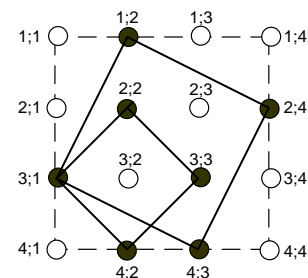


LD_1. Kvadratai kvadratuose.

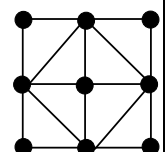
$N \times N$ ($2 \leq N \leq 5$) tuščiaidurių ir pilnavidurių skrituliukų sudaro kvadratą. Parenkite programą, išvedančią kvadratą, gautų sujungus pilnavidurius skrituliukus, viršūnių koordinatas ir skaičiuojančią, kelių kvadratus galima sudaryti. Koordinatų pradžia (1; 1) – kairioji viršutinė didžiojo kvadrato viršūnė.

Pradiniai duomenys saugomi tekstiniame faile U3.txt. Pirmoje eilutėje įrašyta N reikšmė. Tolesnėse N eilučių įrašyta po N reikšmių eilutėje: nulis atitinka tuščiaidurių skrituliuką, vienetą – pilnavidurių. Reikšmės viena nuo kitos skiriamos tarpais.

Rezultatai išvedami į ekraną. Pirmiausia išvedamas susidarančių kvadratų skaičius. Jei nesusidaro nė vienas kvadratas, tuomet į ekraną išvedamas nulis. Po to išvedami kvadratai ir jų viršūnės. Vieno kvadrato viršūnėms skiriama viena eilutė. Kvadrato viršūnės turi būti išdėstytos eilučių (x) didėjimo tvarka. Jei eilutės sutampa, tuomet stulpelių (y) didėjimo tvarka. Kvadratai išvedami viršūnių koordinatų didėjimo tvarka. Jei dviejų ar daugiau kvadratų pirmosios viršūnės sutampa, tuomet lyginamos antrosios kiekvieno kvadrato viršūnės.



U3.txt	Ekranas	U3.txt	Ekranas
4	2	3	6
0 1 0 0	(1;2) ir (2;4) ir (3;1) ir	1 1 1	(1;1) ir (1;2) ir (2;1) ir
0 1 0 1	(4;3)	1 1 1	(2;2)
1 0 1 0	(2;2) ir (3;1) ir (3;3) ir	1 1 1	(1;1) ir (1;3) ir (3;1) ir
0 1 1 0	(4;2)		(3;3)
	Pavyzdys atitinka paveikslėlį šalia		(1;2) ir (1;3) ir (2;2) ir
	uždavinio sąlygos.		(2;3)
			(1;2) ir (2;1) ir (2;3) ir (3;2)
			(2;1) ir (2;2) ir (3;1) ir (3;2)
			(2;2) ir (2;3) ir (3;2) ir (3;3)



LD_2. Kryžiažodis.

Duota k žodžių, kurie surašyti n x m išmatavimų lentelėje – kryžiažodyje.

Parašykite programą, kuri padėtų kryžiažodžių sprendimo mėgėjui išspręsti kryžiažodį – surastų lentelėje duotus žodžius, vietoj kiekvieno surasto žodžio raidžių įrašytų žodžio eilės numerį. Ekране pateikite kontrolines sumas: stulpeliuose surašytų skaičių sumas.

Duomenys įvedami iš tekstinio failo 'Kur3.txt'. Duomenų formatą studentas pasirenka ir suderina su savo dėstytoju. Rezultatai išvedami į ekraną. Rezultatuose pageidautinos spalvos.

L	A	J	I	N	I	K	S	A	1. KABINETAS
A	K	A	M	I	N	U	M	R	2. LAIVELIS
I	S	B	I	S	A	C	E	G	3. GIMINAITIS
V	I	I	G	A	I	S	I	I	4. CUKINIJA
E	L	N	E	T	T	I	K	S	5. KIEMSARGIS

Atsakymas:

2	4	4	4	4	4	4	5	5	1. KABINETAS
2	1	1	3	3	3	4	5	5	2. LAIVELIS
2	2	1	3	1	3	4	5	5	3. GIMINAITIS
2	2	1	3	1	3	3	5	5	4. CUKINIJA
2	2	1	1	1	3	3	5	5	5. KIEMSARGIS
10	11	8	14	10	16	18	25	25	Kontrolinės sumos teisingumui patikrinti

LD_3. Domino.

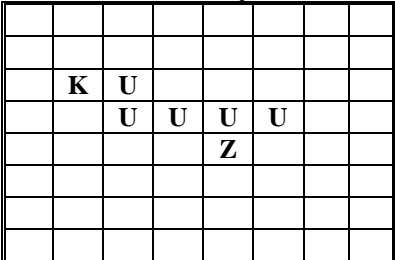
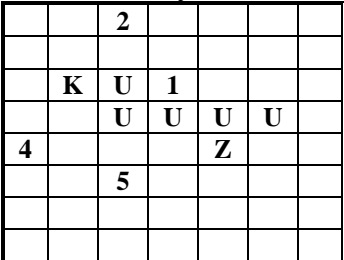
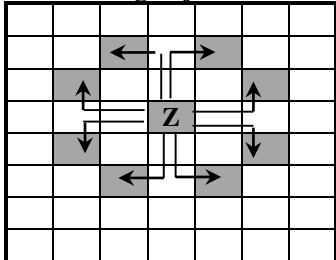
Imami 7 vieno domino rinkinio kauliukai. Vieną domino kauliuką sudaro dvi dalys, kurių kiekvienoje arba nieko nėra (baltas), arba juodi taškai, kurių yra nuo 1 iki 6. Kauliuką galima nusakyti kaip dviženklį skaičių, kurio pirmas skaitmuo nurodo pirmos dalies taškų skaičių, o antrasis – antros. Jeigu dalis tuščia, tai rašomas skaitmuo 0 (nulis). Parašykite programą, kuri sudarytų iš šių 7 kauliukų visas galimas grandines, kai jungiami kauliukai galais su vienodu taškų skaičiumi. Gali būti, kad tokios grandinės visai nėra. Sudarant grandines, kauliukas gali būti apskamas, t.y. kauliukas 35 gali būti padėtas, kaip 53.

Kauliukų duomenys įvedami iš tekstinio failo 'Kur3.txt'. Čia vienoje eilutėje yra parašyti 7 (septyni) dviženkliai skaičiai. Rezultatus surašyti į tekstinį failą eilutėmis po vieną grandinę. Grandinę sudaro 7 kauliukai, tarp kiekvieno kauliuko (dviženkliai skaičiai) paliekamas vieno tarpo ženklas.

