



MINISTERUL EDUCAȚIEI, CERCETĂRII ȘI TINERETULUI  
INSPECTORATUL ȘCOLAR JUDEȚEAN CLUJ  
COLEGIUL NAȚIONAL „MIHAI VITEAZUL” TURDA

CONCURSUL INTERJUDEȚEAN DE MATEMATICĂ ȘI INFORMATICĂ  
„MARIAN ȚARINĂ”

EDIȚIA A VIII-A



16 – 17 MAI 2008

## Clasa 9

### Problema 1 – FERESTRE

Pe ecranul unui calculator sunt deschise mai multe ferestre. Se știe, despre fiecare fereastră, poziția ei pe ecran precum și „nivelul” ei (deasupra căror ferestre se află). Ferestrele pot să se acopere parțial.

O fereastră poate fi închisă dacă colțul dreapta-sus al ei este vizibil. Odată închisă o fereastră, ea încetează să mai acopere ferestrele de sub ea. Ordinea sus-jos a ferestrelor nu poate fi modificată.

O parte dintre ferestrele de pe ecran trebuie să rămână deschise. Se cere să se închidă cât mai multe dintre celelalte ferestre.

### Date de intrare

Datele se vor citi din fișierul **ferestre.in** în următorul format:

- pe primul rând, numărul total  $n$  de ferestre și numărul  $m$  de ferestre ce trebuie să rămână deschise
- următoarele  $n$  rânduri descriu ferestrele, de la cea mai de jos (eventual acoperită de toate celelalte) la cea mai de sus (neacoperită de nicio fereastră). Fiecare rând conține câte 4 numere întregi  $x_1$   $y_1$   $x_2$   $y_2$  reprezentând coordonatele colțurilor ferestrei;  $x_1 < x_2$ ,  $y_1 < y_2$ ; colțul dreapta-sus are coordonatele  $(x_2, y_1)$ .
- pe următoarele  $m$  rânduri, câte un număr natural reprezentând numărul de ordine al câte unei ferestre ce trebuie lăsată deschisă; numerotarea ferestrelor începe de la 1 de la fereastra cea mai de jos.

### Date de ieșire:

În fișierul **ferestre.out** se va scrie:

- pe primul rând, numărul maxim  $k$  de ferestre ce pot fi închise
- pe următoarele  $k$  rânduri, numerele de ordine ale ferestrelor ce pot fi închise, în ordinea în care vor fi închise

### Exemple:

**ferestre.in**

```
4 1
0 1 2 5
0 0 3 4
4 2 5 5
1 1 5 3
2
```

**ferestre.out**

```
2
4
3
```

### Restricții și precizări:

- $n \leq 200$
- coordonatele sunt numere întregi între -30000 și 30000

**Timp maxim de execuție/test:** 1 secundă



MINISTERUL EDUCAȚIEI, CERCETĂRII ȘI TINERETULUI  
INSPECTORATUL ȘCOLAR JUDEȚEAN CLUJ  
COLEGIUL NAȚIONAL „MIHAI VITEAZUL” TURDA

CONCURSUL INTERJUDEȚEAN DE MATEMATICĂ ȘI INFORMATICĂ  
„MARIAN ȚARINĂ”

EDIȚIA A VIII-A



16 – 17 MAI 2008

## Clasa 9

### Problema 2 - ROBOȚI

Într-o hală se găsește o bandă rulantă circulară pe care sunt dispuse piese metalice, la distanțe egale una de cealaltă. Lângă bandă se găsesc două brațe de robot, în poziții fixe. Banda se rotește în sens invers acelor de ceasornic. Fiecare braț ia de pe bandă toate piesele ce ajung în dreptul lui. Procesul se încheie după ce toate piesele de pe bandă au fost luate.

### Cerință

Cunoscându-se numărul  $n$  de piese dispuse pe bandă și greutatele acestora, să se determine unde trebuie poziționate cele două brațe astfel încât diferența, în valoare absolută, dintre suma greutateilor pieselor luate de primul braț și suma greutateilor pieselor luate de al doilea braț să fie minimă.

### Date de intrare

În fișierul **roboti.in** se află:

- pe prima linie un număr  $n$  reprezentând numărul de piese dispuse pe bandă;
- pe cea de-a doua linie, separate prin câte un spațiu,  $n$  numere naturale reprezentând corespunzător greutateile celor  $n$  piese.

