, Tiberiu Dăneș* și *Dan-Ionuț Fechete*.

Primii doi au obținut medalii de aur, următorii doi medalii de argint, iar ceilalți patru medalii de bronz.

În continuare vă prezentăm enunțurile celor șase probleme propuse spre rezolvare.

P060313: Secvența Farey

Secvența *Farey* de ordin N este definită ca fiind o secvență ordonată crescător care conține toate fracțiile ireductibile de forma p/q pentru care $0 < p < q \leq N$. De exemplu, secvența *Farey* de ordin 5 este următoarea:

1/5 1/4 1/3 2/5 1/2 3/5 2/3 3/4 4/5.

Numerotăm toate elementele secvenței începând cu valoarea 1. De exemplu, fracția cu numărul de ordin șase în secvența anterioară este 3/5.

Dându-se două valori N și K , determinați fracția cu numărul de ordine K în secvența *Farey* de ordin N .

Date de intrare

Fișierul de intrare **farey.in** conține o singură linie pe care se află valorile N și K , separate printr-un spațiu.

Date de ieșire

Fișierul de ieșire **farey.out** va conține o singură linie pe care se vor afla numerele întregi p și q , separate printr-un spațiu. Frația p/q trebuie să aibă numărul de ordine K în secvența *Farey* de ordin N .

Restricții și precizări

- $1 \leq N \leq 40000$;
- pentru 40% dintre seturile de date folosite pentru testare vom avea $K \leq 40000$;
- pentru toate seturile de date folosite pentru testare vom avea $N \geq 10000$;
- valoarea K va reprezenta întotdeauna un număr de ordine valid; cu alte cuvinte, va fi mai mare sau egală cu 1 și mai mică sau egală cu numărul elementelor secvenței *Farey* de ordin N ;
- fracția p/q trebuie să fie ireductibilă; cu alte cuvinte, cel mai mare divizor comun al numerelor scrise în fișierul de ieșire trebuie să fie 1.

Exemplu

farey.in

5 6

farey.out

3 5

Timp de execuție: 0,3 secunde/test

P060314: Traversare

Am ajuns la un lac mare și trebuie să îl traversez. Din fericire, sunt un săritor foarte bun, dar lacul este mult prea mare pentru ca să îl pot traversa dintr-o singură săritură. Pe mal găsesc N cutii de diferite înălțimi, amplasate într-o anumită ordine. Dacă arunc o cutie în lac, ea va pluti și va avea înălțimea pe care o are pe mal. Acesta este un lucru bun, deoarece intenționez să arunc câteva cutii în lac și să ajung de la un mal la altul sărind de pe o cutie pe alta.

Trebuie să iau în considerare următoarele:

- lacul este mare, deci trebuie să arunc cel puțin două cutii, ceea ce înseamnă că pentru a traversa lacul, va trebui să sar de cel puțin trei ori;



- nu trebuie să arunc toate cutiile; pot să ignor unele dintre ele;
- pot să arunc cutiile în lac doar în ordinea în care acestea se află pe mal; așadar, trebuie să sar de pe una pe alta în această ordine;
- pot sări de pe o cutie pe alta doar dacă diferența dintre înălțimile celor două cutii este de cel mult H metri deoarece pot să sar destul de mult în lungime, dar am unele probleme cu săriturile în înălțime;
- înălțimea unei cutii nu se modifică în momentul în care sar pe ea;
- pot întotdeauna să sar de pe mal pe o cutie și de pe o cutie pe mal indiferent care este înălțimea cutiei respective;

Confruntându-mă cu numărul mare de posibilități care respectă condițiile descrise anterior, am început să număr variantele în loc să traversez lacul. Am găsit răspunsul foarte repede și mă întreb dacă îl puteți și voi găsi la fel de repede.

Scrieți un program care determină numărul posibilităților de a traversa lacul respectând condițiile anterioare. Datorită faptului că numărul poate fi foarte mare, va trebui să determinați doar restul împărțirii sale la 9901.

Date de intrare

Prima linie a fișierului de intrare **travers.in** conține numerele întregi N și H , separate printr-un spațiu. Aceste numere reprezintă numărul cutiilor, respectiv diferența maximă a înălțimilor a două cutii consecutive.

Următoarele N linii conțin câte un număr reprezentând înălțimea unei cutii. Înălțimile sunt date în ordinea în care se află cutiile pe mal. Cea de-a $(i + 1)$ -a linie a fișierului va conține înălțimea celei de-a i -a cutii.

