

8 savaitė

Ciklas cikle

1 lygis

1. Stebuklų šalies gyventojai

Stebuklų šalyje gyvena stebukladariukai ir stebukladarytės. Vienas stebukliukladariukas per dieną sukuria **s1** stebuklų, viena stebukladarytė – **s2** stebuklų.

Parašykite programą, kuri į failą **stebuklai_res.txt** išvestų **st1** ir stebukladaryčių **st2** derinius, jeigu jie iš viso sukūrė **s** stebuklų. **s**, **s1** ir **s2** reikšmės perskaitomos iš failo **stebuklai_data.txt**.

stebuklai_data.txt	stebuklai_res.txt
21 3 4	3 3 7 0

2. Stačiakampis

Parašykite programą, kuri iš sveikųjų teigiamų skaičių intervalo [**a**; **b**] išrinktų ir išspausdintų visas atkarpas, iš kurių galima sudaryti stačiakampį, kurio viena kraštinė **x** vienetų ilgesnė už kitą.

Pradiniai duomenys **a**, **b** ir **x** įrašyti faile **staciakampis_data.txt**.

Rezultatai turi būti spausdinami rezultatų faile **staciakampis_res.txt**. Vienam stačiakampiui skiriama viena eilutė. Pirma spausdinama trumpesnioji, po to tarpu atskirta ilgesnioji kraštinė.

staciakampis_data.txt	staciakampis_res.txt
3 8 3	3 6 4 7 5 8

3. Keturių skaičių suma

Parašykite programą, kuri tekstiniame faile **skaiciai_res.txt** išspausdintų visus galimus didėjančiai einančius **keturis** skirtingus natūraliuosius skaičius, kurių suma yra lygi **s**. **s** reikšmė įrašyta faile **skaiciai_data.txt**

skaiciai_data.txt	skaiciai_res.txt
15	1 2 3 9 1 2 4 8 1 2 5 7 1 3 4 7 1 3 5 6 2 3 4 6

4. Gražiausios raganos rinkimai

Susirinkę į raganų suvažiavimą, raganos nusprendė išrinkti gražiausią raganą. Rinkimuose dalyvavo **n** raganų, kurių kiekviena turėjo galimybę balsuoti už **m** kandidačių, skirdama kiekvienai kandidatėi skirtingą tinkamumo įvertinimą, išreikštą sveikuoju skaičiumi nuo 0 iki 10. Kuo kandidatė tinkamesnė, tuo įvertinimas yra mažesnis. Pirmoje pradinių duomenų failo **ragana_data.txt** eilutėje įrašyti du sveikieji skaičiai – **n** ir **m** reikšmės. Tolesnėse **m** eilučių įrašyta po **n** sveikųjų skaičių – kaip kiekviena balsuojanti ragana įvertino kandidatę. Parašykite programą, skaičiuojančią, koks buvo kiekvienos kandidatės bendras įvertinimas (visų rinkėjų įvertinimų suma) ir kuri kandidatė laimėjo rinkimus. Jei kelios kandidatės surinko po vienodą įvertinimą, tuomet rinkimus laimės ta, kuri sąraše yra pirmesnė. Rezultatų failo **ragana_res.txt** pirmosiose **m** eilučių turi būti įrašytas kiekvienos kandidatės galutinis įvertinimas, o paskutinėje eilutėje – rinkimus laimėjusio kandidatės numeris.

ragana_data.txt	ragana_res.txt
3 5	1 12
2 3 3 2 2	2 8
2 1 2 1 2	3 8
1 1 2 2 2	2

5. LEGO robotas

Linas nusprendė 6 mėnesius taupyti pinigus LEGO robotui, kuris kainuoja **a** eurų. Kiekvieną mėnesį jis skirtingą kartų kiekį **n** į taupyklę mesdavo pinigų. Duomenų faile **lego_data.txt** yra 6 eilutės, kuriose pirmasis skaičius nurodo, kiek kartų **n** tą mėnesį Linas į taupyklę įmetė pinigų, o kiti skaičiai toje eilutėje nurodo, kokią pinigų sumą **m** (realusis skaičius) Linas įmetė kiekvieną kartą. Duomenų failo paskutiniojoje (septintoje) eilutėje yra nurodyta norimo LEGO roboto kaina **a**.

