



# Faza JUDEȚEANĂ a olimpiadei de INFOR- MATICĂ pentru LICEE

La fel ca anul trecut, subiectele de la faza județeană a olimpiadei de informatică au fost aceleași în întreaga țară. Vă prezentăm în continuare enunțurile celor șase probleme propuse spre rezolvare.

## Clasa a IX-a

### P040309: Text

Vasile lucrează intens la un editor de texte. Un text este format din unul sau mai multe paragrafe. Orice paragraf se termină cu **Enter** și oricare două cuvinte consecutive din același paragraf sunt separate prin spații (unul sau mai multe). În funcție de modul de setare a paginii, numărul maxim de caractere care încap în pagină pe o linie este unic determinat (*Max*).

Funcția pe care Vasile trebuie să o implementeze acum este alinierea în pagină a fiecărui paragraf din text la stânga și la dreapta. Pentru aceasta el va trebui să împartă fiecare paragraf în linii separate de lungime *Max* (fiecare linie terminată cu **Enter**). Împărțirea se realizează punând numărul maxim posibil de cuvinte pe fiecare linie, fără împărțirea cuvintelor în silabe.

Pentru aliniere stânga-dreapta, el trebuie să repartizeze spații în mod uniform între cuvintele de pe fiecare linie, astfel încât ultimul caracter de pe linie să fie diferit de spațiu, iar numărul total de caractere de pe linie să fie egal cu *Max*. Excepție face numai ultima linie din paragraf, care rămâne aliniată la stânga (cuvintele fiind separate printr-un singur spațiu, chiar dacă linia nu este plină).

În general, este puțin probabil ca alinierea să fie realizabilă prin plasarea aceluiași număr de spații între oricare două cuvinte consecutive de pe linie. Vasile consideră că este mai elegant ca, dacă între unele cuvinte consecutive trebuie plasat un spațiu în plus față de alte perechi de cuvinte consecutive, acestea să fie plasate la începutul liniei.

Scrieți un program care să citească lungimea unei linii și textul dat și care să alinieze textul la stânga și la dreapta.

### Date de intrare

Fișierul de intrare **TEXT.IN** conține pe prima linie *Max*, lungimea maximă a unui rând. Pe fiecare dintre următoarele linii se află câte un paragraf al textului care trebuie aliniat stânga-dreapta.

### Date de ieșire

Fișierul de ieșire **TEXT.OUT** va conține textul aliniat stânga-dreapta.

### Restricții și precizări

- $2 \leq \text{Max} \leq 1000$ ;
- lungimea maximă a unui cuvânt din text este de 25 de caractere și nu depășește *Max*;
- lungimea unui paragraf nu poate depăși 1000 de caractere;
- soluția este unică.

### Exemple

#### TEXT.IN

20

Vasile are multe bomboane bune.

#### TEXT.OUT

Vasile are multe  
bomboane bune.

#### TEXT.IN

20

Ana are mere.

Ion are multe pere galbene?



#### TEXT . OUT

Ana are mere.  
Ion are multe pere  
galbene?

#### Explicații

În cazul primului exemplu, pe prima linie au fost plasate câte trei spații între cuvintele consecutive.

În al doilea exemplu, paragraful "Ana are mere." (care are lungimea mai mică decât 20) a rămas aliniat la stânga, iar ultima linie din fiecare paragraf rămâne aliniată la stânga, cuvintele consecutive fiind separate printr-un singur spațiu. În al doilea paragraf, între "Ion" și "are" există două spații, între "are" și "multe" există două spații, iar între "multe" și "pere" un spațiu.

**Timp de execuție:** 1 secundă/test

#### P040310: Numere

Gigel este un mare pasionat al cifrelor. Orice moment liber și-l petrece jucându-se cu numere. Jucându-se astfel, într-o zi a scris pe hârtie 10 numere distincte de câte două cifre și a observat că printre acestea există două submulțimi disjuncte de sumă egală.

Desigur, Gigel a crezut că este o întâmplare și a scris alte 10 numere distincte de câte două cifre și spre surpriza lui, după un timp a găsit din nou două submulțimi disjuncte de sumă egală.

