

Secțiunea 5-6 începători

PROBLEMA 1 DIVLA

100 puncte

Budi este elev în clasa a 6-a și dorește să-și impresioneze profesoara de informatică, așa ca s-a hotărât să rezolve o problemă primită de fratele său mai mare, la un concurs.

Pentru un șir de Q numere, Budi e nevoit să rezolve una din următoarele cerințe :

1. Pentru fiecare număr, va trebui să afișeze “DA” , dacă numărul este prim și palindrom, sau “NU” în caz contrar
2. Să afișeze suma tuturor numerelor DIVLA. Un număr este DIVLA dacă răsturnatul său este divizibil cu suma cifrelor sale.

Cerința

Ajutați-l pe Budi sa rezolve problema dorită !

Date de intrare

Fișierul de intrare **divla.in** va conține pe prima linie un număr C , reprezentând cerința care trebuie rezolvată. Pe următoarea linie numărul Q, cu semnificația din enunț, iar pe următoarea linie cele Q numere.

Date de ieșire

Fișierul **divla.out** va conține:

Daca C = 1 : Q rânduri cu mesaje de tipul “DA” sau “NU”, cu semnificația din enunț, corespunzătoare celor Q numere

Daca C = 2: suma tuturor numerelor DIVLA.

Restricții și precizări

- $1 \leq Q \leq 1000$
- $1 \leq x \leq 1000000$, unde x reprezintă un număr citit .
- Q și x sunt numere naturale.
- $1 \leq \text{Suma} \leq 1000000000$, unde Suma reprezintă valoarea afișata la cerința 2 .
- Se garantează ca niciun număr x nu se divide cu 10 .
- Pentru rezolvarea primei cerințe se acorda 40% din punctaj, iar pentru cea de a 2-a cerința 60% din punctaj.

Exemple:

divla.in	divla.out	Explicatii
1 7 3 121 11 1234 986 1 181	DA NU DA NU NU NU DA	Budi are de rezolvat prima cerința a problemei. 3 este prim și palindrom, la fel 11 și 181, iar celelalte nu îndeplinesc aceste două condiții.
2 8 8 21 48 54 31 27 71 24	182	Budi trebuie să rezolve cerința a 2-a a problemei. Numerele care convin sunt : 8 21 48 54 27 24 Suma acestora este : 182

Timp maxim de execuție: 0.1 secunde/test.

Memorie totală disponibilă: 2 MB, din care 2 MB pentru stivă

Dimensiunea maximă a sursei: 5 Kb

Secțiunea 5-6 începători

PROBLEMA 2 ORA

100 puncte

Andrei a învățat cum se măsoară timpul (ore și minute). Se uită la ceas și vede că e ora 18:12. Ca să arate că a înțeles cum se măsoară timpul, Andrei se gândește să folosească cele patru cifre pentru a găsi ora cea mai mică și ora cea mai mare care se pot forma cu aceste cifre. Găsește că ora cea mai mică este 11:28 și ora cea mai mare este 21:18

Cerință

Dată fiind o oră sub forma unui număr cu patru cifre, să se determine orele corecte cea mai mică respectiv cea mai mare, care se pot forma cu cele patru cifre.

Date de intrare

Fișierul de intrare **ora.in** conține pe prima linie numărul natural n , format din patru cifre, reprezentând ora curentă în forma hhmm.

Date de iesire

Fișierul de ieșire **ora.out** va conține pe prima linie ora minimă și ora maximă formate cu cele patru cifre ale orei curente, afișată sub forma **hh:mm**. Numerele se scriu pe linii separate.

Restricții și precizări

- $0000 \leq n \leq 2359$

Exemplu

ora.in	ora.out	Explicații
1812	11:28 21:18	11:28 este cea mai mică oră care se poate forma cu cele patru cifre 1,8,1,2 ale numărului 1812, iar 21:18 este cea mai mare oră care se poate forma cu aceste cifre

Timp maxim de execuție: 0.1 secunde/test.

Memorie totală disponibilă: 2 MB, din care 2 MB pentru stivă

Dimensiunea maximă a sursei: 5 Kb.

