



CONCURSUL INTERJUDEȚEAN DE MATEMATICĂ ȘI INFORMATICĂ
„MARIAN ȚARINĂ”

Ediția a XII-a, 11– 12 MAI 2012

Clasa a IX-a

Problema 1 – NUMERE

Se dau două șiruri de numere naturale. Să se determine câte elemente ale primului șir se pot exprima ca sumă de unul sau mai multe elemente succesive ale celui de-al doilea șir.

Date de intrare

Fișierul de intrare **numere.in** conține pe prima linie, separate printr-un spațiu, două numere naturale n și m reprezentând, respectiv, numărul de elemente ale primului șir și numărul de elemente ale celui de-al doilea șir. Pe cea de-a doua linie a fișierului se găsesc, separate prin câte un spațiu, cele n elemente ale primului șir, iar pe cea de-a treia linie a fișierului, separate prin câte un spațiu, cele m elemente ale celui de-al doilea șir.

Date de ieșire

Fișierul de ieșire **numere.out** va conține o valoare reprezentând numărul de elemente ale primului șir care se pot exprima ca sumă de elemente succesive ale celui de-al doilea șir.

Restricții și precizări

- $1 \leq n \leq 1000$
- $1 \leq m \leq 1000$
- Toate elementele din cele două șiruri sunt numere naturale mai mici decât 1000000.
- Orice sumă ce exprimă un număr al primului șir trebuie să aibă cel puțin un termen, element al celui de-al doilea șir.

Exemplu

numere.in	numere.out	Explicație
4 5	3	$2 = 2$
2 4 1 8		$4 = 2 + 2$
2 2 6 0 2		$8 = 2 + 6$

Timp maxim de execuție/test: 0.1 secunde.

Total memorie disponibilă: 2 MB.

Dimensiunea maximă a sursei: 5 KB.



CONCURSUL INTERJUDEȚEAN DE MATEMATICĂ ȘI INFORMATICĂ
„MARIAN ȚARINĂ”

Ediția a XII-a, 11– 12 MAI 2012

Clasa a IX-a

Problema 2 – TRIUNGHI

Considerăm o coală de hârtie pe care avem desenată o grilă formată din triunghiuri echilaterale identice. O astfel de grilă este compusă din trei familii de linii drepte; în cadrul fiecărei familii, liniile sunt paralele echidistante, iar liniile din familii diferite se intersectează în unghiuri de 60 de grade. Mai presupunem că prima familie este formată din linii orizontale; a doua va fi formată din linii orientate din stânga-jos spre dreapta-sus, la 60 de grade față de orizontală și 30 de grade față de verticală, iar liniile celei de-a treia familii sunt orientate dinspre dreapta-jos spre stânga-sus.

În cadrul primei și celei de-a doua familii, numerotăm liniile cu numere întregi, de jos în sus și, respectiv, dinspre stânga-sus spre dreapta-jos. Astfel, fiecare vârf al vreunui triunghi din grilă poate fi identificat unic printr-o pereche de numere întregi, reprezentând numărul liniei din prima familie și numărul liniei din a doua familie ce se intersectează în acel punct.

Dându-se două vârfuri distincte ale grilei, se cere să se determine câte triunghiuri sunt tăiate de segmentul de dreaptă determinat de cele două puncte. Un triunghi se consideră tăiat de segment doar dacă segmentul intersectează interiorul triunghiului; dacă segmentul doar trece printr-un vârf sau de-a lungul unei laturi, nu considerăm că ar tăia triunghiul.

Date de intrare

Datele se citesc din fișierul `triunghi.in` având, pe o singură linie, patru numere întregi. Primele două numere descriu primul punct, fiind, respectiv, numărul liniei din prima familie (orizontală) și numărul liniei din a doua familie (oblică stânga-jos spre dreapta-sus) la intersecția cărora se găsește acesta. Următoarele două numere descriu, în același mod, cel de-al doilea punct.

