

PROBLEMA 1**100 puncte****DANS**

De 1 Iunie - Ziua Copilului se organizează un spectacol de dans cu și pentru copii. Acesta este programat să se desfășoare în intervalul orar 10.30 -12.00.

În spectacol se înscriu n trupe de dans, iar pentru fiecare trupă se cunoaște timpul necesar realizării dansului în minute și numărul de copii din trupa.

Cerințe:

Cunoscând n, numărul de trupe înscrise, cele n perechi (t,c) unde t reprezintă timpul în minute (minim 5 minute – maxim 15 minute) și c numărul de copii din trupa scrieți un program care:

- Verifică dacă toate cele n echipe înscrise în spectacol se încadrează în timpul alocat spectacolului și afișează mesajul „NU” dacă timpul este mai mare decât cel programat, în caz contrar afișează mesajul „DA”.
- Calculează cu câte minute este programul incomplet sau depășit.
- Calculează câți copii au fost implicați în realizarea spectacolului.
- Calculează care este cel mai mare și cel mai mic timp alocat unui dans.

Date de intrare:

Pe prima linie a fișierului dans.in se afla n, numărul de trupe înscrise, iar pe următoarele linii cele n perechi de forma de mai sus.

Date de iesire:

Datele conform cerințelor a) - d) se vor afișa în fișierul dans.out, răspunsul fiecărei cerințe fiind scris pe câte o linie. La punctul d) cele 2 valori se vor despărți prin spațiu.

Exemplu:

dans.in	dans.out	Explicații
8	DA	Timpul total al dansurilor este 70 min și se încadrează în intervalul orar indicat. Diferența între timpul folosit și cel alocat este de 20 minute. Sunt 61 de copii implicați. Dansul cel mai lung durează 12 minute iar cel mai scurt durează 5 minute.
10 6	20	
8 5	61	
9 6	12 5	
12 10		
10 12		
5 6		
7 8		
9 8		

Timp maxim de execuție: 1 secundă/test.

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

Dimensiunea maximă a sursei: 10 KB.

PROBLEMA 2

100 puncte

OPERAȚII MATEMATICE

Maria, Cristi și Alex au găsit o modalitate de a-și îmbunătăți viteza de efectuare a operațiilor matematice printr-un joc care să corespundă nivelului de vârstă al fiecăruia. Maria știe doar operațiile de adunare și scădere, Cristi a învățat înmulțirile iar Alex fiind în clasa a 5-a studiază divizibilitatea numerelor.

Jocul se desfășoară în felul următor: Maria alege o cifră – cifra de start (întotdeauna este nenulă). Cristi o înmulțește cu 3. La numărul obținut de Cristi, Maria adaugă o nouă cifră și îi spune lui Cristi suma obținută. Cristi caută cel mai mare multiplu a lui 7 mai mic decât numărul obținut de Maria și îl spune Mariei. Aceasta scade din numărul ei multiplul spus de Cristi și obține un număr nou. Din acest moment jocul se reia, Cristi înmulțește cu 3, Maria alege o cifră și o adaugă la numărul obținut de Cristi s.a.m.d...

Între timp Alex este atent la cifrele pe care Maria le-a introdus în joc și caută să vadă dacă numărul format din aceste cifre este divizibil cu 7.

Cerințe:

- 1) Aflați care este numărul obținut de Maria după adăugarea ultimei cifre.
- 2) Ajutați-l pe Alex să verifice dacă numărul format din cifrele alese de Maria este divizibil cu 7.

Date de intrare: Fișierul cifre.in conține pe prima linie numărul de cifre alese de Maria iar pe următoarea linie separate prin spațiu cifrele în ordinea alegerii acestora.

Date de ieșire: Fișierul cifre.out conține pe prima linie numărul obținut la cerința 1 și pe a doua linie răspunsul "DA" în cazul în care răspunsul lui Alex trebuie să fie afirmativ, respectiv "NU" în caz contrar.

Restricții și precizări:

Numărul de cifre alese de Maria este maxim 50.

Pentru rezolvarea corectă a primei cerințe se obține 50% din punctajul unui test.

