

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

PROGRAMAVIMO PAMOKĖLIŲ UŽDUOTYS

2016-2017 M.M.

Užduotis rengė: Aidas Eglinskas, Ugnius Skučas, Arnas Valenta, Irmantas Lukenskas, Deividas Andrikis, Deimantas Makrickas, Jurgis Gečys, Edvinas Šulčius, Mangirdas Kazlauskas, Lukas Skamarakas, Paulius Melaika, Karolina Šarauskaitė, Gytis Vaičekauskas, Renata Burbaitė

Kaunas

2016-2017

Turinys

1. TIESINIAI ALGORITMAI	6
1. Rubiko kubas	6
2. Slyvos	6
3. Rankinis laikrodis	6
4. Vaflių kepimas	7
5. Alpinariumas	7
6. Žaidimas su skaičiaus skaitmenimis	7
7. Ruloninės vejos kaina	8
2. ŠAKOTIEJI ALGORITMAI	8
1. Kompasas	8
2. Keturženklis skaičiaus formavimas	8
3. Lenktynės	9
4. Žalos skaičiavimas	9
5. Stačiakampio ketvirtoji viršūnė	10
3. CIKLINIAI ALGORITMAI. ŽINOMO KARTOJIMŲ SKAIČIAUS CIKLAS	10
1. Superfibonačio skaičiai	10
2. Ar tiesės susikerta?	10
3. Laimingi keturženkliai skaičiai	11
4. Miegalius	11
5. UEFA Europos futbolo čempionatai	12
4. CIKLINIAI ALGORITMAI. ŽINOMO KARTOJIMŲ SKAIČIAUS CIKLAS. SANDAUGOS, SUMOS, KIEKIO, DIDŽIAUSIOS/MAŽIAUSIOS REIKŠMĖS SKAIČIAVIMAS	12
1. Faktorialas	12
2. Atvirkštinis faktorialas	12
3. Ar skaičius pirminis?	13
4. Fejerverkai	13
5. Slibinų dresuotojas	13
6. Skaičių patikra	14

5. CIKLINIAI ALGORITMAI. NEŽINOMO KARTOJIMŲ SKAIČIAUS CIKLAS	14
1. Namų darbai	14
2. Akmenuko skrydis	15
3. Pyragų kepimas.....	15
4. Kovos simuliacija	15
5. Skaičiavimo sistemos	16
6. CIKLAS CIKLE.....	17
1. Skaičių eilutės suma	17
2. Daugybės lentelė	17
3. Akučių suma	18
4. Palindromai.....	18
5. Pitagoro trejetai.....	18
7. FUNKCIJOS, GRAŽINANČIOS APSKAIČIUOTĄ REIKŠMĘ PER FUNKCIJOS VARDA.....	19
1. Lagranžo dėsnis	19
2. Laimingi skaičiai	20
3. Užrašas ant Egipto piramidės	20
4. Berberų kalendorius.....	21
Reikalavimai.....	22
8. FUNKCIJOS, GRAŽINANČIOS APSKAIČIUOTAS REIKŠMES PER PARAMETRUS	22
1. Bulviakasis.....	22
2. Atspėk skaičių.....	22
3. Vampyro skaičiai.....	23
4. Sieksniai, varstai ir mylios.....	24
9. MASYVAI. SUMOS, KIEKIO, VIDURKIO SKAIČIAVIMAS.....	24
1. Kalėdinės dovanos	24
2. Sodininko derlius	25
3. Miesto autobusai.....	27
4. Detalių rūšiavimas	28

10.	MASYVAI. DIDŽIAUSIOS IR MAŽIAUSIOS REIKŠMIŲ PAIEŠKA	29
1.	Kompiuterinių žaidimų populiarumas	29
2.	Vasaros kelionės	30
3.	Juodasis penktadienis	32
4.	Miesto autobusų maršrutai.....	33
11.	MASYVAI. ELEMENTŲ ŠALINIMAS IR ĮTERPIMAS, RIKIAVIMAS, NAUJO SĄRAŠO FORMAVIMAS	34
1.	Sodo apšvietimas	34
2.	Virtualios realybės akiniai	35
3.	Vairuotojų pažymėjimai	36
4.	Paguodos savaitė.....	37
5.	Laimingi loterijos bilietai	38
12.	KĄ SLEPIA MASYVO INDEKSAS?	39
1.	Temperatūra.....	39
2.	Statistikos protokolas.....	40
3.	Rezultatyviausias žaidėjas	41
4.	Pinigai.....	41
13.	SIMBOLIAI. SIMBOLIŲ EILUTĖS	42
1.	Heteroliteralai	42
2.	Mažiausiai kartų panaudoti simboliai.....	42
3.	Valdovės	43
4.	Drugio efektas.....	44
5.	Anagrama.....	44
14.	STRUKTŪROS, STRUKTŪRŲ MASYVAI	45
1.	KMI	45
2.	Nauji mokesčiai	46
3.	Juodasis penktadienis	47
15.	DIDŽIAUSIOS IR MAŽIAUSIOS REIKŠMIŲ PAIEŠKA STRUKTŪRŲ MASYVE	50
1.	Akimirkos tyrimas	50
2.	Grojaraštis.....	50
3.	Mokinių sąrašas	52
4.	Žaidimas „Laimingas ūgis“	53

16.	STRUKTŪRŲ MASYVŲ ELEMENTŲ RIKIAVIMAS	54
1.	Asmens kodas	54
2.	Pažangumas	55
3.	Pasiklydę skaičiai	56
4.	Biblioteka.....	57
17.	STRUKTŪRŲ MASYVŲ ELEMENTŲ ŠALINIMAS IR ĮTERPIMAS	58
1.	Krepšinio komanda.....	58
2.	Informacijos saugojimas.....	59
3.	Mokinių ekskursija	61
4.	Studentų sąrašas.....	62
18.	ĮVAIRŪS STRUKTŪRŲ MASYVŲ UŽDAVINIAI.....	64
1.	Perkamiausios prekės	64
2.	Savaitgalis.....	65
3.	Tarptautinė migracija.....	67
4.	Sportininkų ir sporto trenerių skaičius.....	69

1. TIESINIAI ALGORITMAI



1. Rubiko kubas

Rokui labai patinka susigalvoti užduotis programavimui. Kartą dėliojant Rubiko kubą, jam pasidarė įdomu: kiek iš viso kubelių yra išorėje ir kiek kubelių dar tilptų viduje.

Pradiniai duomenys: vienoje eilėje esančių kubelių kiekis.

Padėkite Rokui pabaigti rašyti programą, kuri surastų, kiek iš viso yra kubelių Rubiko kubo išorėje ir kiek tilptų tokių kubelių viduje.

Pradiniai duomenys	Rezultatai
3	26 1

2. Slyvos

Trys draugės Rasa, Rita ir Rima kartu paėmus suvalgė s slyvų. Rita suvalgė k slyvų daugiau, negu Rasa, o Rima m slyvų mažiau, negu kitos dvi mergaitės kartu paėmus.

Parašykite programą, skaičiuojančią, po kelias slyvas suvalgė kiekviena mergaitė. Spręsdami uždavinį laikykite, kad k reikšmė visada bus tokia, kad mergaitėms išsidalinus slyvas, nė vienos slyvos neliks.

Pradiniai duomenys	Rezultatai
Kelios slyvos suvalgytos? 17 Keliomis slyvomis Rita suvalgė daugiau, negu Rasa? 3 Keliomis slyvomis Rima suvalgė mažiau, negu kitos dvi mergaitės? 5	Rasa: 4 Rita: 7 Rima: 6

3. Rankinis laikrodis

Jonas nori būti tikru verslininku. Kad suspėtų į visus posėdžius, jis nusipirko mechaninį rankinį laikrodį. Skubėdamas susidūrė su praeiviu ir laikrodžio ciferblatas pasisuko dešinėn 90 laipsnių kampą.

Duotas po ciferblato pasisukimo rodomas laikas laikrodyje valandomis ir minutėmis. Padėkite Jonui nustatyti, koks laikas buvo rodomas prieš susidūrimą valandomis ir minutėmis.

Pradiniai duomenys	Rezultatai
10 13	1 28

4. Vaflių kepimas

Marytė vakarienei užsinorėjo vaflių. Susiradusi keturkampę vaflių keptuvę ir pradėjusi kepti vaflius pastebėjo, kad jos vaflių keptuvės paviršius yra iškilus lygiai per vidurį ir kepami vafliai apskrunda iš visų pusių tolygiai.

Duotas vaflių keptuvės aukštis, plotis ir vaflio apskrudimo dydis. Visi skaičiai yra sveikieji, neneigiami. Apskrudimo plotas neviršys vaflių keptuvės ploto.

Apskaičiuokite neapskrudusio kepinio plotą.

Pradiniai duomenys	Rezultatai
10 10 4	4

5. Alpinariumas

Kamilė savo sodelyje daro kvadrato formos alpinariumą su nedideliu fontanu jo centre. Norėdama pajavairinti vaizdą, Kamilė sugalvojo vieną pusę alpinariumo padengti šviesiai mėlynos spalvos akmenukais, o kitą – šviesiai raudonos spalvos akmenukais. Apskaičiuokite kokį plotą uždengs šviesiai raudonos spalvos akmenukai, jeigu žinomas alpinariumo kraštinės ilgis **A** ir fontano skersmuo **R**.

Pradiniai duomenys

Pirmoje eilutėje alpinariumo kraštinės ilgis (**A**), antroje – fontano skersmuo (**R**). **A**, **R** – priklauso natūraliųjų skaičių aibei. $A > R > 0$

Išvestis

Į ekraną išvedamas tik raudonų akmenukų plotas, suapvalintas iki 3 skaitmenų po kablelio.