Parašykite programą, kuri suskaičiuotų, kiek Linas per šešis mėnesius sutaupė pinigų ir ar jų užteks nusipirkti norimą LEGO robotą (TAIP, jei užteks; NE, jei pritrūks), o rezultatus įrašytų į rezultatų failą **lego_res.txt**.

lego_data.txt	lego_res.txt
4 6 1.7 4.3 5	152.59
6 7 5.5 6.2 6.1 2.2 6	TAIP
3 8.56 3.33 7	
5 6 8 3.3 6 1.4	
5 7 9.2 2.5 6 5	
6 7 3.3 5 2 5 7	
150.99	

6. Krepšinio komandos

Pirmoje pradinių duomenų failo **komandos_data.txt** eilutėje įrašytas krepšinio komandų skaičius **n**. Tolesnėse **n** failo eilučių įrašyti tarpais atskirti skaičiai: pirmasis skaičius yra sveikasis ir reiškia komandos naujokų skaičių **m**, tolesni **m** realiųjų skaičių rodo krepšininkų ūgi metrais.

Rezultatų faile **komandos_res.txt** įrašykite:

- pirmose **n** eilučių – komandos numerį ir vidutinį komandos narių ūgį dešimtųjų tikslumu;
- paskutinėje failo eilutėje – vidutinį visų komandų sportininkų naujokų ūgį dešimtųjų tikslumu ir kiek sportininkų naujokų žaidžia visose komandose iš viso.

komandos_data.txt	komandos_res.txt
3	1 193.6
5 180.5 195.2 210.2 195.1 187.2	2 194.5
3 195.0 196.2 192.2	3 191.1
2 189.7 192.4	193.4 10

7. Elektronikos elementų ir komponentų rinkinys

Elektrinėse grandinėse naudojami įvairūs puslaidininkiniai elementai, integriniai grandynai (mikroschemos), elektroniniai komponentai bei elektriniai prietaisai, kitos dalys. Visi jie turi savo paskirtį ir atlieka grandinėse atitinkamas funkcijas. Mokyklos 7-10 klasių mokiniai gali mokytis jungti elektrines grandines per technologijų pamokas. Mokytojui reikia suformuoti užsakymą elektrinių grandinių komponentams. Mokytojas tekstinio failo **grandines_data.txt** pirmoje eilutėje įrašė, kelių tipų **t** komponentus reikia užsakyti. Toliau seka **t** eilučių, kurių kiekvienoje įrašyta, kiek skirtingų vieno tipo komponentų **s** reikia užsakyti ir po **s** skaičių dvejetų. Pirmasis dvejeto skaičius rodo komponentų skaičių (sveikasis skaičius) **k** ir komponento vieneto kainą **vk** (realusis skaičius). Parašykite programą, kuri tekstiname faile **grandines_res.txt** išvestų kiekvieno tipo bendrą komponentų skaičių ir jų kainą dviejų skaitmenų po kablelio tikslumu. Vienam komponentui skiriama viena eilutė. Paskutinėje eilutėje turi būti išvestas bendras įsigytų komponentų skaičius ir jų kaina dviejų skaitmenų po kablelio tikslumu.

grandines_data.txt	grandines_res.txt
3	19 8.85
4 5 0.50 4 0.05 5 1.20 5 0.0	16 1.07
3 3 4 0.03 5 0.05 7 0.10	12 0.84
1 12 0.07	47 10.76

2 lygis

8. Pitagoro teoremos diena

Pitagoro teoremos diena vadinama data, kurios dienos, mėnesio ir metų paskutiniųjų dviejų skaitmenų rinkinys tenkina Pitagoro teoremą. Pavyzdžiui, 2017 metų rugpjūčio 15 d. buvo Pitagoro teoremos diena, nes $15^2 + 8^2 = 17^2$.

Parašykite programą, kuri išrinktų metų intervalo **[m1; m2]** visas Pitagoro dienas ir jas išspausdintų po vieną eilutėje tokia tvarka: metai, mėnuo, diena.

Pradiniai duomenys įrašyti faile **diena_data.txt**, rezultatus įrašykite į tekstinį failą **diena_res.txt**.

diena_data.txt	diena_res.txt
2010 2015	2010 6 8 2010 8 6 2012 5 13 2012 9 15 2013 5 12 2013 12 5 2015 8 17 2015 9 12 2015 12 9

9. Pitagoro teoremos paros laikas

Pitagoro teoremos paros laiku vadinamas laikas, kurio valandos, minutės ir sekundės tenkina Pitagoro teoremą. Pavyzdžiui, 21 valanda 20 minučių 29 sekundės yra Pitagoro teoremos laikas, nes $21^2 + 20^2 = 29^2$.

