



Problema 1 – Decodificarea mesajelor Morse

Enunț: Alfabetul Morse codifică fiecare literă a alfabetului englez printr-un șir de puncte și linii, astfel:

A	. -	J	. - - -	S	. . .
B	- . . .	K	- . -	T	- . .
C	- . - .	L	. - . .	U	. . -
D	- . .	M	- -	V	. . . -
E	.	N	- .	W	. - -
F	. . - .	O	- - -	X	- . . -
G	- - .	P	. - - .	Y	- . - -
H	Q	- - . -	Z	- - . .
I	. .	R	. - .		

Mesajul codificat Morse este reprezentat printr-un șir de biți, după regulile:

1) . este codificat prin 1

- este codificat prin 111

Oricare două coduri consecutive sunt separate printr-un 0.

Exemplu: M este reprezentat prin 1110111, iar B prin 111010101

2) Literele interioare aparținând aceluiași cuvânt sunt separate prin 000 (3 de 0).

3) Cuvintele sunt separate prin 00000 (5 de 0).

Exemplu: ALB ROSU se codifică prin:

1011100010111010100011101010100000101110100011101110111000101010001010111

Intrare: Fișierul text '**MORSE.IN**' conține una sau mai multe linii. Fiecare linie conține o succesiune de 0 și 1. Primul 1 de pe linie poate fi precedat de o serie de 0 ne semnificativi. Fiecare linie se termină cu 7 de 0. Dacă un caracter de pe o linie nu respectă una din regulile anterioare, linia se decodifică până la apariția primei erori, după care se scrie '?'.
Ieșire: Fișierul text '**MORSE.OUT**' ce conține textul decodificat.

Cerință: Fiind dat un text în cod Morse în fișierul text '**MORSE.IN**', să se scrie un program care scrie textul decodificat *cu majuscule* în fișierul text de ieșire '**MORSE.OUT**'. Cuvintele vor fi separate printr-un singur spațiu, fără semne de punctuație.

Restricții: Lungimea maximă a unei linii din fișierul de intrare este 240.

Exemplu:

Dacă fișierul '**MORSE.IN**' este:

```
00000001011100011101010000011101011101000000
00110000000
000101110000000
000000001011100011101010000011101011101000000
11101010001000101011101000101000000
```

Atunci fișierul de ieșire '**MORSE.OUT**' va fi:

AD C

?

A

AD C

DEFI

Punctaj: 45 puncte

Timp maxim de execuție: 1 secundă/test



Problema 2 – Numere superprime

Definiție: Numim număr **superprim** un număr prim pentru care orice prefix al său este de asemenea număr prim.

De exemplu numărul 719 este superprim deoarece 7, 71, 719 sunt numere prime.

Cerință:

Să se scrie un program care să determine toate numerele superprime cu N cifre.

Restricții:

$$1 \leq N \leq 9$$

Numerele generate vor fi afișate în ordine crescătoare.

Intrare:

Fisierul de intrare se numește **DATE.IN** și conține pe prima linie numărul N .

Ieșire:

Fisierul de ieșire se numește **DATE.OUT** și conține pe o singură linie, separate prin spațiu, numerele superprime determinate sau valoarea 0, dacă nu există soluție.

Exemplu:

<i>DATE.IN</i>	<i>DATE.OUT</i>
2	23 29 31 37 53 59 71 73 79

Observație:

Numărul 1 nu este număr prim.