Kur3.txt	Rezultatas
13 01 02 24 14 12 25	31 10 02 24 41 12 25 52 21 14 42 20 01 13 31 12 24 41 10 02 25 52 20 01 14 42 21 13 ir t.t.

LD_4. Atilos žirgas.

Šachmatų lentoje stovi baltasis žirgas ir juodasis karalius. Kai kurie lentos laukai yra užimti. Žirgas privalo pasiekti karalių, jį kirsti ir sugrįžti į pradinę padėtį. Kaip žirgas gali eiti parodyta pavyzdyje. Jis negali eiti į užimtus langelius, o taip pat į jau vaikščiotus langelius Pavyzdžiui, šachmatų lenta, kur K – karalius, Z – žirgas, U – užimtas langelis. Atsakymo dalyje skaičiais pažymėti žirgo ėjimai, 3 nėra, nes tas ėjimas kerta karalių, ir nėra 6, nes tuo ėjimu žirgas grįžta į pradinę padėtį.

Duomenys	Atsakymas	Žirgo ėjimai
		

Duomenys. Tekstiniame faile **U3.txt** yra 8x8 lentelė, kuri užpildyta simboliais: 0 – laisvas langelis, K – karalius, Z – žirgas, U – užimtas langelis. Simboliai skiriami vienu tarpu.

Rezultatai. Ekrane spausdinkite atsakymo lentelę, kurioje sunumeruokite žirgo ėjimus, kaip parodyta pavyzdyje. Kad lentelė būtų lengviau skaitoma vietoje nulių spausdinkite brūkšnelį. Stulpeliai vienas nuo kito skiriami vienu tarpu. Sunumeruokite stulpelius bei eilutes.

U3.txt	Rezultatas
0 0 0 0 0 0 0 0	1 2 3 4 5 6 7 8
0 0 0 0 0 0 0 0	1 - - 2 - - - -
0 K U 0 0 0 0	2 - - - - - - -
0 0 U U U U 0 0	3 - K U 1 - - -
0 0 0 0 Z 0 0 0	4 - - U U U U -
0 0 0 0 0 0 0 0	5 4 - - - Z - -
0 0 0 0 0 0 0 0	6 - - 5 - - - -
0 0 0 0 0 0 0 0	7 - - - - - - -
0 0 0 0 0 0 0 0	8 - - - - - - -

LD_5. Žodžių paieška.

Parašykite programą, kuri perskaitytų iš tekstinio failo "Trecias.txt" tekstą po vieną simbolį ($1 < \text{simbolių kiekis} \leq 2000$) ir jais užpildytų masyvą $A[n, n]$. Į masyvą nerašomi eilutės pabaigos, naujos eilutės ir failo pabaigos simboliai. n parenkamas toks mažiausias, kad tekstas tilptų į kvadratinę matricą. Jei paskutinei eilutei trūksta simbolių, užpildote tarpais. Po to programa iš tekstinio failo "Zodziai.txt", kuriame kiekvienas žodis yra atskiroje eilutėje ir prasideda nuo 1 pozicijos, perskaito eilinių žodį, kurio ilgis $k \leq n/2$. Reikia nustatyti, kiek kartų šis žodis kartojasi lentelėje $A[n, n]$. Paieška turėtų būti atliekama horizontaliai iš kairės į dešinę, vertikalčiai iš viršaus žemyn ir pagal dešinę diagonalę žemyn ir į dešinę (ne tik pagrindinė diagonalė). Žodžiai iš eilutės (stulpelio) į eilutę (stulpelį) nekeliami. Žodžiai nepersidengia, bet trumpesnis žodis gali būti ilgesniojo žodžio dalimi.

Ekrane atspausdinkite parinktą n reikšmę ir kiekvieno žodžio pasikartojimų skaičių.

Trecias.txt	Matrica
Berzas,sula;;sula;;klevu saldial lapasula	B e r z a s , s u
a aula, ar suart zemes vaikai du	l a ; ; s u l a ;
Zodziai.txt	; k l e v u s a
Sula	l d i a l l a p
Alus	a s u l a a
Atsakymas	a u l a , a
n = 9	r s u a r t
sula 3	z e m e s v a i
alus 2	k a i d u

LD_6. Apskritimas.

Turime kvadrato formos languoto popieriaus lapą. Į lapo centre esantį langelių linijų susikirtimą statoma skriestuvo kojėlė. Brėžiamas apskritimas, siekiantis lapo kraštus. Sunumeruokite, pradedant nuo 1, languoto popieriaus lapo langelius, pilnai patenkančius į apskritimo vidų, o kitiems langeliams priskirkite po 0. Numeravimo tvarka nėra svarbi. Atvaizduokite languoto popieriaus lapą matrica.

Spindulio r ($r \in \mathbb{N}$) dydis nurodomas langelių kiekiu. Pavyzdžiui, kai r yra 5, matrica gali būti užpildoma taip:

0	0	0	0	0	0	0	0	0	0
0	0	25	26	27	28	29	30	0	0
0	17	18	19	20	21	22	23	24	0
0	9	10	11	12	13	14	15	16	0
0	1	2	3	4	5	6	7	8	0
0	31	32	33	34	35	36	37	38	0
0	39	40	41	42	43	44	45	46	0
0	47	48	49	50	51	52	53	54	0
0	0	55	56	57	58	59	60	0	0
0	0	0	0	0	0	0	0	0	0

Duomenys. r įvedamas klaviatūra ($1 \leq r \leq 16$).

Rezultatai. Gautą matricą spausdinkite ekrane.

LD_7. Cilindras.

Matrica $A(n, m)$ vaizduoja languoto popieriaus lapą. Iš lapo suklijavom n aukščio cilindą. Cilindro klijavimui lapo langelių nepanaudojome. Į kiek gabaliukų subyrės cilindras, jei iš jo iškirpsime kai kuriuos langelius. Iškerpami langeliai matricoje žymimi 0.

Duomenys. Tekstinio failo 'U3.txt' pirmoje eilutėje nurodyti dydžiai n ($2 \leq n \leq 30$) ir m ($2 \leq m \leq 30$), tolimesnėse eilutėse išdėstytos tvarkingai, skiriant vienu tarpu, matricos $A(n, m)$ reikšmės.