Exemple:

cifre.in	cifre.out	Explicații
4 2 3 4 5	14 DA	Maria alege 2 ca cifra de start. $2 \times 3 = 6$; Cristi a înmulțit-o cu 3. $6 + 3 = 9$; Maria a adăugat următoarea cifră și Cristi a căutat cel mai mare multiplu a lui 7 mai mic decât 9 $9 - 7 = 2$; Maria scade din numărul ei multiplul spus de Cristi și se reia jocul $2 \times 3 = 6$; $6 + 4 = 10$; $10 - 7 = 3$; $3 \times 3 = 9$; $9 + 5 = 14$; Numărul format conform ordinii cifrelor este 2345 și este divizibil cu 7.
3 3 4 5	23 NU	$3 \times 3 = 9$; $9 + 4 = 13$; $13 - 7 = 6$; $6 \times 3 = 18$; $18 + 5 = 23$; Numărul 345 nu este divizibil cu 7

Timp maxim de execuție: 1 secundă/test.

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

Dimensiunea maximă a sursei: 10 KB.

PROBLEMA 3
PEȘTI**100 puncte**

Andrei își dorește un acvariu cu pești. Găsește în oraș un singur magazin ZOO unde se vând doar peștișori ciudați. Fiecare peștișor se îngrașă în fiecare zi cu câte un număr de grame. Cu fiecare săptămâna ce trece, peștișorii vor lua în greutate același număr de grame ca și săptămâna precedentă, la care se adaugă greutatea pe care o luau în prima săptămână. Nu toți peștișorii sunt de același tip, deci nu au neapărat aceeași greutate și nici nu se îngrașă neapărat cu același număr de grame.

Andrei se hotărăște totuși să cumpere n peștișori, pe care-i numește A,B,C,D,... în ordinea în care îi pune în acvariu. Vrând să știe în permanență ce greutate are fiecare, își notează câți pești a pus în acvariu, litera atribuită fiecărui pește, câte grame are fiecare pește când a fost pus în acvariu și cu câte grame se îngrașă în ziua în care este pus în acvariu.

Cerință

Scrieți un program care afișează, în ordine alfabetică, toți peștii care au cel puțin greutatea G , dată în grame, după ce au trecut z zile. Pentru fiecare pește se va afișa greutatea în grame și litera ce i-a fost atribuită.

Date de intrare

Fișierul **pesti.in** conține pe prima linie, separate prin spațiu, valorile n , G și z .

Urmează n linii, pe fiecare linie fiind scrise separate prin spațiu: o literă mare, greutatea peștelui în grame și numărul de grame cu care se îngrașă în ziua în care este pus în acvariu.

Date de ieșire

Fișierul **pesti.out** va conține mai multe linii, pe fiecare linie fiind scrise, greutatea în grame și o literă mare, separate prin spațiu. Dacă în z zile niciun pește nu depășește greutatea G fișierul **pesti.out** va conține valoarea 0.

Restricții și precizări

- $0 < n < 27$; G – număr natural ; $0 < z < 365$
- Pentru fiecare pește : greutatea inițială în grame și numărul de grame cu care se îngrașă nu depășesc valoarea **10**
- Andrei codifică peștii în ordine, cu litere mari consecutive
- La scrierea fișierului **pesti.in** Andrei respectă ordinea alfabetică

Exemple

pesti.in	pesti.out	Explicații
2 500 25 A 5 2 B 4 3	0	Greutatea în grame a peștișorului A după 25 de zile va fi $5(\text{initial}) + 7 \times 2(\text{săptămâna } 1) + 7 \times (2+2)(\text{săptămâna } 2) + 7 \times (2+2+2)(\text{săptămâna } 3) + 4 \times (2+2+2+2)(\text{săptămâna } 4) = 121 \text{g} < 500 \text{g}$ Greutatea în grame a peștișorului B după 25 de zile va fi $4 + 21 + 42 + 63 + 48 = 178 \text{g} < 500 \text{g}$ (0,5 kg)
pesti.in 5 1530 100 A 5 2 B 4 1 C 2 3 D 4 2 E 5 4	pesti.out 1535 A 2297 C 1534 D 3065 E	Peștișorul B nu atinge greutatea indicată

Timp maxim de execuție: 1 secundă/test.

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

Dimensiunea maximă a sursei: 10 KB.

PROBLEMA 4

100 puncte

BOSUMFLAT

Vrăjitorul informatician Arpsod a făcut un farmec asupra unui șir de N numere naturale, fiecare număr având exact 8 cifre (doar vrăjitorul știe de ce a ales cifra 8). În urma farmecului, numerele au început să prindă sentimente. Un număr X se numește *bosumflat* dacă există un alt număr Y , printre cele N , cu proprietatea că, numărul format din cifrele de pe poziții impare ale lui X este strict mai mic decât numărul format din cifrele de pe poziții pare ale lui Y și numărul format din cifrele de pe poziții pare ale lui X este strict mai mare decât numărul cifrele de pe poziții impare ale lui Y .