Timp maxim de execuție: 0.1 secunde/test

Punctaj: 45 puncte



Problema 1 – Portocalosport

Jocul Portocalosport are reguli destul de simple. Mai întâi 100 de portocale sunt etichetate, folosind o cerneală netoxică, cu numere distincte de la 1 la 100. Portocalele sunt aruncate în aer și doi jucători, care încep cu scorul 1, se întrec să mănânce portocalele care cad și în același timp își multiplică scorul cu numărul înscris pe fiecare portocală mâncată. După 10 minute, ei nu mai au voie să mănânce portocale și fiecare concurent își raportează scorul, adică produsul numerelor de pe portocalele mâncate. Câștigătorul neoficial este jucătorul care raportează cel mai mare scor. Inevitabil, în aceste momente apar dispute și adevărul câștigător se anunță abia atunci când aceste dispute sunt rezolvate. Astfel, jucătorul cu cel mai mic scor raportat va putea să câștige partida în anumite condiții. În rezolvarea disputei, se presupune că jucătorul cu scorul cel mai mic spune adevărul, dacă numărul său se poate descompune ca produs de numere diferite din intervalul 1..100. Problema este de a afla dacă scorul mai mare poate fi obținut din portocale care nu au fost mâncate de celălalt jucător. Dacă există o astfel de posibilitate atunci câștigă jucătorul cu scorul mai mare, altfel câștigă cel cu scorul mai mic (dacă scorul are vreo descompunere validă).

Cerință: Se cere să se decidă cu ajutorul unui program care jucător câștigă și, corespunzător acestui jucător să se determine o modalitate de a fi mâncat portocalele (cea mai mică în sens lexicografic).

Exemplul 1

Dacă primul jucător spune 343 și al doilea spune 49, atunci sigur primul jucător minte: singura cale de a avea 343 puncte este de a mânca portocalele 7 și 49, iar singura cale de a avea 49 de puncte este portocala 49. De unde deducem că cel care a declarat 343 puncte minte.

Exemplul 2

Dacă primul jucător spune 194 și al doilea spune 178 atunci primul jucător are doar posibilitatea de a fi mâncat portocalele 2 și 97, al doilea jucător are doar posibilitatea 2 și 89. Deoarece știm că al doilea spune adevărul rezultă că primul minte, deci câștigă jucătorul cu 178 de puncte.

Exemplul 3

Scoruri declarate 138 258

Pentru a obține scorul 138 avem posibilitățile: (6, 23), (3, 46), (2, 69), (2, 3, 23)

Pentru a obține scorul 258 avem posibilitățile: (6, 43), (3, 86), (2, 3, 43)

Putem considera că primul a mâncat portocalele 6 și 23, al doilea a mâncat portocalele 3 și 86, deci și al doilea a spus adevărul, deci câștigă.

Exemplul 4

Scoruri declarate 941 2234

Pentru nici unul din scoruri nu avem vreo posibilitate de a-l scrie ca produs de numere diferite din 1..100, deci ambii mint.

Exemplul 5

Scoruri declarate 1236 100

Nu există nici o posibilitate de a obține scorul 1236, deci primul minte.

Pentru a obține scorul 100 există posibilitățile: (100), (5, 20), (4, 25), (2, 50) și (2, 5, 10). Deci al doilea spune adevărul și câștigă.

Date de intrare: În fișierul **PORTO.IN** se află pe prima linie scorul declarat de jucătorul 1 și scorul declarat de jucătorul 2, separate prin spațiu:

N1 N2

Date de ieșire: Fișierul **PORTO.OUT** conține una sau două linii.

Pe **prima linie** se va scrie un număr cuprins între 1 și 5, cu semnificația:

- 1 - CASTIGA PRIMUL JUCATOR!
- 2 - CASTIGA AL DOILEA JUCATOR!
- 3 - PRIMUL MINTE! CASTIGA AL DOILEA!
- 4 - AL DOILEA MINTE! CASTIGA PRIMUL!
- 5 - AMBII MINT! REMIZA!

Observați că primele două mesaje se referă la cazul când ambele scoruri se pot descompune în produs de portocale, al treilea mesaj se referă la cazul când primul scor declarat nu se poate descompune în produs de portocale diferite, iar al doilea scor da. Al patrulea mesaj se referă la cazul în care al doilea scor nu se poate descompune și primul se poate descompune. Ultimul se referă la cazul când nici unul din scoruri nu se poate descompune în "produs de portocale".