Date de ieșire

Fișierul de ieșire **travers.out** va conține o singură linie pe care se va afla un singur număr, reprezentând restul împărțirii numărului de posibilități la 9901.

Restricții și precizări

- $2 \leq N \leq 100000$;
- $1 \leq H \leq 100000000$;
- înălțimile cutiilor sunt numere întregi strict pozitive, cel mult egale cu 100000000;
- pentru 40% dintre seturile de date folosite pentru teste, înălțimile cutiilor vor fi numere întregi cel mult egale cu 100000.

Exemplu

travers.in	travers.out
4 2	4
1	
3	
7	
5	

Explicație

Cele patru posibilități sunt:

```
1 3
1 3 5
3 5
7 5
```

Timp de execuție: 0,5 secunde/test

P060315: Compania

O companie internațională are N angajați, numerotați de la 1 la N . Aceștia sunt organizați într-o structură ierarhică cu următoarele proprietăți:

- fiecare angajat are un superior direct, cu excepția angajatului cu numărul 1 care este managerul companiei;
- fiecare angajat poate fi superiorul direct a 0, 1 sau cel mult 2 alți angajați.

Pentru a îmbunătăți eficiența afacerii lor, deținătorii companiei au decis să o împartă în trei companii mai mici. Pe lângă alte probleme pe care le-au avut de rezolvat a apărut una foarte importantă: cum trebuie distribuiți angajații la cele trei companii nou create? Trebuie luate în considerare următoarele condiții:

- fiecare angajat trebuie distribuit la una dintre cele trei companii;
- un angajat și superiorul său nu trebuie să fie distribuiți la aceeași companie;
- dacă doi angajați au același superior, atunci ei trebuie să fie distribuiți la companii diferite.

Pentru ca cele trei companii să fie la fel de eficiente a mai apărut o condiție suplimentară. Să presupunem că N_1 este numărul de angajați distribuiți la prima companie, N_2 este numărul de angajați distribuiți la cea de-a doua companie și N_3 este numărul de angajați distribuiți la cea de-a treia companie ($N_1 + N_2 + N_3 = N$). Cu aceste notații, trebuie ca diferența $\max(N_1, N_2, N_3) - \min(N_1, N_2, N_3)$ să fie cât mai mică posibil.

Date de intrare

Fișierul de intrare **comp.in** conține pe prima linie un singur număr natural N , care reprezintă numărul de angajați ai companiei.

Fiecare dintre următoarele $N - 1$ linii conține două numere naturale a și b cuprinse între 1 și N , separate printr-un singur spațiu, care semnifică faptul că angajatul b este superiorul direct al angajatului a .

Date de ieșire

Fișierul de ieșire **comp.out** va avea o singură linie pe care se vor afla N numere, separate prin spații, din mulțimea $\{1, 2, 3\}$.

Primul număr reprezintă numărul companiei la care a fost distribuit angajatul cu numărul 1, al doilea număr reprezintă numărul companiei la care a fost distribuit an-

gajatul cu numărul 2, al treilea număr reprezintă numărul companiei la care a fost distribuit angajatul cu numărul 3 etc.

Restricții și precizări

- $0 \leq N \leq 16.000$;
- $\max(a, b, c)$ reprezintă maximum dintre valorile a, b și c ;
- $\min(a, b, c)$ reprezintă minimum dintre valorile a, b și c ;
- este acceptată orice soluție care minimizează diferența specificată în enunț;
- datele de intrare sunt corecte.

Exemplu

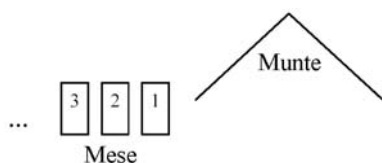
comp.in	comp.out
6	3 1 3 2 1 2
2 1	
3 2	
4 2	
5 3	
6 3	

Timp de execuție: 0,1 secunde/test

P060316: Consiliu de trib

Tribul *Kazooba* are în componența sa N bărbați care vor să se întâlnească pentru un consiliu. Toți bărbații sunt descendenții unei singure persoane care este șeful tribului. De când tribul s-a format, toți strămoșii bărbaților sunt în viață.