Pradiniai duomenys	Rezultatai
4 1	7.607
Pradiniai duomenys	Rezultatai
10 3	46.468

6. Žaidimas su skaičiaus skaitmenimis

Yra daug žaidimų, susijusių su naujų skaičių sudarymu panaudojant žinomo natūraliojo skaičiaus skaitmenis. Parenkite programą, kuri iš įvesto natūraliojo triženklis skaičiaus x suformuotų naują skaičių s , tenkinantį žemiau nurodytas sąlygas:

a. Pirmasis naujo skaičiaus skaitmuo gaunamas padidinus pirmąjį pradinio skaičiaus skaitmenį dviem. Jei padidinus gaunamas dviženklis skaičius, tuomet pirmuoju skaitmeniu tampa antrasis dviženklis skaičiaus skaitmuo.

b. Antrasis naujo skaičiaus skaitmuo gaunamas sudėjus pradinio skaičiaus pirmąjį ir paskutinįjį skaitmenis. Jei gaunamas dviženklis skaičius, tuomet antruoju skaitmeniu tampa antrasis dviženklis skaičiaus skaitmuo.

c. Trečiasis naujo skaičiaus skaitmuo gaunamas padidinus antrąjį pradinio skaičiaus skaitmenį 2 kartus. Jei gaunamas dviženklis skaičius, tuomet trečiuoju skaitmeniu tampa antrasis dviženklis skaičiaus skaitmuo.

Pradiniai duomenys	Rezultatai
759	960

7. Ruloninės vejės kaina

Saulė aptvėrė aptvėrė lygiakraštį trikampį, kurio perimetras yra a metrų (a - sveikasis skaičius), kuriame planuoja užveisti ruloninę veją. Kiek kainuos vejės įrengimas, jei vieno kvadratinio vejės metro kaina yra k (k realusis skaičius) eurų. Atsakymą pateikite 2 ženklų po kablelio tikslumu be aiškinamojo teksto.

Pradiniai duomenys: $a = 15$, $k = 3.5$. Rezultatas: vejės kaina bus lygi 37.89.

2. ŠAKOTIEJI ALGORITMAI

1. Kompasas

Jonas gavo užduotį: reikia sukurti programą, kuri apskaičiuotų, kokių mažiausių kampų turi pasisukti kompasas rodyklė, kad iš dabartinės pozicijos pasiektų norimą poziciją.

Pradiniai duomenys: dabartinė rodyklės pozicija laipsniais ir norima rodyklės pozicija laipsniais (sveikieji skaičiai). Apskaičiuokite trumpiausią pokytį laipsniais tarp pradinės pozicijos ir norimos. Teigiamas skaičius parodo, jog kompasas rodyklė suksis pagal laikrodžio rodyklę, neigiamas – prieš laikrodžio rodyklę. Jei pradinė ir galutinė kompasas rodyklės pozicijos sudaro 180 laipsnių kampą, tuomet kompasas rodyklė turėtų suktis pagal laikrodžio rodyklę.

Pradiniai duomenys	Rezultatai
315 45	90
180 270	90

2. Keturženklis skaičiaus formavimas

Parenkite programą, kuri iš įvesto natūraliojo keturženklis skaičiaus x suformuotų naują skaičių s , tenkinantį nurodytas sąlygas:

a. Naujai suformuotame skaičiuje pirmiausia eina lyginiai skaitmenys, po jų - nelyginiai. Nulis laikomas lyginiu skaitmeniu.

b. Naujame skaičiuje skaitmenys turi būti išdėstyti mažėjimo tvarka.

Pradiniai duomenys	Rezultatai
7590	957
1230	2031

3. Lenktynės

Trys draugai susirungė žaisdami kompiuterinį žaidimą - mašinų lenktynes. Važiavo tris ratus ir fiksavo, per kiek laiko kiekviena mašina įveikė ratą.

Duotos trys eilutės, kiekvienoje eilutėje yra dalyvavusiojo žaidime vardas (naudokite duomenų tipą string) ir kiekvieno rato įveikimo laikas (sekundėmis).

Ekrane turi būti rodomos prizinės vietos nuo pirmos iki trečios su žaidėjų vardais.

Pastaba: Laikykite, kad lygiųjų niekada nebus.

Pradiniai duomenys	Rezultatai
Aidas 536 546 587	1) Aidas
Rokas 587 566 573	2) Lukas
Lukas 556 597 536	3) Rokas

4. Žalos skaičiavimas

Žaidime yra 3 žaidėjų klasės: karys, magas ir lankininkas. Kiekviena iš šių klasių turi skirtingą sugebėjimą, nuo kurio priklauso žala priešininkui.

Sugebėjimai:

- Karys. Smūgio žala: **124 + 7%** priešininko turimų gyvybių.
- Magas. Burto žala: **240 + 120 jeigu priešininkas turi 2000 gyvybių ir daugiau.**
- Lankininkas. Strėlės žala: 180; **Lankininkai vienu šūviu šauna 3 strėles iš karto.**

Apskaičiuokite, kiek žalos padarys žaidėjas, žinodami jo klasę ir priešininko gyvybių skaičių.

Pradiniai duomenys. Pirmoje eilutėje pateikiama žaidėjo klasė, antroje - priešininko gyvybių skaičius.

Rezultatai. Išvesti žaidėjo padaromą žalą 2 skaitmenų po kablelio tikslumu.

Duomenys	Rezultatai
Karys 1000	194.00
Magas 2000	360.00
Lankininkas 5150	540.00

5. Stačiakampio ketvirtoji viršūnė

Lukas konstruoja stačiakampį, kurio kraštinės yra lygiagrečios koordinačių ašims. Berniukas pasirinko 3 stačiakampio viršūnių koordinates. Padėkite Lukui tinkamai pasirinkti ketvirtosios viršūnės koordinates.

Pradiniai duomenys: Luko pasirinktų 3 stačiakampio viršūnių koordinatės x ir y (sveikieji skaičiai ne didesni už 100).

Ekrane vienoje eilutėje turi būti rodomos 4-osios viršūnės koordinatės x ir y atskirtos tarpu.

Pradiniai duomenys	Rezultatai
5 5 5 7 7 5	7 7

3. CIKLINIAI ALGORITMAI. ŽINOMO KARTOJIMŲ SKAIČIAUS CIKLAS

1. Superfibonačio skaičiai

Tikriausiai esate girdėję Fibonačio skaičių seką: 1 1 2 3 5 8. Ji apibrėžiama tokiu būdu: $F_1 = 1$, $F_2 = 1$, $F_n = F_{n-1} + F_{n-2}$, t.y kiekvienas narys, pradedant trečiuoju, lygus prieš jį esančių dviejų narių sumai. Panašiai apibrėžkime *superfibonačio* skaičius: $F_1 = 1$, $F_2 = 1$, $F_3 = 1$, $F_n = F_{n-1} + F_{n-2} + F_{n-3}$, t.y pirmieji trys sekos nariai yra vienetai, o kiekvienas tolesnis narys gaunamas sudėjus tris paskutinius sekos narius. Parenkite programą, kuri rastų n -ąją *superfibonačio* sekos narį.

Eilutėje įrašytas vienas sveikasis skaičius n ($1 \leq n < 20$) - ieškomo *superfibonačio* sekos nario numeris. Apskaičiuokite n -ąją *superfibonačio* sekos narį.

Pradiniai duomenys	Rezultatai	Paaškinimas
6	9	1, 1, 1, 3, 5, 9

2. Ar tiesės susikerta?

Per matematikos pamokas tenka spręsti uždavinius, kuriuose prašoma surasti tašką, kuriame susikerta dvi tiesės.

Parenkite programą, kuri, įvedus tiesių $y = kx + b$ ir $t = mx + a$ koeficientų reikšmes k , b , m ir a , bei sveikųjų skaičių intervalą $[x_1; x_2]$, kuriame kinta x reikšmės, į ekraną išvestų rezultatus kaip pateikta pavyzdyje. Visi duomenys ir rezultatai yra sveikąjo tipo.

```

Pirmosios tieses koeficientai k ir b
2 3
Antrosios tieses koeficientai m ir a
3 2
Intervalo pradzia ir pabaiga
-5 5
x      y      t      Susikirtimas
-5     -7     -13    ***
-4     -5     -10    ***
-3     -3     -7     ***
-2     -1     -4     ***
-1     1      -1     ***
0      3      2     ***
1      5      5     susikerta
2      7      8     ***
3      9      11    ***
4      11     14    ***
5      13     17    ***

```

3. Laimingi keturženkliai skaičiai

Keturženklis skaičius x vadinamas laimingu, jei jo pirmųjų dviejų skaitmenų ir paskutiniųjų dviejų skaitmenų sumos kvadratas yra lygus x .

Parašykite programą, kuri ekrane išspausdintų keturženklių skaičių intervalo $[a; b]$ laimingus skaičius.

Jei $a = 1000$, $b = 5000$, tuomet ekrane turi būti rodoma:

2025

3025

4. Miegalius

Romas labai mėgsta rytais keltis paskutinę minutę. Romo mama n dienų registravo, kuriuo laiku berniuką žadina žadintuvas (valanda ir minutė), ir kada Romas atsikelia (valanda ir minutė).

Parenkite programą, kuri apskaičiuotų, keliomis minutėmis Romas kiekvieną dieną atsikeldavo vėliau, negu nuskambėdavo žadintuvas. Pirmiausia įvedamas dienų skaičius, po to iš eilės kiekvienos dienos duomenys: žadintuvo skambėjimo valanda ir minutė ir Romo atsikėlimo valanda ir minutė. Ekrane turi būti rodomas dienos numeris ir skirtumas tarp žadintuvo skambėjimo ir Romo kėlimosi laiko minutėmis.

