

1. "KRĀSAINAIS SKAITLIS"

Ir doti k naturāli skaitļi. Katra skaitļa visi cipari ir nokrāsoti vienā krāsā. Katrs skaitlis ir nokrāsots citā krāsā. No šo skaitļu cipariem nepieciešams izveidot jaunu ("krāsainu") skaitli tā, lai

- būtu izmantoti visi dotajos skaitļos ietilpstošie cipari (un nekādi citi);
- katrai krāsai ciparu secība jaunajā skaitlī būtu tāda pati kā dotajos skaitļos;
- iegūtais skaitlis būtu lielākais iespējamais.

Piemēram, ja dotie skaitļi ir 712 , **35319** un 2090, tad krāsainais skaitlis ir 735321912090 (pirmās krāsas cipari ir pasvītroti, otrās izcelti ar trekno fontu, bet trešās rakstīti parastā fontā).

Uzrakstiet programmu, kas dotiem skaitļiem atrod šo krāsaino skaitli!

Ievaddati

Teksta faila `skaitlis.dat` pirmajā rindā dots krāsu skaits k ($k \leq 20$).

Katrā no nākošajām k faila rindām dots viens naturāls skaitlis, kura vērtība nepārsniedz 10^9 . Faila $i+1$ -ajā rindā dots i -tās krāsas skaitlis.

Izvaddati

Teksta faila `skaitlis.rez` vienīgajā rindā jāizvada iegūtais krāsainais skaitlis.

Piemērs

Ievaddati (<code>skaitlis.dat</code>)	Izvaddati (<code>skaitlis.rez</code>)
3	735321912090
712	
35319	
2090	

2. "SIMETRIJAS"

Aplūkojot sekojošus latīņu alfabēta lielos burtus un ciparus,

ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789

viegli pamanīt, ka dažiem no tiem piemīt apbrīnojamas īpašības:

- burtu "A" atspoguļojot pret spoguļi, kas novietots vertikāli, iegūsim to pašu burtu "A". Tāda pat īpašība ir arī burtiem "H", "I", "M", "O", "T", "U", "V", "W", "X", "Y" un cipariem 0,8.
- burtu "B" atspoguļojot pret spoguļi, kas novietots horizontāli, iegūsim to pašu burtu "B". Tāda pat īpašība ir arī burtiem "C", "D", "E", "H", "I", "K", "O", "X" un cipariem 0,3,8.
- burtu "S" pagriežot par 180 grādiem, iegūsim to pašu burtu "S". Tāda pat īpašība ir arī burtiem "H", "I", "N", "O", "X", "Z" un cipariem 0 un 8. Savukārt, ciparu 6 pagriežot par 180 grādiem, iegūsim ciparu 9, un otrādi.



Skaidrs, ka apbrīnojamas īpašības piemīt ne vien atsevišķiem burtiem, bet arī veseliem vārdiem. Tā, vārds "OTO" nemainās, ja to atspoguļo pret vertikālu spoguļi, vārds "DEKO" nemainās, ja to atspoguļo pret horizontālu spoguļi, bet "6SOS9" nemainās, ja to pagriež par 180 grādiem. Vienam un tam pašam vārdam var piemist arī vairākas no nosauktajām īpašībām vienlaikus. Tā, vārdam "OHO" piemīt visas trīs iepriekš aplūkotās īpašības.

Uzrakstiet programmu, kas ievadītam vārdam nosaka, vai šis vārds nemainās, ja to atspoguļo pret vertikālu vai horizontālu spoguļi, vai apgriež otrādi!

Ievaddati

Faila `vards.dat` vienīgajā rindā dots vārds, kas sastāv no latīņu alfabēta lielajiem burtiem un cipariem. Zināms, ka vārdā ir vismaz viens un ne vairāk kā 250 simboli.

Izvaddati

Teksta faila `vards.rez` vienīgajai rindai jāsaturs tieši trīs simboli bez atdalošiem tukšumiem. Pirmajam simbolam jābūt "+", ja, atspoguļojot šo vārdu pret vertikālu spoguļi, vārds nemainās, vai "-", ja mainās. Otrajam simbolam jābūt "+", ja, atspoguļojot šo vārdu pret horizontālu spoguļi, vārds nemainās, vai "-", ja mainās. Trešajam simbolam jābūt "+", ja, pagriežot šo vārdu par 180 grādiem, vārds nemainās, vai "-", ja mainās.

