



Faza LOCALĂ a olimpiadei de INFORMATICĂ

În acest an, problemele propuse spre rezolvare la acest concurs, în județul Cluj au fost alese în colaborare cu redacția GInfo. Vă prezentăm în continuare enunțurile celor nouă probleme cu care s-au confruntat participanții.

P020307: Ecuație

Se consideră ecuația de forma $x^2 - y^2 = k$. Cunoscându-se valoarea întreagă nenulă k , să se determine toate perechile de numere întregi (x, y) pentru care ecuația anterioară este satisfăcută.

Date de intrare

Fișierul de intrare **ECUATIE.IN** va conține o singură linie pe care se va afla un număr întreg care reprezintă valoarea k .

Date de ieșire

În fișierul de ieșire **ECUATIE.OUT** vor fi scrise toate perechile (x, y) , câte una pe o linie. Cele două numere de pe aceeași linie vor fi separate printr-un singur spațiu. În cazul în care nu există nici o astfel de pereche, fișierul de ieșire va conține o singură linie pe care va fi scris mesajul **NU EXISTA SOLUTII**.

Restricții și precizări

- $-1000 < k < 1000, k \neq 0$;
- nu contează ordinea în care sunt scrise perechile în fișierul de ieșire;
- o pereche poate apărea în fișierul de ieșire o singură dată.

Exemple

ECUATIE.IN

8

ECUATIE.OUT

3 1
-3 1
-3 -1
3 -1

ECUATIE.IN

1

ECUATIE.OUT

1 0
-1 0

Timp de execuție: 1 secundă/test

P020308: Număr

Dându-se un număr natural N , să se determine cel mai mic număr M care este par și este format din cifrele numărului N .

Date de intrare

Fișierul de intrare **NUMAR.IN** conține o singură linie pe care se află un număr întreg care reprezintă valoarea N .

Date de ieșire

Fișierul de ieșire **NUMAR.OUT** trebuie să conțină o singură linie pe care se va afla valoarea M . În cazul în care nu există nici un număr par format din cifrele numărului N , pe linia din fișierul de ieșire se va scrie mesajul **NU EXISTA**.

Restricții și precizări

- $1 \leq N \leq 1.000.000.000.000$ (o mie de miliarde);
- numărul M nu poate conține zerouri nesemnificative.

Exemple

NUMAR.IN

3216756

NUMAR.OUT

1235676

NUMAR.IN

45098060

NUMAR.OUT

40005698



NUMAR . IN
3333333333

NUMAR . OUT
NU EXISTA

Timp de execuție: 1 secundă/test

P020309: Sumă

Se consideră un număr $N = \overline{a_1 a_2 \dots a_n}$ (cifrele numărului sunt a_1, a_2, \dots, a_n). Determinați valoarea sumei:

$$S = \overline{a_1 a_2 \dots a_n} + \overline{a_2 a_3 \dots a_n a_1} + \dots + \overline{a_n a_1 \dots a_{n-1}}.$$

Date de intrare

Fișierul de intrare **SUMA . IN** conține o singură linie pe care se află un număr întreg care reprezintă valoarea N .

Date de ieșire

Fișierul de ieșire **SUMA . OUT** trebuie să conțină o singură linie pe care se va afla valoarea S .

Restricții și precizări

- $1 \leq N \leq 100.000.000$ (o sută de milioane);
- termenii sumei pot conține zerouri ne semnificative;
- numărul N nu conține zerouri ne semnificative.

Exemplu

SUMA . IN
1234

SUMA . OUT
11110

Timp de execuție: 1 secundă/test

P020310: Numere

Se consideră un număr natural N cuprins între 1 și 16. Va trebui să construiești un șir care conține toate numerele cuprinse între 1 și 2^N ; acest șir este construit astfel:

- se pornește cu șirul care conține numerele 1 și 2;
- înaintea acestui șir este inserat numărul 3, iar la sfârșitul șirului este inserat numărul 4, obținându-se șirul 3, 1, 2, 4;
- înaintea acestui șir sunt inserate numerele 5 și 6 iar la sfârșitul șirului sunt inserate numerele 7 și 8, obținându-se șirul 5, 6, 3, 1, 2, 4, 7, 8;
- procedeul continuă până la inserarea tuturor celor 2^N numere.

Așadar, la fiecare pas i șirul va conține primele 2^i numere și înaintea sa vor fi inserate următoarele 2^{i-1} numere, iar la sfârșitul său vor fi inserate următoarele 2^{i-1} numere.

Date de intrare

Fișierul de intrare **NUMERE . IN** va conține o singură linie pe care se va afla valoarea N .