Pradiniai duomenys	Rezultatai
3	1 20
7 30 7 50	2 5
7 10 7 15	3 15
8 10 8 25	

5. UEFA Europos futbolo čempionatai

UEFA Europos futbolo čempionatai pradėti rengti 1960 metais ir vyksta kas ketveri metai. Parašykite programą, kuri iš nurodyto metų intervalo [m; k] išrinktų ir kompiuterio ekrane išvestų tuos metus, kuriais vyko futbolo čempionatas ir nurodytų, kelintas čempionatas vyko tais metais.

Kai $m = 1960$, $k = 1970$, tuomet ekrane turi būti rodoma:

1960 1

1964 2

1968 3

4. CIKLINIAI ALGORITMAI. ŽINOMO KARTOJIMŲ SKAIČIAUS CIKLAS. SANDAUGOS, SUMOS, KIEKIO, DIDŽIAUSIOS/MAŽIAUSIOS REIKŠMĖS SKAIČIAVIMAS

1. Faktorialas

Natūraliojo skaičiaus n faktorialu vadinama visų natūraliųjų skaičių nuo **1 iki n sandauga imtinai**. Apskaičiuokite skaičiaus n faktorialą. Pradiniai duomenys visada bus teigiamas skaičius (įskaitant nulį) ir neviršys 10.

Pradiniai duomenys (n)	Rezultatai
0	1
1	1
2	2
3	6

2. Atvirkštinis faktorialas

Natūraliojo skaičiaus n atvirkštiniu faktorialu vadinamas skaičius x , kurio faktorialas yra lygus n . T.y. $x!$ yra lygus n . Apskaičiuokite skaičių x ir išspausdinkite jį. Teigiamas kintamasis n mažesnis už 4 milijonus ir visi duomenys bus pateikti taip, jog faktorialas visada egzistuos.

Pradiniai duomenys (n)	Rezultatai
1	0
1	1
2	2
6	3

3. Ar skaičius pirminis?

Pirminis skaičius yra bet kuris natūralusis skaičius, didesnis nei 1, kuris dalinasi tik iš savęs ir vieneto. Vienetas nelaikomas nei pirminiu skaičiumi, nei sudėtinium.

Keletas mažiausių pirminių skaičių:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, ...

Pirminių skaičių yra be galo daug.

Parašyti programą, patikrinančią, ar duotasis skaičius yra pirminis.

Pradiniai duomenys (n)	Rezultatai
1	false
3	true
4	false
7	true

4. Fejerverkai

Švenčių proga į orą šaudomi fejerverkai. Rokas žiūri į fejerverkus ir naudodamas specialią įrangą nustato, į kokį aukštį pakilo kiekvienas iššautas fejerverkas.

Pirmoje eilutėje duotas skaičius n , kuris žymi kiek iš viso buvo iššauta fejerverkų. Kiti n skaičių parodo aukštį, kurį pasiekė kiekvienas fejerverkas.

Raskite, kelintas buvo iššautas fejerverkas, kuris pakilo į aukščiausią aukštį, koks tai aukštis ir kiek vidutiniškai pakilo vienas fejerverkas.

Pradiniai duomenys	Rezultatai
5 15.41 12.3 16.32 13.03 11.01	Aukščiausiai pakilęs fejerverkas buvo iššautas: 3 Pakilo į 16.32 m aukštį. Vienas fejerverkas vidutiniškai pakilo: 13.614 m.

5. Slibinų dresuotojas

Senolis slibinų dresuotojas Jonas, turintis 15 metų patirtį savo srityje, pastebėjo, jog nebespėja atlikti visų reikalingų skaičiavimų sekti slibinų pažangai, tad jis prašo tavo pagalbos. Kiekvieną dieną jo slibinas yra dresuojamas t minučių, t.y pirmą t_1 , antrąją - t_2 ir t.t. Parenk programą, kuri apskaičiuotų, kiek minučių buvo dresuojamas slibinas, jei Jonas jį dresavo n dienų, ir kiek vidutiniškai minučių slibinas buvo dresuojamas per dieną.

Pirmoje pradinių duomenų eilutėje įrašytas dienų skaičius n ($0 \leq n \leq 10$). Tolesnėse n eilučių įrašyta po vieną sveikąjį skaičių t ($0 \leq t \leq 1440$) - kiek minučių buvo dresuojamas slibinas kiekvieną dieną.

Išveskite vieną po kito kiek minučių buvo dresuojamas slibinas per **n** dienų ir kiek vidutiniškai minučių slibinas buvo dresuojamas per dieną (sveikasis skaičius).

Pradiniai duomenys	Rezultatai
4 20 12 13 15	60 15

6. Skaičių patikra

Programai pateikiama tam tikra skaičių seka, kurios dydis žinomas iš anksto. Programa privalo kiekvieną skaičių, kuris prasidenda antroje sekos pozicijoje, sulyginti su prieš juo esančiu skaičiumi. Jeigu skaičiai sutampa, programa iškart spausdina **true**, jeigu ne - **false**. Sulyginus visus duotos sekos skaičius programos pabaigoje reikia išspausdinti didžiausią sekoje esantį skaičių.

Pradiniai duomenys. Pirmoje eilutėje pateikiamas skaičių sekos dydis **x**. Toliau **x kartų** pateikiami sveikieji skaičiai.

Rezultatai. Sutapus dviem sulygintiems skaičiams eilutėje išspausdinti **true**, priešingu atveju – **false**. Programos išvesties pabaigoje išspausdinti didžiausią tos sekos skaičių.

Ivestis	Išvestis	Ivestis	Išvestis
5	false	5	true
1	false	1	false
2	false	1	true
3	false	2	false
4	5	2	3
5		3	

5. CIKLINIAI ALGORITMAI. NEŽINOMO KARTOJIMŲ SKAIČIAUS CIKLAS

1. Namų darbai

Juožiaus informatikos mokytojas uždavė namų darbą: parašyti algoritmą, kuris suskaičiuotų skaičiaus skaitmenų kiekį. Kadangi Juozis nelabai išmano programavimą, jis pasikvietė tave padėti išspręsti šią problemą. Salygoje parašyta jog skaičius gali būti ir teigiamas, ir neigiamas, bet jį sudaro ne daugiau kaip dešimt skaitmenų.

Pradiniai duomenys	Rezultatai
-1	1
0	1
1	1

2. Akmenuko skrydis

Rokas su keliais klasiokais sukonstravo specialią įrangą, kuri leido nustatyti, kaip toli nuskrieja mestas akmenukas. Visi draugai metė po akmenuką, o kompiuteris viską registruoja ir duomenų įvestis baigiama užrašant "-1". Padėkite parašyti programą, kuri surastų toliausiai nuskridusį akmenuką, bei išspausdinkite rezultatą.

Pradiniai duomenys	Rezultatai
16.41	16.41
13.30	
14.32	
12.03	
16.00	
-1	

3. Pyragų kepimas

Laimonas sukūrė žaidimą apie pyragų kepimą. Žaidžiant šį žaidimą buvo gauti rezultatai, kuriuos reikia apdoroti.

Pirmoje eilutėje duotas laikas, kuris nurodo, kiek laiko sekundėmis buvo skirta pyragams kepti. Kitose eilutėse pateikiama po vieną natūralųjį skaičių, nurodantį, kiek sekundžių kepė kiekvienas pyragas. Pasitaikydavo atvejų, kai pyragai buvo kepami ir praėjus leistinam laikui, tačiau atvejų, kada pyragai buvo kepami trumpiau, niekada nepasitaikė.

Parašykite programą, kuri išvestų, kiek pyragų spėjo iškepti, jeigu visus juos kepė iš eilės.

Pradiniai duomenys	Rezultatai
10	2
4	
5	
4	
3	

4. Kovos simuliacija

Žaidimo kovos principas pagrįstas ėjimais. Kova baigiasi tada, kai vienas iš kovotojų nebegali tęsti kovos, t. y. nebeturi gyvybių. Kovotojas, kuris pasibaigus kovai turi daugiausiai gyvybių paskelbiamas laimėtoju. Žaidžiant žaidimą kiekvienas žaidėjas priskiriamas tam tikrai klasei.

Žaidėjų klasių aprašai

Karys

Kiekvieną ėjimą kerta po 87 gyvybės taškus. Jeigu kova tęsiasi daugiau nei 5 ėjimus, aktyvuojasi šios klasės pasyvus gebėjimas, leidžiantis kirsti dvigubai daugiau.

Magas

Kiekvieną ėjimą kerta po 47 + 1.5% priešininko maks. gyvybių. Kas 7-tą ėjimą magas įgyja 30 gyvybių vertės skydą, kuris nuo žalos jį apsaugo tik tam ėjimui.

Tankas

Kiekvieną ėjimą kerta po 64 gyvybės taškus. Kas 2 ėjimus jo smūgis įgyja 14% nuo jo turimų gyvybių taškų žalos premiją ir tuo pačiu ėjimu ignoruoja 21% gautos žalos.

Monstras

Kiekvieną ėjimą kerta po 32 gyvybės taškus. 5% nuo patirtos žalos sugeria ir prideda prie savo gyvybių.

Pradiniai duomenys. Pirmoje įvesties eilutėje pateikiama žaidėjo klasė. Antroje - gyvybių skaičius. Trečioje eilutėje nurodomas priešininko gyvybių skaičius. Priešininko klasė visada bus **Monstras**.

Rezultatai. Pirmoje eilutėje spausdinti **true**, jeigu laimi žaidėjas, priešingu atveju – **false**. Antroje eilutėje spausdinti įvykusių ėjimų skaičių.