Piemēri

Ievaddati (<code>vards.dat</code>)	Izvaddati (<code>vards.rez</code>)
OHO8OHO	+++
Ievaddati (<code>vards.dat</code>)	Izvaddati (<code>vards.rez</code>)
RUKSIS	---
Ievaddati (<code>vards.dat</code>)	Izvaddati (<code>vards.rez</code>)
OZO99N66OZO	---+
Ievaddati (<code>vards.dat</code>)	Izvaddati (<code>vards.rez</code>)
TOMATO69	---

3."APAKŠKOPAS"

Aplūkosim kopu, kurā ir visi naturālie skaitļi no 1 līdz N.

Ja $N=3$, šo kopu iespējams tikai vienā veidā sadalīt divās apakškopās tā, ka skaitļu summa abās apakškopās ir vienāda: $\{1,2\}$ un $\{3\}$.

Arī, ja $N=4$, to arī iespējams izdarīt tikai vienā veidā: $\{1,4\}$ un $\{2,3\}$.

Ja $N=5$ vai $N=6$, to izdarīt nav iespējams, jo kopējā visu elementu summa ir nepāra skaitlis.

Ja $N=7$, iespējami četri dažādi dalījuma veidi divās apakškopās ar vienādu elementu summu: $\{2,3,4,5\}$ un $\{1,6,7\}$, $\{1,3,4,6\}$ un $\{2,5,7\}$, $\{1,2,5,6\}$ un $\{3,4,7\}$, $\{3,5,6\}$ un $\{1,2,4,7\}$.

Ievadītam naturālam skaitlim N nosakiet, cik dažādos veidos skaitļu kopu iespējams sadalīt divās apakškopās ar vienādu elementu summu!

Ievaddati

Teksta faila `kopas.dat` vienīgajā rindā dota skaitļa N ($1 \leq N \leq 74$) vērtība.

Izvaddati

Teksta faila `kopas.rez` vienīgajā rindā jāizvada vesels nenegatīvs skaitlis - dažādo veidu skaits, kādā iespējams sadalīt kopu.

Piemēri

Ievaddati (<code>kopas.dat</code>)	Izvaddati (<code>kopas.rez</code>)
3	1

Ievaddati (<code>kopas.dat</code>)	Izvaddati (<code>kopas.rez</code>)
5	0

Ievaddati (<code>kopas.dat</code>)	Izvaddati (<code>kopas.rez</code>)
7	4

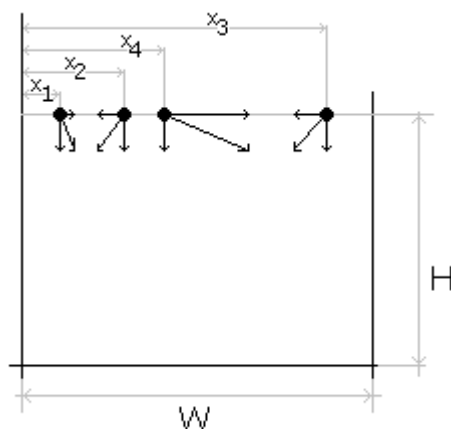
Ievaddati (<code>kopas.dat</code>)	Izvaddati (<code>kopas.rez</code>)
71	2681644149792639400

4."BUMBAS"

Kad jauniešiem nav ko darīt, viņi izgudro sev jaunas spēlītes. Tā tika izdomāta jauna spēle "BUMBAS". Spēles noteikumi ir šādi:

Vienā rindā tiek saliktas un vienlaicīgi no viena augstuma tiek mestas N bumbas tā, ka tās kustas ar savstarpēji vienādu vertikālu ātrumu v_y uz leju, un ar kādu ātrumu v_{xi} horizontālā virzienā (pa kreisi vai pa labi). Ja krītot divas bumbas vienlaicīgi nonāk vienā punktā, tad tās atsitas viena no otras un maina savu kustības virzienu tā, ka pirmā bumba lido tur, kur bija jālido otrai, bet otrā tur, kur bija jālido pirmajai. Telpā ir arī divas vertikālas sienas, tāpēc bumbas var atsisties arī no sienām (atcerieties, ka atsišanās leņķis ir vienāds ar krišanas leņķi). Bet, kad galu galā bumbas nonāk līdz grīdai, tās apstājas.

Uzrakstiet programmu, kura pēc dotajiem sākuma datiem pēc iespējas precīzi nosaka bumbu secību pēc visu bumbu apstāšanās!



Ievaddati

Faila `bumbas.dat` pirmajā rindā doti trīs reāli skaitļi, kas atdalīti ar tukšumsimboliem: H , W , v_y , kur H – telpas augstums ($0 < H \leq 10^5$), W – telpas platums ($0 < W \leq 10^5$), v_y – ātrums, ar kādu visas bumbas krīt uz leju.

Faila otrajā rindā dots naturāls skaitlis $N(0 < N \leq 10^5)$ – bumbu skaits.