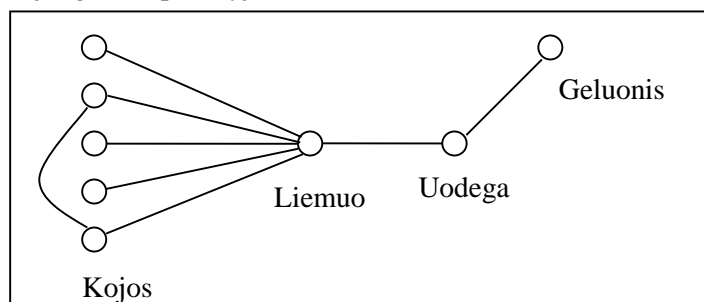
Rezultatai. Į ekraną išveskite pradinius duomenis. Žemiau išveskite skaičių, į kiek gabaliukų subyrės cilindras.

U3.txt	Ekranas
4 5 1 1 0 1 1 1 1 0 1 0 1 0 1 0 1 1 1 0 1 1	Pradiniai duomenys: 1 1 0 1 1 1 1 0 1 0 1 0 1 0 1 1 1 0 1 1 2 gabaliukai

LD_8. Skorpionas.

Grafas – tai viršūnių ir jas jungiančių briaunų visuma. Tarp dviejų viršūnių gali būti tik viena briauna. Brėžiniuose grafo viršūnės dažnai vaizduojamos mažais apskritimais, o briaunos – linijomis, jungiančiomis šiuos apskritimus.

Grafas, turintis n viršūnių yra „skorpionas“, jei yra viena viršūnė (geluonis), sujungta viena briauna su kita viršūne (uodega). Uodega dar viena briauna turi būti sujungta su trečia viršūne (liemuo). Viršūnė-liemuo jungiama su likusiomis viršūnėmis (kojomis) atskiromis briaunomis. Kai kurios viršūnės-kojos gali būti sujungtos tarpusavyje.



Duomenys surašyti tekstiniam failui 'U3.txt'. Pirmoje failo eilutėje yra parašytas sveikasis skaičius n ($5 \leq n \leq 50$). n nurodo grafo viršūnių skaičių. Toliau eilutėmis, kurių kiekvienoje yra n simbolių, užrašyta grafo matrica $V(n, n)$. $V[i, j] = '-'$, jei tarp i -osios ir j -osios viršūnių nėra briaunos ir $V[i, j] = '+'$, jei tarp i -osios ir j -osios viršūnių yra briauna. $V[i, i] = '*'$.

Rezultatai. Į ekraną išvesti pranešimą, ar įvestą matricą atitinkantis grafas yra „skorpionas“, ar ne.

Jei taip, nurodyti, kuri viršūnė yra „geluonis“, kuri „uodega“, kuri „liemuo“, kurios viršūnės yra „kojos“.

U3.txt	Rezultatų ekrane pavyzdys
5 *+--+ +*--- --*+- +-+*+ ---+*	Grafas yra "skorpionas" Geluonis: 2 virsune Uodega: 1 virsune Liemuo: 4 virsune 1 koja: 3 virsune 2 koja: 5 virsune

LD_9. Kurjeris.

Studentas įsidarbino kurjeriu firmoje, turinčioje n punktų, į kuriuos jis laiku privalo pristatyti gaminius. Kurjerio algos dydis yra pastovus ir nepriklauso nuo kelionės trukmės. Norėdamas įvertinti bendrą kelionių kainą, studentas apskaičiavo kelionės kainą tarp bet kurių dviejų punktų. Užduotis – surasti tokį kelionės maršrutą, kuris praeitų tik po vieną kartą pro kiekvieną punktą ir kurio bendra punktų apvažiavimo kaina būtų mažiausia, t.y. kad studentui daugiau liktų pelno. Studento kelionės pradžia visą laiką yra pirmasis punktas. Ten jis ir turi grįžti.

Duomenys surašyti tekstiniam failui 'U3.txt'. Pirmoje failo eilutėje parašytas sveikasis skaičius n ($5 \leq n \leq 50$), nurodantis kvadratinės matricos dydį. Toliau eilutėmis užrašyta matrica $K(n, n)$, kurioje išvardinta kelionių kaina tarp atskirų punktų, kur $K[i, j] = K[j, i]$ ir reiškia kelionės kaštus tarp punktų i ir j . **Rezultatus** išvesti į ekraną, kuriame būtų išvardinti punktai jų aplankymo eilės tvarka ir kelionės kaina.

U3.txt	Rezultatų pavyzdys
5 0 1 3 4 2 1 0 4 2 6 3 4 0 7 1 4 2 7 0 7 2 6 1 7 0	1, 5, 3, 2, 4, 1 = 13

LD_10. Kurmiai.

Pavasariį sode apsigyveno kurmiai. Kiekvienas kurtis išsirausė sau atskirą urvą. Suskaičiuokite, kiek kurmių apsigyveno sode ir koks yra kiekvieno kurmio išrausto urvo dydis.

Duomenys. Failu U3.txt yra pateikta sodo plokštuminė kurmių urvų schema – atvaizduota dvimačiu simbolių masyvu. Pirmoje failo eilutėje yra užrašytas schemos dydis: eilučių skaičius n ($5 \leq n \leq 500$) ir stulpelių skaičius m ($5 \leq m \leq 500$). Tolesnėse n eilučių yra užrašyta po m simbolių: 'z' (žemė) arba 'u' (urvas). Vienas simbolis atitinka 5 cm^2 plotą. Du urvo simboliai ('u') priklauso tam pačiam urvui, jeigu jie yra greta toje pačioje eilutėje arba greta tame pačiame stulpelyje.

Rezultatai. Ekrane atskirose eilutėse spausdinkite sode apsigyvenusių kurmių skaičių ir kurmių urvų dydžius (cm^2) surikiuotus mažėjimo tvarka.

U3.txt	Ekranas	Paaškinimai
6 15 zzzzzzzzzzuuzzz uuuuuuuuuuuuuu zuuuzuuuuuuuu zuuuzuuuuuuuu zuuuzuuuuuuuu zuuuzuuuuuuuu zzzzzzuuuuuuuu	2 110 70	2 kurmiai $22 \times 5 \text{ cm}^2 = 110$ $14 \times 5 \text{ cm}^2 = 70$

LD_11. Pilis.