Įvestis	Išvestis
Karys 750 245	true 4
Magas 448 320	true 7
Tankas 1000 750	true 8

5. Skaičiavimo sistemos

Petriukas ruošiasi informatikos olimpiadai. Benaršydamas internete užklydo į puslapį (<http://www.mif.vu.lt/~ragaisis/InfIvadas/SkSistemas.htm>) su skaičiavimo sistemomis ir nutarė pabandyti parašyti universalų algoritmą, kuris dešimtainės skaičiavimo sistemos skaičius pervestų į bet kokią kitą skaičiavimo sistemą. Pastebėjęs jog jam gali būtų per sunku, nutarė ignoruoti visas skaičiavimo sistemas, kurių pagrindas didesnis už 10. Padėkite Petriukui sukurti algoritmą, kuris apskaičiuotų ir išvestų rezultatą, kuris gaunamas dešimtainį skaičių pervedus į kitas skaičiavimo sistemas, kurių pagrindas ne didesnis už 10.

Duotas dešimtainis skaitmuo ir skaičiavimo sistemos pagrindas. Visi duomenys yra teisingi ir rezultatas neviršys int tipo ribų.

Pradiniai duomenys	Rezultatai
0 2	0
1 2	1
7 3	21
7 5	12

6. CIKLAS CIKLE

1. Skaičių eilutės suma

Parašykite programą, kuri rastų eilutės sumą

$1^k + 2^k + 3^k + \dots + n^k$. Čia n ir k – sveikieji skaičiai.

Kai $n = 3$, $k = 2$, tuomet suma lygi 14.

2. Daugybės lentelė

Sudaryti daugybės lentelę, kurioje yra n eilučių ir m stulpelių. Spausdindami reikšmes taikykite kairinę lygiuotę, rezultatui skirkite 5 pozicijas.

Kai $n = 10$ ir $m = 10$, tuomet lentelė turėtų būti tokia:

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

3. Akučių suma

Grupė draugų, kurią sudaro n žmonių, paėliui meta kauliuką ir užsirašo kiekvieną kartą iškritusių akučių skaičių. Kauliuką kiekvienas žmogus meta po k kartų. Parašykite programą, skaičiuojančią, kokia kiekvieno draugo iškritusių akučių suma.

Pradiniai duomenys: pirmiausia įvedamos n ir k reikšmės. Toliau įvedama n eilučių sveikųjų skaičių $[1; 6]$, kurių kiekvienoje eilutėje yra po k .

3 5

1 1 2 1 2

3 2 1 2 2

5 1 1 1 1

Rezultatai - kiekvieno draugo akučių suma. Vieno draugo sumai skiriama viena eilutė.

7

10

9

4. Palindromai

Skaičiai, vienodai skaitomi iš kairės į dešinę ir iš dešinės į kairę, vadinami palindromais. Pvz., 121, 1221, 12221 yra palindromai. Parašykite programą, rodančią visus intervalo $[m; n]$ palindromus po vieną eilutėje.

Kai $m = 45$, $n = 105$, tuomet palindromai bus:

```
55
66
77
88
99
101
```

5. Pitagoro trejetai

Jeigu natūraliųjų skaičių a , b ir c trejetas (a,b,c) tenkina lygybę $a^2 + b^2 = c^2$, tai jis vadinamas Pitagoro trejetu.

Parenkite programą, spausdinančią visus natūraliųjų skaičių intervalo $[x; y]$ Pitagoro trejetus, kai trejeto skaičiai išsidėstę nuo mažiausio iki didžiausio. Vienam trejetui skiriama viena eilutė.

Kai $x = 1$, $y = 20$, tuomet Pitagoro trejetai yra:

3 4 5
5 12 13
6 8 10
8 15 17
9 12 15
12 16 20

7. FUNKCIJOS, GRAŽINANČIOS APSKAIČIUOTĄ REIKŠMĘ PER FUNKCIJOS VARDĄ

1. Lagranžo dėsnis

Remiantis Lagranžo dėsniu, kiekvieną natūralųjį skaičių galima išreikšti keturių sveikųjų skaičių kvadratų suma. Pvz.: $5 = 0^2 + 0^2 + 1^2 + 2^2$.

Parenkite programą, kuri duotą natūralųjį skaičių išreikštų keturių sveikųjų skaičių kvadratų suma visais galimais būdais.

Pradiniai duomenys	Rezultatai
7	1 1 1 2 1 1 2 1 1 2 1 1 2 1 1 1
12	0 2 2 2 1 1 1 3 1 1 3 1 1 3 1 1 2 0 2 2 2 2 0 2

Reikalavimai

Būtina loginė funkcija, tikrinanti, ar skaičių galima išreikšti keturių sveikųjų skaičių kvadratų suma.

2. Laimingi skaičiai

Natūralusis skaičius x laikomas laimingu, kai jį sudaro lyginis skaitmenų skaičius $2k$ ir kurio pirmųjų k skaitmenų suma yra lygi paskutiniųjų k skaitmenų sumai. Parenkite programą, surandančią visus intervalo $[m; n]$ laimingus skaičius.

Pradiniai duomenys	Rezultatai
1000 1100	1001 1010
100 1000	Nebus nė vieno laimingo skaičiaus

Reikalavimai

Būtinios funkcijos, grąžinančios apskaičiuotą reikšmę per funkcijos vardą:

- 1) skaitmenų kiekio skaičiavimo funkcija;
- 2) skaitmenų sumos skaičiavimo funkcija.

Programoje gali būti ir daugiau funkcijų, grąžinančių apskaičiuotą reikšmę per funkcijos vardą.

3. Užrašas ant Egipto piramidės

Viename sename programavimo uždavinyne Julius surado uždavinį apie ant piramidės akmeninės antkapio plokštės hieroglifais iškaltą natūralųjį skaičių n . Uždavinio autorius teigia, kad ant piramidės iškaltas skaičius be liekanos dalijasi iš visų natūrinių skaičių nuo 1 iki m (yra visų skaičių nuo 1 iki m mažiausias bendras kartotinis (MBK)).

Julius nusprendė patikrinti, ar galima intervalą susiaurinti, t.y. sumažinti m reikšmę, kad skaičius n liktų sutrumpėjusio intervalo MBK. Kol kas Juliaus programavimo žinių nepakanka uždaviniui išspręsti. Pagelbėkite jam.

Ekране reikia išspausdinti visus intervalo $[1; m]$ skaičius, kuriais pradėdant MBK yra lygus n .

Pradiniai duomenys: ant piramidės antkapio plokštės iškaltas skaičius n ir intervalo pabaigos skaičius m .

Pavyzdžiui, jei $n = 2520$, $m = 10$, tuomet MBK lygus 2520 yra šių intervalo skaičių: 9 ir 10 (žr. paaiškinantį paveikslėlį).

```
2520 10
[1;2] 2
[1;3] 6
[1;4] 12
[1;5] 60
[1;6] 60
[1;7] 420
[1;8] 840
[1;9] 2520
[1;10] 2520
```

Ekране turi būti spausdinama po vieną skaičių eilutėje:

9

10

Reikalavimai

Dviejų skaičių mažiausias bendras kartotinis gaunamas sudauginus du skaičius ir padalinus juos iš tų skaičių didžiausio bendro daliklio.

Spręsdami uždavinį parašykite 2 funkcijas, grąžinančias apskaičiuotą reikšmę per funkcijos vardą:

- didžiausio bendro daliklio;
- mažiausio bendro kartotinio.

4. Berberų kalendorius

Šiais laikais Šiaurės Afrikoje naudojamas berberų kalendorius. Tai yra šiuolaikinis kalendorius, kuris remiasi Julijaus kalendoriumi:

- **Tagrest:** žiema
 1. Jember: nuo gruodžio 14 d. iki sausio 13 d.;
 2. Yennayer: nuo sausio 14 d. iki vasario 13 d.;
 3. Furar: nuo vasario 14 d. iki kovo 13 d.
- **Tafsut:** pavasaris
 1. Meghres: nuo kovo 14 d. iki balandžio 13 d.;
 2. Ibrir: nuo balandžio 14 d. iki to gegužės 13 d.;
 3. Mayyu: nuo gegužės 14 d. iki birželio 13 d.
- **Iwilen:** vasara
 1. Yunyu: nuo birželio 14 d. iki liepos 13 d.;
 2. Yulyu: nuo liepos 14 d. iki rugpjūčio 13 d.;
 3. Ghust ou Awussu: nuo rugpjūčio 14 d. iki rugsėjo 13 d.
- **Amewan:** ruduo
 1. Shtember: nuo rugsėjo 14 d. iki spalio 13 d.;
 2. Tuber: nuo spalio 14 d. iki lapkričio 14 d.;
 3. Wamber: nuo lapkričio 14 d. iki gruodžio 13 d.

Parenkite programą, kuri, įvedus datą (mėnesį ir dieną), išreikštą sveikaisiais skaičiais, praneštų, koks metų laikas yra pagal berberų kalendorių.

Pavyzdžiui, jei mėnuo = 6, diena = 12, tuomet ekrane turi būti rodoma:

Tafsut: pavasaris

Jei bus įvesta neteisinga data, pvz., balandžio mėnuo turi 30 dienų, o bus įvesta balandžio 31 d., tuomet ekrane turi būti rodoma, kad įvesta neteisinga data. Arba mėnuo = 15 – taip pat įvesta neteisinga data. Vasario mėnesio didžiausia leidžiama diena – 29.

Reikalavimai

Parašyti funkcijas, gražinančias apskaičiuotas reikšmes per funkcijos vardą:

- string tipo rezultatą – metų laiką pagal berberų kalendorių;
- loginę reikšmę true, jei įvesta diena teisinga, arba false, jei diena įvesta neteisingai;
- loginę reikšmę true, jei mėnuo įvestas teisingai, arba false, jei mėnuo įvestas neteisingai.