Dându-se 10 numere distincte formate din câte două cifre fiecare, determinați numărul de perechi de submulțimi disjuncte de sumă egală care se pot forma cu numere din cele date, precum și una dintre aceste perechi pentru care suma numerelor din fiecare dintre cele două submulțimi este maximă.

#### Date de intrare

Fișierul de intrare **NUMERE.IN** conține o singură linie pe care se află 10 numere naturale distincte,  $x_1, x_2, \dots, x_{10}$ , separate prin câte un spațiu.

#### Date de ieșire

Fișierul de ieșire **NUMERE.OUT** va conține trei linii. Pe prima linie se va afla numărul de perechi de submulțimi de sumă egală, precum și suma maximă obținută, separate printr-un spațiu.

Pe linia a doua se vor afla elementele primei submulțimi, iar pe cea de-a treia linie se vor afla elementele celei de a doua submulțimi, separate prin câte un spațiu.

#### Restricții și precizări

- $10 \leq x_i \leq 99, \forall 1 \leq i \leq 10$ ;
- fiecare submulțime va avea cel puțin un element și cel mult 9;
- ordinea submulțimilor în perechi nu contează;
- perechea de submulțimi determinată nu este întotdeauna unică.

#### Exemplu

##### NUMERE . IN

60 49 86 78 23 97 69 71 32 10

##### NUMERE . OUT

130 276  
78 97 69 32  
60 49 86 71 10

#### Explicație

Pentru fișierul de intrare de la exemplul anterior, există 130 de soluții, suma maximă este 276. Din cele 10 numere date s-au folosit 9, prima submulțime având patru elemente, iar cea de-a doua cinci elemente.

**Timp de execuție:** 1 secundă/test

## Clasa a X-a

#### P040311: Spirala

Se consideră un automat de criptare format dintr-un tablou bidimensional cu  $n$  linii și  $n$  coloane. Tabloul conține toate numerele cuprinse între 1 și  $n^2$ , așezate "șerpuit" pe linii, de la prima la ultima linie, pe liniile impare pornind de la stânga spre dreapta, iar pe cele pare de la dreapta spre stânga (ca în figura 1).

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

Figura 1

Numim "amestecare" operația de desfășurare în spirală a valorilor din tablou în ordinea indicată de săgeți și de reaşezare a acestora în același tablou, "șerpuit" pe linii, ca și în cazul precedent.

De exemplu, desfășurarea tabloului ducă la obținerea șirului: 1 2 3 4 5 12 13 14 15 16 9 8 7 6 11 10, iar reaşezarea acestuia în tablou conduce la obținerea unui nou tablou reprezentat în figura 2.

După orice operație de amestecare se poate relua procedeul, efectuând o nouă amestecare. S-a observat un fapt interesant: că după un număr de amestecări, unele valori ajung din nou în poziția inițială (pe care o aveau în tabloul de pornire).

De exemplu, după două amestecări, tabloul de dimensiuni  $4 \times 4$  conține 9 dintre elementele sale în exact aceeași poziție în care se aflau inițial (vezi elementele marcate din figura 3).

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

Figura 3

Dându-se numerele naturale  $n$  și  $k$ , scrieți un program care să determine numărul minim de amestecări, ale unui tablou cu  $n$  linii și  $n$  coloane, necesare pentru a ajunge la un tablou cu exact  $k$  elemente aflate din nou în poziția inițială.

#### Date de intrare

Fișierul de intrare **SPIRALA.IN** conține pe prima linie cele două numere  $n$  și  $k$  despărțite printr-un singur spațiu.

**Date de ieșire**

Fișierul de ieșire **SPIRALA.OUT** va conține o singură linie pe care se va afla numărul de amestecări determinat.

**Restricții și precizări**

- $3 \leq n \leq 50$ ;
- numărul minim de amestecări necesare nu va depăși valoarea  $2 \cdot 10^9$ .

**Exemple****SPIRALA.IN**

4 9

**SPIRALA.OUT**

2

**SPIRALA.IN**

6 36

**SPIRALA.OUT**

330

**Timp de execuție:** 1 secundă/test**P040312: Taxe**

Într-o țară în care corupția este în floare și economia la pământ, pentru a obține toate aprobările necesare în scopul demarării unei afaceri, investitorul trebuie să treacă prin mai multe camere ale unei clădiri în care se află birouri.