Date de ieșire

Fișierul de ieșire **NUMERE . OUT** va conține o singură linie pe care se vor afla cele 2^N elemente ale șirului construit potrivit regulilor prezentate anterior. Elementele șirului vor fi separate prin câte un spațiu.

Exemplu

NUMERE . IN
4

NUMERE . OUT
9 10 11 12 5 6 3 1 2 4 7 8 13 14 15 16

Timp de execuție: 1 secundă/test

P020311: Triunghi

Se consideră un triunghi (determinat de trei puncte necoliniare din plan) și alte N puncte din plan.

Să se determine câte dintre aceste N puncte se află în interiorul triunghiului. Un punct este considerat a fi în interiorul triunghiului și în cazurile în care se află pe una dintre cele trei laturi sau este unul dintre cele trei vârfuri ale triunghiului.

Date de intrare

Primele trei linii ale fișierului de intrare **TRIUNGHI . IN** conțin câte două numere întregi reprezentând coordonatele vârfurilor triunghiului. Următoarea linie conține numărul N al celorlalte puncte. Fiecare dintre următoarele N linii va conține câte două numere întregi reprezentând coordonatele unui punct.

Date de ieșire

Fișierul de ieșire **TRIUNGHI . OUT** trebuie să conțină o singură linie pe care se va afla numărul punctelor din interiorul triunghiului.

Restricții și precizări

- $1 \leq N \leq 100.000$;
- coordonatele punctelor sunt numere naturale cuprinse între 0 și 255;
- coordonatele tuturor punctelor sunt date în ordinea: coordonata pe axa Ox , coordonata pe axa Oy .

Exemple

TRIUNGHI . IN
0 0
10 0
0 10
3
1 1
5 5
10 1

TRIUNGHI . OUT
2

TRIUNGHI . IN

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

TRIUNGHI . OUT

```
0
```

Timp de execuție: 1 secundă/test

P020312: Găuri

Se consideră o matrice ale cărei elemente fac parte din mulțimea $\{0, 1\}$. O **regiune** este o parte a matricei care conține elemente cu aceeași valoare astfel încât se poate ajunge din orice element al regiunii în oricare altul prin deplasări pe orizontală sau verticală, trecând doar prin elemente ale regiunii. O **gaură** este o regiune de zerouri care nu "atinge" marginile matricei. Va trebui să determinați numărul de elemente pe care le conține cea mai mare gaură (cea care conține un număr maxim de elemente).

Date de intrare

Prima linie a fișierului de intrare **GAURI . IN** conține două numere întregi M și N care reprezintă numărul liniilor, respectiv numărul coloanelor matricei. Fiecare dintre următoarele M linii conțin câte N cifre din mulțimea $\{0, 1\}$. Acestea reprezintă valorile elementelor matricei. Valorile de pe o linie **nu** vor fi separate prin spații.

Date de ieșire

Fișierul de ieșire **GAURI . OUT** trebuie să conțină o singură linie pe care se va afla numărul elementelor din cea mai mare gaură.

Restricții și precizări

- $1 \leq M, N \leq 100$;
- dacă nu există nici o gaură, se consideră că cea mai mare gaură are 0 elemente;
- pot exista mai multe găuri care au același număr maxim de elemente.

Exemplu**GAURI . IN**

```
8 8
00000000
00100111
01110101
01011011
01001010
01101001
00111111
00000000
```

GAURI . OUT

```
4
```

Timp de execuție: 1 secundă/test

P020313: Exponent

Se consideră două numere întregi m și n . Să se determine cea mai mare valoare k astfel încât numărul $m!$ este multiplu al numărului n^k .

Date de intrare

Fișierul de intrare **EXPONENT . IN** conține o singură linie pe care se află două numere întregi, separate printr-un spațiu, reprezentând valorile m și n .

Date de ieșire

Fișierul de ieșire **EXPONENT . OUT** va conține o singură linie pe care se va afla valoarea k .

Restricție

- $2 \leq m, n \leq 2.000.000.000$ (două miliarde).