8. FUNKCIJOS, GRAŽINANČIOS APSKAIČIUOTAS REIKŠMES PER PARAMETRUS

1. Bulviakasis

Rudenį ūkininkas su talkininkais kasė bulves. Kiekvieną dieną ūkininkas užsirašydavo, kiek bulvių prikasta ir keli žmonės tą dieną dirbo.

Parenkite programą, skaičiuojančią: 1) kelias dienas buvo kasamos bulvės; 2) kiek iš viso bulvių prikasta; 3) kiek bulvių vidutiniškai prikastė vienas žmogus per dieną.

Pradiniai duomenys įvedami klaviatūra tokia tvarka: kiek bulvių prikasta per dieną (realūs skaičiai) ir keli talkininkai dirbo tą dieną. Pradinių duomenų įvedimas baigiamas įvedus bent vieną nulį.

Rezultatai rodomi ekrane po vieną reikšmę eilutėje. Prikastų bulvių kiekis ir per dieną žmogaus prikastų bulvių kiekis rodomi 2 skaitmenų po kablelio tikslumu.

Pradiniai duomenys	Rezultatai
50.5 3	2
30.5 1	81.00
0 5	20.25

Reikalavimai

Būtina funkcija **void**, gražinanti per parametrus: kelias dienas buvo kasamos bulvės, kiek bulvių prikasta iš viso ir kiek bulvių vidutiniškai prikastė vienas žmogus per dieną.

2. Atspėk skaičių

Sukurkite kompiuterinį žaidimą „Atspėk skaičių“. Žaidimo algoritmas:

1. Pirmiausia kompiuteris turi sugeneruoti atsitiktinį sveikąjį skaičių iš intervalo [0; 99]. Žaidėjas sugeneruoto skaičiaus nemato.
2. Žaidėjas įveda, kelis kartus norės spėti sugeneruotą skaičių.
3. Žaidėjas pradeda įvedinėti sugalvotus skaičius.
4. Toliau viskas vyksta pagal tokią schemą: Jei žaidėjas įveda skaičių, kuris mažesnis už sugeneruotą skaičių, kompiuteris išveda pranešimą PER MAŽAS, jei didesnę už sugeneruotą

skaičių - PER DIDELIS, jei sutampanti su sugeneruotu - TINKAMAS, jei pasiekia k - PASIEKĖTE SPĖJIMŲ RIBĄ.

Pvz., jei kompiuteris sugeneruoja skaičių 12, o žaidėjas pasirenka, kad spės 3 kartus ir įveda:

5. Ekране rodomas pranešimas: PER MAŽAS, spėjimas 1

14. Ekране rodomas pranešimas: PER DIDELIS, spėjimas 2

7. Ekране rodomas pranešimas: PER MAŽAS, spėjimas 3. PASIEKĖTE SPĖJIMŲ RIBĄ.

Jei kompiuteris sugeneruoja skaičių 12, o žaidėjas pasirenka, kad spės 3 kartus ir įveda:

5. Ekране rodomas pranešimas: PER MAŽAS, spėjimas 1

12. Ekране rodomas pranešimas: TINKAMAS, spėjimas 2

Reikalavimai

Programoje turi būti funkcija **void**, grąžinanti kintamųjų reikšmes per parametrus: pranešimą apie žaidėjo įvestą skaičių (PER MAŽAS, PER DIDELIS, TINKAMAS), kelintas spėjimas tai buvo ir apie pasiektą spėjimų ribą.

3. Vampyro skaičiai

Sveikieji skaičiai, sudaryti iš lyginio skaitmenų skaičiaus ir kurių visi skaitmenys yra skirtingi, vadinami vampyro skaičiais. Iš skaičiaus skaitmenų sudaryti daugikliai yra vadinami vampyro skaičiaus iltimis. Pavyzdžiui, vampyro skaičius $1260 = 21 * 60$, čia daugikliai 21 ir 60 - vampyro skaičiaus iltys. Kaip pastebėjote, vampyro skaičiaus iltys sudarytos iš vampyro skaičiaus skaitmenų.

Parenkite programą, spausdinančią visus keturženklį skaičių intervalo [m; n] vampyro skaičius ir jų iltis. Iltis išdėstykite pradedant nuo mažesnės iki didesnės.

Jei $m = 1000$, $n = 2000$, tuomet vampyro skaičiai:

1260 21 60

1395 15 93

1435 35 41

1530 30 51

1827 2187

Reikalavimai

Loginė funkcija **ArNevienodi()**, grąžinanti reikšmę true, jei keturženklį skaičių sudaro nevienodi skaitmenys, o jei yra vienodų skaitmenų – false.

Funkcija **void Iltys ()**, spausdinanti vampyro skaičių ir jo iltis.

4. Sieksniai, varstai ir mylios

Lietuvoje iki Pirmojo pasaulio karo buvo plačiai naudojami senoviniai ilgio matai: sieksnis, 500 sieksnių sudaro varstą, 7 varstai sudaro mylią.

Parašykite programą, skaičiuojančią dviejų ilgių, išreikštų myliomis, varstais ir sieksniais, sumą ir skirtumą, išreikštą myliomis, varstais ir sieksniais.

Pradiniai duomenys: pirmoje eilutėje įrašyta, kelių ilgių rinkinių sumą ir skirtumą reikia apskaičiuoti. Tolesnėse n eilučių įrašyta po 6 sveikuosius skaičius: pirmieji trys skaičiai reiškia pirmojo ilgio mylias m_1 , varstus v_1 ir sieksnius s_1 , paskutiniai trys skaičiai – antrojo ilgio mylias m_2 , varstus v_2 ir sieksnius s_2 .

Rezultatai: ekrane turi būti n eilučių su 6 sveikaisiais skaičiais kiekvienoje eilutėje: pirmieji trys skaičiai reiškia sumos mylias sm , sumos varstus sv , sumos sieksnius ss ; paskutiniai trys skaičiai – skirtumo mylias skm , skirtumo varstus skv , skirtumo sieksnius sks .

Pradiniai duomenys	Rezultatai
2	4 1 149 0 1 151
2 1 150 1 6 499	6 1 497 1 3 1
3 5 499 2 2 498	

Reikalavimai

Turi būti funkcija **void Suma()**, skaičiuojanti dviejų ilgių sumą.

Turi būti funkcija **void Skirtumas()**, skaičiuojanti dviejų ilgių skirtumą.

9. MASYVAI. SUMOS, KIEKIO, VIDURKIO SKAIČIAVIMAS

1. Kalėdinės dovanos

Jūs planuojate n ($n \leq 20$) draugų nupirkti kalėdines dovanas. Yra žinoma kiekvienos dovanos kaina (realusis skaičius).

Parenkite programą, skaičiuojančią:

- kiek iš viso kainavo dovanos;
- kokia vidutinė vienos dovanos kaina;
- kelios dovanos kainavo daugiau kaip k eurų (k yra realusis skaičius, kurio reikšmė įvedama klaviatūra pagrindinėje funkcijoje).

Pradiniai duomenys	Rezultatai
10	Visos dovanos kainavo: 268.26
10	Vidutine dovanos kaina: 26.83
15	Dovanu kiekis: 7
50	
15.15	
30.40	
10.50	
8.10	
100.67	
20.45	
5.49	
12.5	
k reikšmė lygi 12	

Reikalavimai

- Naudokite realiųjų skaičių masyvus.
- Pradinių duomenų įvedimo funkcija **void**.
- sumos skaičiavimo funkcija, grąžinanti apskaičiuotą reikšmę per funkcijos vardą.
- kiekio skaičiavimo funkcija, grąžinanti apskaičiuotą reikšmę per funkcijos vardą.

2. Sodininko derlius

Sodininkas mėgėjas augina obelis, kriaušes, vyšnias ir slyvas. Sodo derlių jis parduoda žemdirbių turguje. Sodininkas turguje prekiavo n dienų. Kiekvieną dieną – tik vienos rūšies vaisiais arba uogomis.

Pirmoje eilutėje įrašytas dienų skaičius n . Tolesnėse n eilučių įrašyti vienos dienos duomenys: kokios rūšies vaisiais ar uogomis prekiavo – pirmasis vaisiaus (uogos) pavadinimo simbolis, kelis kilogramus vaisių (uogų) pardavė (sveikasis skaičius), kiek kainavo vienas kilogramas vaisių (uogų) (realusis skaičius).

Parenkite programą, kuri į ekraną išvestų prekybos rezultatus taip, kaip pateikta pavyzdyje. Jei kurios nors rūšies vaisiais (uogomis) sodininkas neprekiavo, tuomet šalia vaisiaus (uogos) turi būti nurodyta: neprekiavo.