### Date de ieșire

Fișierul **roboti.out** va conține o singură linie pe care vor fi scrise, separate printr-un spațiu, două numere distincte reprezentând pozițiile celor două brațe de robot. Pozițiile se dau prin numerele de ordine ale celor două piese în dreptul cărora se vor găsi inițial cele două brațe (și care vor fi totodată primele piese luate de cele două brațe). Piesele se numerotează de la 1.

### Restricții și precizări

- $2 \leq n \leq 10000$ .
- Greutatea fiecărei piese este un număr natural nenul mai mic decât 200.
- Dacă există mai multe perechi de poziții care duc la o diferență minimă se va alege cea mai mică în ordine lexicografică (fiind date două perechi  $(a_1, b_1)$  și  $(a_2, b_2)$  se spune că prima este mai mică lexicografic decât a doua dacă fie  $a_1 < a_2$ , fie  $a_1 = a_2$  și  $b_1 < b_2$ ).

### Exemplu

<b>roboti.in</b>	<b>roboti.out</b>	<b>Explicație</b>
5	2 4	Primul braț de robot ridică piesele 2 și 3 (cu greutatea totală 16), iar al doilea braț de robot ridică piesele 4, 5 și 1 (cu greutatea totală 17).
7 14 2 2 8		

**Timp maxim de execuție/test:** 1 secundă



MINISTERUL EDUCAȚIEI, CERCETĂRII ȘI TINERETULUI  
INSPECTORATUL ȘCOLAR JUDEȚEAN CLUJ  
COLEGIUL NAȚIONAL „MIHAI VITEAZUL” TURDA

CONCURSUL INTERJUDEȚEAN DE MATEMATICĂ ȘI INFORMATICĂ  
„MARIAN ȚARINĂ”

EDIȚIA A VIII-A



16 – 17 MAI 2008

## Clasa 10

### Problema 1 - NUMERE

Un șir  $X=(x_1, x_2, \dots, x_n)$  este secvență a șirului  $Y=(y_1, y_2, \dots, y_m)$  dacă există un indice  $k$ , cu  $1 \leq k \leq m-n+1$  cu proprietatea că  $x_i = y_{k-1+i}$ ,  $1 \leq i \leq n$ .

### Cerință

Fiind dat un șir de cifre binare, să se determine cel mai mic număr natural nenul a cărui reprezentare în baza 2 nu se regăsește ca secvență a șirului dat.

### Date de intrare

În fișierul **numere.in** se află:

- pe prima linie un număr  $n$  reprezentând dimensiunea șirului;
- pe cea de-a doua linie,  $n$  cifre binare, fără spații între ele.

### Date de ieșire

Fișierul **numere.out** va conține o singură linie pe care va fi scrisă reprezentarea zecimală a numărului căutat.

### Restricții și precizări

- $1 \leq n \leq 200000$ .

### Exemplu

numere.in	numere.out	Explicație
5 10110	4	Reprezentarea numărului 4 în baza 2 (100) nu este secvență a șirului.

**Timp maxim de execuție/test:** 1 secundă



MINISTERUL EDUCAȚIEI, CERCETĂRII ȘI TINERETULUI  
INSPECTORATUL ȘCOLAR JUDEȚEAN CLUJ  
COLEGIUL NAȚIONAL „MIHAI VITEAZUL” TURDA

CONCURSUL INTERJUDEȚEAN DE MATEMATICĂ ȘI INFORMATICĂ  
„MARIAN ȚARINĂ”

EDIȚIA A VIII-A



16 – 17 MAI 2008

## Clasa 10

### Problema 2 – Trenuri

Într-un nod feroviar intră un număr de  $m$  linii de intrare, paralele. De-a lungul zonei nodului există un număr de macaze, unul după altul. Fiecare macaz fie unește două linii vecine într-una singură, fie desparte o linie în două. Astfel, la finalul zonei nodului feroviar iese un număr de linii de ieșire egal cu numărul liniilor de intrare plus numărul macazelor ce despart linii minus numărul de macaze ce unesc linii.

Se mai dau niște trenuri, pentru fiecare tren se știe pe ce linie intră și pe ce linie trebuie să iasă.

Se cere să se determine ce traseu trebuie să urmeze fiecare tren astfel încât traseele trenurilor să fie disjuncte două câte două. Se garantează existența unei soluții. Dacă există mai multe soluții, se va da oricare dintre ele.

