

CLASA A IX-a

ROBOSONG

PROBLEMA 1

100 puncte

R2D2, simpaticul roboțel din STARWARS și-a făcut un nou prieten, Robosong. Robosong transformă mesajele numerice în mesaje muzicale, utilizând robo10 (cifrele 0...9). R2D2 cunoaște foarte bine aplicațiile teoremei împărțirii cu rest a două numere întregi, iar pentru el orice două numere întregi reprezintă o adevărată simfonie. R2D2 este și el foarte pasionat în ultima vreme de muzică și de codificarea sunetelor muzicale. Pe scara robo10 el încearcă să compună melodii cu ajutorul a două numere naturale, pornind de la teorema împărțirii cu rest. Astfel, pentru generarea unei note din robo10, cu numerele naturale a și b , împarte produsul $a * 10$ la b , obținând o notă; apoi îl înlocuiește pe a cu restul împărțirii și continuă procesul. R2D2 este foarte încântat de compozitiile prietenului său, dar observă că în unele partituri sunt sevențe care se repetă identic (refren), unele au uvertură, altele nu. R2D2 vrea acum, să rezolve programarea generării compozitiilor pentru Robosong.

Cerință

Scrieți un program care, citind două numere naturale a și b ($a < b$), va determina cifrele uverturii și cifrele refrenului. Se vor afișa cifrele uverturii, în continuare cifrele refrenului apoi un spațiu urmat de un număr reprezentând câte cifre are refrenul.

Date de intrare

Fișierul de intrare **robosong.in** conține pe prima linie două valori: a și b numere naturale, separate printr-un spațiu.

Date de ieșire

Fișierul de ieșire **robosong.out** va contine o singură linie cu cifrele uverturii urmate în continuare de cifrele refrenului și, după un spațiu, numărul de cifre ale refrenului.

Restricții și precizări

$1 < a, b < 1.000.000.000$
 a și b sunt distințe, $a < b$.

Exemplu

robosong.in	robosong.out	Explicație
164 824	19902912621359223300970873786407766 34	1 este uvertura 99029126213... este refrenul
13 32	406250 1	40625 este uvertura 0 este refrenul
6 11	54 2	nu există uvertură 54 este refrenul

Timp maxim de execuție: 0.1 secunde/test.

CLASA A IX-a
LISIOM

PROBLEMA 2
100 puncte

Un grup de cercetători explorează planeta Lisiom. Ei au descoperit n tipuri de fibre, din componența unor microorganisme, care trebuie colectate în dispozitive speciale, epro-uri. Eprourile pot fi prelungite la capete, spre stânga și/sau spre dreapta. O fibră este dată de coordonatele ei, de pe axa numerelor întregi [stânga,dreapta]. Într-un epro pot fi colectate oricâte fibre. Ca dispozitivul să fie modificat spre stânga sau/și spre dreapta trebuie ca fibra ce urmează a fi colectată să aibă o „secvență” de mai multe puncte comună cu dispozitivul.

Cerință

Scrieți un program care să determine numărul minim de dispozitive necesare pentru colectarea fibrelor, limitele acestor dispozitive și numărul fibrelor colectate de fiecare dispozitiv.

Date de intrare

Fișierul de intrare **lisi.in** conține:

- pe prima linie numărul natural **n**, care reprezintă numărul de fibre descoperite;
- pe fiecare dintre următoarele **n** linii se află **stg drt** (două numere întregi separate printr-un spațiu); numerele de pe linia **x+1** reprezintă coordonatele de colectare a fibrei **x**.

Date de ieșire

Fișierul de ieșire **lisi.out** va conține pe prima linie numărul minim de dispozitive de stocare **p**, iar pe următoarele **p** linii, cate trei numere **st, dr, nr**, reprezentând limitele și numărul de fibre din fiecare dispozitiv.