Parašykite programą, kuri išrinktų valandų **[v1; v2]**, minučių **[m1; m2]** ir sekundžių **[s1; s2]** intervalų visus Pitagoro laikus ir juos išspausdintų po vieną eilutėje tokia tvarka: valanda, minutė, sekundė.

Pradiniai duomenys įrašyti faile **para_data.txt**, rezultatus įrašykite į tekstinį failą **para_res.txt**.

para_data.txt	para_res.txt
21 23	21 20 29
15 45	21 28 35
20 50	21 29 20 21 35 28

10. Palindrominis laikas nuo vidurnakčio

Laikas vadinamas palindrominiu, jei apskaičiavus, kiek laiko praėjo sekundėmis nuo vidurnakčio, pavertus laiką sekundėmis, sekundžių skaičius skaitant iš kairės ir iš dešinės gaunamas vienodas. Pvz., jei laikrodis rodo 21 valandą, 26 minutes ir 17 sekundžių, tai nuo vidurnakčio praėjo $21 * 3600 + 26 * 60 + 17 = 77177$ sekundės. Skaičius 77177 yra palindromas. Parašykite programą, kuri išrinktų valandų [v1; v2], minučių [m1; m2] ir sekundžių [s1; s2] intervalų visus palindrominius laikus ir juos išspausdintų po vieną eilutėje tokia tvarka: valanda, minutė, sekundė.

Pradiniai duomenys įrašyti faile **palindrominis_data.txt**, rezultatus įrašykite į tekstinį failą **palindrominis_res.txt**.

palindrominis_data.txt	palindrominis_res.txt
15 16	15 12 25
10 20	15 19 15
15 25	16 11 25
	16 16 25

11. Palindrominės datos

Data vadinama palindromine, jei ją perskaičius iš dešinės į kairę ir iš kairės į dešinę, gaunama ta pati data. Pvz., 2001 metų spalio mėnesio 2 diena yra palindrominė data: 20011002.

Parašykite programą, kuri išrinktų metų intervalo [m1; m2] visas palindromines datas ir jas išspausdintų po vieną eilutėje tokia tvarka: metai, mėnuo, diena.

Pradinius duomenis perskaitykite iš failo **datos_data.txt**, rezultatus įrašykite į tekstinį failą **datos_res.txt**.

datos_data.txt	datos_res.txt
2000 2017	20011002
	20100102
	20111102

12. Biblioteka

Bibliotekoje prasidėjus naujiems kalendoriniams metams registruojamos knygos, kurios bus išduodamos skaitytojams, paimtos ir dar negrąžintos knygos, bei knygos, kurios turės būti skaitomos tik skaitykloje. Taip pat fiksuojama, kelios knygos grąžintos pirmąją metų darbo dieną, bei kelias knygas užsisakė skaitytojai. Vieno pavadinimo knygoms skiriama viena lentyna. Pradinių duomenų failo **biblioteka_data.txt** pirmoje eilutėje įrašytas lentynų skaičius **n**. Tolesnėse **n** eilučių įrašyta po 5 sveikuosius skaičius: (1) lentynoje esančių knygų skaičius, (2) knygų, skaitomų tik skaitykloje, skaičius, (3) pirmąją darbo dieną grąžintų knygų skaičius, (4) likusių pas skaitytojus knygų skaičius, (5) knygų, užsakytų pirmąją bibliotekos darbo dieną, skaičius.

Parašykite programą, skaičiuojančią:

- kelios kiekvieno pavadinimo knygos bus bibliotekoje pirmosios darbo dienos pabaigoje, kai skaitytojams bus išduotos užsakytos tą dieną knygos;

- kelių kiekvieno pavadinimo knygų trūksta skaitytojų užsakymų poreikiui patenkinti pirmosios darbo dienos pabaigoje;
- kurio numerio (knygų lentynos numeruojamos iš eilės, pradedant nuo 1) knygų, likusių pas skaitytojus, skaičius yra didžiausias;
- kelios kiekvieno pavadinimo knygos bus vidutiniškai užsakytos pirmąją darbo dieną. Išveskite sveikąją dalmens dalį;
- kiek knygų iš viso bus skaitoma skaitykloje;
- kiek iš viso knygų trūksta, kad būtų patenkinti visų skaitytojų užsakymai.