Bărbații vin la consiliu într-o anumită ordine și se așează la mese în momentul în care ajung. Bărbații stau într-un rând de mese paralele. Prima masă se află la baza *Muntelui Sfânt* și rândul se extinde nedefinit către vest (mai departe de munte). Din păcate, nici un bărbat nu stă la aceeași masă cu fiii sau tatăl lui. Pe de altă parte, fiecare bărbat dorește să stea cât mai aproape posibil de divinitate, astfel, când un bărbat sosește la consiliu el se așează la masa cea mai apropiată de *Muntele Sfânt*, la care nu s-a așezat nici un fiu și nici tatăl său.



După cum se obișnuiește, zeii tribului au anumite înclinații pentru grandoare, așa că aceștia comandă ca bărbații să vină într-o ordine care să maximizeze numărul de mese ocupate. Din păcate, zeii nu au înclinații spre programare și v-au angajat să scrieți un program pentru ei.

Sarcina voastră este de a scrie un program care să determine numărul maxim de mese care pot fi ocupate după sosirea tuturor bărbaților care fac parte din trib.

Date de intrare

Prima linie a fișierului de intrare **tribe.in** conține un singur număr natural N , care reprezintă numărul bărbați-

lor din tribul *Kazooba*. Fiecare bărbat este numerotat de la 1 la N , șeful tribului având numărul 1.

Fiecare dintre următoarele linii conține două numere naturale A și B cuprinse între 1 și N , separate printr-un singur spațiu, cu semnificația că bărbatul cu numărul de ordine A este tatăl bărbatului cu numărul B .

Date de ieșire

Fișierul de ieșire **tribe.out** va trebui să conțină un singur număr natural, care reprezintă numărul maxim de mese care sunt ocupate după sosirea la consiliu a tuturor bărbaților care fac parte din trib.

Restricții

- $1 < N \leq 100.000$;
- pentru 30% din teste $N \leq 1.000$;
- fiecare masă are capacitate infinită;
- datele de intrare sunt corecte.

Exemplu

tribe.in	tribe.out
5	3
1 4	
3 2	
1 3	
3 5	

Explicație

Dacă bărbații vin în ordinea 2 4 1 5 3, atunci:

- bărbatul cu numărul 2 se așează la prima masă;
- bărbatul cu numărul 4 se așează la prima masă;
- bărbatul cu numărul 1 are un fiu (4) la prima masă, deci acesta se așează la cea de-a doua masă;
- bărbatul cu numărul 5 se așează la prima masă;
- bărbatul cu numărul 3 are doi fii (2 și 5) la prima masă și a doua masă, deci acesta se așează la cea de-a treia masă.

Timp de execuție: 0,5 secunde/test

P060317: Euro

Tocmai v-ați deschis un cont la o bancă de unde veți primi banii în *euro*. Când v-ați deschis contul nu aveți nici *euro* în cont și nici *lei* în posesie. În fiecare dintre următoarele N zile vei primi sume în *euro*; acestea pot fi și negative, caz în care suma totală din contul tău va scădea. Este, de asemenea, posibil ca la un moment dat suma din cont să fie negativă.

La sfârșitul fiecărei zile, banca vă permite să transformați toată suma din cont, din *euro* în *lei*. Cursul de schimb se modifică zilnic după următoarea regulă: în timpul celei de-a K -a zi, un *euro* poate fi schimbat pentru K *lei*. Banca ia de asemenea T *lei* pentru fiecare conversie. Astfel, dacă la sfârșitul zilei aveți S *euro* în cont și vă decideți să schimbați toată suma în *lei* atunci veți primi $S \cdot K - T$ *lei* (S poate fi și negativ). La sfârșitul celei de-a N -a zi, trebuie să schimbați toată suma din *euro* în *lei*, chiar dacă este negativă.





Scopul vostru este acela de a maximiza suma totală în lei pe care o aveți la sfârșitul perioadei de N zile.

Date de intrare

Prima linie a fișierului de intrare **euro.in** conține două numere întregi N și T , separate printr-un singur spațiu, care reprezintă numărul de zile, respectiv suma pe care banca o ia la fiecare conversie.

Următoarea linie conține N numere întregi, separate prin spații, care reprezintă sumele în *euro* pe care le primiți la începutul fiecărei zile.

Date de ieșire

Pe prima linie a fișierului de ieșire **euro.out** va trebui să scrieți un singur număr întreg, care reprezintă suma maximă în lei pe care o puteți obține.