Pradiniai duomenys			Rezultatai
10			VYSNIOS:
v	4	3.5	Is viso parduota, kg: 7
s	4	1.5	Parduota uz suma, Eur: 23.60
s	4	1.3	Vidutine kilogramo kaina, Eur: 3.37
			SLYVOS:
			Is viso parduota, kg: 14
o	3	0.5	Parduota uz suma, Eur: 18.70
			Vidutine kilogramo kaina, Eur: 1.34
k	3	0.8	OBUOLIAI:
			Is viso parduota, kg: 7
o	4	0.6	Parduota uz suma, Eur: 3.90
			Vidutine kilogramo kaina, Eur: 0.56
k	2	0.9	KRIAUSES:
			Is viso parduota, kg: 5
v	3	3.2	Parduota uz suma, Eur: 4.20
			Vidutine kilogramo kaina, Eur: 0.84
s	3	1.2	VISI VAISIAI:
			Is viso parduota, kg: 33
s	3	1.3	Parduota uz suma, Eur: 50.40
			Vidutine kilogramo kaina, Eur: 1.53

Reikalavimai

- Naudokite sveikųjų ir realiųjų skaičių masyvus.
- Pradinių duomenų skaitymo funkcija **void Skaitymas()**.
- Parduoto vaisių (uogų) kiekio (kg) skaičiavimo funkcija **int Kiek()**, grąžinanti apskaičiuotą reikšmę per funkcijos vardą. Turi būti mažiausiai 4 kreipiniai į funkciją: skaičiuojant skirtingų rūšių parduotų vaisių (uogų) kiekius.
- Funkcija **double Suma()**, skaičiuojanti, už kokią sumą parduota vaisių (uogų). Funkcija grąžina apskaičiuotą reikšmę per funkcijos vardą. Turi būti mažiausiai 4 kreipiniai į funkciją: skaičiuojant sumas, gautas pardavus skirtingų rūšių vaisius (uogas).
- Vidurkių skaičiavimą užrašykite naudodami jau parašytas funkcijas.
- Rezultatų išvedimo funkcija **void Rasymas()**. Į funkciją reikia kreiptis 5 kartus – su kiekvienos rūšies vaisiais (uogomis) ir visais vaisiais (uogomis).

3. Miesto autobusai

Mieste yra n autobusų maršrutų. Skirtingų maršrutų autobusai aplanko skirtingą stotelių skaičių (tarp stotelių yra skirtingi atstumai), bei perveža skirtingus keleivių skaičius. Autobusų parko administracija atlieka tyrimą, kurio tikslas – sugrupuoti autobusus į 2 grupes. Pirmai grupei turi priklausyti tų maršrutų autobusai, kuriais pervežama ne mažiau kaip k keleivių ir maršruto ilgis didesnis kaip m metrų, antrai – visi likusieji.

Pradinių duomenų pavyzdyje pirmoje eilutėje įrašyti 3 sveikieji skaičiai: n – maršrutų skaičius, bei k ir m reikšmės. Tolesnėse n eilučių įrašyta informacija apie kiekvieną maršrutą – maršruto numeris (gali būti, pvz., 10A, rekomenduojama naudoti string tipą), pervežtų keleivių skaičius, stotelių skaičius ir atstumai tarp gretimų stotelių (sveikieji skaičiai, metrais): 1-2, 2-3, 3-4 ir t.t.

Parenkite programą, kuri į ekraną išvestų rezultatus taip, kaip pateikta pavyzdyje: nurodyti maršruto numerį, pervežtų keleivių skaičių ir bendrą maršruto ilgį. Jei kuriai nors grupei priskirtų maršrutų nėra, tuomet turi būti įrašyta „Pirmos grupės maršrutu nėra“ („Antros grupės maršrutu nėra“).

<i>Pradiniai duomenys</i>	<i>Rezultatai</i>
5 20 2500 5 14 5 500 700 400 200 7 25 8 800 450 600 250 400 600 700 6A 12 7 450 220 400 750 560 420 10 42 9 450 520 600 320 290 560 470 300 10A 15 5 490 250 360 540	Pirmos grupės maršrutai: 7 25 3800 10 42 3510 Antros grupės maršrutai: 5 14 1800 6A 12 2800 10A 15 1640

Reikalavimai

- Naudokite masyvus.
- Pradinių duomenų skaitymo funkcija **void Skaitymas()**.
- Funkcija **int Ilgis()**, skaičiuojanti, koks yra **vieno** maršruto ilgis. Funkcija grąžina apskaičiuotą reikšmę per funkcijos vardą.
- Funkcija **bool Tikrinimas()**, kuri nustato, ar **vienas** maršrutas atitinka pirmos grupės maršrutams keliamus reikalavimus. Funkcija grąžina apskaičiuotą reikšmę per funkcijos vardą.
- Rezultatų išvedimo funkcija **void Rasymas()**. Į funkciją reikia kreiptis 2 kartus – su pirmos ir su antros grupės maršrutais.

4. Detalių rūšiavimas

„Lego“ robotui konstruoti skirtos detalės yra sugrupuotos į keletą grupių: K – konstrukcijai kurti, J – jungiamosios, S – sutvirtinamosios, V – varikliai ir jutikliai.

Pradinių duomenų pavyzdžio pirmoje eilutėje įrašyti 4 sveikieji skaičiai – kelios kiekvienos grupės detalės sudaro pilną rinkinį. Pirmasis skaičius nurodo konstrukcijai kurti skirtų detalių skaičių, antrasis – jungiamųjų, trečiasis – sutvirtinamųjų, ketvirtasis – variklių ir jutiklių. Antroje eilutėje įrašytas dėžių, į kurias sudėtos detalės, skaičius n . Vienoje dėžėje saugomos vienos rūšies detalės. Tolesnėse n eilučių įrašyta informacija apie dėžėje saugomas detales: simbolis, nurodantis, kuriai grupei priklauso dėžėje esančios detalės ir detalių skaičius dėžėje.

Ekране turi būti parodyta informacija apie kiekvieną detalių grupę, nurodant, kiek iš viso yra tos grupės detalių, kelis pilnus rinkinius galima sudaryti iš turimų kiekvienos grupės detalių ir kiek detalių liks nepanaudota. Paskutinėje eilutėje reikia išvesti, kelis pilnai sukomplektuotus rinkinius galima sudaryti iš turimų detalių. Jei kurios nors grupės detalių nėra, tuomet turi būti įrašyta: „nera“.

<i>Pradiniai duomenys</i>	<i>Rezultatai</i>
32 47 25 8	Yra K detalių: 98
12	Is K detalių galima sudaryti: 3 komplektai
J 15	Lieka nepanaudotu: 2
S 12	Yra J detalių: 100
K 56	Is J detalių galima sudaryti: 2 komplektai
V 3	Lieka nepanaudotu: 6
V 2	Yra S detalių: 44
J 58	Iš S detalių galima sudaryti: 1 komplektai
S 32	Lieka nepanaudotu: 19
J 12	Yra V detalių: 10
V 3	Is V detalių galima sudaryti: 1 komplektai
K 42	Lieka nepanaudotu: 2
V 2	Is viso galima sudaryti pilnu komplektu: 1
J 15	

Reikalavimai

- Naudokite masyvus.
- Pradinių duomenų skaitymo funkcija **void Skaitymas()**.
- Funkcija **int Suma()**, skaičiuojanti vienos grupės detalių skaičių. Funkcija grąžina apskaičiuotą reikšmę per funkcijos vardą.
- Funkcija **void Rinkinys()**, skaičiuojanti, kiek pilnų rinkinių galima sudaryti iš vienos grupės detalių ir kiek tos grupės detalių lieka nepanaudota. Funkcija grąžina apskaičiuotas reikšmes per parametrus. Į funkciją turi būti kreipiniai skaičiuojant kiekvienos grupės rinkinių ir likusių detalių skaičius.
- Funkcija **int Kiek()**, kuri nustato, kiek pilnų rinkinių susidaro. Funkcija grąžina apskaičiuotą reikšmę per funkcijos vardą.
- Rezultatų išvedimo funkcija **void Rasymas()**. Į funkciją reikia kreiptis su kiekvienos grupės detalių rinkiniu.
- Pilnų komplektų skaičių išveskite pagrindinėje funkcijoje **main()**.

10. MASYVAI. DIDŽIAUSIOS IR MAŽIAUSIOS REIKŠMIŲ PAIEŠKA

1. Kompiuterinių žaidimų populiarumas

Ketrios kompiuterinių žaidimų kūrėjų kompanijos (Electronic Arts, Valve Coporation, Capcom, Rockstar North) surašė penkių kompanijoje sukurtų populiariausių žaidimų parduotų kopijų skaičius milijonais. Išrinkite kiekvienos kompanijos populiariausią žaidimą (kurio kopijų parduota daugiausiai) ir nepopuliariausią (kurio kopijų parduota mažiausiai), bei sudarykite du naujus masyvus, kurių viename saugomi populiariausių, kitame – nepopuliariausių žaidimų numeriai. Kiekvienos kompanijos žaidimai numeruojami nuo nulio. Jei keli žaidimai yra vienodai populiarūs, reikia pateikti žemiausiai sąrašė esantį žaidimą. Jei keli žaidimai yra vienodai nepopuliarūs, turi būti pateiktas aukščiausiai sąrašė esančio žaidimo numeris.

Pradiniai duomenys	Rezultatai
10	Populiariausi:
20	1. 4
30	2. 3
40	3. 4
50	4. 4
11	Nepopuliariausi:
22	1. 0
33	2. 0
44	3. 3
12	4. 2
12	

23	
34	
4	
56	
14	
15	
5	
100	
200	

Reikalavimai

- Pradinių duomenų įvedimo funkcija **void Skaityti()**, kuri grąžina suformuotus populiariausių ir nepopuliariausių žaidimų masyvus.
- Didžiausios reikšmės vietos nustatymo funkcija, grąžinanti apskaičiuotą reikšmę per funkcijos vardą. Į funkciją reikia kreiptis su kiekvienos žaidimų kompanijos žaidimais.
- Mažiausios reikšmės vietos nustatymo funkcija, grąžinanti apskaičiuotą reikšmę per funkcijos vardą. Į funkciją reikia kreiptis su kiekvienos žaidimų kompanijos žaidimais.
- Rezultatų išvedimo į ekraną funkcija **void Rodyti()**. Į funkciją reikia kreiptis 2 kartus: su populiariausių ir su nepopuliariausių žaidimų masyvais.