Skaičiavimų rezultatus įrašykite į tekstinį failą **biblioteka_res.txt**.

biblioteka_data.txt	biblioteka_res.txt
3	1 8 4
8 8 8 8 12	2 7 0
7 3 8 9 8	3 8 0
12 5 5 10 9	----
	3 10

	9

	16

	4

13. Skaniausi pietūs

Populiariame šou „Skaniausi pietūs“ varžosi geriausi virtuvės šefai, kurie gamina įvairius patiekalus. Kiekvieną dieną vienas dalyvis gamina pietus likusiems, o likusieji įvertina jo darbą sveikųjų skaičių skalėje nuo 1 iki 5. Kiekvieno dalyvio galutinis įvertinimas - likusių dalyvių įvertinimų suma, atėmus geriausią ir blogiausią įvertinimus. Jei geriausi ir/ar blogiausi įvertinimai yra keli, tuomet atimama tik po vieną geriausią ir po vieną blogiausią įvertinimą. Jei geriausias ir blogiausias įvertinimai sutampa, tuomet iš sumos atimamos dvi vienodos vertės. Šou laimi daugiausia taškų surinkęs dalyvis.

Pirmoje pradinių duomenų **pietus_data.txt** eilutėje įrašytas šou dalyvių skaičius **n** ($n > 3$). Tolesnėse **n** eilučių įrašyta po **n-1** sveikųjų skaičių - kaip kažkurį dalyvį vertina likusieji šou dalyviai. Dalyviai numeruojami nuo 1 iki **n**. Jei keli nors šou dalyviai surinko po vienodą taškų skaičių, tuomet nugalėtojas yra tas, kuris pietus gamino anksčiau (turi mažesnę numerį). Rezultatų faile **pietus_res.txt** turi būti įrašyti du sveikieji skaičiai, vienas nuo kito atskirti tarpu: šou laimėtojo numeris ir jo galutinis įvertinimas.

pietus_data.txt	pietus_res.txt
5	4 10
5 4 4 5	
5 4 4 4	
5 5 4 4	
5 5 5 4	
4 4 4 5	

14. Lošimas kauliukais

Vyksta lošimo kauliukais žaidimas: žaidėjų yra **n**, kiekvienas meta kamuoliuką tol, kol atsiverčia lyginis akučių skaičius, jeigu iškrenta nelyginis - metama toliau. Surinkti taškai yra sumuojami. Parašykite programą, kuri rastų daugiausiai taškų surinkusi žaidėją, išvestų, kiek jis surinko taškų, bei išvestų visų žaidėjų surinktus taškus.

Jeigu yra du ar daugiau žaidėjų, surinkusių daugiausiai taškų, išvesti tik pirmąjį sąrašė. Žinoma, jeigu iškrenta lyginis kauliukų skaičius, žaidėjo žaidimas baigiamas, todėl toliau esami akučių skaičiai nesumuojami, nes tada yra sukčiaujama.

Pirmoje duomenų failo **kauliukai_data.txt** eilutėje įrašytas žaidėjų skaičius **n**. Tolesnėse **n** eilučių parašyta, kiek kartų žaidėjas metė kauliuką ir atsivertusių akučių skaičiai.

Rezultatų faile **kauliukai_res.txt** įrašyti, po kiek taškų surinko kiekvienas žaidėjas ir laimėjusio žaidėjo numeris.

kauliukai_data.txt	kauliukai_res.txt
4	1 16
3 5 5 6	2 7
2 1 6	3 14
3 3 5 6	4 2
1 2	1

3 lygis

15. Gucikai, mucikai ir fucikai (XVI olimpiada, 2004)

Toli visatoje esančioje planetoje egzistuoja gyvybė. Ten gyvena trijų rūšių būtybės: *gucikai*, *mucikai* ir *fucikai*. Skirtingų rūšių būtybės gali turėti nevienodą skaičių kojų, rankų ir akių, o vienos rūšies būtybės jų turi po vienodą skaičių. Žinoma, kad bet kuri būtybė turi bent vieną ranką, bent vieną koją ir bent vieną akį. Pavyzdžiui, *gucikai* gali turėti po dvi kojas, po dvi rankas ir po tris akis, *mucikai* – po keturias kojas, po vieną ranką ir po vieną akį, *fucikai* – po vieną koją, penkias rankas ir po tris akis. Žinoma, po kiek kojų, po kiek rankų ir po kiek akių turi *gucikai*, *mucikai* ir *fucikai*. Taip pat žinoma, kiek kojų, rankų ir akių turi visos planetoje gyvenančios būtybės kartu. Parenkite programą, kuri apskaičiuotų, kiek planetoj gyvena *gucikų*, *mucikų* ir *fucikų*.

