



Etapa FINALĂ

A doua ediție a concursului de programare organizat de Gazeta de Informatică s-a încheiat. Primii zece în clasamentul după cele șase runde desfășurate prin e-mail au participat la faza finală care a avut loc pe data de 26 mai 2001 la Liceul de Informatică din Cluj.

Cei zece participanți calificați la faza finală au fost:

Csaba András, Oradea
Mugurel-Ionuț Andreica, București
Ciprian Cană, Vaslui
Adrian Cârțu, Bistrița
George Drumea, București
Liviu Lalescu, Craiova
Maximilian Machedon, București
Cosmin-Silvestru Negrușari, Bistrița
Mihai Stroe, București
Radu Ștefan, Brașov

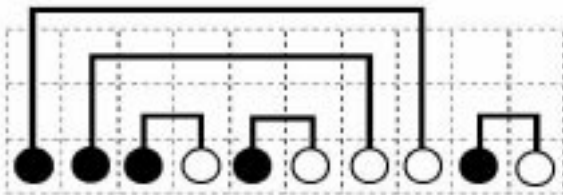
Singurul care nu a putut să se prezinte la faza finală a fost *Csaba András*, așadar am avut doar nouă participanți.

Până la urmă, bursa de 50\$/lună oferită de *Computer Press Agora* a fost câștigată de *Mugurel-Ionuț Andreica*. El a fost singurul care a reușit să obțină punctajul maxim la toate cele cinci probleme propuse spre rezolvare. Pe locul al doilea s-a clasat *Mihai Stroe*, care a câștigat o bursă de 30\$/lună.

În continuare vă prezentăm enunțurile celor cinci probleme; vă propunem să încercați să le rezolvați. Soluțiile le veți găsi în numărul următor al revistei noastre.

CPAR7P1: Bile

Se consideră n bile negre și n bile albe înșirate pe bară dreaptă. Aceste bile urmează să fie legate între ele două câte două prin alte bare care nu pot să se intersecteze și să se suprapună. În acest scop se vor folosi, în cazul fiecărei perechi, trei bucăți de bare: două verticale și una orizontală, conform figurii. O pereche este formată dintr-o bilă albă și una neagră, sau dintr-una neagră și una albă.



Date de intrare

Fișier de intrare: BILE.IN

Linia 1: n

- număr natural nenul, reprezentând numărul bilelor albe (și care este totodată și numărul bilelor negre);

Linia 2: s

- succesiune de cifre egale cu 0 și 1, neseparate prin spații, reprezentând bilele fixate pe bară; valoarea 0 corespunde bilelor albe, iar valoarea 1 corespunde bilelor negre.

Date de ieșire

Fișier de ieșire: BILE.OUT

Linia 1: L

- număr natural nenul, reprezentând lungimea minimă totală a barelor folosite;
- Liniile 2.. $n+1$: $x \ y$
- aceste n linii conțin fiecare, două numere naturale nenule, separate printr-un spațiu, reprezentând numărul de ordine a două bile legate în pereche; perechile care se vor scrie în fișier vor fi ordonate crescător în funcție de valoarea x .

Restricții

- $1 \leq n \leq 100$

Exemplu

BILE.IN	BILE.OUT
10	31
1110100010	1 8
	2 7
	3 4
	5 6
	9 10

Timp maxim de execuție/test: 1 secundă

CPAR7P2: Plăți și încasări

Casierul unei firme trebuie să parcurgă zilnic un traseu liniar prin cele N magazine ale sale. Magazinele sunt numerotate de la 1 la N și se parcurg zilnic în ordinea numerelor

lor de ordine, începând cu primul magazin și terminând cu al N -lea. (Evident casierul va trece doar o singură dată prin fiecare magazin.)

Scopul parcurgerii acestui traseu este efectuarea unor plăți, respectiv a unor încasări. Se știe că există cel puțin un magazin de la care casierul trebuie să încaseze bani. De asemenea se cunosc sumele care trebuie plătite, respectiv încasate.

Într-o zi, patronul nu are bani pentru plăți și îi cere casierului să aleagă două magazine m_i și m_j de pe traseu, astfel încât, parcurgând traseul începând cu m_i și încheindu-l cu m_j , încasările să reprezinte suma maximă posibilă de adus înapoi la firmă.

Casierul trebuie să intre în toate magazinele aflate pe traseul ales, chiar dacă în unele trebuie să efectueze plăți. În plus, pentru a nu crea suspiciuni printre angajații magazinelor, patronul dorește să fie vizitate cât mai multe magazine.