Pilis yra suskirstyta į $M \times N$ ($M \leq 50$, $N \leq 50$) kvadratinį celių. Kiekviena celė gali turėti nuo 0 iki 4 sienų. Reikia surasti, kiek pilyje yra kambarių. Kambarys – tai erdvė, sudaryta iš vienos ar daugiau celių ir apribota iš visų keturių pusių sienomis.

Duomenys. Pilies planas yra tekstiniam failui 'U3.txt'. Pirmasis skaičius – tai celių skaičius šiaurės-pietų kryptimi, antrasis skaičius – celių skaičius vakarų-rytų kryptimi, tolesnėse eilutėse kiekvienai celei pažymėti yra skirtas skaičius p ($0 \leq p \leq 15$). Ši skaičių sudaro tokių skaičių suma: 1 (jei celė turi vakarinę sieną), 2 (šiaurinę sieną), 4 (rytinę sieną), 8 (pietinę sieną). Vidinė siena priklauso abiem celėms. Piešinyje yra pilies planas pagal duomenų failo duomenis:

	1	2	3	4	5	6	7
1							
2							

3							
4							

Rezultatai. Ekrane spausdinkite pradinis duomenis ir kambarių kiekį.

U3.txt	Rezultatas
4	11 6 11 6 3 10 6
7	7 9 6 13 5 15 5
11 6 11 6 3 10 6	1 10 12 7 13 7 5
7 9 6 13 5 15 5	13 11 10 8 10 12 13
1 10 12 7 13 7 5	
13 11 10 8 10 12 13	Kambarių kiekis 5

LD_12. Žaidimas „Sudoku 6×6“.

Duota dalinai užpildyta lentelė su skaičiais nuo 1 iki 6. Lentelėje gali būti n ($1 \leq n \leq 9$) neužpildytų (tuščių) langelių. Žaidimo tikslas – pilnai užpildyti lentelę taip, kad bet kurioje eilutėje, stulpelyje ir bet kuriame stačiakampyje, įrėmintame dviguba linija, skaičiai nuo 1 iki 6 nesikartotų. Parašykite programą, kuri surastų trūkstamus lentelės skaičius.

Duomenys. Faile U3.txt yra surašyti žaidimo lentelės skaičiai – 6 eilutės. Kiekvienoje eilutėje po 6 skaičius. Tose vietose, kur lentelės langelis yra tuščias, faile yra užrašyta 0 (nulis).

Rezultatai. Ekrane spausdinkite pilnai užpildytos lentelės skaičius. Jeigu egzistuoja keli teisingi lentelės užpildymo variantai spausdinkite tik vieną variantą (Pastaba: visuomet bus bent vienas lentelės užpildymo variantas).

U3.txt	Ekranas	Duomenų pavyzdžio iliustracija																																				
0 0 2 4 5 3 4 3 5 0 1 2 5 2 6 1 3 0 3 0 1 2 6 5 1 5 0 3 0 6 2 6 0 5 4 1	6 1 2 4 5 3 4 3 5 6 1 2 5 2 6 1 3 4 3 4 1 2 6 5 1 5 4 3 2 6 2 6 3 5 4 1	<table><tr><td></td><td></td><td>2</td><td>4</td><td>5</td><td>3</td></tr><tr><td>4</td><td>3</td><td>5</td><td></td><td>1</td><td>2</td></tr><tr><td>5</td><td>2</td><td>6</td><td>1</td><td>3</td><td></td></tr><tr><td>3</td><td></td><td>1</td><td>2</td><td>6</td><td>5</td></tr><tr><td>1</td><td>5</td><td></td><td>3</td><td></td><td>6</td></tr><tr><td>2</td><td>6</td><td></td><td>5</td><td>4</td><td>1</td></tr></table>			2	4	5	3	4	3	5		1	2	5	2	6	1	3		3		1	2	6	5	1	5		3		6	2	6		5	4	1
		2	4	5	3																																	
4	3	5		1	2																																	
5	2	6	1	3																																		
3		1	2	6	5																																	
1	5		3		6																																	
2	6		5	4	1																																	

LD_13. Sniego valymas.

Sniegas užklojo miesto gatves. Jas reikia valyti. Mieste yra N mazgų ir M vienos krypties kelių. Kiekvienas kelias jungia du skirtingus mazgus x_i ir y_i , kryptis yra iš x_i į y_i . Ant i -tojo kelio yra w_i tonų sniego. Kiekvieną dieną valymo mašina startuoja iš mazgo A, važiuoja iki mazgo B pagal kelių kryptis ir sustoja. Bet kurį mazgą (A ir B taip pat) galima pravažiuoti per dieną daug kartų. Vienu pravažiuoimu nuvaloma viena tono sniego. Ten, kur sniego nėra, mašina važiuoti negali. Kai kurie keliai turi istorinę vertę, todėl juos nuvalyti būtina. Be to, iš mazgo A galima pasiekti bet kurį istorinės vertės kelią, važiuojant tik istorinės vertės keliais. Parašykite programą, kuri nurodytų, kokia tvarka reiktų valyti kelius, kad valymas vyktų kuo daugiau dienų ir būtų nuvalyti visi istorinės vertės keliai.

Duomenys. Tekstinio failo 'U3.txt' pirmoje eilutėje nurodytas N ($2 \leq N \leq 100$) – mazgų kiekis, M ($0 \leq M \leq 1000$) – kelių kiekis, mazgai A ir B ($1 \leq A, B \leq N$; $A \neq B$). Tolesnėse M eilučių įvardinti vienos krypties keliai 4 sveikaisiais skaičiais x_i (pradžios mazgas), y_i (pabaigos mazgas), w_i (sniego tonos), t_i (1 – istorinis kelias, 0 – ne) ($1 \leq x_i, y_i \leq N$, $x_i \neq y_i$, $0 \leq w_i \leq 100$, $0 \leq t_i \leq 1$).

Rezultatai. Į ekraną pirmoje eilutėje išveskite maksimalų dienų skaičių, o tolesnėse eilutėse – kiekvienos dienos kelius, išvardijant pravažiuojamus mazgus pagal kelio kryptį.

U3.txt	Ekranas
4 7 1 4 1 2 3 1 2 1 100 0 2 4 1 0 1 3 1 0 3 4 4 0 2 3 2 1 1 4 2 0	6 1 3 4 1 4 1 4 1 2 4 1 2 3 4 1 2 3 4
3 3 1 2 1 3 2 0	3 1 3 2

3 2 3 0	1 3 2
1 2 1 0	1 2

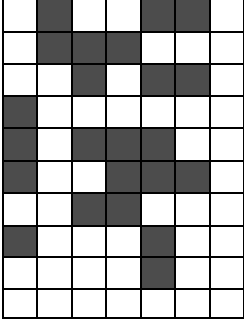
LD_14. Salos.