Restricții și precizări

- $1 \leq N \leq 34.567$;
- $1 \leq T \leq 34.567$;
- pentru 80% din teste, T va fi mai mic sau egal cu 255;
- suma în *euro* pe care o veți primi în fiecare zi este un număr întreg cuprins între -1.000 și 1.000;
- puteți converti doar *euro* în lei, nu și invers;
- pentru rezultat se recomandă folosirea tipurilor `int64` sau `extended` în limbajul *Pascal* și `long long` sau `double` în limbajul *C/C++* (în cazul în care efectuați calculele folosind numere întregi sau reale).

Exemplu

euro.in	euro.out
7 1	17
-10 3 -2 4 -6 2 3	

Explicație

La sfârșitul primei zile schimbați -10 *euro* în lei și veți primi -11 = -10 · 1 - 1 lei.

A doua conversie are loc la sfârșitul celei de cincia zi, când aveți -1 *euro* în cont (3 - 2 + 4 - 6) și veți primi -6 = -1 · 5 - 1 lei.

Ultima conversie are loc la sfârșitul celei de șaptea zi când aveți 5 *euro* în cont (2 + 3) și veți primi pentru aceștia 34 = 5 · 7 - 1 lei.

Suma finală pe care o aveți în lei este 17 = -11 - 6 + 34.

TimP de execuție: 0,4 secunde/test

P060318: Nava spațială

Asociația Spațială Română (Romanian Space Association - RSA) a descoperit un asteroid nou pe care l-a numit BOI2003. RSA este nerăbdătoare să exploreze asteroidul și, în acest scop a construit o navă spațială care să-i transporte pe cercetători pe acesta. Nava spațială are câteva caracteristici neobișnuite. În momentul în care nava spațială părăsește pământul pentru prima dată ar trebui să transporte exact S oameni de știință. Când nava spațială ajunge

pe asteroidul BOI2003, un singur cercetător aterizează și nava se întoarce pentru a mai aduce S cercetători.

Primul cercetător care a ajuns pe BOI2003, suspectează existența unui virus periculos pe acesta. Datorită acestui fapt, el sugerează ca nici un cercetător care a călătorit pe BOI2003 să nu părăsească nava spațială când ajunge pe pământ, până după terminarea cercetărilor.

Pentru a fi mai exacți, să presupunem că cercetătorii sunt numerotați începând cu 1. Nava spațială funcționează după cum urmează:

- prima dată când părăsește pământul, nava transportă cercetătorii 1, 2, ..., S ; unul dintre aceștia coboară pe asteroid, iar ceilalți $S - 1$ se întorc pe pământ, fără însă a părăsi nava;
- a doua oară părăsește pământul, după ce cercetătorii $S + 1$, $S + 2$, ..., $2 \cdot S$ se alătură celor $S - 1$ cercetători care s-au întors, deci nava transportă $2 \cdot S - 1$ cercetători; unul dintre aceștia coboară pe asteroid, iar ceilalți $2 \cdot S - 2$ se întorc pe pământ fără însă a părăsi nava;
- a treia oară, cercetătorii $2 \cdot S + 1$, $2 \cdot S + 2$, ..., $3 \cdot S$ împreună cu cei $2 \cdot S - 2$ cercetători care s-au întors sunt transportați la asteroid și, din nou, doar unul dintre aceștia coboară, iar ceilalți $3 \cdot S - 3$ se întorc pe pământ fără însă a părăsi nava;
- ș.a.m.d

După N transporturi, o echipă formată din N cercetători care au fost transportați se află pe asteroid și îl cercetează.

Sarcina voastră este de a scrie un program care să determine numărul total de posibilități de a forma echipa de cercetători.

Date de intrare

Fișierul de intrare **ship.in** conține o singură linie pe care se află două numere întregi S și N , separate printr-un singur spațiu, care reprezintă capacitatea navei spațiale, respectiv numărul de transporturi care se vor efectua.

Date de ieșire

Fișierul de ieșire **ship.out** va conține un singur număr întreg K , care reprezintă numărul total de posibilități de a forma echipa de cercetători.

Restricții și precizări

- $1 \leq S \leq 10$;
- $1 \leq N \leq 40$;
- capacitatea navei spațiale este nelimitată;
- ordinea în care membrii echipei ajung la asteroid nu este importantă.

Exemplu

ship.in	ship.out
2 3	14

TimP de execuție: 0,4 secunde/test