Exemplu**EXPONENT . IN**

```
20 24
```

EXPONENT . OUT

```
6
```

Timp de execuție: 1 secundă/test

P020314: Soldați

Se consideră n soldați care fac parte dintr-un detașament și n valori care reprezintă numărul de pachete de hrană pe care le poate duce fiecare dintre soldați (prima valoare corespunde primului soldat, a doua valoare celui de-al doilea soldat etc.). Soldații trebuie dispuși într-un număr nenul b de bastioane, respectându-se condițiile:

- nu pot exista bastioane fără soldați;
- bastioanele sunt identificate prin numere cuprinse între 1 și b și pot fi distruse, dar numai unul după altul, începând cu primul (cu alte cuvinte, un bastion nu poate fi distrus dacă nu sunt distruse toate bastioanele din fața lui).
- pentru orice număr d de bastioane ($0 \leq d \leq b - 1$) care pot fi distruse, pachetele de hrană purtate de soldații din bastionul $d + 1$ trebuie să poată fi împărțite exact (fără fracțiuni de pachet) la toți soldații rămași în bastioanele identificate prin numere cuprinse între $d + 1$ și b (cele care nu au fost distruse).

Date de intrare

Prima linie a fișierului de intrare **SOLDATI . IN** va conține numărul n al soldaților. Cea de-a doua linie va conține cele n valori care reprezintă numărul de pachete de hrană pe care le poate duce fiecare soldat. Aceste valori vor fi separate prin câte un spațiu.





Date de ieșire

Fișierul de ieșire **SOLDATI.OUT** va conține un număr de linii egal cu numărul bastioanelor. Fiecare linie va conține numărul pachetelor pe care le pot duce soldații din bastionul respectiv. Aceste numere vor fi separate prin câte un spațiu.

Restricții și precizări

- $2 \leq n \leq 1.000$;
- un soldat va duce cel puțin unul și cel mult 1000 de pachete;
- dacă există mai multe soluții, trebuie generată doar una dintre ele;
- pentru datele de test folosite va exista întotdeauna cel puțin o soluție.

Exemplu

SOLDATI.IN

```
6
4 10 3 9 2 3
```

SOLDATI.OUT

```
2 4
3 9
10
3
```

Explicație

Detașamentul este format din șase soldați. Soldații de la primul bastion pot duce 6 pachete care pot fi împărțite la cei șase soldați (cazul în care nu cade nici un bastion).

Soldații de la al doilea bastion pot duce 12 pachete care pot fi împărțite la cei patru soldați rămași (cazul în care nu cade primul bastion).

Soldatul de la al treilea bastion poate duce 10 pachete care pot fi împărțite la cei doi soldați rămași (cazul în care cad primele două bastioane).

Soldatul de la ultimul bastion poate duce 3 pachete. Acestea nu mai trebuie împărțite deoarece a rămas un singur soldat. *Dacă la ultimul bastion rămas ar fi fost plasați mai mulți soldați, atunci numărul total de pachete pe care le duc soldații de la acest bastion trebuie să poată fi împărțit la numărul soldaților de la acest bastion.*

Disponerea tuturor soldaților într-un singur bastion nu este posibilă, deoarece numărul total al pachetelor de hrană este 31, care nu este multiplu de 6.

Timpe de execuție: 1 secundă/test

P020315: Graf

Se consideră un graf conex cu n vârfuri și m muchii. Nodurile sunt identificate prin numere cuprinse între 1 și n . Să se determine nodul aflat la cea mai mare distanță față de nodul identificat prin numărul 1. Distanța este dată de lungimea minimă a unui drum de la nodul respectiv la nodul identificat prin numărul 1.

Date de intrare

Prima linie a fișierului de intrare **GRAF.IN** va conține numărul n al nodurilor și numărul m al muchiilor, separate printr-un spațiu. Fiecare dintre următoarele m linii va descrie o muchie a grafului; ea va conține două numere x și y , separate printr-un spațiu, cu semnificația: *există o muchie între nodul identificat prin numărul x și nodul identificat prin numărul y .*

Date de ieșire

Fișierul de ieșire **GRAF.OUT** va conține o singură linie pe care se va afla numărul prin care este identificat nodul aflat la cea mai mare distanță față de nodul identificat prin numărul 1.

Restricții și precizări

- $1 \leq n \leq 100$;
- $n \leq m \leq 1000$;
- nu pot exista două sau mai multe muchii între aceleași două noduri ale grafului;
- dacă există mai multe noduri aflate la distanța maximă, poate fi ales oricare dintre ele.

Exemple

GRAF.IN

```
7 8
1 2
1 3
2 3
2 4
3 4
1 5
1 6
1 7
```

GRAF.OUT

```
4
```

GRAF.IN

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

GRAF.OUT

```
2
```

Timpe de execuție: 1 secundă/test

Observație

Evaluarea soluțiilor participanților din județul Cluj a fost centralizată fiind coordonată de redacția **GInfo**. Pentru evaluare s-a folosit cea mai nouă versiune a sistemului de evaluare creat de redacția **GInfo**, care a fost utilizat la ediția din acest an a concursului de programare *Bursele Agora*.