Katrā no nākošajām N faila rindām doti divi reāli skaitļi – vienas bumbas ātrums horizontālā virzienā v_{xi} ($-10^5 \leq v_{xi} \leq 10^5$), ja vērtība ir negatīva, tad bumba kustas pa kreisi, bet, ja pozitīva, tad pa labi), un x_i – sākotnējais bumbas attālums līdz kreisajai sienai ($0 \leq x_i \leq W$). i -tās bumbas apraksts ($1 \leq i \leq N$) dots faila $i+2$ -ajā rindā.

Izvaddati

Failam `bumbas.rez` jāsaturs N rindas, katra no rindām ir jābūt reālam skaitlim ar piecām zīmēm aiz decimālā punkta. Faila i -tajā rindā jābūt i -tās bumbas attālumam no kreisās sienas.

Piezīmes

- Nekādas trīs bumbas vienā laika momentā nenonāk vienā punktā.
- Nekādas divas bumbas vienā laika momentā nenonāk vienā sienas punktā.

Piemērs

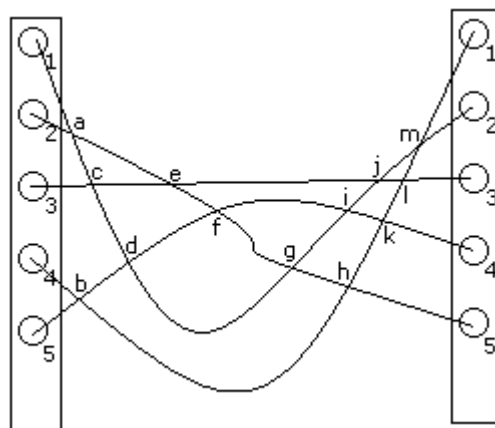
Ievaddati(<code>bumbas.dat</code>)	Izvaddati(<code>bumbas.rez</code>)
4.0 5.0 1.0	4.00000
2	1.00000
-1.0 3.0	
1.0 2.0	

5."KABEĻI"

Mūsdienās dažādas iekārtas tiek savienotas ar kabeļu palīdzību. Tā kā kabeļi ir vienādi pelēki un laika gaitā tiek gan pieslēgti, gan atslēgti, gan no jauna pieslēgti, tad beigu beigās kabeļi var veidot visai nopietnu mudžekli. Pēc īsāka vai garāka laika sprīža vairs neviens nevar saprast, kurš kabelis uz kuriem aiziet, un galvenais tādos brīžos ir neaiztikt strādājošu sistēmu.

Ir dotas divas iekārtas, katrai no kurām ir N pieslēgvietas, kas izvietotas vertikāli vienā rindā.

Pieslēgvietas ir sanumurētas no augšas uz leju pēc kārtas ar skaitļiem no 1 līdz N .



Vienas iekārtas katra pieslēgvietā ar kabeļu palīdzību ir savienota ar otras iekārtas kādu pieslēgvietu (skat.zīmējumu). Nepieciešams noskaidrot, kura vienas iekārtas pieslēgvietā ir savienota ar kuru. Lai to noskaidrotu, asistente Gunta noliek visus kabeļus vienā plaknē un pieraksta pēc kārtas visus kabeļu krustošanās punktus (zīmējumā šie punkti apzīmēti ar burtiem alfabētiskā secībā). Tā kā kabeļi ir vienādi, tad Gunta var pierakstīt tikai, kuri divi blakus kabeļi pēc kārtas, skatoties no augšas uz leju, šajā vietā krustojas. Piemēram, punktā "d" krustojas trešais un ceturtais kabelis. Par laimi, nevienā punktā vienlaicīgi nekrustojas vairāk par diviem kabeļiem.

Uzrakstiet programmu, kas dotām kabeļu krustošanās vietām nosaka, kuras pieslēgvietas savā starpā ir savienotas!

Ievaddati

Teksta faila `kabeļi.dat` pirmajā rindā doti divi veseli skaitļi N (pieslēgvietu skaits, $1 \leq N \leq 10000$) un K (krustošanās vietu skaits, $0 \leq K \leq 100000$), kas atdalīti ar tukšumsimbolu.

Katrā no nākošajām K faila rindām dots naturāls skaitlis n_i ($1 \leq n_i \leq N-1$), kas norāda, ka šajā punktā krustojas n_i -tais un n_i+1 -ais kabelis (skaitot no augšas). i -tais krustpunkts ir aprakstīts faila $i+1$ -ajā rindā.

Izvaddati

Teksta failam `kabeļi rez` jāsaturs tieši N rindas. Faila i -tajā rindā jāizvada viens naturāls skaitlis - tās otrās iekārtas pieslēgvietas numurs, ar kuru ir savienota pirmās iekārtas i -tā pieslēgvietā.