PROBLEMA 3 SECVCOD

100 puncte

Fiind un băiat harnic și conștiincios, Budi are grijă să își facă mereu toate temele. Însă dorind să lucreze mai mult, într-o zi, s-a gândit să analizeze mai bine tema la matematică. Și-a notat pe o hârtie indicii celor N probleme primite și a început să le codeze în felul următor: dacă din cifrele indicelui unei probleme se poate obține un număr palindrom, atunci problema va fi codată cu "1", dacă nu, atunci va fi codată cu "0". Din acest șir de 0 și 1, și-a notat toate lungimile secvențelor de 1 care pot fi obținute. De exemplu, pentru șirul 001110010011101100, Budi va obține secvențe de lungimi 3,1,3 și 2. În final, cu aceste lungimi de secvențe, lungimi ce vor fi ≤ 9 , Budi vrea să construiască un număr cât mai mic. În exemplul anterior, răspunsul este numărul 1233.

Cerința

Budi este însă un băiat care se plictisește foarte repede, așa că vă roagă pe voi să găsiți cel mai mic număr care se poate construi cu lungimile secvențelor de "1" obținute după codarea problemelor.

Secțiunea 5-6 începători

Date de intrare

Fișierul de intrare **secvcod.in** va conține pe prima linie numărul N, urmat de indicii celor N probleme pe care Budi le-a rezolvat.

Date de iesire

Fișierul de ieșire **secvcod.out** va conține numărul minim pe care Budi poate să îl formeze din lungimile secvențelor de "1" obținute după codare.

Restricții și precizări

- $N \leq 500000$;
- Indicii problemelor ≤ 999999999 ;
- Lungimea unei secvențe de "1" este maxim 9;

secvcod.in	secvcod.out	Explicații
15 101 202 1343 111 5 123321 99 10 25 37 77 670076 33 21 7	1234	După codare, se obține șirul: 110111100011101. Lungimile secvențelor sunt 2, 4, 3, 1 => numărul 1234.

Timp maxim de execuție: 0.5 secunde/test.

Memorie totală disponibilă: 5 MB, din care 2 MB pentru stivă

Dimensiunea maximă a sursei: 5 Kb

PROBLEMA 4 TAXA**100 puncte**

Adrian este un programator începător și își dorește să câștige bani pentru a își putea cumpăra un ceas, așa că se angajează. Prima sarcină este scrierea unui program pentru aparatul de taxat al unei parcuri.

Pentru a folosi această parcare, un șofer solicită acest lucru prin apăsarea unui buton la aparatul de la intrare. Acesta va elibera un tichet de parcare care conține data și ora intrării.

La plecare, șoferul scanează tichetul la aparat. În urma scanării, automatul calculează și afișează suma de plată. Calculul sumei se face astfel:

- se citește de la ceasul automatului data și ora plecării
- se calculează diferența de timp dintre ora plecării și ora intrării în parcare
- diferența calculată se rotunjește la număr întreg de ore, în sus
- se înmulțește timpul exprimat în ore cu tariful orar

Cerința

Cunoscându-se data și ora intrării în parcare, data și ora plecării din parcare și tariful orar, să se determine cât a stat mașina în parcare și care este suma care trebuie plătită.

Date de intrare

Fișierul de intrare **taxa.in** conține pe prima linie data și ora intrării în parcare, sub forma zi luna ora minut separate printr-un spațiu.

Secțiunea 5-6 începători

Pe linia a doua a fișierului de intrare se găsesc data și ora plecării din parcare, sub forma zi luna ora minut separate printr-un spațiu.

Linia a treia conține un număr natural q reprezentând tariful orar exprimat în lei. Se cere determinarea timpului în care mașina a stat în parcare, timp exprimat în minute și determinarea sumei de plată pentru parcare.

Date de ieșire

Fișierul de ieșire **taxa.out** conține pe prima linie două valori naturale reprezentând timpul staționării mașinii în parcare și suma de plată, separate printr-un spațiu.

Restricții și precizări

- Mașina stă în parcare cel puțin 1 minut dar nu poate sta mai mult de 7 zile.
- Dacă timpul de staționare nu depășește 15 minute, nu se percepe taxă.
- Cele două date sunt din același an, care nu este bisect.
- Minutul de plecare din parcare nu se contorizează.
- Minutele unei ore sunt numerotate de la 0 la 59.
- $1 \leq q \leq 100$

Exemplu

taxa.in	taxa.out	Explicații
30 01 00 00 30 01 23 59 33	1439 792	Pe 30 ianuarie la ora 00:00, un șofer intra în parcare. Pe 30 ianuarie la ora 23:59, șoferul a ieșit din parcare. Acesta a petrecut astfel, 1439 minute. $1439/60 = 23.98$ Deci șoferul a petrecut 24 de ore în parcare, plătiind $24*33 = 792$.

Timp maxim de execuție: 0.1 secunde/test.

Memorie totală disponibilă: 2 MB, din care 2 MB pentru stivă

Dimensiunea maximă a sursei: 5 Kb