### Date de intrare

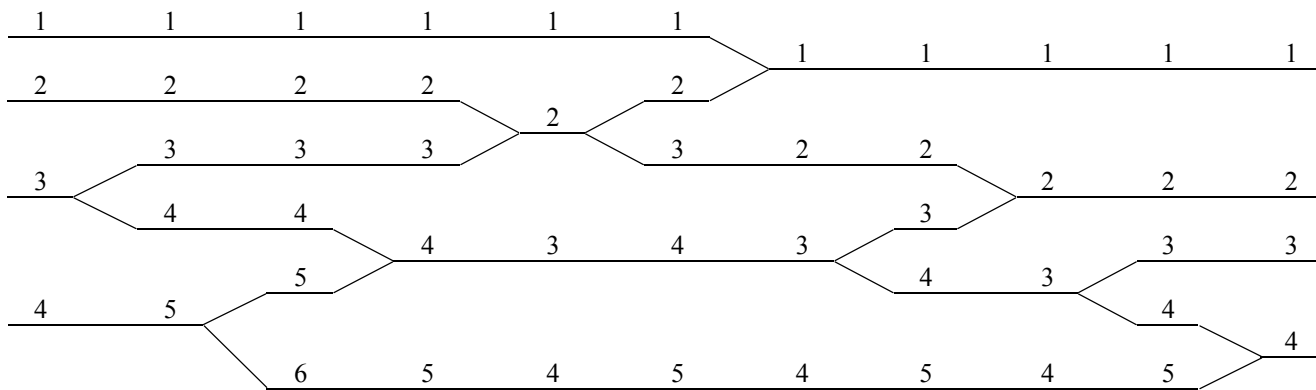
Datele se vor citi din fișierul **tren.in** având următorul format:

- pe primul rând, numărul  $n$  de macaze, numărul  $m$  de linii de intrare și numărul  $t$  de trenuri
- următoarele  $n$  rânduri descriu cele  $n$  macaze, în ordine, de la intrare spre ieșire. Pe fiecare rând se află o literă și un număr. Litera este **S** dacă macazul separă o linie în două și **U** dacă macazul unește două linii. Litera este urmată de numărul liniei ce este bifurcată, respectiv numărul primeia dintre liniile unite. Liniile sunt numerotate consecutiv de la 1 și sunt renumerotate după fiecare macaz.
- pe următoarele  $t$  rânduri, câte o pereche de numere reprezentând linia de intrare și linia de ieșire a trenului.

### Date de ieșire:

În fișierul **tren.out** se vor scrie  $t$  rânduri, câte unul pentru fiecare tren, descriind traseul trenului. Rândul va conține un număr de litere egal cu numărul de macaze de separare întâlnite de tren. Fiecare literă este **S** dacă trenul trebuie să meargă "în sus" la macazul respectiv (spre linii cu număr de ordine mic), respectiv **J** dacă trenul trebuie să meargă în jos.

**Exemplu:**



**tren.in**

10 4 3  
 S 3  
 S 5  
 U 4  
 U 2  
 S 2  
 U 1  
 S 3  
 U 2  
 S 3  
 U 4  
 2 1  
 3 2  
 4 4

**tren.out**

S  
 J S  
 J

**Restricții și precizări:**

$m, n \leq 1000$

$t \leq 100$

**Timp maxim de execuție/test:** 1 secundă



MINISTERUL EDUCAȚIEI, CERCETĂRII ȘI TINERETULUI  
INSPECTORATUL ȘCOLAR JUDEȚEAN CLUJ  
COLEGIUL NAȚIONAL „MIHAI VITEAZUL” TURDA

CONCURSUL INTERJUDEȚEAN DE MATEMATICĂ ȘI INFORMATICĂ  
„MARIAN ȚARINĂ”

EDIȚIA A VIII-A



16 – 17 MAI 2008

## Clasele 11-12

### Problema 1 - PROBLEME

Elevii unei clase s-au hotărât să rezolve cât mai repede problemele propuse pentru examenul de bacalaureat la disciplina informatică. Urmărind aceste probleme, elevii și-au dat seama că multe dintre ele sunt relativ ușor de rezolvat sau se repetă în diverse forme și s-au hotărât să selecteze, să rezolve și să discute mai apoi doar pe acelea care le ridică dificultăți (se presupune că celelalte probleme, neselectate, vor fi rezolvate de fiecare elev în parte individual).