Robotas iš didelio aukščio nufotografavo stačiakampį Žemės paviršiaus plotą. Gautas vaizdas buvo sudalintas kvadratiniais langeliais $n \times m$ ($1 \leq n, m \leq 100$). Langelis tamsus, jeigu jis atitinka sausumos dalelę ir šviesus, jeigu atitinka vandens plotelį.

Parašykite programą, kuri suskaičiuotų, kiek yra salų ir kokio dydžio didžiausia sala. Sala – tai gretimų tamsių langelių visuma, iš visų pusių apribota šviesiais. Sala taip pat laikoma tamsių langelių visuma, esanti ant nufotografuoto ploto ribos (ne iš visų pusių supa vanduo).

Programa turi analizuoti k ($1 \leq k \leq 10$) tokių žemės plotų nuotraukų.

Duomenys surašyti tekstiname faile **U3.txt**, kur pirmoje eilutėje yra k . Toliau – kiekvieno stačiakampio duomenys: pirmoje stačiakampio aprašo eilutėje yra n ir m reikšmės, o kitose eilutėse – stačiakampio langelių reikšmės eilutėmis. Tamsų langelį atitinka simbolis '1', o šviesų '0'.

Fotografijos pavyzdys	U3.txt	Ekrane
Fotografija 10 x 7 prieš ją koduojant skaitmenimis (faile trečia koduotė) 	3	2 2
	2 6	3 8
	110100	4 10
	000000	
	9 4	
	1000	
	1000	
	0110	
	1100	
	1100	
	0000	
	1010	
	1000	
	0000	
	10 7	
	0100110	
	0111000	
	0010110	
	1000000	
	1011100	
	1001110	
	0011000	
	1000100	
	0000100	
	0000000	

Rezultatai. Ekrane spausdinti kiekvienoje eilutėje po du skaičius: kiekvieno paviršiaus ploto salų skaičių ir didžiausios salos langelių skaičių.

LD_15. Moduliai.

Studentas studijuoja informatiką universitete. Visi informatikos moduliai yra pasirenkami. Visus modulius reikia išklaudyti. Kai kuriuos modulius galima rinktis tik tuomet, kai išklaudyti jau tam tikri moduliai. Studentui reikia susidaryti nuosavą studijų programą, kurioje moduliai būtų išdėstyti į sąrašą, kuriame toliau stovintys moduliai būtų priklausomi tik nuo anksčiau stovinčių modulių. Parašykite programą, kuri sudarytų visus galimus modulių priklausomybės sąrašus.

Duomenys. Duomenų faile **U2.txt** yra informacija apie modulius. Pirmoje failo eilutėje nurodytas modulių kiekis N ($1 \leq N \leq 50$). Tolesnėse eilutėse informacija apie modulius: modulio kodas (4 simboliai), modulio pavadinimas (30 simbolių), kiekis modulių, nuo kurių priklauso šio modulio studijos, priklausomų modulių kodai. Eilutėje vienus duomenis nuo kitų skiria vienas tarpas

Rezultatai. Spausdinkite modulių priklausomybės sąrašus, kuriuose nurodysite modulio kodą ir modulio pavadinimą.

U2.txt	Vieno sąrašo rezultatai
9	
IF01 Programavimo įvadas 0	IF01 Programavimo įvadas
IF02 Diskrečioji matematika 1 IF01	IF04 Skaitmeninė logika
IF03 Duomenų struktūros 2 IF01 IF02	IF02 Diskrečioji matematika
IF04 Skaitmeninė logika 0	IF03 Duomenų struktūros
IF05 Matematinė logika 1 IF04	IF08 Programavimo kalbų teorija
IF06 Operacijų optimizavimas 1 IF05	IF05 Matematinė logika
IF07 Algoritmų analizė 2 IF03 IF06	IF06 Operacijų optimizavimas
IF08 Programavimo kalbų teorija 1 IF03	IF07 Algoritmų analizė
IF09 Operacinės sistemos 2 IF07 IF08	IF09 Operacinės sistemos

LD_16. Pažintis.

Įvairių miesto mokyklų geriausi moksleiviai važiuoja į ekskursiją. Nors moksleiviai yra iš skirtingų mokyklų, tačiau yra tokių, kurie pažįsta vieni kitus. Moksleiviai nori užmegzti naujas pažintis, tačiau su nepažįstamu moksleiviu galima susipažinti tik tuomet, jeigu yra pažįstamų moksleivių grandinė (pirmas pažįsta antrą, antras pažįsta trečią, trečias pažįsta ketvirtą, tuomet pirmas gali susipažinti su ketvirtu), kuri veda iki nepažįstamo moksleivio. Pirmame tekstiname faile 'U31DUOM.TXT' apie moksleivius pateikta tokia informacija: moksleivio vardas, jo pažįstamų moksleivių kiekis, pažįstamų moksleivių vardai. Kiekvienam

moksleiviui tekstiniame faile yra skirta po vieną eilutę. Antrame tekstiniame faile 'U32DUOM.TXT' vienoje eilutėje nurodyti dviejų moksleivių vardai. Tokių eilučių gali būti keletas. Abiejuose failuose moksleivių duomenys skiriami bent vienu tarpu.

Nustatykite kiekvienai moksleivių porai iš antrojo failo ar jie jau yra pažįstami, ar jie gali susipažinti (jeigu gali, reikia nurodyti visus bendrus pažįstamus moksleivius), ar jie negali susipažinti (bendro pažįstamo moksleivio neturi). Rezultatų faile 'U3REZ.TXT' reikia spausdinti poros vardus, šalia nurodant atsakymą, kaip žemiau pateiktame pavyzdyje.

Pirmasis duomenų failas 'U31DUOM.TXT':

Rūta	1	Arnoldas
Agnė	3	Nerijus Neda Antanas
Nerijus	1	Agnė
Antanas	2	Agnė Marius
Marius	2	Antanas Neda
Neda	3	Marius Rūta Agnė
Arnoldas	1	Rūta

Antrasis duomenų failas 'U32DUOM.TXT':

Rūta	Nerijus
Agnė	Antanas
Neda	Nerijus

Rezultatų failas 'U3REZ.TXT':

Rūta	Nerijus	negali susipažinti
Agnė	Antanas	jau pažįstami
Neda	Nerijus	bendri pažįstami: Agnė

LD_17. Mozaika.