Date de ieșire

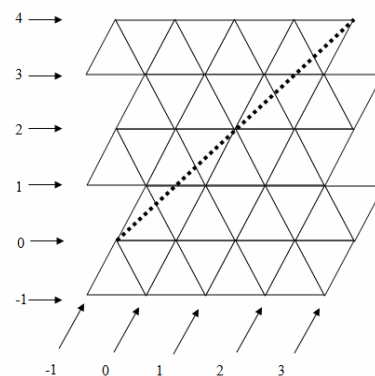
În fișierul `triunghi.out` se va scrie numărul de triunghiuri tăiate de segment.

Restricții

- Numerele date la intrare sunt întregi cuprinse între -500000000 și 500000000 .

Exemplu 1

<code>triunghi.in</code>	<code>triunghi.out</code>
0 -1 4 1	8



Exemplu 2

```
triunghi.in      triunghi.out
0 1 1 0         0
```

Timp maxim de execuție/test: 0.1 secunde.

Total memorie disponibilă: 5 MB.

Dimensiunea maximă a sursei: 5 KB.



CONCURSUL INTERJUDEȚEAN DE MATEMATICĂ ȘI INFORMATICĂ
„MARIAN ȚARINĂ”

Ediția a XII-a, 11– 12 MAI 2012

Clasa a X-a

Problema 1 – CUVINTE

Ionică și Vasilică, prieteni buni, caută mereu să-și testeze îndemânarea și logica. Cei doi se găsesc în fața aceluiași calculator și tastează, cât pot de repede, câte un cuvânt. Textul rezultat în urma tastării literelor din cele două cuvinte va avea literele cuvintelor amestecate între ele, deși apar în aceeași ordine ca în cuvintele inițiale. Mai mult, grăbindu-se, cei doi băieți pot apăsa la un moment dat o aceeași tastă, acțiune care va avea ca rezultat apariția unei singure litere în șirul final tastat.

Cunoscându-se cele două cuvinte pe care cei doi băieți le tastează și textul rezultat, să se determine numărul de variante distincte în care băieții au apăsat tastele și o astfel de variantă.

Date de intrare

Fișierul de intrare **cuvinte.in** conține pe prima linie cuvântul pe care l-a tastat Ionică, pe cea de-a doua linie cuvântul pe care l-a tastat Vasilică, iar pe cea de-a treia linie textul rezultat.

Date de ieșire

Fișierul de ieșire **cuvinte.out** va conține pe prima linie numărul de variante în care băieții au apăsat tastele, iar pe cea de-a doua linie o astfel de variantă.

Restricții și precizări

- Lungimea fiecărui șir din fișierul de intrare este cel mult 200.
- Șirurile conțin doar litere mici ale alfabetului englez.
- O variantă de apăsare a tastelor va fi descrisă printr-un șir de valori 1, 2 sau 3, fără spații între ele, reprezentând persoana care a apăsat tasta la momentul corespunzător poziției din șir (1 – Ionică, 2 – Vasilică, 3 – tasta a fost apăsată în același timp de ambii băieți).
- Numărul total de variante este cel mult 1000000000.
- Pentru rezolvarea corectă doar a primei cerințe se acordă 60% din punctaj. Pentru rezolvarea corectă doar a celei de-a doua cerințe se acordă 40% din punctaj.

Exemplu

cuvinte.in	cuvinte.out
abbc	8
abcb	1213122
aabbccb	

Timp maxim de execuție/test: 0.2 secunde.
Total memorie disponibilă: 32 MB.
Dimensiunea maximă a sursei: 5 KB.