Cerința

Scrieți un program care determină suma maximă posibilă de încasat din magazinele firmei. De asemenea, determinați secvența cea mai lungă de magazine care permite obținerea acestei sume. Dacă aceeași sumă maximă se poate obține pentru mai multe alegeri diferite, vizitând același număr maxim de magazine, atunci în fișierul de ieșire se va scrie una singură, și anume cea care începe de la magazinul cu numărul de ordine cel mai mic.

Date de intrare

Fișier de intrare: CASIER.IN

Linia 1: N

- număr natural nenul, reprezentând numărul magazinelor de pe traseul inițial.

Linia 2: $Suma_1 \quad Suma_2 \quad \dots \quad Suma_N$

- N numere întregi, separate prin câte un spațiu, reprezentând sumele pe care casierul ar trebui să le plătească sau să le încaseze în/din cele N magazine; numerele pozitive reprezintă sume care se vor încasa, iar numerele negative corespund sumelor care se vor plăti.

Date de ieșire

Fișier de ieșire: CASIER.OUT

Linia 1: $SumaMax$

- număr natural nenul, reprezentând suma maximă cu care casierul se întoarce la firmă.

Linia 2: $i \quad j$

- două numere naturale nenule, care aparțin intervalului $[1, N]$ și care reprezintă numerele de ordine ale magazinelor alese de casier pentru începutul și sfârșitul porțiunii de traseu ($i \leq j$).

Restricții

- $1 \leq N \leq 1000$
- $-1000000 \leq Suma_i \leq 1000000, i=1, 2, \dots, N$
- $1 \leq SumaMax \leq 1000000000$

Exemplu

CASIER.IN

```
10
2 3 -6 5 -6 6 7 -2 2 -1
```

CASIER.OUT

```
13
6 9
```

Timp maxim de execuție/test: 1 secundă

CPAR7P3: Robot

Un robot se poate mișca în plan doar pe direcțiile verticală, notate cu *up* ('U') și *down* ('D') respectiv orizontală, notate cu *left* ('L') și *right* ('R'). Robotul pornește dintr-un anumit punct și după ce își îndeplinește sarcinile, se întoarce tot în acest punct. Mișcarea robotului se descrie printr-o secvență de comenzi, conform căreia acesta se va deplasa de-a lungul unor linii care formează un caroi în care pătrățelele au laturile egale cu 1.

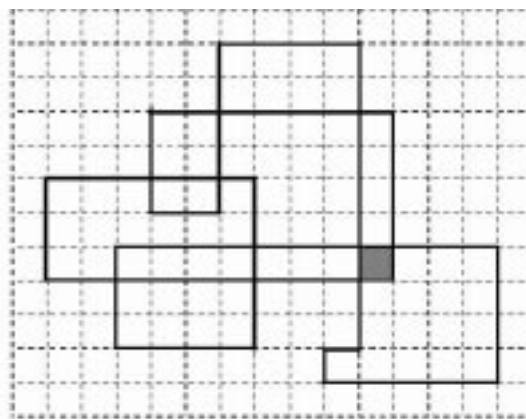
Robotul, în timp ce își parcurge drumul, trasează o linie în urma lui. Datorită faptului că traseul se sfârșește în același punct din care a pornit, se pot forma figuri geometrice, printre care și dreptunghiuri.

Vom numi *dreptunghi vid* acel dreptunghi care se formează din segmente aparținând traseului și care nu este intersectat de alte segmente din traseu.

Cerință

Determinați cel mai mic dreptunghi vid care se formează din traseul robotului.

În imaginea următoare este prezentat un posibil traseu al robotului. Cel mai mic dreptunghi vid care se formează este hașurat.



Date de intrare

Fișier de intrare: ROBOT.IN

Linia 1: $x \quad y$

- două numere naturale, separate printr-un spațiu, reprezentând coordonatele punctului de pornire;

Linia 2: c

- număr natural nenul, reprezentând numărul de comenzi pe care le execută robotul;

Linii 3..c+2: caracter număr

- unde *caracter* este una din literele 'U', 'D', 'L', 'R', având semnificația descrisă mai sus.



**Date de ieșire****Fișier de ieșire:** ROBOT . OUT**Linia 1:** c_1 c_2

- două numere naturale, separate printr-un spațiu, reprezentând coordonatele colțului stânga jos, al dreptunghiului vid având aria minimă.

Linia 2: c_3 c_4

- două numere naturale, separate printr-un spațiu, reprezentând coordonatele colțului dreapta sus, al dreptunghiului vid având aria minimă.

Restricție

- coordonatele în care se va afla robotul vor fi întotdeauna în intervalul $[0, 100]$.