Pirmoje pradinių duomenų failo eilutėje **gucikai_data.txt** įrašyti trys skaičiai: kiek iš viso planetoje gyvenančios būtybės turi rankų, kojų ir akių. Antroje eilutėje nurodyti vieno *guciko*,

trečiojoje – vieno *muciko*, ketvirtojoje – vieno *fuciko* rankų, kojų ir akių skaičiai. Planetoje gyvena ne daugiau kaip 500 kiekvienos rūšies būtybių ir kiekviena būtybė gali turėti ne daugiau kaip 20 rankų, ne daugiau kaip 20 kojų ir ne daugiau kaip 20 akių.

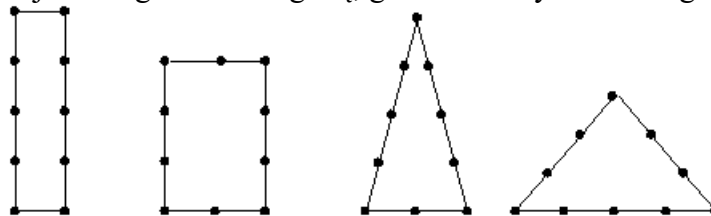
Rezultatų faile **gucikai_res.txt** turi būti įrašyti planetoje gyvenančių *gucikų*, *mucikų* ir *fucikų* skaičiai.

gucikai_data.txt	gucikai_res.txt
39 19 20 2 2 3 4 1 1 1 5 3	3 8 1

16. Figūros iš virvutės (X olimpiada, 1999 m.)

Ilgoje virvutėje kas 10 cm užmegzta po mazgelį (virvutės pradžioje užrištas mazgelis, o gale jo nėra). Lenkdami per mazgelius iš virvutės galime sudaryti tokias geometrines figūras: trikampius, kvadratus, stačiakampius.

Pavyzdžiui, kai virvelėje užmegzta 10 mazgelių, galime sudaryti tokias figūras:



Žinodami užrištą virvutėje mazgelių skaičių pabandykite sudaryti kuo daugiau skirtingų anksčiau minėtų geometrinių figūrų (sutampančių figūrų neturi būti). Vienodos formos figūros laikomos skirtingomis, jei jų kraštinių ilgių yra skirtingi.

Parašykite programą šiam uždaviniui išspręsti, kai *pradinis duomuo* – mazgelių skaičius, įrašytas faile **figuros_data.txt**, o *rezultatai* – sudarytų trikampių, kvadratų ir stačiakampių skaičiai atskirai. Skaičiuodami stačiakampių skaičių, kvadratų nelaikykite stačiakampiais. Rezultatai įrašomi į failą **figuros_res.txt**.

figuros_data.txt	figuros_res.txt
16	5 1 3

17. Miegas (XVIII olimpiada, 2006-2007 m.m.)

Adomas neturi žadintuvo, taigi, norėdamas nepramiegoti egzaminų, naudojami mobiliojo telefono žadintuvo funkcija. Tačiau keli telefono klaviatūros mygtukai sugedo ir jis nebegali įvesti tikslaus skambėjimo laiko. Telefone ir valandai, ir minutei įvesti skiriama po du skaitmenis (pavyzdžiui, 9:58 įvedama kaip 0 9 5 8). Padėkite Adomui įvesti tokį laiką, kad jis nepavėluotų atsikelti, tačiau galėtų miegoti kuo ilgiau.

Parašykite programą, randančią, kokį laiką reikėtų įvesti veikiančiais mygtukais, kad praėjęs laikas nuo telefono žadintuvo įsijungimo iki laiko, kada Adomas norėtų atsikelti, būtų kuo trumpesnis.

Laikoma, kad Adomas žadintuvą nustato likus parai iki laiko, kada norėtų atsikelti.

Pradiniai duomenys pateikti faile **miegas_data.txt**. Pirmoje eilutėje įrašyti keturi tarpais atskirti skaitmenys, kur pirmi du nurodo valandą, kiti du – minutes. Tai laikas, kada Adomas norėtų atsikelti. Antroje eilutėje įrašytas veikiančių mygtukų skaičius N ($1 \leq N \leq 9$). Trečioje eilutėje įrašyta N skirtingų tarpais atskirtų sveikųjų skaičių (iš intervalo $[0..9]$), kurie nurodo veikiančius mygtukus.