CONCURSUL INTERJUDEȚEAN DE MATEMATICĂ ȘI INFORMATICĂ
„MARIAN ȚARINĂ”

Ediția a XII-a, 11– 12 MAI 2012

Clasa a X-a

Problema 2 – PERECHI

Lui Gigel îi plac numerele care au doua sau mai multe cifre consecutive egale. De exemplu, îi plac numerele 11, 22, 100, 110, 111. El dorește acum să afle câte numere naturale, formate dintr-un număr dat de cifre, au cel puțin două cifre consecutive identice. De asemenea, el dorește să poată face acest lucru și pentru numerele scrise în altă bază decât 10. Scrierea numerelor se face intotdeauna fără zerouri în față (prima cifră a numărului nu poate fi 0).

Dându-se două numere naturale N , și B , se cere să se afle cate numere naturale au scrierea în baza B formată din exact N cifre și au cel puțin 2 cifre consecutive identice.

Date de intrare

Fișierul **perechi.in** contine, pe o singură linie, cele două numere naturale, N și B , ambele reprezentate în baza 10 și separate printr-un spațiu.

Date de ieșire

În fișierul **perechi.out** se va scrie, pe o singură linie, numărul de numere ce conțin cel puțin o pereche de cifre consecutive identice, modulo 2012.

Restricții

- $1 \leq N \leq 1000$.
- $1 \leq B \leq 1000$.

Exemplu 1

perechi.in	perechi.out	Explicație
2 10	9	Numerele căutate sunt: 11, 22, 33, 44, 55, 66, 77, 88, 99.

Exemplu 2

perechi.in	perechi.out	Explicație
3 3	10	Numerele căutate, reprezentate în baza 3, sunt: 100, 110, 111, 112, 122, 200, 211, 220, 221, 222

Timp maxim de execuție/test: 1 secundă.

Total memorie disponibilă: 5 MB.

Dimensiunea maximă a sursei: 5 KB.



CONCURSUL INTERJUDEȚEAN DE MATEMATICĂ ȘI INFORMATICĂ

„MARIAN ȚARINĂ”

Ediția a XII-a, 11– 12 MAI 2012

Clasele XI-XII

Problema 1 – DISCURI

Ionuț a găsit în garajul tatălui său, mecanic de profesie, mai multe discuri cu dinți, toate identice între ele.

Băiatul vrea să facă un mic experiment cu aceste discuri. Astfel, el conectează mai multe perechi de discuri între ele. În acest mod obține un sistem de discuri interconectate în care rotirea unui disc într-un anumit sens determină rotirea tuturor discurilor direct conectate la acest disc în sens opus (vezi imaginea alăturată). În final, atașează fiecărui disc din sistem câte un motorăș capabil să producă discului căruia îi este atașat într-o secundă o rotație completă în sensul acelor de ceasornic.

Pentru a verifica sistemul construit, Ionuț își cheamă câțiva prieteni și îi roagă să acționeze motorășele. Evident, prietenii lui, încântați de sistemul construit, acționează motorășele fără a aștepta oprirea completă a celor anterior acționate, ceea ce poate duce la blocarea întregului sistem de discuri.



Cunoscându-se structura sistemului de discuri, să se determine:

- dacă sistemul este funcțional, adică dacă toate discurile pot fi rotite. Dacă cel puțin un disc nu se poate roti, întrucât rotirea lui într-un sens ar provoca, direct sau indirect, rotirea unui același disc simultan în sensuri contrare, sistemul este nefuncțional;
- numărul maxim de discuri care pot fi rotite în sensul acelor de ceasornic prin acționarea unui singur motorăș din sistem;
- cunoscând, în plus, intervalele de timp în care este acționat fiecare motorăș de către prietenii băiatului, se cere să se determine primul moment de timp în care sistemul se blochează. Sistemul se blochează dacă există cel puțin un disc care este forțat, direct sau indirect, de către motorășe în funcțiune în acel moment, să se rotească simultan în ambele sensuri.