Restricții

- $1 \leq N \leq 100000$
- $-1000000 \leq \text{st}_{\text{x}} < \text{drt}_{\text{x}} \leq 1000000$ (numere întregi, reprezentând coordonate), pentru orice **x** de la 1 la **N**

Exemple

lisi.in	lisi.out	lisi.in	lisi.out	lisi.in	lisi.out
3	2	4	3	5	2
-10 10	-10 10 2	2 5	2 7 2	-10 10	-20 10 4
-2 5	20 50 1	4 7	10 20 1	10 12	10 12 1
20 50		10 20	30 40 1	-20 10	
		30 40		7 10	
				7 8	

Timp maxim de execuție/test: 0.1 secunde

CLASA A X-a
MESAJ

PROBLEMA 1
100 puncte

Locotenentul DATA, încearcă să intre în rețea intergalactică de comunicații. Receptorul său înregistrează la un moment dat niște mesaje ciudate, dar foarte exacte, formate din cifre binare. Studiindu-le cu atenție, constată că pe grupuri de 1 octet, unele din ele reprezintă de fapt caracterele cu codul ASCII din intervalul [32,127], numai că bitul 7 nu este întotdeauna 0. Reluând analiza mesajelor, el constată că bitul 7 al unui octet are valoarea 0, dacă octetul are un număr par de cifre binare 1 și 1 în caz contrar. De asemenea mai întâlnescă în unele mesaje caracterul cu codul ASCII 10(00001010), care presupune el că ar fi un sfârșit de linie.

Unele mesaje însă nu le poate descifra. Ele par a nu fi corect transmise. Pentru că sunt multe, DATA vă roagă să-l ajutați să faceți selecția mesajelor ce pot fi decodificate și să le și decodificați, iar pe cele care nu pot fi decodificate să-i transmită poziția octetilor din mesaj care nu se pot decodifica.

Cerință

Să se scrie un program care să verifice dacă un mesaj poate fi sau nu poate fi decodificat.

Date de intrare

Fișierul de intrare mesaj.in are pe prima linie o succesiune de caractere '0' și '1' care reprezintă mesajul recepționat. Între caractere nu există spații. Linia se termină cu caracterul marcaj de sfârșit de linie (newline).

Date de ieșire

Fișierul de ieșire mesaj.out are pe prima linie mesajul DA dacă textul poate fi cod sau NU în caz contrar. În cazul în care mesajul de pe prima linie este DA liniile următoare vor conține textul decodificat în clar. În cazul în care mesajul de pe prima linie este NU linia următoare va conține numerele de ordine ale caracterelor care nu pot fi decodificate, în ordine strict crescătoare, separate prin câte un spațiu.

Restricții și precizări

- Cei 8 biți ai codului ASCII a unui caracter se numerotează de la 0 la 7, de la dreapta la stânga, cel mai din stânga bit fiind bitul 7 iar cel mai din dreapta bitul 0.
- Textul recepționat are cel mult 60000 caractere.
- Numărul de caractere '0' și '1' din prima linie a fișierului de intrare este multiplu de 8.
- Codurile ASCII ale caracterelor din mesaj aparțin mulțimii {10}U{[32,127]}, codul 10 însemnând trecerea la începutul unui rând nou.
- Nici o linie din fișierul de ieșire nu va avea mai mult de 255 caractere.
- Octetii din mesajul recepționat sunt numerotați începând de la 0.
- Mesajele DA/NU din prima linie a fișierului de ieșire se scriu cu majuscule.

Exemple

mesaj.in	mesaj.out	Explicație
0101000011100001011100100110100101110100111000010111010001100101	DA Paritate	Toate codurile sunt decodificabile
mesaj.in 110100001110000111110010011010010100101110100111000010111010011100101	mesaj.out NU 0 2 7	Primul octet a fost recepționat ca succesiunea de biți 11010000 ceea ce înseamnă că fără bitul 7 ar fi trebuit să existe un număr impar de cifre 1, ceea ce este fals. Deci octetul nu se poate decodifica. Același lucru se verifică și pentru octetii cu numerele de ordine 2 și 7
mesaj.in 01000001111101001101001000010100110010100001010101100101011000	mesaj.out DA Azi e 28	Toți octetii sunt decodificabili. În mesaj există două caractere cu cod ASCII 10