Pentru primele 4 mesaje se va scrie pe **a doua linie** cea mai mică posibilitate (în sens lexicografic) a câștigătorului de a obține scorul declarat.

Exemple:



OLIMPIADA DE INFORMATICĂ
Etapa județeană - IAȘI, 10 martie 2001

Clasa a X-a

Exemplu	PORTO.IN	PORTO.OUT
1	110 119	2 7 17
2	294 202	4 2 3 49
3	941 2234	5

Restricții:

$1 \leq N1, N2 \leq 1.000.000.000$

$N1 \neq N2$

Observații:

- Două șiruri de numere a_1, a_2, \dots, a_n și b_1, b_2, \dots, b_m sunt în ordine lexicografică (șirul a este mai mic decât șirul b) dacă $\exists i \in \{1, 2, \dots, n\}$ astfel încât $a_1 = b_1, a_2 = b_2, \dots, a_{i-1} = b_{i-1}$ și $a_i < b_i$ (ordinea "din dicționar").
Ex. $(2, 3, 49) < (2, 7, 21) < (14, 21)$
- Pe baza observației de mai sus rezultă că soluția care trebuie scrisă în fișierul de ieșire PORTO.OUT este unică.

Timp maxim de execuție: 1 secunda/test

Punctaj: 90 puncte + 10 puncte din oficiu



Problema 2 – JOCUL CULORILOR

Adrian a primit de ziua sa de naștere un joc interesant de la prietenul său Radu. Jocul conține două cartonase cu câte 6 linii și 6 coloane și două creioane colorate diferite (cate unul pentru fiecare jucător). Pe fiecare cartonăș căsuțele conțin numere distincte ce au valori de la 1 la 36, așezate aleator. Distribuția numerelor de pe cele două cartonase diferă.

Jocul se desfășoară astfel :

- Inițial toate căsuțele sunt necolorate.
- Jucătorii colorează alternativ câte o casuță de pe fiecare cartonăș.
- La rândul său, fiecare jucător alege un număr necolorat și îl marchează cu culoarea sa pe ambele cartonășe.
- Jocul se termină când toate casutele sunt colorate.

Stabilirea scorului

La sfârșitul partidei se stabilește punctajul fiecărui jucător pe fiecare cartonăș astfel:

1. Pentru fiecare se punctează toate grupele **maximale** având cel puțin 4 căsuțe consecutive dispuse fie pe aceeași linie, fie pe aceeași coloană, fie pe aceeași diagonală.

O grupă de 4 căsuțe se punctează cu 1 punct.

O grupă de 5 căsuțe se punctează cu 3 puncte.

O grupă de 6 căsuțe se punctează cu 5 puncte .

NOTĂ. Fiecare grupă având 6 căsuțe consecutive se va puncta numai cu 5 puncte. Nu se vor puncta și grupele de 4 sau 5 căsuțe conținute de aceasta. Această regulă se aplică și pentru grupe de 5 căsuțe consecutive.

2. Jucătorii pot primi **punctaj suplimentar** pentru fiecare cartonăș. Pentru aceasta, pentru fiecare cartonăș și pentru fiecare jucător se determină regiunea maximă formată din căsuțe alăturate pe linii și coloane, colorate în culoarea jucătorului respectiv. Dimensiunea unei regiuni este exprimată în numărul de căsuțe ale regiunii. Un jucător primește punctaj suplimentar la cartonăș dacă pentru cartonășul respectiv, dimensiunea regiunii sale maxime este mai mare decât a celuilalt jucător. Punctajul suplimentar primit este diferența dintre dimensiunea regiunii sale maxime și dimensiunea regiunii maxime a celuilalt jucător.

Cerinta

Scrieți un program care să determine punctajul fiecărui jucător, pentru o configurație de joc dată.