Exemplu

ROBOT . IN	ROBOT . OUT
4 7	7 3
8	10 5
R 6	
D 4	
L 3	
U 2	
R 5	
D 4	
L 8	
U 6	

Timp maxim de execuție/test: 1 secundă**CPAR7P4: Star Trek**

N membri ai echipajului navei stelare *USS Enterprise* sunt teleportați pe o planetă în poziții diferite pentru a cerceta suprafața (considerată plană) a planetei.

Fiecare membru posedă un tricorder cu ajutorul căruia poate primi informații referitoare la zona în care a fost teleportat.

Raza de acțiune a tricorder-ului poate fi stabilită de fiecare membru. Suprafața cercetată de fiecare membru este deci un disc cu centrul în poziția în care este teleportat și cu raza egală cu raza de acțiune a tricorder-ului său.

Pentru ca fiecare membru să muncească la fel de mult, *Cpt. Jean Luc Picard* decide ca fiecare membru să aibă aceeași rază de acțiune.

Pentru a nu apărea interferențe, nu este permisă intersecția între suprafețele studiate de două tricorder-e diferite. În plus, raza de acțiune a tricorder-elor este obligatoriu un număr natural.

Cerință

Stabiliți raza maximă de acțiune a tricorder-elor, astfel încât zonele în care acționează oricare două dintre ele să nu aibă nici un punct comun.

Date de intrare**Fișier de intrare:** STARTREK . IN**Linia 1:** N

- număr natural nenul, reprezentând numărul membrilor echipajului;

Liniiile 2..N+1: x_i y_i

- pe aceste N linii se află câte două numere naturale nenule, separate printr-un spațiu, reprezentând coordonatele la care sunt teleportați membrii echipajului.

Date de ieșire**Fișier de ieșire:** STARTREK . OUT**Linia 1:** $n \pi$

- număr natural, reprezentând raza maximă de acțiune a tricorder-elor.

Restricții

- $2 \leq N \leq 10000$
- $0 \leq x_i, y_i \leq 20000000000$ $i=1, 2, \dots, N$
- Nu pot exista doi membri ai echipajului care să fie teleportați în aceeași poziție.

Exemplu

STARTREK . IN	STARTREK . OUT
4	2
0 0	
6 6	
0 6	
6 0	

Timp maxim de execuție/test: 1 secundă**CPAR7P5: Windows XP**

Sistemul de operare *Microsoft Windows XP* va fi lansat oficial pe data de 25 octombrie 2001. După achiziționarea produsului, mulți utilizatori vor dori să instaleze noul sistem de operare "pe curat".

Ei vor formata discul hard, vor instala noul sistem de operare, apoi vor dori să instaleze aplicațiile preferate. După cum știți, instalarea unei aplicații necesită, uneori, un anumit număr de resetări ale sistemului. Sarcina voastră este de a-i ajuta pe utilizatori să instaleze toate aplicațiile, resetând calculatorul de cât mai puține ori.

Pentru fiecare program de instalare este cunoscut numărul x de resetări necesare, precum și lista aplicațiilor care trebuie să fie deja instalate pentru ca acesta să poată fi lansat în execuție.

Un program de instalare funcționează astfel:

- este lansat în execuție;
- se execută primul pas al instalării (eventual, în paralel cu execuția altor programe de instalare);
- se așteaptă finalizarea execuției pasului curent de către toate programele de instalare aflate în execuție;
- se resetează sistemul;
- se trece la al doilea pas, etc.

La un moment dat se pot afla în execuție oricâte programe de instalare, dacă aplicațiile necesare pentru execuția lor au fost deja instalate.



Să presupunem că dorim să instalăm aplicațiile *Microsoft Office XP*, *Microsoft Visual Studio.NET*, *Microsoft DirectX 8.0* și *Microsoft Internet Explorer 6.0*. Pentru a instala *Office* este nevoie de patru resetări, pentru *Visual Studio* este nevoie de trei resetări, pentru *DirectX* două resetări și pentru *Internet Explorer* o resetare. *Internet Explorer* și *DirectX* pot fi instalate fără să fie instalată nici o altă aplicație. *Visual Studio* poate fi instalat numai dacă sunt instalate *DirectX* și *Internet Explorer*, iar *Office* dacă a fost instalat *Internet Explorer*.

La primul pas se vor lansa programele de instalare pentru *Internet Explorer* și *DirectX*. După resetare, *Internet Explorer* este instalat, iar pentru *DirectX* se trece la al doilea pas al instalării.