2. Vasaros kelionės

Kelionių agentūra „Novaturas“ siūlo platų kelionių pasirinkimą. Trys draugai rengiasi vasarą keliauti po Europą. Pirmasis draugas kelionei gali skirti $d1$ eurų, antrasis – $d2$, trečiasis – $d3$. Reikia surasti, kurios kelionės tiks kiekvienam iš draugų ir kurios bus tinkamos visiems trims draugams.

Pirmoje eilutėje įrašyti 3 sveikieji skaičiai: pinigų suma, kurią kiekvienas draugas gali skirti kelionei. Antroje eilutėje įrašytas „Novaturo“ siūlomas kelionių po Europą skaičius n . Tolesnėse eilutėse nurodyta informacija apie kiekvieną kelionę: aplankomos šalys, kelionės trukmė dienomis, kelialapio kaina (sveikasis skaičius), kiek pinigų reikia turėti papildomoms išlaidoms (sveikasis skaičius).

Parenkite programą, kuri į ekraną išvestų rezultatus taip, kaip pateikta pavyzdyje, nurodant šalies pavadinimą, kelionės trukmę ir bendras kelionės išlaidas. Jei kuriam nors draugui nėra tinkamos kelionės, tuomet turi būti išvedama: „Pirmam draugui tinkamų kelionių nėra“ („Antram draugui tinkamų kelionių nėra“, „Trečiam draugui tinkamų kelionių nėra“, „Visiems draugams tinkamų kelionių nėra“.

Taip pat programa turi surasti:

- Į kurias šalis vykstant kelionių trukmė yra trumpiausia? Šalis išvardinti tokia tvarka, kokia jos pateiktos sąrašė. Vienai šaliai skiriama viena eilutė.
- Į kurias šalis vykstant kelionių trukmė yra ilgiausia? Šalis išvardinti tokia tvarka, kokia jos pateiktos sąrašė. Vienai šaliai skiriama viena eilutė.

Pradiniai duomenys	Rezultatai
700 650 1000 5 Italija 10 550 200 Prancūzija 12 620 150 Kroatija 8 440 120 Vokietija 10 560 170 Graikija 12 570 100	Pirmam draugui tinkamos kelionės: Kroatija 8 560 Graikija 12 670 Vidutinė kelionės trukmė dienomis yra: 10 Antram draugui tinkamos kelionės: Kroatija 8 560 Vidutinė kelionės trukmė dienomis yra: 8 Trečiam draugui tinkamos kelionės: Italija 10 750 Prancūzija 12 770 Kroatija 8 560 Vokietija 10 730 Graikija 12 670 Vidutinė kelionės trukmė dienomis yra: 10 Visiems draugams tinkamos kelionės: Kroatija 8 560 Vidutinė kelionės trukmė dienomis yra: 8 *** Trumpiausios kelionės: Kroatija Ilgiausios kelionės: Prancūzija Graikija

Reikalavimai

- Naudokite simbolių eilučių (string) ir sveikųjų skaičių masyvus.
- Pradinių duomenų įvedimo funkcija **void Skaitymas()**.
- Funkcija **int Suma()**, skaičiuojanti bendras vienos kelionės išlaidas. Funkcija grąžina apskaičiuotą reikšmę per funkcijos vardą.
- Funkcija **int Trukme()**, skaičiuojanti, kokia yra vieno keliautojo (bendrų kelionių) vidutinė trukmė. Funkcija grąžina apskaičiuotą reikšmę per funkcijos vardą. Skaičiuodami vidurkį naudokite sveikųjų skaičių dalybą. Į funkciją turi būti kreipiniai skaičiuojant kiekvieno draugo ir visiems draugams tinkančių kelionių vidutinę trukmę.
- Funkcija **bool Tikrinimas()**, kuri nustato, ar kelionė tinka visiems draugams. Funkcija grąžina apskaičiuotą reikšmę per funkcijos vardą.

- Ilgiausios trukmės kelionės paieškos funkcija **int Ilgiausia()**, grąžinanti masyvo indeksą.
- Trumpiausios trukmės kelionės paieškos funkcija **int Trumpiausia()**, grąžinanti masyvo indeksą.
- Rezultatų išvedimo funkcija **void Rasymas()**. Į funkciją reikia kreiptis su kiekvieno draugo kelionių rinkiniu ir bendru kelionių rinkiniu.
- Išrinktų ilgiausių (trumpiausių) kelionių išvedimo funkcija **void Rasymas1()**. Į funkciją reikia kreiptis 2 kartus: išvedant ilgiausios ir trumpiausios trukmės keliones.

3. Juodasis penktadienis

„Juodoju penktadieniu“ Jungtinėse Amerikos Valstijose vadinamas penktadienis po Padėkos dienos. Šį penktadienį prasideda kalėdinė prekyba.

Kadangi pirkėjams „juodąjį penktadienį“ taikomos didelės nuolaidos, susidaro dideli žmonių srautai, kuriuos labai sunku suvaldyti. Dalis pirkėjų sukčiauja norėdami greičiau patekti į parduotuvę - patys užsirašo tokį eilės numerį, koks jau buvo suteiktas kitam sąžiningai laukusiam pirkėjui.

IT prekių parduotuvės vadybininkai sugalvojo apsaugą nuo sukčių: pirkėjai grupuojami. Iš grupelės išrenkamas pirkėjas, turintis didžiausią unikalų (nesikartojantį) numerį ir jis patenka į parduotuvę.

Parenkite programą, surandančią, kelintas pirkėjas iš n žmonių grupelės ($1 \leq n \leq 100$) pateks į parduotuvę. Pirkėjų numeriai – sveikieji skaičiai iš intervalo $[1; n]$.

Pirmiausia įvedamas grupelėje esančių žmonių skaičius n . Toliau įvedama n sveikųjų skaičių – kiekvieno pirkėjo numeris.

Rezultatas – kokia buvo išrinkto pirkėjo vieta eilėje. Jei pirkėjo su unikaliu numeriu nėra, turi būti išvedamas pranešimas „nėra“.

Pradiniai duomenys:

8

1 1 1 5 3 4 6 6

Rezultatas: 4

Reikalavimai

- Naudokite sveikųjų skaičių masyvus.
- Pradinių duomenų įvedimo funkcija **void Skaitymas()**.
- Pirkėjo paieškos funkcija **int Geriausias ()**, grąžinanti surastą reikšmę per funkcijos vardą.

4. Miesto autobusų maršrutai

Mieste yra n stotelių, kuriose sustoja miesto autobusai, kurių numeriai yra sveikieji skaičiai nuo 1 iki 20.

Parenkite programą, surandančią:

- Trumpiausio maršruto numerį. Jei tokių maršrutų yra daugiau, negu vienas, turi būti spausdinami visų maršrutų numeriai vienoje eilutėje didėjimo tvarka. Maršrutai vienas nuo kito atskiriami tarpais.
- Ilgiausio maršruto numerį. Jei tokių maršrutų yra daugiau, negu vienas, turi būti spausdinami visų maršrutų numeriai vienoje eilutėje mažėjimo tvarka. Maršrutai vienas nuo kito atskiriami tarpais.

Pirmoje eilutėje įrašytas stotelių skaičius n . Tolesnėse n eilučių duomenys išdėstyti tokia tvarka: sveikasis skaičius m , nurodantis kelių maršrutų autobusai stoja stotelėje ir m sveikųjų skaičių – autobusų maršrutų numeriai, nebūtinai iš eilės. Vienai stotelei skiriama viena eilutė.

Pradiniai duomenys	Rezultatai
5	Trumpiausi:
4 14 15 10 16	2 14 16
5 3 10 2 15 16	Ilgiausi:
4 3 14 15 10	15 10
3 3 15 10	
4 15 3 2 10	

Reikalavimai

- Naudokite vienmačius sveikųjų skaičių masyvus.
- Pradinių duomenų įvedimo funkcija void.
- Mažiausios reikšmės paieškos funkcija, grąžinanti apskaičiuotą reikšmę per funkcijos vardą.
- Didžiausios reikšmės paieškos funkcija, grąžinanti apskaičiuotą reikšmę per funkcijos vardą.
- Rezultatų išvedimo funkcijos void: 1) trumpiausių maršrutų; 2) ilgiausių maršrutų.

11. MASYVAI. ELEMENTŲ ŠALINIMAS IR ĮTERPIMAS, RIKIAVIMAS, NAUJO SĄRAŠO FORMAVIMAS

1. Sodo apšvietimas

Sodo plote išdėstyta n ($2 \leq n \leq 100$) sodo LED žibintų su saulės baterijomis. Dalis žibintų nebešviečia. Juos reikia pakeisti naujais. Šeimininkai žibintus sunumeravo ir iš sąrašo išbraukė nešviečiančius. Parenkite programą, kuri:

- iš esamų žibintų sąrašo pašalintų nešviečiančius žibintus,
- sudarytų sąrašą, kuriose vietose nešviečiančius žibintus reikia pakeisti naujais.

Pradiniai duomenys. Pirmiausia įvedama, keli žibintai n yra sode. Tolesnėse n eilučių įvedamas žibinto numeris (numeriai gali eiti ne iš eilės) ir žibinto būseną. Jei žibintas šviečia, jo būseną žymima vienetu, jei ne – nuliu. Vienam žibintui skiriama viena eilutė.

Rezultatai išvedami į ekraną. Pirmoje eilutėje išvedamas šviečiančių žibintų sąrašas, gautas iš pradinių duomenų sąrašo pašalinus nešviečiančius žibintus. Žibintų numeriai vienas nuo kito atskiriami tarpais.

Antroje eilutėje – naujai suformuotas nešviečiančių žibintų numerių sąrašas. Žibintų numeriai vienas nuo kito atskiriami tarpais.

Jei kuriame nors sąrašė nėra nė vieno žibinto, toje eilutėje turi būti žodis NĖRA.