Clădirea are un singur nivel în care birourile sunt lipite unele de altele, formând un caroiaj pătrat de dimensiune  $n \times n$ . Pentru a facilita accesul în birouri, toate camerele vecine au uși între ele. În fiecare birou se află un funcționar care pretinde o taxă de trecere prin cameră (taxă ce poate fi, pentru unele camere, egală cu 0). Investitorul intră încrezător prin colțul din stânga-sus al clădirii (cum se vede de sus planul clădirii) și dorește să ajungă în colțul opus al clădirii, unde este ieșirea, plătiind o taxă totală cât mai mică.

Știind că el are în buzunar  $S$  euro și că fiecare funcționar îi ia taxa de cum intră în birou, se cere să se determine dacă el poate primi aprobările necesare și, în caz afirmativ, care este suma maximă de bani care îi rămâne în buzunar la ieșirea din clădire.

**Date de intrare**

Fișierul de intrare **TAXE.IN** conține pe prima linie cele două numere  $S$  și  $n$  despărțite printr-un spațiu, iar pe următoarele  $n$  linii, câte  $n$  numere separate prin spații care reprezintă taxele cerute de funcționarii din fiecare birou.

**Date de ieșire**

Fișierul de ieșire **TAXE.OUT** conține o singură linie pe care se află numărul maxim de euro care îi rămân în buzunar sau valoarea  $-1$ , dacă investitorului nu-i ajung banii pentru a obține aprobarea.

**Restricții și precizări**

- $3 \leq n \leq 100$ ;
- $1 \leq S \leq 10000$ ;
- valorile reprezentând taxele cerute de funcționarii din birouri sunt numere naturale, o taxă nedepășind valoarea de 200 de euro.

**Exemplu****TAXE.IN**

10 3

1 2 5

1 3 1

0 8 1

**TAXE.OUT**

3

**Timp de execuție:** 1 secundă/test**Clasele a XI-a și a XII-a****P040313: Compus**

La ultima expediție pe *Marte* a fost descoperit un compus organic necunoscut. Acest compus este acum studiat în laboratoarele NASA.

Cercetătorii au descoperit că acest compus este format numai din atomi de *hidrogen* (*H*), *ixigen* (*I*) și *carbin* (*C*) și are masa moleculară  $M$ .

Se știe că regulile de formare a compușilor organici pe *Marte* sunt următoarele:

- un atom de *carbin* se poate lega de oricare dintre atomii de *C*, *H* și *I* cu oricâte dintre cele 4 legături pe care le are (astfel, în combinația  $H-C-C$  primul atom de *carbin* se leagă prin două legături de alt atom de *carbin* și cu o legătură de alt atom de *hidrogen*);
- un atom de *hidrogen* se poate lega numai de un atom de *carbin* cu singura legătură pe care o posedă;
- un atom de *ixigen* se poate lega numai de atomi de *carbin* cu cele două legături pe care le posedă;
- un compus este un ansamblu cu proprietatea că toți atomii de *carbin* sunt legați conex între ei și nu există vreun atom cu una sau mai multe legături libere (nelegate de un alt atom).

Combinația  $H-C-C$  nu este un compus, deoarece atomii de *carbin* mai au legături libere.

Cercetătorii au în vedere studiul categoriilor de compuși, făcând distincție între doi compuși numai dacă aceștia diferă prin numărul de atomi de *carbin*, de *ixigen* sau de *hidrogen*.

Scrieți un program care să determine câți compuși distincți formați din atomi de *carbin*, *hidrogen* și *ixigen* (cel puțin unul din fiecare) și care au masa moleculară  $M$  există.

**Date de intrare**

Fișierul de intrare **COMPUS.IN** conține pe prima linie masa moleculară a compusului.



## Date de ieșire

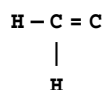
Fișierul de ieșire **COMPUS.OUT** va conține o singură linie pe care se va afla numărul de compuși determinat.

## Restricții și precizări

- $30 \leq M \leq 1000000$ ;
- masa atomului de  $H$  este 1, masa atomului de  $C$  este 5, iar masa atomului de  $I$  este 3. Masa moleculară a unui compus este egală cu suma maselor atomilor din care este constituit compusul respectiv;
- ordinea în care sunt "utilizate" legăturile unui atom nu contează. De asemenea, nici ordinea atomilor sau legăturile interne dintre ei nu contează, atâta timp cât respectă regulile de formare enunțate.