Datorită faptului că *Internet Explorer* este instalat, se poate lansa în execuție și programul de instalare pentru *Office*. La acest al doilea pas se află în execuție programele de instalare pentru *DirectX* (al doilea pas) și *Office* (primul pas).

După resetare, *DirectX* este instalat, iar pentru *Office* se trece la al doilea pas. Datorită faptului că *DirectX* este instalat, se poate lansa în execuție și programul de instalare pentru *Visual Studio*. La acest al treilea pas se află în execuție programele de instalare pentru *Office* (al doilea pas) și *Visual Studio* (primul pas).

După resetare se vor afla în execuție programele de instalare pentru *Office* (al treilea pas) și *Visual Studio* (al doilea pas). După o nouă resetare se vor afla în execuție programele de instalare pentru *Office* (al patrulea pas) și *Visual Studio* (al treilea pas).

Va avea loc o ultimă resetare, după care și aceste două programe sunt instalate. În total au avut loc cinci resetări.

Date de intrare

Fișier de intrare: WINXP.IN

Linia 1: N

- număr natural nenul, reprezentând numărul aplicațiilor care trebuie instalate;

Liniiile 2..N+1: nr na a₁ a₂ ... a_{na}

- aceste N linii conțin fiecare mai multe numere naturale, separate prin câte un spațiu; nr reprezintă numărul de resetări pentru aplicația respectivă, na reprezintă numărul de aplicații care trebuie deja să fie instalate, iar a₁, a₂, ..., a_{na} reprezintă numerele de ordine ale aplicațiilor care trebuie să fie deja instalate (dacă este cazul).

Date de ieșire

Fișier de ieșire: WINXP.OUT

Linia 1: R

- număr întreg pozitiv, reprezentând numărul total de resetări;

Liniiile 2..R+1: nr₁ nr₂ ...

- aceste R linii conțin fiecare un număr oarecare de numere naturale nenule, reprezentând numerele de ordine ale aplicațiilor a căror programe de instalare se află în execuție la pasul respectiv.

Exemplu

WINXP.IN

```
4
4 1 4
3 2 3 4
2 0
1 0
```

WINXP.OUT

```
5
3 4
1 4
1 2
1 2
1 2
```

Timp maxim de execuție/test: 1 secundă

Despre concurs

Concurenții au considerat că toate cele cinci probleme au fost relativ simple, dovadă fiind faptul că au fost obținute multe punctaje de 100 de puncte la diferite probleme. Totuși, doar câștigătorul concursului le-a rezolvat corect pe toate cinci.

Problema **Bile** a fost rezolvată corect de *Mugurel-Ionuț Andreica*, *Adrian Cârnu* și *Mihai Stroe*.

La problema **Plăți și încasări** au fost obținute punctaje maxime de către *Mugurel-Ionuț Andreica*, *Ciprian Cană*, *Adrian Cârnu*, *Cosmin-Silvestru Negrușeri*, *George Drumea*, *Maximilian Machedon* și *Ștefan Radu*.

A treia problemă, **Robot**, a fost rezolvată corect de doar doi concurenți: *Mugurel-Ionuț Andreica* și *Ștefan Radu*.

Tot doi concurenți, *Mugurel-Ionuț Andreica* și *Mihai Stroe*, au obținut punctaje maxime la problema **Star Trek**.

În fine, problema **Windows XP** a fost rezolvată corect de *Mugurel-Ionuț Andreica*, *Adrian Cârnu*, *Cosmin-Silvestru Negrușeri*, *George Drumea*, *Ștefan Radu* și *Mihai Stroe*.

Programele celor nouă participanți au fost evaluate folosind un calculator dotat cu procesor *Intel Celeron 600*, 56 MB de memorie internă și folosind sistemul de operare *Microsoft Windows '98 Second Edition*. Limitele de timp indicate în revistă sunt valabile pentru acest tip de calculator.

Diplomele primite de concurenți au fost înmânate în prezența domnului **Cristian Nagy**, reprezentatul editurii **Computer Press Agora**.

Toți participanții au primit din partea organizatorilor câte un abonament pe un an la *GInfo*.

Concluzii

Această ediție a **Concursului de Programare Agora** s-a dovedit a fi un real succes, dovadă fiind numărul mare al celor înscriși (peste 400, dintre care aproape 100 din afara granițelor țării).

Îi felicităm pe toți cei care au participat și le mulțumim pentru interesul acordat concursului nostru. Le urăm tuturor mult succes la ediția 2001/2002 a concursului și sperăm că aceasta va fi cel puțin la fel de interesantă și îi va ajuta la fel de mult pe participanți la pregătirea pentru diferite concursuri naționale și internaționale de programare.