Date de intrare

Fișierul de intrare **discuri.in** conține pe prima linie, separate prin câte un spațiu, trei numere naturale n , m și p reprezentând, respectiv, numărul de discuri, numărul de perechi de discuri direct conectate și numărul de acționări ale motorășelor. Pe următoarele m linii se găsesc câte două numere naturale, separate între ele prin câte un spațiu, reprezentând numerele de ordine ale două discuri direct conectate. Ultimele p linii conțin, separate între ele prin câte un spațiu, câte trei numere întregi d i f reprezentând, corespunzător, numărul de ordine al motorășului acționat, momentul în care motorășul este acționat și momentul în care acesta este oprit. La momentul inițial, se consideră că toate motorășele sunt oprite.

Date de ieșire

Fișierul de ieșire **discuri.out** va conține pe prima linie numărul 1 sau 0 după cum sistemul este funcțional sau nu. Pe cea de-a doua linie a fișierului se va scrie o valoare reprezentând numărul

maxim de discuri care se rotesc în sensul acelor de ceasornic la acționarea unui motorăș oarecare sau 0 dacă orice motorăș acționat duce la blocarea sistemului. Pe cea de-a treia linie, se va scrie o valoare reprezentând primul moment de timp în care sistemul se blochează sau valoarea -1 dacă sistemul nu se blochează prin acționarea motorășelor precizate.

Restricții și precizări

- Fiecare motorăș are numărul de ordine identic cu cel al discului pe care îl acționează.
- $3 \leq n \leq 1000$
- $1 \leq m \leq 10000$
- $2 \leq p \leq 1000$
- $1 \leq d \leq n$
- $0 \leq i < f \leq 10000$
- Oprirea unui motorăș determină oprirea concomitentă din rotație a tuturor discurilor din sistem care sunt legate direct sau indirect de discul corespunzător motorășului respectiv, exceptând situația în care unul sau mai multe alte motorășe sunt încă în funcțiune.
- Dacă un motorăș este pornit concomitent cu oprirea altui motorăș, nu se consideră că aceste două motorășe ar fi simultan în funcțiune (ca atare, chiar dacă acestea ar determina un același disc să se rotească în sensuri opuse, nu se consideră că sistemul se blochează).
- Mai multe discuri care determină rotirea unui disc în același sens nu modifică viteza de rotație a acestuia.
- Se garantează că acționarea unui motorăș nu se face decât dacă acesta este oprit.
- Pentru rezolvarea corectă doar a primei cerințe se acordă 40% din punctaj, pentru rezolvarea corectă doar a celei de-a doua cerințe se acordă 30% din punctaj, iar pentru rezolvarea corectă doar a celei de-a treia cerințe se acordă 30% din punctaj.

Exemple

discuri.in	discuri.out	Explicație
5 5 3	1	Sistemul este funcțional.
2 3	3	Prin acționarea motorășului cu numărul de ordine 1 vor fi 3 discuri (1, 2 și 4) care se rotesc în sensul acelor de ceasornic.
1 5	-1	Sistemul nu se blochează.
1 3		
4 3		
4 5		
2 1 6		
3 9 10		
4 4 7		

discuri.in	discuri.out	Explicație
4 3 3	1	Sistemul este funcțional.
2 3	3	Prin acționarea motorășului cu numărul de ordine 1 vor fi 3 discuri care se rotesc în sensul acelor de ceasornic.
1 3	3	Sistemul se blochează la momentul 3.
4 3		
2 3 6		
3 2 4		
4 4 7		

Timp maxim de execuție/test: 0.5 secunde.

Total memorie disponibilă: 5 MB.

Dimensiunea maximă a sursei: 5 KB.