Date de intrare



Fișierul de intrare **JOC.IN** are formatul:

JOC.IN	Semnificație
C ₁ C ₂ P ₁₁ P ₁₂ P ₁₃ P ₁₄ P ₁₅ P ₁₆ P ₂₁ P ₂₂ P ₂₃ P ₂₄ P ₂₅ P ₂₆ P ₃₁ P ₃₂ P ₃₃ P ₃₄ P ₃₅ P ₃₆ P ₄₁ P ₄₂ P ₄₃ P ₄₄ P ₄₅ P ₄₆ P ₅₁ P ₅₂ P ₅₃ P ₅₄ P ₅₅ P ₅₆ P ₆₁ P ₆₂ P ₆₃ P ₆₄ P ₆₅ P ₆₆ D ₁₁ D ₁₂ D ₁₃ D ₁₄ D ₁₅ D ₁₆ D ₂₁ D ₂₂ D ₂₃ D ₂₄ D ₂₅ D ₂₆ D ₃₁ D ₃₂ D ₃₃ D ₃₄ D ₃₅ D ₃₆ D ₄₁ D ₄₂ D ₄₃ D ₄₄ D ₄₅ D ₄₆ D ₅₁ D ₅₂ D ₅₃ D ₅₄ D ₅₅ D ₅₆ D ₆₁ D ₆₂ D ₆₃ D ₆₄ D ₆₅ D ₆₆	C ₁ – un caracter ce reprezintă culoarea primului jucător C ₂ – un caracter ce reprezintă culoarea celui de-al doilea jucător P _{ij} –șir de caractere format din numărul situat în poziția (i,j) a primului cartonaș și un caracter ce reprezintă culoarea jucătorului care a ales acest număr D _{ij} –șir de caractere format din numărul situat în poziția (i,j) al celui de-al doilea cartonaș și un caracter ce reprezintă culoarea jucătorului care a ales acest număr

Observati ca tabloul P codifică primul cartonaș de joc , iar tabloul D codifică al doilea cartonaș de joc, după ce partida a fost încheiată . Cele două tablouri se separă printr-o linie goală în fișierul de intrare. Datele de intrare se consideră corecte.

Pe fiecare linie din cele doua tablouri sirurile de caractere sunt separate printr-un singur spatiu. Caracterele care codifica culorile nu sunt separate prin spatiu.

Date de ieșire

Fișierul de iesire **JOC.OUT** contine scorul partidei pe două linii consecutive în formatul următor:

p1

p2

unde **p₁** reprezinta punctajul total stabilit pentru primul jucător si **p₂** este punctajul total stabilit pentru al doilea jucător.

Exemplu

Pentru fișierul de intrare **JOC.IN** următor

gr

9g 18r 7r **12g** 26r 5r
23g **2g** 20r **1g** **10g** 30r
16g **4g** 28r **6g** 19r 25r
32g **21g** 13r 15r 33r 14r
11g 24r 3r **29g** **22g** 27r
31g 36r 17r **34g** **8g** **35g**

36r 27r **32g** **10g** **34g** 13r
24r **1g** **16g** **35g** **21g** **29g**
15r 19r **9g** **2g** **8g** **31g**
25r 30r **12g** **4g** **22g** 7r
28r 3r **23g** **6g** **11g** 26r
5r 18r 14r 20r 17r 33r

se va obține fișierul de ieșire **JOC.OUT**:

21

32



Observație. În acest exemplu, pe primul cartonaș primul jucător are regiunea maximală formată din 9 căsuțe și al doilea are regiunea maximală formată din 18 căsuțe, deci la punctajul jucătorului al doilea se adaugă 9 puncte ($18 - 9 = 9$).

Pe al doilea cartonaș, primul jucător are regiunea maximală formată din 18 căsuțe și al doilea are regiunea maximală formată din 17 căsuțe, deci la punctajul primului jucător se adaugă 1 punct ($18 - 17 = 1$).

Timp de execuție : maxim 2 secunde /test.

Punct: 90 puncte + 10 puncte din oficiu.



Problema 2 – Fazan

Într-un fișier se găsește un text, structurat pe mai multe linii, format din cuvinte scrise cu litere mici ale alfabetului englez, separate prin spații sau/și marcate de sfârșit de linie.

