

Secțiunea 7-8 începători

PROBLEMA 1 ID

100 puncte

Calculatoarele trebuie să se recunoască în rețeaua de Internet printr-un ID. În prezent, există 2 metode de identificare a ID-ului folosite la scară globală: IPv4 și IPv6. Adresele (sau ID-ul) IPv4 au o lungime de 32 de biți (4 octeți) și se reprezintă binar ca 4 grupe de 8 biți separați prin puncte. Spre exemplu, adresa IPv4 11000000.10101000.00000000.00000001 în sistem zecimal este 192.168.0.1. IPv6 este un protocol dezvoltat pentru a înlocui IPv4 în Internet. Scrierea adreselor IPv6 diferă fata de IPv4, fiind scrise sub forma a 8 grupuri de cate 4 cifre hexazecimale, grupurile fiind separate prin doua puncte (:), un exemplu fiind

2001:0DB8:85A3:08D3:1319:8A2E:0370:7334.

Se definesc următorii termeni:

1. Viteza adresei (IPv4 sau IPv6): diferență dintre maximul și minimul numerelor zecimale obținute din cele 4 grupe de 8 biți pentru o adresa IPv4, respectiv cele 8 grupe de 4 cifre hexazecimale pentru o adresa IPv6.
2. Coeficient de înlocuire: dacă se înlocuiește o adresa IPv4 cu una IPv6, acesta are valoarea egală cu diferența în modul dintre viteza IPv4 și IPv6.
3. Eficienta înlocuirii unei rețele (notata cu E): suma coeficienților de înlocuire ai calculatoarelor.

Un departament de logistica dispune de N calculatoare conectate între ele prin Internet. Informaticienii acestui departament vor sa schimbe toate adresele IPv4 în adrese IPv6 și au cerut organizației ce se ocupa cu alocarea adreselor IPv6 să le ofere N adrese de acest tip valide pentru a le distribui pe calculatoarele lor. Însă, ei credeau că adresele o să vina în sistem zecimal, dar organizația le-a trimis în reprezentarea hexazecimala. În funcție de buget, ei doresc să facă una din următoarele două modificări:

Modificarea 1: cunoscând cele N adrese IPv4 deja existente pe calculatoare și adresele IPv6 permise să determine corespondentele lor în sistemul zecimal.

Modificarea 2: să atribuie fiecărei adrese IPv4 una IPv6 astfel încât înlocuirea rețelei să fie cât mai eficientă. Înlocuirea rețelei este MAI EFICIENTĂ cu cât valoarea lui E este MAI MICĂ.

Cerință

Realizați un program care să îndeplinească cele două modificări în funcție de ce anume își dorește departamentul.

Date de intrare

Pe prima linie a fișierului **id.in** se află numărul v egal cu una din valorile 1 sau 2. Valoarea 1 reprezintă îndeplinirea prime modificări, iar valoare 2 a doua modificare. Pe următoarea linie se află numărul N . Pe următoarele N linii se afla adresele IPv4. De la linia $N + 3$ până la $2*N + 2$ se afla adresele IPv6.

Date de ieșire

În fișierul **id.out** pentru cerința 1 se vor afișa adresele mai întâi adresele IPv4 iar după cele IPv6. Fiecare adresă se va afla pe o linie. Pentru cerință 2 se va afișa pe prima linie numărul E pentru înlocuirea cea mai eficientă a rețelei.

Secțiunea 7-8 Începători

Restricții și precizări

- $1 \leq N \leq 100000$.
- Se garantează că adresele furnizate sunt valide.
- Pentru 40% din cazuri $v = 1$, iar pentru celelalte $v = 2$.
- Adresele IPv4 și IPv6 nu conțin separatori pentru grupurile de 8 biți respectiv 4 cifre hexazecimale, adică NU vor exista puncte (.) sau doua puncte (:).
- În afișarea adreselor (IPv4 și IPv6) se va pune punct (.) după fiecare grupă rezultată (IPv4 grupa de 8 biți, IPv6 grupa de 4 cifre hexazecimale) din transformarea în baza zecimală cu EXCEPȚIA ultimei.
- Adresele IPv4 și IPv6 sunt unice și pot exista mai multe viteze de adrese egale.
- Se poate demonstra matematic că pentru oricare 2 adrese IPv4 și respectiv 2 adrese IPv6, înlocuirea cea mai eficientă este pentru minimul vitezei IPv4 asociat minimului IPv6 și maximul vitezei IPv4 asociat maximului vitezei IPv6.
- Sistemul hexazecimal conține 16 "cifre" cu următoarele egalități (primul număr este cel din baza 10 iar cel din dreapta egalului cel din sistemul hexazecimal): 0 = 0; 1 = 1; 2 = 2; 3 = 3; 4 = 4; 5 = 5; 6 = 6; 7 = 7; 8 = 8; 9 = 9; 10 = A; 11 = B; 12 = C; 13 = D; 14 = E; 15 = F