## Observații

Există un singur compus cu masa moleculară 10: cel format cu un atom de  $C$ , doi atomi de  $H$  și un atom de  $I$  ( $5 + 2 \cdot 1 + 3 = 10$ ), compus ale cărui legături pot fi reprezentate astfel:



Se pot obține trei compuși cu masa moleculară 40: ( $5C$ ,  $6H$ ,  $3I$ ), ( $6C$ ,  $4H$ ,  $2I$ ) și ( $7C$ ,  $2H$ ,  $1I$ ) care sunt prezentați în figura 4.

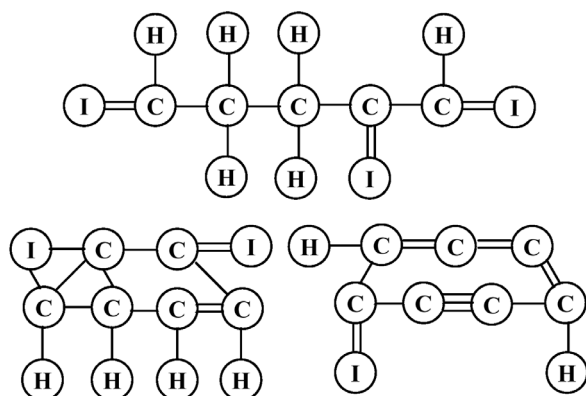


Figura 4

Reprezentarea cu legături a oricăruia dintre compuși nu este unică. Orice altă combinație corespunzătoare aceluiași triplet nu este considerată ca fiind un compus distinct.

## Exemple

COMPUS.IN	COMPUS.OUT
40	3
120	28

**Timpe de execuție:** 1 secundă/test

## P040314: Zmeu

Un *zmeu* cu  $n$  capete călătorește din poveste în poveste, iar în poveștile tradiționale întâlnește câte un *Făt-Frumos* care-l mai reteză de câteva capete, în timp ce în poveștile moderne salvează omenirea mâncând în timp record, cu toate capetele lui, insecte ucigașe apărute prin mutații genetice. Într-o seară, el își planifică o succesiune de povești cărora să le dea viață. El știe  $p$  povești numerotate de la 1 la  $p$ , durata fiecăreia și numărul de capete pe care le pierde în fiecare poveste. Mai știe o mulțime de  $k$  perechi de povești, semnificând faptul că a doua poveste din pereche nu poate fi spusă după prima poveste din pereche.

Știind că trebuie să înceapă cu povestea 1 și să încheie succesiunea cu povestea  $p$ , ajutați bietul zmeu să aleagă una sau mai multe povești intermediare, astfel încât durata totală să fie minimă și să rămână cu cel puțin un cap la sfârșitul tuturor poveștilor.

## Date de intrare

Fișierul de intrare **ZMEU.IN** conține pe prima linie numerele  $n$ ,  $p$  și  $k$  despărțite prin câte un spațiu. Pe fiecare din următoarele  $p$  linii se află câte o pereche de numere  $d_i$  și  $c_i$  (separate prin câte un spațiu) care reprezintă durata și numărul de capete tăiate pentru fiecare poveste. Iar pe ultimele  $k$  linii se află câte o pereche de numere  $p_i$  și  $p_j$  (separate prin câte un spațiu) care semnifică faptul că povestea  $p_i$  nu poate fi spusă după povestea  $p_j$ .

## Date de ieșire

Fișierul de ieșire **ZMEU.OUT** va conține o singură linie pe care se va afla un număr natural care reprezintă durata minimă a succesiunii de povești sau valoarea  $-1$  dacă nu există o astfel de succesiune.

## Restricții și precizări

- $2 \leq n \leq 500$ ;
- $1 \leq p \leq 200$ ;
- $1 \leq k \leq 30000$ ;
- valorile reprezentând duratele și numărul de capete sunt numere naturale (duratele fiind strict pozitive), nedepășind valoarea 10.

## Exemplu

### ZMEU.IN

```
10 4 2
2 6
4 0
1 3
3 3
3 2
4 3
```

### ZMEU.OUT

```
9
```

**Timpe de execuție:** 1 secundă/test