Vom defini *gradul de bosumflare* al unui număr X ca fiind egal cu numărul de numere dintre cele N , care îl bosumflă pe X .

Pentru că vrăjitorul este prea ocupat cu alți bosumflați, vă roagă pe voi să determinați gradul de bosumflare pentru fiecare dintre cele N numere.



Cerințe

Cunoscându-se N , numărul de numere precum și numerele efective, determinați gradul de bosumflare pentru fiecare număr în parte.

Date de intrare

Pe prima linie a fișierului **bosumflat.in** se găsește numărul natural N . Pe cea de-a doua linie se găsesc N numere naturale (nu neapărat distincte), fiecare având exact 8 cifre.

Date de ieșire

Pe prima linie a fișierului **bosumflat.out** se vor afișa N numere naturale separate prin spațiu cu semnificația că al i -lea număr afișat reprezintă gradul de bosumflare al celui de-al i -lea număr din șirul inițial.

Restricții și precizări

- $2 \leq N \leq 5000$
- Cele N numere sunt naturale și au exact 8 cifre
- Dacă un număr nu este bosumflat atunci acesta are gradul de bosumflare 0.
- Se garantează că primele două cifre ale fiecărui număr sunt nenule.
- Nu încercați să înțelegeți sentimentele numerelor, deoarece sunt foarte dificile.

Exemplu

bosumflat.in	bosumflat.out	Explicație
5 15629013 29032000 19970808 33331111 86000000	3 4 4 3 2	Dacă $X = 15629013$ și $Y = 29032000$, X este bosumflat ($1691 < 9300$ și $5203 > 2020$, cifrele marcate se găsesc pe poziții pare).

Timp maxim de execuție: 0.2 secunde/test.

Memorie totală disponibilă 1 MB, din care 0,5 MB pentru stivă

Dimensiunea maximă a sursei: 5 KB.

PROBLEMA 5

100 puncte

SILVER

O mulțime “Silver” este o mulțime de numere formată după 3 reguli simple.

1. dacă a este un număr “silver” atunci $a*3$ este tot un număr “silver”.
2. dacă a este un număr “silver” atunci $a*6+1$ este tot un număr “silver”.
3. dacă a este un număr “silver” atunci a este un număr natural strict mai mare decât 1.

Provocarea pe care o primește Andrei este să găsească numere silver cunoscând un singur număr silver indicat de profesorul său.

Cerințe: Scrieți un program care să citească două numere A și N și să determine:

- 1) cel mai mic număr silver din mulțimea care îl conține pe A
- 2) toate numerele silver mai mici sau egale cu numărul N din mulțimea care îl conține pe A

Date de intrare

În fișierul **silver.in** se afla 2 numere A și N unde A este numărul silver cunoscut pe baza căruia trebuie determinate celelalte numere, iar N are semnificația specificată în cerința.

Date de ieșire

Fișierul **silver.out** va conține pe prima linie cel mai mic număr silver din mulțimea care îl conține pe A iar pe a doua linie **șirul ordonat crescător** al numerelor silver mai mici sau egale cu N . Dacă A se găsește în mai multe mulțimi atunci pentru cerința 1 se va afișa cel mai mic număr posibil.

Restricții și precizări

Pentru rezolvarea corectă a unei singure cerințe se obține 50% din punctajul unui test.

$10 < N < 1000000$;

$A > 1$;

Se garantează că A este un număr dintr-o mulțime silver.

Exemplu

silver.in	silver.out	Explicații
91 150	5 5 15 31 45 91 93 135	Șirurile obținute din: $a=2$: 2, 6,13, 18,37, 39,79, 54,109, 111... $a=3$: 3, 9,19, 27,55, 57,115 81,163.. $a=4$: 4, 12,25 , 36, 73, 75,151, 108... $a=5$: 5, 15,31, 45, 91 , 93,135... 5 este cel mai mic număr din mulțimea silver care îl conține pe 91 se afișează doar numerele mai mici decât 150.
73 50	4 4 12 25 36	73 apare în mulțimea care pornește cu 4 4,12,25,36,73,75... se afișează doar numerele mai mici decât 50.

Timp maxim de execuție/test: 0.5 sec/test

Memorie totală: 15 MB din care 5 MB pentru stivă

Dimensiunea maximă a sursei: 10 KB