CONCURSUL INTERJUDEȚEAN DE MATEMATICĂ ȘI INFORMATICĂ
„MARIAN ȚARINĂ”

Ediția a XII-a, 11– 12 MAI 2012

Clasele XI-XII

Problema 2 – SALTURI

Prințul Persiei este un joc pe calculator, aflat în vogă acum 20 de ani. Personajul jocului trebuie să se deplaseze prin culoarele unui castel, să evite diverse capcane și să deschidă porțile prin care intenționează să treacă. Adesea, el trebuie să parcurgă un culoar în timp limitat, înainte ca o poartă aflată la capătul culoarului să se închidă. Dacă culoarul conține capcane, el trebuie să sară peste acestea. În jocul original, personajul poate să fugă, să sară de pe loc, sau să sară din fugă. Saltul din fugă este mai rapid (mai rapid chiar decât fuga pe aceeași distanță) și mai lung (putând astfel evita capcane mai întinse), dar are nevoie de mai mult spațiu pentru elan și pentru oprire, comparativ cu saltul de pe loc. Ca atare, pe un culoar dat, un jucător bun trebuie să găsească secvența optimă de salturi și de porțiuni parcurse în fugă pentru traversarea cât mai rapidă a acesteia.

Culoarul de traversat este compus din N porțiuni de lungime 1; fiecare astfel de porțiune poate fi fie culoar normal, fie o capcană. În problema noastră, generalizăm puțin jocul, permițând un număr oarecare de tipuri de salturi (precum și deplasarea în fugă). Despre fiecare tip de salt, se cunoaște lungimea necesară pentru elan, lungimea saltului propriu-zis și lungimea necesară pentru oprire, precum și timpul necesar efectuării lui. Porțiunile utilizate pentru elan și pentru oprire trebuie să fie fără capcane; porțiunile peste care se sare pot fi oricum. Lungimea saltului este exact cea specificată; nu poate fi micșorată. După parcurgerea porțiunii de oprire a unui salt se poate începe imediat elanul saltului următor sau personajul poate fugi mai întâi o anumită distanță. Porțiunile parcurse în fugă pot fi oricât de lungi, dar nu au voie să conțină capcane. La finalul culoarului, personajul jocului nu are voie să fie în cursul unui salt, nici măcar în cadrul elanului sau opririi.

Dându-se configurația culoarului și caracteristicile salturilor posibile, se cere să se determine timpul minim necesar parcurgerii culoarului, precum și o secvență de salturi și porțiuni parcurse în fugă prin care se obține acest minim.

Date de intrare

Datele se citesc din fișierul **salturi.in** având următorul format:

- pe prima linie, trei numere naturale nenule, L , N și T_0 , separate prin spații, reprezentând lungimea culoarului, numărul de tipuri de salturi și timpul necesar parcurgerii în fugă a unei porțiuni de culoar de lungime 1;
- pe a doua linie, L numere 0 sau 1 separate prin spații, reprezentând configurația culoarului; 0 reprezintă culoar liber, iar 1 reprezintă o capcană;
- pe fiecare din următoarele N linii, câte patru numere naturale nenule, reprezentând lungimea elanului, lungimea saltului propriu-zis, lungimea opririi și durata totală a saltului.

Date de ieșire

Rezultatele se vor scrie în fișierul **salturi.out** în următorul format:

- pe prima linie se vor scrie două numere naturale, T și M , reprezentând timpul minim necesar și suma dintre numărul de salturi și numărul de porțiuni de lungime 1 parcurse în fugă
- pe a doua linie se vor scrie M numere, fiecare cuprins între 0 și N inclusiv, reprezentând tipul saltului, valoarea 0 indicând parcurgerea în fugă a unei porțiuni de lungime 1.

Restricții și precizări

- $1 \leq L \leq 1000$.
- $1 \leq N \leq 1000$.
- Se garantează existența unei soluții.

Exemplu

salturi.in

```
15 2 1
0 0 0 1 0 1 1 0 0 0 0 0 0 0 0
0 2 1 6
3 3 1 5
```

salturi.out

```
19 5
0 0 1 1 2
```

Timp maxim de execuție/test: 1 secundă.

Total memorie disponibilă: 5 MB.

Dimensiunea maximă a sursei: 5 KB.