Exemple

id.in	id.out
1 2 10101100110110010000001101100100 11001100111110010000011101100101 20010DB8AC10FE010000000000000000 20300000130E0000000009C0876A130A	172.217.3.100 204.249.7.101 8193.3512.44048.65025.0.0.0.0 8240.0.4878.0.0.2496.34666.4874
2 2 10101100110110010000001101100100 11001100111110010000011101100101 20010DB8AC10FE010000000000000000 20300000130E0000000009C0876A130A	99235

Explicație $v = 1$, deci se va rezolva prima cerință. Prima adresa IPv4 este formată din grupele 10101100, 11011001, 00000011, 01100100 pe care transformându-le în baza 10 se obțin : 172, 217, 3, 100. Analog pentru cea de-a doua adresă. Pentru a treia adresa (prima adresa IPv6) avem grupele 2001, 0DB8. AC10, FE01, 0000, 0000, 0000, 0000 care transformate în baza 10 rezultă: 8193, 3512, 44048, 65025, 0, 0, 0, 0. Analog pentru a patra adresa (a doua IPv6).

Explicație $v=2$: se va rezolva a doua cerință.

Vitezele de conectare la Internet, în ordine, sunt: 214,242, 65025, 34666.

Există 2 atribuiri: prima 214 cu 65025, 242 cu 34666, coeficienții de înlocuire sunt 64811 și 34424, deci $E = 64811 + 34424 = 99235$, iar cealaltă: 242 cu 65025, 214 cu 34666, coeficienții de înlocuire: 64783 și 34452. $E = 99235$.

Timp maxim de execuție: 2 secunde/test.

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

Dimensiunea maximă a sursei: 5 Kb

Secțiunea 7-8 Începători

PROBLEMA 2 MASINUTE

100 puncte

Un grup de n copii, fiecare cu o mașinuță, având la dispoziție o pistă împărțită în segmente de aceeași lungime (1 dm), numerotate de la 1 la m și jetoane pe care le trag la sorți, joacă un joc, după următoarele reguli:

- fiecare copil trage la sorți un jeton pe care se află înscrise poziția de pe pistă unde își amplasează mașinuța (un număr întreg de la 1 la m) și direcția spre care se va deplasa (S pentru stânga sau D pentru dreapta);
- copiii își așează mașinuțele pe pistă în poziția și direcția indicate de jeton și pornesc toți, în același timp, deplasându-se cu aceeași viteză, 1 decimetru pe secundă, unii spre stânga, alții spre dreapta, în funcție de direcția trasă la sorți.
- dacă un copil se întâlnește cu alt copil, cei doi își schimbă instantaneu direcția de mers;
- dacă un copil ajunge la un capăt al pistei, acesta iese din joc (pista nu este circulară);
- câștigă copilul care stă cel mai mult pe pistă.

Cerință

Se cere să se afle după cât timp se termină jocul.

Date de intrare

Fișierul de intrare **masinute.in** conține pe prima linie două numere naturale n și m separate printr-un spațiu.

Următoarele n linii conțin cele 2 valori existente pe jetonul extras de fiecare copil: p_i și d_i separate printr-un spațiu;

p_i reprezintă poziția unde își amplasează mașinuța copilul i iar d_i arată direcția de deplasare a mașinuței acestuia (S pentru stânga și D pentru dreapta). Numărul liniei corespunde numărului de identificare al fiecărui copil.

Date de ieșire

Fișierul de ieșire **masinute.out** va conține timpul de joc al copiilor.