Piemērs

Ievaddati (<code>kabeļi.dat</code>)	Izvaddati (<code>kabeļi rez</code>)
5 13	2
1	5
4	3
2	1
3	4
1	
2	
3	
4	
2	
1	
3	
2	
1	

6. "ČETRSTŪRIS"

Uzrakstiet programmu, kas dotiem četriem plaknes punktiem nosaka, kāda veida četrstūri iegūst, šos punktus dotajā secībā (pirmo ar otro, otro ar trešo, trešo ar ceturto un ceturto ar pirmo) savienojot savā starpā!

Ievaddati

Teksta failā `cetrst.dat` ir četras rindas. Katrā rindā doti divi veseli skaitļi x_i un y_i ($|x_i| \leq 10000, |y_i| \leq 10000$), kas atdalīti ar tukšumsimbolu - vienas virsotnes koordinātas. Faila i-tajā rindā dotas i-tās virsotnes koordinātas.

Izvaddati

Teksta faila `cetrst.rez` vienīgajā rindā jāizvada viens burts:

- "K", ja iegūtā figūra ir kvadrāts,
- "T", ja iegūtā figūra ir taisnstūris,
- "R", ja iegūtā figūra ir rombs,
- "P", ja iegūtā figūra ir paralelograms,
- "A", ja iegūtā figūra ir trapece,
- "C", ja iegūtā figūra ir cits četrstūris,
- "N", ja iegūtā figūra nav četrstūris.

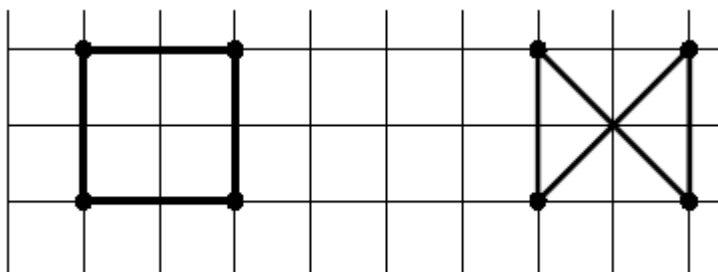
Jāizvada ir **tikai viens** burts (informācija par pirmo derīgo četrstūra veidu no dotā saraksta). Piemēram, ja dotie punkti veido kvadrātu, tad jāizvada ir tikai "K" (lai gan kvadrāts, protams, ir arī rombs, taisnstūris un paralelograms).

Piemēri

Ievaddati (<code>cetrst.dat</code>)	Izvaddati (<code>cetrst.rez</code>)
1 1 1 3 3 3 3 1	K

Ievaddati (<code>cetrst.dat</code>)	Izvaddati (<code>cetrst.rez</code>)
7 1 7 3 9 1 9 3	N

Dotajiem piemēriem atbilstošs zīmējums:



7."CEĻI"

Kādā valstī ir N pilsētas, un tās savā starpā savieno M ceļi. Katrs ceļš savieno tieši divas pilsētas, un ceļi ārpus pilsētām nekrustojas.

Uzrakstiet programmu, kas katrai pilsētai aprēķina attālumu līdz tuvākajai un tālākajai no pārējām pilsētām! Uzskatīsim, ka pārvietoties drīkst tikai pa ceļiem un attālums starp pilsētām ir īsākais iespējamais ceļu kopgarums.

Ievaddati

Teksta faila `celi.dat` pirmajā rindā dotas divu naturālu skaitļu $N(2 \leq N \leq 1000)$ un $M(1 \leq M \leq 10000)$ vērtības, kas atdalītas ar tukšumsimbolu. Visas pilsētas ir sanumurētas ar naturāliem skaitļiem no 1 līdz N pēc kārtas. Katrā no nākošajām M rindām doti trīs veseli skaitļi formātā $p_1 p_2 d$ ($1 \leq p_1, p_2 \leq N$, $p_1 \neq p_2$, $1 \leq d \leq 1000$) un nozīmē, ka dotā ceļa garums ir d kilometri un tas savieno pilsētas p_1 un p_2 . Starp katriem diviem blakus skaitļiem ir viens tukšumsimbols. Zināms, ka no katras pilsētas ir iespējams aizbraukt uz jebkuru citu.

Izvaddati

Teksta failam `celi rez` jāsaturs tieši N rindas. Katrā rindā jāizvada divi naturāli skaitļi, kas atdalīti ar tukšumsimbolu. Faila i -tajā rindā jāizvada attālums no pilsētas i līdz tuvākajai un līdz tālākajai kaimiņu pilsētai.

Piemērs

Ievaddati (<code>celi.dat</code>)	Izvaddati (<code>celi rez</code>)
4 4	10 20
1 3 10	10 22
1 4 15	10 12
2 3 10	12 22
3 4 12	