Pradiniai duomenys yra visada korektiški. Tai reiškia, kad veikiančiais mygtukais visada įmanoma įvesti taisyklingą laiką.

Rezultatus programa turi įrašyti į failą **miegas_res.txt**. Pirmoje ir vienintelėje eilutėje turi būti įrašyti keturi tarpais atskirti skaitmenys: pirmi du nurodo valandą, kiti du - minutes.

miegas_data.txt	miegas_res.txt	Paaiškinimai
0 9 0 0 9 0 1 2 3 4 5 6 7 8	0 8 5 8	Tikslaus (9:00) laiko įvesti neįmanoma, nes neveikia skaičius 9. Artimiausias laikas, kurį galima įvesti yra 8:58.
1 1 1 1 1 1	1 1 1 1	Įmanoma įvesti tikslų laiką.

18. Karuselės (XI olimpiada, 2000 m.)

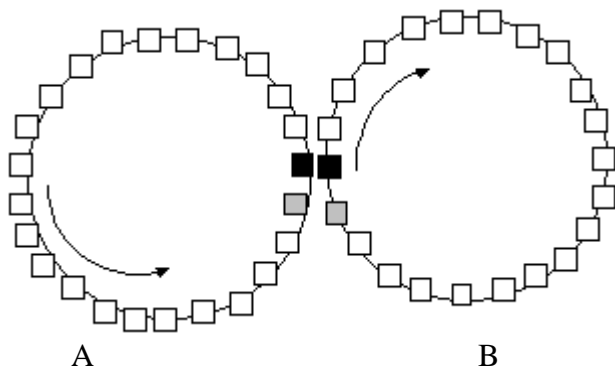
Priešingomis kryptimis sukasi dvi karuselės A ir B (žr. pav.). Jos turi a ir b vagonėlių.

Toje pačioje karuselėje negalima pereiti iš vieno vagonėlio į kitą. Tačiau karuselės pastatytos taip arti viena kitos, kad jų sąlyčio vietoje galima peršokti iš vienos karuselės į kitą.

Kiekvienu laiko momentu liečiasi tik po vieną kiekvienos karuselės vagonėlį. Karuselės sukasi tokiu greičiu, kad praėjus vienam laiko momentui liečiasi vagonėliai, gretimi vagonėliams, kurie lietėsi ankstesniu laiko momentu.

Žemiau esančiame paveiksle liečiasi juodai nuspalvinti vagonėliai. Praėjus vienam laiko momentui liesis pilki vagonėliai.

Pavyzdžiui, karusele A važiuoja keleivis. Jis gali peršokti į karuselės B vagonėlį ir apsisukus karuselei B vėl grįžti į kažkurį karuselės A vagonėlį.



Užduotis. 1) Žinomi vagonėlių skaičiai abiejose karuselėse a ir b ($20 \leq a, b \leq \text{maxlongint}$). Pradiniu momentu keleivis yra karuselės A vagonėlyje. Reikia suskaičiuoti, keliuose skirtinguose karuselės A vagonėliuose gali pabuvoti keleivis pasinaudodamas karusele B.

2) Žinomas karuselės A vagonėlių skaičius a . Reikia rasti,

koks turi būti mažiausias karuselės B vagonėlių skaičius b , kad keleivis iš bet kurio karuselės A vagonėlio per karuselę B galėtų pakliūti į bet kurį kitą karuselės A vagonėlį.

Parašykite vieną programą, kuri rastų abu skaičius (t. y. drauge išspręstų uždavinio pirmą ir antrą dalis).

Pradiniai duomenys įrašyti pradinių duomenų faile **karuseles_data.txt**. Pirmą pateiktas karuselės A vagonėlių skaičius *a*, po to – karuselės B vagonėlių skaičius *b*. Abu skaičiai yra toje pačioje eilutėje.

Skaičius *a* yra pradinis duomuo abejoms uždavinio dalims, skaičius *b* – tik pirmajai daliai.

Rezultatus – pirmosios bei antrosios uždavinio dalies atsakymus – spausdinkite į dvi rezultatų failo **karuseles_res.txt** eilutes: pirmoje eilutėje – pirmosios dalies, antroje – antrosios dalies rezultata.

karuseles_data.txt	karuseles_res.txt
24 22	12
	23