Turime daug vienodo dydžio kubelių, kurių kiekvienas šonas nudažytas kokia nors spalva: r (raudona), z (žalia), g (geltona). Kubeliai paberiami ant stalo ir sustumdomi taip, kad gulėtų vienu sluoksniu ir sudarytų stačiakampį NxM. Padaryti kubelių generatorių, kuris nuspaltotų kubelius ir juos sudėtų ant stalo NxM stačiakampiame plote. Čia N – eilučių skaičius, o M – stulpelių skaičius. Kubelių iš viso turi būti NxM. Čia $1 \leq N \leq 20$ ir $1 \leq M \leq 30$. N ir M įvedami klaviatūra.

Surasti didžiausią vienos spalvos plotą, kuriam priklauso visi kaimyniniai kubeliai, susieti ta pačia spalva viršuje ir apačioje. Kubelio kaimynu laikomas tas kubelis, kuris su juo liečiasi bent vienu tašku. Viršutinis spalvotas plotas ir apatinis spalvotas plotas laikomi vienu plotu, jeigu turi bent vieną kubelį, kurio viršus ir apačia tos pačios nagrinėjamos spalvos.

Ekrane vieną šalia kitos parodyti viršutinę ir apatinę stačiakampio puses. Surasto didžiausio vienos spalvos ploto langelius pažymėti žvaigždute *. Apačioje parašyti, iš kiek langelių sudarytas tas plotas: atskirai viršuje ir apačioje. Parašyti bent vieno kubelio, jungiančio tuos plotus (ta pati spalva viršuje ir apačioje) koordinatas: eilutės numerį ir stulpelio numerį.

- Sugeneruoto lauko ir rezultato pavyzdys, kai N = 6 ir M = 10.

	1	2	3	4	5	6	7	8	9	10
1	Red	Yellow	Red	Red	Yellow	Yellow	Yellow	*	*	*
2	Green	Green	Yellow	Red	Yellow	Yellow	Yellow	*	*	*
3	Green	Green	Yellow	Green	Green	Green	*	*	Yellow	Yellow
4	Yellow	*	Yellow	Green	Green	Green	*	*	Green	Green
5	Yellow	*	*	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Green
6	Yellow	*	*	*	*	*	*	*	Green	Green

	1	2	3	4	5	6	7	8	9	10
1	Green	Green	Green	Green	Green	*	Yellow	Yellow	Yellow	Yellow
2	Green	Green	Green	Green	Green	*	Yellow	Yellow	Yellow	Yellow
3	*	*	*	*	*	*	*	*	*	*
4	Yellow	Yellow	*	Green	Green	*	*	Green	Green	*
5	Yellow	Yellow	*	Yellow	Yellow	Yellow	Yellow	Green	Green	*
6	Yellow	Yellow	Green	Green	Green	Green	*	Green	Green	*

- Didžiausią plotą sudaro viršuje 20 ir apačioje 20 langelių. Langelis: 5 eil., 3 st.

LD_18. Miestas.

Esame mieste, kurio gatvės sudaro kvadratinį tinklą. Mieste yra tam tikras skaičius gėlių parduotuvių. Visos parduotuvės išdėstytos gatvių kampuose. Viename iš gatvės kampų stovi vyras ir galvoja, ar nepadovanoti žmonai gėlių. Bet vyras yra alkanas ir vedęs jau daugelį metų, todėl gėles pirks tik tuo atveju, jei iki artimiausios parduotuvės reikės eiti ne daugiau kaip 5 kvartalus. Tačiau mieste vyksta gatvių remonto darbai, ir

kai kurie kvartalai nepraeinami. Kad nevaikščioti veltui, vyras skambina į miesto savivaldybės informacijos tarnybą ir klausia, ar toli nuo esamo taško yra gėlių parduotuvė. Jei netoli, tai kaip ją pasiekti. Parašykite programą, kuria galėtų naudotis savivaldybės informacijos tarnybos darbuotojas.

Duomenys tekstiname faile “Duom3.txt” išdėstyti tokia tvarka:

pirmoji eilutė – matricos dydis (vienas skaičius ≤ 50), vyro koordinatės (du skaičiai), antroji ir tolimesnės eilutės - pati miesto matrica. Matricoje duomenys yra simboliai. Kvartalus skiria gatvės, kurios pažymėtos taškais. Vieną kvartalo kraštinę žymi 3 simboliai: pirmas gatvės kampas (G – yra gėlių parduotuvė, 0 – nėra gėlių parduotuvės), kliūtis (0 – nėra, 1 – yra), antras gatvės kampas (G – yra gėlių parduotuvė, 0 – nėra gėlių parduotuvės). Rezultatas turėtų būti atsakymas, ar yra tokia parduotuvė, iki kurios ne daugiau kaip 5 kvartalai. Jei yra tokia parduotuvė, tai reikia pažymėti bent vieną kelią iki jos. Rezultatą parodykite ekrane.

Pastabos. Praėjimas viena kvartalo kraštine laikomas kvartalo įveikimu. Negalima vaikščioti kiemais, nes ten gali būti plėšikų. Sankryžas galima kirsti tik stačiu kampu.

Pavyzdys:

“Duom3.txt”	Rezultatai
11 11 9	Parduotuvė yra už 3 kvartalų .
G00.000.000	G00.000.000
010.010.010	010.010.010
G00.010.010	G00.010.010
.....
G00.010.010	G00.010.010
010.010.010	010.010.010
G00.010.010	K00.010.010
.....	K.....
000.000.000	KKKKKKKKK00
010.010.010	010.010.K10
000.000.000	000.000.K00

LD_19. Dėmė.