Timp maxim de execuție/test: 1 secundă

**CLASA A X-a
PARC****PROBLEMA 2
100 puncte**

În călătoriile lui, Gulliver ajunge în Araugia, țara cercurilor colorate. Araugenii au obiceiul să-și ornamenteze parcurile și grădinile cu parcele circulare, fiecare cerc fiind cultivat cu flori de o singură culoare. Pe Gulliver îl uimește talentul arhitectural al araugenilor de a-și amenaja grădinile și tot minunându-se de frumusețea acestora, el constată că armonia coloristică este dată pe alocuri de o proiectare geometrică, structurată în dreptunghiuri, cu laturile paralele cu axe de coordonate, având toate patru vârfurile în centrele unor cerculete de aceeași culoare. În timp ce admira armonia geometrică dintr-o grădină, în imediata lui apropiere araugenii se pregăteau să amenajeze un astfel de parc. Pe suprafața parcului pot fi amenajate n cerculete, fiecare cerculeț fiind identificat de coordonatele centrului, x, y - numere întregi și culoarea florii. Observându-l pe Gulliver, îl întreabă dacă poate să-i ajute și să le răspundă la întrebarea câte dreptunghiuri, care să îndeplinească condițiile descrise mai sus, pot proiecta.

Cerință

Ajutați-l pe Gulliver să determine numărul maxim de dreptunghiuri care se pot forma cu cele n cerculete din parc.

Date de intrare

Pe prima linie a fișierul text **parc.in** se găsesc două numere n și c_n reprezentând numărul de cerculete din parc și numărul de culori asociate cerculețelor. Pe următoarele n linii se citesc câte trei numere x y c reprezentând în ordine coordonata pe axa OX (abscisa), coordonata pe axa OY (ordonata) și codul culorii asociate cerculețului. Nu există două cerculete cu același centru.

Date de ieșire

Pe prima linie a fișierului text **parc.out** se va scrie un singur număr cu semnificația numărul maxim de dreptunghiuri, cu toate 4 vârfurile în centre ale cerculețelor de aceeași culoare, cu laturile paralele cu axe de coordonate, care se pot proiecta.

Restrictii

- $1 \leq N \leq 1000$
- $1 \leq C_n \leq 5$
- $-1000 \leq x, y \leq 1000$

Exemplu

Parc.in	parc.out	Explicație
9 2	3	Vârfurile celor trei dreptunghiuri sunt:
3 10 1		(3, 0) (3, 4) (6, 4) (6, 0)
3 8 2		(3, 0) (3, 10) (6, 10) (6, 0)
3 6 1		
3 4 1		
3 0 1		
6 0 1		
6 4 1		
6 8 2		
6 10 1		

Timp de execuție: 0.2 secunde pe test

CLASA A XI-a

PROBLEMA 1

MAP

Fata cea mofturoasă a lui Verde Împărat nu vrea să se mărite. Ca să scape de cicaleala părinților săi i-a supus pe toți pretendenții de până acum la tot felul de probe, care mai de care mai ciudate și mai greu de îndeplinit. Iată că la poarta împăratului ajunge și Făt Frumos, care vrea să-și încerce și el norocul. Fata i-a înmânat lui Făt Frumos o hartă care descrie o parte a Împărației, de formă pătrată, cu latura de **n** dam. Pe hartă este marcat un castel. Harta este împărțită în **n × n** zone pătrate, cu latura de un dam. O astfel de zonă poate conține o construcție, o locuință, o pădure, etc. sau o zonă aridă liberă. Sarcina lui Făt Frumos, este de a construi o cale de acces directă și continuă din exterior spre castel, cu covoare de gazon pătrate cu latura de 1 dam, utilizând pentru acest lucru doar zone libere. Întrucât împărația este în criză financiară și gazonul este foarte scump, el trebuie să folosească un număr minim de covoare. Ajutați-l voi pe Făt Frumos, măcar să determine numărul minim de covoare, că dacă nu-și va termina sarcina în timp util, își va pierde capul.

Cerință

Scrieți un program care să determine numărul minim de covoare necesare pentru construirea unei căi de acces continue de la marginea la castel. Dacă există mai multe căi de acces care utilizează același număr minim de covoare, specificați și numărul variantelor optime.