### Cerință

Cunoscându-se numărul  $n$  de elevi ai clasei care susțin examenul de bacalaureat din informatică și numărul  $m$  de probleme care au fost selectate, să se determine numărul de modalități distincte de împărțire a problemelor astfel încât fiecare problemă să fie rezolvată de cel puțin un elev și fiecare elev să rezolve exact o problemă. Două modalități de împărțire a problemelor sunt distincte dacă cel puțin un elev rezolvă probleme distincte.

### Date de intrare

În fișierul **probleme.in** se află pe unica linie a sa, separate printr-un spațiu, cele două numere  $n$  și  $m$  cu semnificațiile din enunț.

### Date de ieșire

Fișierul **probleme.out** va conține o singură linie pe care va fi scris numărul modalităților distincte de împărțire a problemelor.

### Restricții și precizări

- $1 \leq m \leq n \leq 30$ .
- În 60% din teste valoarea determinată va avea cel mult 9 cifre.

### Exemple

probleme.in	probleme.out	Explicație
3 2	6	Există șase variante de împărțire a problemelor. Specificând, pentru fiecare elev, problema pe care o rezolvă, ele sunt: (1,1,2), (1,2,1), (2,1,1), (2,2,1), (2,1,2), (1,2,2).
3 3	6	Soluțiile sunt: (1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2), (3,2,1).

Timp maxim de execuție/test: 1 secundă



MINISTERUL EDUCAȚIEI, CERCETĂRII ȘI TINERETULUI  
INSPECTORATUL ȘCOLAR JUDEȚEAN CLUJ  
COLEGIUL NAȚIONAL „MIHAI VITEAZUL” TURDA

CONCURSUL INTERJUDEȚEAN DE MATEMATICĂ ȘI INFORMATICĂ  
„MARIAN ȚARINĂ”

EDIȚIA A VIII-A



16 – 17 MAI 2008

## Clasele 11-12

### Problema 2 - OBSERVIUM

Tehnologia locuitorilor de pe planeta Observium a avansat foarte mult în ultima eră. Ea le permite acestora să exploreze galaxia și să construiască stații de observare pe aproape orice corp ceresc solid. Ultima descoperire a acestora este planeta P1000. Pe acesta, observienii doresc construirea unei stații de observare.

Construirea stației constă în desfășurarea mai multor activități. Fiecare activitate durează exact o zi, durând de la începutul până la sfârșitul zilei observiene în care este planificată. Pe parcursul unei zile se pot desfășura oricâte activități în paralel. Unele activități sunt condiționate de alte activități; o activitate poate începe doar după terminarea tuturor activităților de care depinde.

### Cerință

Cunoscându-se numărul  $n$  de activități necesare construirii stației de observare și activitățile care condiționează începerea desfășurării fiecărei activități în parte, să se determine numărul minim de zile în care se poate finaliza construirea stației și activitățile care nu pot fi desfășurate mai repede de ultima zi a acestei perioade minime.

### Date de intrare

În fișierul **observ.in** se află:

- pe prima linie un număr  $n$  reprezentând numărul de activități;
- pe fiecare linie  $i+1$  o valoare întreagă  $k_i$  reprezentând numărul de activități care condiționează realizarea activității  $i$  urmată de  $k_i$  valori distincte reprezentând activitățile care condiționează începerea activității  $i$ ,  $1 \leq i \leq n$  (fiecare două valori succesiv scrise pe orice linie a fișierului sunt separate prin câte un spațiu).

### Date de ieșire

În fișierul **observ.out** se va scrie:

- pe prima linie o valoare reprezentând numărul minim de zile;
- pe cea de-a doua linie, separate prin câte un spațiu, în ordine crescătoare, activitățile care trebuie desfășurate în ultima zi.

### Restricții și precizări

- $1 \leq n \leq 4000$ .
- $0 \leq k_i \leq 10$ .
- Se garantează că condiționarea activităților nu împiedică construirea stației.

## Exemplu

**observ.in**

5

0

1 1

3 1 2 5

1 1

0

**observ.out**

3

3

**Explicație**

O planificare optimă este: în prima zi se planifică activitățile 1 și 5, în a doua zi activitățile 2 și 4, iar în a treia zi activitatea 3.

**Timp maxim de execuție/test: 1 secundă**