Turime popieriaus lapą langeliais. Langelių eilutės ir stulpeliai sunumeruoti, pradedant vienetu. Numeracijos pradžia yra kairysis viršutinis langelis. Vaikas padažė teptuką dažuose ir krestelėjo virš lapo. Suskaičiuokite, kiek dėmių lape ir iš kiek langelių sudaryta didžiausia dėmė. Langeliai arba tušti, arba dažyti. Pusiau dažytų nėra. Tai pačiai dėmei priklauso visi dažyti langeliai, kurių bent viena koordinatė skiriasi vienetu nuo gretimo dažyto langelio. Reikia parašyti dėmių generatorių, kuris $N \times M$ lape nudažytų atsitiktinai trečdalį langelių, kai $1 \leq N \leq 20$ ir $1 \leq M \leq 70$. Čia N – eilučių skaičius, o M – stulpelių skaičius. N ir M reikšmės įvedamos klaviatūra. Ekrane parodykite dėmėmis išmargintą lapą. Tuščius langelius žymėkite tašku, dažytus langelius žymėkite žvaigždute * arba spalva. Apačioje parašykite, kiek yra dėmių, kiek langelių sudaro didžiausią dėmę ir didžiausios dėmės vieno bet kurio langelio koordinatės: eilutės numerį ir stulpelio numerį.

Sugeneruoto lauko ir rezultatų pavyzdys, kai N 10 ir M 15.

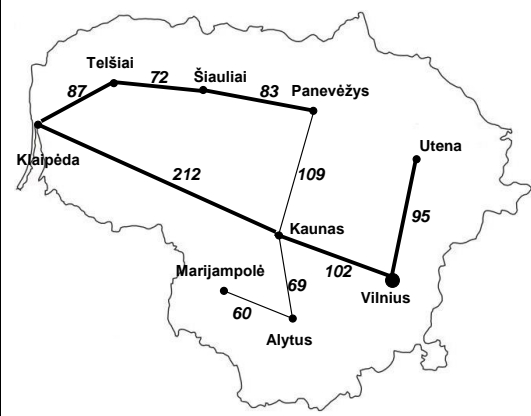
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	*	*	*	.	.	.	*	*	*
2	*	*	*	.	.	*	.	.	*	*
3	*	.	.	.	*	.	.	.	*	*	.
4	*	*	.	.	*	.	.	.	*	.	.	.	*	.	.
5	*	.	.	*	.	.	*
6	.	*	*	*	*
7	.	*	*	*	*
8	*	.	.	.	*	*	.
9	*	*	*	*	*	*	.
10	*	*	*	*	*	*	*	.	*	*

Dėmių skaičius: 6. Didžiausia dėmė: 21. Eilutė: 3. Stulpelis: 5.

LD_20. Maršrutai.

Turizmo agentūra organizuoja kelionę po Lietuvą. Reikia parašyti programą, kuri pasiūlytų ilgiausios kelionės maršrutą iš nurodyto miesto. Tekstiname faile 'U3.txt' surašyti duomenys apie kelius tarp miestų. Pirmoje failo eilutėje yra kelių skaičius n ($1 \leq n \leq 100$), antroje eilutėje miesto pavadinimas, iš kur prasideda kelionė. Sekančiose n failo eilučių surašyta: pirmojo miesto pavadinimas, antrojo miesto pavadinimas, kelio tarp pirmojo ir antrojo miesto ilgis kilometrais. Miesto pavadinimas ne ilgesnis kaip 15 simbolių.

Išvykus iš vieno miesto galima nukeliauti į bet kurį kitą miestą. Tarp miestų gali būti daugiau kaip vienas kelionės maršrutas. Kelionės metu tas pats miestas gali būti aplankytas tik vieną kartą. Maršrutas nebūtinai turi apimti visus duotus miestus. Atspausdinkite ekrane ilgiausios kelionės maršrutą ir jo ilgį kaip parodyta pavyzdyje.

	U3 .txt		
	9		
Utena			
Vilnius	Kaunas		102
Panevėžys	Šiauliai		83
Utena	Vilnius		95
Klaipėda	Kaunas		212
Marijampolė	Alytus		60
Telšiai	Šiauliai		72
Kaunas	Alytus		69
Klaipėda	Telšiai		87
Kaunas	Panevėžys		109
Rezultatų pavyzdys			
Utena – Vilnius – Kaunas – Klaipėda – Telšiai – Šiauliai – Panevėžys (651 km)			

LD_21. Egzaminų tvarkaraštis.

Universiteto studentams sudaromas studijų planas, į kurį įtraukiami studijų moduliai. Studentui, išklausiusiam atitinkamą modulį, privaloma laikyti egzaminą. Studentas gali būti išklausęs kelis skirtingus modulius, bet negali laikyti kelių egzaminų tuo pačiu laiku. Skirtingų modulių egzaminai gali vykti tuo pačiu laiku, jei tuose moduluose nėra bendrų studentų. Egzaminavimo laikų (visi laikai skirtingi) skaičius yra mažesnis už modulių skaičių. Jei visuose moduluose nėra bendrų studentų, tuomet visų modulių egzaminai vyksta vienu laiku. Paskirtas egzaminavimo laikas gali likti laisvas, jei modulį išklausę studentai galėjo laikyti egzaminą kitu laiku. Taip užtikrinama, kad užimtų egzaminavimo laikų skaičius būtų minimalus.

Parašykite programą, kuri suformuotų modulių egzaminų tvarkaraštį. Jei galimi keli tvarkaraščio sudarymo būdai, nurodomas tas, kuriame moduliai nagrinėjami paeiliui, pradedant nuo pirmojo.

Duomenys. Faile U3 .txt pirmoje eilutėje pateiktas egzaminavimo laikų skaičius m ($1 \leq m \leq 10$), o antroje eilutėje – modulių skaičius n ($1 \leq n \leq 100$). Toliau surašyti kiekvieno modulio sąsają pagal bendrus studentus su kitais moduliais duomenys (n eilučių ir n stulpelių matrica).

Matricos elemento reikšmė yra lygi 1, jei du atskiri moduliai, kurių numeriai i ir j ($1 \leq i, j \leq n, i \neq j$), turi bendrų (tų pačių) studentų, priešingu atveju matricos elemento reikšmė lygi 0. Pavyzdžiui, pirmame duomenų pavyzdyje pirmas modulis turi bendrų studentų su antru, trečiu bei ketvirtu moduliais, antras – su pirmu ir trečiu ir t.t.

Rezultatai. Ekrane dviem stulpeliais spausdinkite modulių paskirstymą pagal egzaminavimo laikus, pradedant nuo pirmojo modulio, kaip parodyta pavyzdyje. Moduliai ir egzaminavimo laikai numeruojami nuo vienetų. Pirmame stulpelyje išveskite modulio numerį, o antrame – egzaminavimo laiko numerį. Jei pagal turimus duomenis tvarkaraščio suformuoti neįmanoma, spausdinkite „Sprendinys neegzistuoja“.