Date de intrare

Fișierul de intrare **map.in** conține pe prima linie patru valori naturale **n**, **m**, **x1**, **y1** separate prin spațiu, **n** reprezentând dimensiunea hărții, **m** numărul de zone ocupate care se găsesc pe hartă, poziția castelului pe hartă (**x1** reprezentând coloana, iar **y1** linia). Fiecare dintre următoarele **m** linii conține câte două numere naturale **x** și **y** separate printr-un spațiu, reprezentând pozițiile zonelor ocupate pe hartă (**x** reprezintă coloana, iar **y** reprezintă linia zonei ocupate).

Date de ieșire

Fișierul de ieșire **map.out** va conține pe prima linie două numere naturale **a** și **b** separate de un spațiu, primul număr reprezentând numărul minim de covoare, iar cel de-al doilea numărul de căi de acces ce se pot construi cu un număr minim de covoare de gazon.

Restricții

- $1 \leq n \leq 250$
- $1 \leq m < n^2$
- Calea de acces este continuă dacă oricare două covoare consecutive au o latură comună.
- Drumul începe cu o zonă liberă de pe marginea hărții și se termină cu zona unde se găsește castelul.
- Pentru datele de test există întotdeauna cel puțin o soluție.

Exemplu

map.in	map.out	Explicație
8 6 4 4	4 2	Modalitățile de a construi calea cu număr minim de covoare sunt:
7 2		-----
3 3		-----x-
6 4		--x-----
4 5		OOOO-x--
3 7		---x----
5 7		-----
		(cu x am marcat zonele ocupate, cu - zonele libere, iar cu o covoare de gazon).

Timp maxim de execuție/test: 0,1 secunde

CLASA A XI-a

PROBLEMA 2

ALGOP

100 puncte

AlgoPrim și AlgoPar au descoperit un nou joc, de tras la țintă. Țintele sunt generate de un computer; la țintă se trage cu numere. Ca să doboare cât mai multe ținte, cei doi jucători pot folosi doar secvențe de numere, în care pentru oricare două numere a, b există alte două numere x, y , astfel încât $a*x+b*y=1$. Cu cel mai mare divizor comun a două numere întregi, cei doi se descurcă destul de bine, numai că șirurile de numere generate de computer sunt foarte lungi. Lor le trebuie să calculeze exact numărul de secvențe din sir, astfel încât într-o secvență, oricare două numere să fie prime între ele. Având acest număr calculat ei pot doborâ oricâte ținte.

Date de intrare

Prima linie a fișierului de intrare **algop.in** conține numărul natural N reprezentând dimensiunea șirului. Pe următoarea linie se află N numere naturale.

Date de ieșire

Prima linie a fișierului de ieșire **algop.out** va conține numărul de secvențe cerut.

Restricții

- $1 \leq N \leq 10^5$
- Numerele din sir vor fi cuprinse între 1 și 1.000.000 (10^6)
- Secvențele de lungime 1 nu se iau în considerare

Exemplu

algop.in	algop.out	Explicație
4 3 2 6 5	2	Secvențele sunt : 3 2 și 6 5.

Timp de execuție: 1 secundă pe test

CLASA A XII-a
ENERG

PROBLEMA 1
100 puncte

În goana după energie ieftină și neconvențională, un grup de cercetători au creat un sistem, prin care circulă n particule, aflate inițial în cele n locații ale sistemului. Pentru a genera o cantitate maximă de energie, particulele trebuie să se deplaseze prin canalele de comunicație, care există între oricare două locații, spre una din cele n locații, numită „energmax”. Canalele de comunicație pot fi directe sau indirecte. Un canal de comunicație este direct dacă el nu trece prin alte locații. Între oricare două locații există o singură succesiune de canale de comunicație, prin care se poate ajunge de la o locație la alta. O particulă care se deplasează pe un canal de comunicație directă consumă o unitate de energie.

Toate particulele trebuie să ajungă în locația „energmax”. La deplasarea pe anume q canale particulele nu pierd energie. Locația „energmax” se poate stabili pe traseul celor q canale. Problema cercetătorilor este aceea de a alege cele q canale, astfel încât prin folosirea lor pierderea de energie în sistem să fie minimă. O particulă se deplasează o singură dată de la locația ei la locația energmax.