Cerință

Scrieți un program care să determine cea mai lungă înșiruire de cuvinte din text, în ordinea în care acestea apar în textul dat, construită astfel încât pentru oricare două cuvinte consecutive ultima literă din primul cuvânt să coincidă cu prima literă din următorul cuvânt.

Intrare

Numele fișierului de intrare este **IN.TXT**.

Ieșire

Fișierul de ieșire se numește **OUT.TXT**. Pe prima linie în fișierul de ieșire se află **LgMax**, numărul de cuvinte din înșiruirea determinată.

Pe următoarele **LgMax** linii cuvintele din înșiruirea de lungime maximă găsită, câte un cuvânt pe linie.

Restricții

- Orice cuvânt are maxim 15 litere.
- În text există maxim 1000 de cuvinte.

Exemplu

Pentru fișierul de intrare **IN.TXT**:

```
in universul nostru dens si mic  
ursii      mananca si nu fac nimic
```

Fișierul de ieșire **OUT.TXT** poate conține:

```
3  
in  
nostru  
ursii
```

Timp maxim de execuție: 1 secundă/test

Punctaj: 45 puncte.



Problema 1 - Școlari mici, gălăgie mare

Elevii claselor a V-a de la Școala de Informatică “Grigore C. Moisil” au câștigat concursul “Un joc pe calculator – o șansă în plus în viitor” și vor pleca **toți** în excursie de studiu la Disneyland. Trebuie organizate două grupuri de elevi însoțiți de câte un profesor (de informatică, evident). După cum se știe, micuții sunt gălăgioși și se ceartă pentru tot felul de lucruri mai mult sau mai puțin posibile. Pentru a beneficia de o călătorie liniștită, profesorii vor încerca să separe perechile de copii între care au observat că izbucnesc des conflicte și să formeze două grupuri cât mai echilibrate numeric.

Cerința:

Să se scrie un program care să verifice dacă este posibilă împărțirea copiilor în două grupuri în care să nu apară nici un conflict. Dacă există soluție, să se determine o variantă de repartizare astfel încât să rezulte două grupuri cât mai echilibrate numeric (diferența **în modul** dintre numărul de copii din primul grup și numărul de copii din cel de-al doilea grup să fie minimă).

Observații: Copii sunt identificați prin numere distincte de la **1** la **N**.

Profesorii însoțitori nu fac parte din soluție.

Date de intrare:

Datele de intrare se citesc din fișierul text **ELEVI . IN** cu următoarea structură

N K // *N- numărul de elevi și K-numărul de perechi de copii care pot fi în conflict*
x1 y1 // *perechile de certăreși, cu semnificația Xi și Yi pot fi în conflict*
x2 y2
...
xk yk

Datele de ieșire:

Datele de ieșire se vor scrie în fișierul text **ELEVI . OUT** cu următoarea structură: pe prima linie va apare, scris cu majuscule, răspunsul **DA** (dacă pot fi repartizați în două grupuri) sau **NU** (altfel). Dacă pe prima linie se află mesajul **DA** atunci fișierul de ieșire conține pe liniile următoare:

Min // *diferența minimă în modul*
i1 i2 i3 ... ip // *copiii repartizați în primul grup (separați prin câte un spațiu)*
j1 j2 j3 ... jq // *copiii repartizați în al doilea grup (separați prin câte un spațiu)*

Restricții: $2 \leq N \leq 200$

$p+q=N$

Exemple

Exemplul 1		Exemplul 2	
ELEVI . IN	ELEVI . OUT	ELEVI . IN	ELEVI . OUT



OLIMPIADA DE INFORMATICĂ
Etapa județeană - IAȘI, 10 martie 2001

Clasele a XI-a, a XII-a

4 2	DA	4 3	NU
1 2	0	1 2	
4 2	1 4	2 4	
	2 3	1 4	

Timp de executie: 1 secundă/test

Punctaj: 45 puncte

Observație: Datele de intrare sunt corecte.