U3 .txt	Ekranas
3	1 1
4	2 2
0 1 1 1	3 3
1 0 1 0	4 2
1 1 0 1	
1 0 1 0	
2	Sprendinys neegzistuoja
3	
0 1 1	
1 0 1	
1 1 0	

LD_22. Kelias tarp vietovių.

Gūdučių universiteto informatikos fakulteto I kurso studentai nutarė dalyvauti orientavimosi dviračiais varžybose. Jie sudarė komandą ir atvyko į vietovę Preivai, kur bus duotas startas. Buvo pranešta, kad finišas Balkuose. Kaip ir kitų komandų atstovai, jie gavo vietovės žemėlapi, kuriame pažymėti visi keliai ir surašyti kelių ilgiai. Padėkite studentams surasti trumpiausią kelią tarp nurodytų vietovių, jei žinoma, kad kelias tarp starto ir finišo vietovių gali būti tiesioginis (be tarpinių vietovių) arba tarpe jų gali būti ne daugiau kaip 5 tarpinės vietovės.

U3.txt	Rezultatai
5 8	Minimalus atstumas tarp vietovių
Preivai	Preivai ir Balkai 6 km
Saukai	Trasa eina per vietas:
Salai	Preivai

Duomenys. Tekstinio failo 'U3.txt' pirmoje eilutėje nurodytas vietovių skaičius N ($2 \leq N \leq 10$) ir visų kelių kiekis M ($1 \leq M \leq 50$). Tolimesnėse N eilutėse surašytos visos galimos vietovės po vieną eilutėje. Po to eilutėje surašytos starto ir finišo vietovių pavadinimai. Šiai eilutei iš viršaus ir apačios

Rekai			Saukai
Balkai			Rekai
			Balkai
Preivai	Balkai		
Preivai	Saukai	2	
Preivai	Salai	8	
Preivai	Balkai	10	
Saukai	Balkai	7	
Saukai	Rekai	2	
Salai	Rekai	4	
Salai	Balkai	6	
Rekai	Balkai	2	

palikta po vieną tuščią eilutę. Po antros tuščios eilutės M eilutėse surašyti visi keliai po vieną eilutėje. Tokios eilutės struktūra: pradinė vietovė, galinė vietovė, atstumas tarp jų. Vietovės pavadinimas – iki 10 simbolių.

Rezultatai. Rezultatus išveskite į ekraną, kuriame nurodykite trumpiausio kelio ilgį ir paeiliui išvardinkite visas vietoves, per kurias eis trasa.

LD_23. Parlamentas.

Naujosios Demokratijos saloje kiekvienas gyventojas įkūrė po partiją, kurios vadovu pats ir tapo. Kiekvienas salos gyventojas gali būti bet kurios kitos partijos nariu. Visų nustebimui net ir pačioje mažiausioje partijoje buvo ne mažiau 2 žmonių. Tačiau finansiniai sunkumai neleido įgyvendinti Konstitucijos projekto, pagal kurį į parlamentą turėjo būti išrinkti visų partijų vadovai. Pasitarę, salos gyventojai nusprendė, kad bus gerai, jei parlamente bus nors po vieną kiekvienos partijos atstovą. Nustatyti minimalaus dydžio parlamentą, kuriame bus atstovaujamos visos partijos.

Duomenys. Tekstinio failo 'U3.txt' pirmoje eilutėje nurodytas partijų skaičius N ($2 \leq N \leq 30$), tolimesnėse eilutėse – pirmasis skaičius nurodo partijos narių kiekį, toliau skaičiais nurodomi partijos nariai (partija ir jos vadovas turi tą patį numerį).

Rezultatai. Į ekraną išveskite pradinį duomenį, kur parlamentarų numeriai būtų surikiuoti didėjimo tvarka. Suskaičiuotus rezultatus išveskite taip pat į ekraną, kur išvardinsite parlamento narius numerių didėjimo tvarka.

U3. txt	Ekranas
4	Pradiniai surikiuoti duomenys:
4 1 2 3 4	1 2 3 4
2 2 3	2 3
4 3 1 4 2	1 2 3 4
2 4 2	2 4
	Parlamentą sudaro:
	2

LD_24. Susitikimas.

Grupė draugų nusprendė susitikti mieste, o po to kartu eiti valgyti picos. Bet tuomet jie susiginčijo, kur geriausia susitikti, ir kurioje picerijoje valgyti. Galiausiai draugai nusprendė, jog patogiausia išsirinkti tokią susitikimo vietą ir piceriją, kad jų nueitų kelių iki susitikimo vietos, po to iki picerijos ir atgal į pradinį tašką suma būtų mažiausia. Parašykite programą, kuri rastų patogiausią susitikimo vietą ir piceriją.

Duomenys. Pirmoje „U3.txt“ eilutėje duoti du skaičiai n ir m — miesto žemėlapis aukštis ir plotis ($2 \leq n, m \leq 10$). Toliau pateiktas miesto žemėlapis — n eilučių, kiekvienoje jų — m simbolių. Galimi šie simboliai:

U3. txt	Ekranas
5 5	1 1
P.X.P	5 2
..XS.	Neįmanoma
S.X..	
..X.D	
D.X..	
5 5	1 1
D...P	1 3
XX...	1 5
D.S.P	Susitikimo vieta 3 3
XX...	Picerija 5 3

. — langelis **pereinamas**.

X — langelis **nepereinamas** (pastatas, tvora...)

P — picerija. Langelis **nepereinamas**, į piceriją galima įeiti iš gretimo langelio ir išeiti į bet kurį gretimą langelį. Įeiti galima tik į tą piceriją, kurioje bus valgoma pica.

S — susitikimo vieta. Langelis **pereinamas**.

D — vieno iš draugų pradinė buvimo vieta. Langelis **pereinamas**.

Judėti galima tik aukštyn, žemyn, į kairę arba į dešinę (ne įstrižai).

Rezultatai. Ekrane atskirose eilutėse atspausdinkite pradines draugų koordinatas (koordinačių pradžia – apatinis kairysis langelis, numeruojama nuo 1, pirmiausiai nurodoma x koordinatė, po to - y).

Atskirose eilutėse atspausdinkite: „Susitikimo vieta“ ir susitikimo vietos koordinatas; „Picerija“ ir picerijos koordinatas; „Nueita“ ir draugų nueitų kelių sumą. Jei susitikimo vietos ar picerijos, kurias visi draugai galėtų pasiekti, nėra, atspausdinkite žodį „Neįmanoma“.