Cerință

Scrieți un program care determină pierderea minimă de energie, care se poate obține prin alegerea adecvată a celor q canale și a locației „energmax”.

Date de intrare

Fișierul **energ.in** conține

- pe prima linie două valori n q separate printr-un spațiu reprezentând numărul total de particule și numărul de canale pe care nu se consumă energie.
- pe următoarele n-1 linii se află câte două valori i j separate printr-un spațiu, reprezentând numerele de ordine a două locații între care există un canal de comunicație direct.

Date de ieșire

Pe prima linie a fișierului **energ.out** se va scrie numărul total minim de unități de energie pierdută, prin deplasarea particulelor în sistemul de generare a energiei, iar pe următoarele q linii, câte două valori i j, separate prin spatiu, reprezentând capetele celor q canale alese care au consum zero de energie (una din soluții, dacă sunt mai multe).

Restricții

- $1 < n \leq 10000$, $0 < q < n$
- $1 \leq i, j \leq n$, $i \neq j$
- Oricare două perechi de valori de pe liniile 2,3,...,n din fișierul de intrare reprezintă două locații distințe.
- Perechile din fișierul de intrare sunt date astfel încât respectă condițiile din problemă.

Exemplu

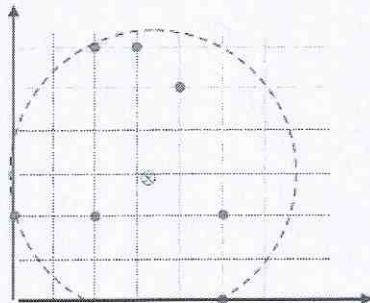
energ.in	energ.out	Explicație
13 3	11	
1 2	5 7	
2 3	7 8	
2 8	8 10	
7 8		
7 5		
5 4		
5 6		
8 9		
8 10		
10 11		
10 12		
10 13		

Timp maxim de executare: 0.1 sec/test

CLASA A XII-a
SKYWALKER

Luke, R2D2 și C3PO sunt într-o mare primejdie. Fără negativă le-a întins o capcană și încearcă să-i domine. Cele n puncte energetice, emit unde radiale de soc, din centrul cercului de rază minimă, care le înconjoară. Jedi îi transmite lui Luke, că pentru a primi un ajutor de la un alt SKYWALKER, trebuie să comunice repede poziția cât mai exactă a centrului cercului determinat de cele n puncte și raza acestuia. Toate bune, numai că între timp lui R2D2 îi pocnesc circuitele, iar C3PO rezistă foarte greu la radiațiile emise. Ajutați-l voi pe Jedi să poată să-i salveze pe Luke și cei doi prieteni.

PROBLEMA 2



Cerință

Scrieți un program care să determine poziția centrului cercului care cuprinde în interior sau pe circumferință cele n puncte cât și raza minimă a acestui cerc.

Date de intrare

Fisierul de intrare **skywalk.in** conține pe prima linie un număr natural n , reprezentând numărul de puncte energetice. Pe următoarele n linii se află pozițiile în plan a acestora. Mai exact, pe linia $i+1$ se află două numere întregi separate printr-un spațiu x y , ce reprezintă abscisa și respectiv ordonata punctului energetic i . Două puncte diferite nu au aceeași poziție.

Date de ieșire

Fisierul de ieșire **skywalk.out** conține pe prima linie două numere reale separate printr-un spațiu x y , reprezentând abscisa și ordonata poziției centrului cercului. Pe cea de a doua linie se va scrie un numar real reprezentând raza cercului.

Restricții și precizări

$2 < N < 15001$

$-15000 < x, y < 15001$

Numerele reale din fisierul de ieșire trebuie scrise cu trei zecimale cu rotunjire.

*La evaluare, se verifică dacă diferența dintre soluția afișată și cea corectă (în valoare absolută) este <0.01 .

Exemplu

skywalk.in	skywalk.out	Explicații
7	3.250 2.875	
5 0	3.366	
2 6		
4 5		
2 2		
0 2		
3 6		
5 2		

Timp maxim de executie/test: 0.2 secunde