Restricții și precizări:

- $0 < n \leq 100000$
- $1 < m \leq 1000000$
- $0 \leq p_i \leq m$

Exemplu

masinute.in	masinute.out	Explicații
3 12 4 D 7 S 1 S	9	Pista are lungimea de 12 dm și pe pistă vor fi amplasate 3 mașinuțe în pozițiile de 4, 7, 1 și vor avea următoarele direcții de deplasare: dreapta, stânga respectiv stânga. Primele două mașini se vor întâlni după 2 secunde la intersecția segmentului 5 cu segmentul 6 și își vor schimba direcția de deplasare. După schimbarea de sens a doua mașinuță va parcurge încă 7 segmente către stânga și va ieși din joc, în timp ce prima mașinuță după schimbarea de direcție va parcurge încă 5 segmente către stânga, până la ieșire. În acest timp a treia mașinuță a ieșit de pe pistă după 1 secundă. Pe pistă nu vor mai fi mașinuțe după 9 secunde.

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 7-8 începători

PROBLEMA 3 KMARE

100 puncte

La concursul regional EmpowerSoft de la Ploiesti, se propune următoarea problemă: dându-se un număr cu multe cifre să se determine al k-lea număr mai mare decât el, format din aceleași cifre.

Date de intrare

Fișierul de intrare **kmare.in** conține pe prima linie K. Pe a doua linie este numărul cu multe cifre nedespărțite.

Date de ieșire

În fișierul de ieșire **kmare.out** se va scrie, pe prima linie, numărul cerut.

Restricții și precizări

- $K < 1000$
- Numărul de cifre este mai mic decât 1000
- Există totdeauna al k-lea număr mai mare decât numărul dat pe linia a doua

Exemplu

kmare.in	kmare.out	Explicații
12 341567	346157	Următoarele numere mai mari decât 341567, formate tot cu aceste cifre, sunt în ordine: 341576, 341657, 341675, 341756, 341765, 345167, 345176, 345617, 345671, 345716, 345761, 346157.
1 341567	341576	

Timp execuție pe test: 1 secundă/test.

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

Dimensiunea maximă a sursei: 5 Kb

PROBLEMA 4 TZ

100 puncte

În excursie la grădina botanică TZ vrea să viziteze toate speciile de plante. Grădina are formă dreptunghiulară (nxm) și are două intrări care se găsesc în două colțuri diagonal opuse. Fiecare specie este plantată într-o porțiune pătratică de mărime 1x1. TZ vrea să facă un traseu oblic șerpuit intrând pe prima poartă și să iasă pe a doua poartă. TZ în fiecare unitate de timp parcurge o porțiune pătratică de mărime 1 x 1 chiar și atunci când se deplasează oblic.

De exemplu grădina este de 5 x 9 va fi parcursă astfel:

1	2	6	7	15	16	25	26	35
3	5	8	14	17	24	27	34	36
4	9	13	18	23	28	33	37	42
10	12	19	22	29	32	38	41	43
11	20	21	30	31	39	40	44	45

Cerința

Pentru că este foarte nerăbdător, ajutați-l pe tz să știe la ce moment ajunge să vadă specia de pe linia L și coloana C.

Secțiunea 7-8 începători

Date de intrare

Fișierul de intrare **tz.in** conține pe prima linie două numere N, M despărțite prin spațiu ce reprezintă dimensiunea grădinii (numărul de linii și coloane), iar pe următoarele linii sunt scrise două numere: L, C despărțite prin spațiu – coordonatele speciei.

Date de ieșire

În fișierul **tz.out** se vor scrie unul sub altul timpii corespunzători datelor de intrare când TZ se află la acea specie.

Restricții și precizări

- $1 \leq N, M \leq 10000$
- $1 \leq L \leq N, 1 \leq C \leq M$
- $N \leq M$
- numărul de linii din fișier este mai mic decât 1000
- TZ în fiecare unitate de timp parcurge o porțiune pătratică de mărime 1×1 chiar și atunci când se deplasează oblic.
- Prima poartă este în poziția 1×1 iar cea de-a doua poartă este în poziția $n \times m$.

Exemplu

tz.in	tz.out	Explicații
5 9	1	TZ intră pe poarta unu la timpul 1, la specia de pe poziția 3×5 ajunge la timpul 23, la specia de pe poziția 4×4 ajunge la timpul 22 și așa mai departe
1 1	23	
3 5	22	
4 4	44	
5 8	5	
2 2	4	
3 1	23	
3 5	31	
5 5	40	
5 7	41	
4 8		

Timp execuție pe test: 0.1 secunde/test.

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

Dimensiunea maximă a sursei: 5 Kb.