Pradiniai duomenys	Rezultatai
5	3 4 5
3 1	2 1
2 0	
1 0	
4 1	
5 1	

Reikalavimai

- Naudojami sveikųjų skaičių masyvai.
- Pradinių duomenų įvedimo funkcija **void**.
- Masyvo elemento šalinimo funkcija **void**.
- Naujo sąrašo formavimo funkcija **void**.
- Rezultatų išvedimo funkcija **void**, į kurią reikės kreiptis 2 kartus: išvedant šviečiančių žibintų numerius ir išvedant nešviečiančių žibintų numerius.

2. Virtualios realybės akiniai

Mokykloje vienos klasės draugai nusprendė nusipirkti „Playstation“ VR (Virtualios realybės akiniai). Visi susidėjo po lygiai pinigų. Iš pradžių sąraše buvo n žmonių. Tačiau atsirado abejojančių, ar nori šio įrenginio. Sąrašas sumažėjo. Vėliau kiti draugai, išgirdę apie virtualios realybės akiniai, užsimanė prisidėti prie pirkinio, nes kas prisideda, tas ir galės naudotis VR. Parašykite programą, kuri į ekraną išvestų: 1) pradinį pagal abėcėlę surikiuotą draugų sąrašą; 2) draugų sąrašą pašalinus atsisakiusius prie pirkinio prisidėti draugus. **Šalinti reikia pagal numeraciją pradiniame surikiuotame sąraše**; 3) draugų sąrašą, kuris gautas įterpus naujus draugus.

Pradiniai duomenys. Pirmiausia įvedamas pradinis draugų skaičius n. Toliau įvedami draugų vardai. Po to įvedami draugų, atsisakiusių prisidėti prie pirkinio, eilės numeriai (pradiniame sąraše numeracija prasideda nuo 1). Numeriai įvedami didėjimo tvarka. Numerių įvedimas baigiamas nuliu.

Po to įvedama, keli nauji draugai nori prisijungti įsigijant pirkinį, ir kiekvieno draugo vardas. Nauji draugai į sąrašą turi būti įterpti toje vietoje, kur jie tinka pagal abėcėlę.

Pradiniai duomenys	Rezultatai
10	Pradinis sąrašas:
Domas	Aivaras
Tomas	Arnoldas
Martynas	Domas
Rokas	Gabrielius
Gabrielius	Jonas
Aivaras	Kazimieras
Jonas	Linas
Kazimieras	Martynas
Arnoldas	Rokas
Linas	Tomas
3 5 0	Sąrašas pašalinus atsisakiusius dalyvauti:
3	Aivaras
Edvardas	Arnoldas
Antanas	Gabrielius
Laurynas	Kazimieras
	Linas
	Martynas
	Rokas
	Tomas

	Atnaujintas sąrašas: Aivaras Antanas Arnoldas Edvardas Gabrielius Kazimieras Laurynas Linas Martynas Rokas Tomas
--	---

Reikalavimai

- Naudokite eilučių (string) ir sveikųjų skaičių masyvus.
- Pradinių duomenų įvedimo funkcija(-os) **void**.
- Masyvo elementų rikiavimo pagal abėcėlę funkcija **void**.
- Rezultatų išvedimo funkcija **void**. Į rezultatų išvedimo funkciją reikia kreiptis 3 kartus.
- Masyvo elemento šalinimo funkcija **void**.
- Elemento įterpimo į masyvą funkcija **void**.

3. Vairuotojų pažymėjimai

Regitra turi duomenis apie vairuotojų pažymėjimų išdavimo datą. Pažymėjimų išduota n ($2 \leq n \leq 100$). Pažymėjimas laikomas nebegaliojančiu jei jis išduotas daugiau nei prieš 10 metų. Parenkite programą, kuri:

- nebegaliojančius pažymėjimus pašalintų iš duomenų bazės
- sudarytų sąrašą savininkų, kurios reikia informuoti apie tai, kad jiems reikia atvykti ir užsisakyti naujus pažymėjimus.

Pradiniai duomenys. Pirmiausia įvedama, kiek pažymėjimų n yra duomenų bazėje ir patikros data: metai, mėnuo, diena. Tolesnėse n eilučių įvedamas vairuotojo asmens kodas ir data, kada išduotas pažymėjimas: metai, mėnuo, diena. Vienam pažymėjimui skiriama viena eilutė.

Rezultatai išvedami į ekraną. Pirmoje eilutėje išvedamas galiojančių pažymėjimų savininkų asmens kodai, gauti iš pradinių duomenų sąrašo pašalinus negaliojančius. Asmens kodai vienas nuo kito atskiriami tarpais.

Antroje eilutėje – naujai suformuotas savininkų, kurios reikia informuoti, sąrašas. Asmens kodai vienas nuo kito atskiriami tarpais.

Jei kuriame nors sąrašė nėra nė vieno asmens, turi būti žodis NĖRA.

Pradiniai duomenys	Rezultatai
2 2017 01 25 45612150012 2001 12 15 36601191475 2016 01 12	Galiojantys: 36601191475 2016 01 12 Negaliojantys: 45612150012 2001 12 15
2 2017 01 25 45612150012 2001 12 15 36601191475 2006 01 12	Galiojantys: NĖRA Negaliojantys: 45612150012 2001 12 15 36601191475 2006 01 12

Reikalavimai

- Naudokite sveikųjų skaičių masyvus.
- Turi būti pradinių duomenų įvedimo funkcija **void**.
- Turi būti masyvo elemento šalinimo funkcija **void**.
- Turi būti naujo sąrašo formavimo funkcija **void**.
- Turi būti rezultatų išvedimo funkcija **void**.
- Turi būti funkcija, grąžinanti reikšmę **true**, jei vairuotojo pažymėjimas galioja, arba reikšmę **false**, jei vairuotojo pažymėjimas negalioja.

4. Paguodos savaitė

Pusmečio pabaigoje informatikos mokytoja nusprendė moksleiviams padaryti „paguodos savaitę“. Visiems, kurie yra atsiskaitę visus pusmečio darbus vietoj vieno blogiausio pažymio bus įrašytas pažymys, kuris gaunamas apskaičiavus pusmečio pažymių vidurkį. Parenkite programą, kuri esamo pažymių sąrašo blogiausią pažymį pakeistų vidurkiu, jei moksleivis turi visus pažymius. Jei moksleivis atsiskaitęs ne visus darbus – išvesti „Pakeitimai nebuvo atlikti“;

Pradiniai duomenys. Pirmiausia įvedamas mokinių skaičius n . Kitose eilutėse įvedamas kiekvieno mokinio gautų pažymių kiekis p ($2 \leq p \leq 10$) ir pažymiai. Jei mokinys darbo neatsiskaitęs, toje vietoje rašomas nulis.

Rezultatai. Ekrane išvedamas kiekvieno mokinio galutinių pažymių sąrašas. Jei mokinys yra neatsiskaitęs kai kurių darbų, išvedama „Pakeitimai nebuvo atlikti“. Vienam mokiniui skiriama viena eilutė

Pradiniai duomenys	Rezultatai
3	5 8 7 7 9 9
6 5 8 7 4 9 9	10 9 9 9
4 10 8 9 9	Pakeitimai nebuvo atlikti
5 4 0 2 0 5	

Reikalavimai

- Vieno mokinio duomenų įvedimo funkcija **void**.
- Mokinio pažymių vidurkio skaičiavimo funkcija, gražinanti apskaičiuotą reikšmę per funkcijos vardą, laikantis apvalinimo taisyklių.
- Funkcija, surandanti blogiausią pažymį, kurį reikės pakeisti apskaičiuotu vidurkiu.
- Loginė funkcija, gražinanti reikšmę **true**, jei mokinys atsiskaitęs visus darbus ir **false**, jei neatsiskaitęs bent vieno darbo.
- Naują pažymių sąrašą suformuojanti funkcija **void**.
- Vieno mokinio galutinio pažymių sąrašo išvedimo funkcija.

5. Laimingi loterijos bilietai

Loterijos bilietai laikomi laimingais, jei jų numeriai yra palindromai, sudaryti iš k skaitmenų ($3 \leq k \leq 10$) (palindromas – skaičius, kuris perskaitytas iš dešinės į kairę ir iš kairės į dešinę nepasikeičia, pvz., 12121 - palindromas, 1212 – ne palindromas). Tačiau spausdinant n ($1 \leq n \leq 100$) laimingų bilietų dėl prastos spausdinimo kokybės kai kurie skaičiai dešinėje palindromo pusėje buvo išspausdinti nekokybiškai (pvz., vietoj 8 išspausdintas 3). Gali nutikti taip, kad pakeitus nekokybiškai išspausdintą skaičių kokybišku, palindromo gauti nepavyksta. Parenkite programą, kuri:

- Iš esamų bilietų numerių sąrašo pašalintų nekokybiškai išspausdintus bilietus.
- Į pašalintųjų vietą įterptų naujai suformuotus bilietų numerius, jei, pakeitus nekokybiškus skaitmenis kokybiškais, gaunamas palindromas.

Pradiniai duomenys. Pirmoje eilutėje yra įvedamas laimingų bilietų kiekis n . Kitose n eilučių yra išvedami bilietų numeriai. Vienam bilietui skiriama viena eilutė. Paskutinėse 10 eilučių įrašyta po du skaitmenis, kurių pirmasis nurodo, koks skaičius buvo išspausdintas nekokybiškai, antrasis – koku skaitmeniu nekokybiškas skaitmuo keičiamas.

Rezultatai. Į ekraną pirmoje eilutėje yra išvedamas sąrašas gerai išspausdintų palindromų. Antroje – pataisytas bilietų numerių sąrašas. Bilietų numeriai vienas nuo kito atskiriami tarpais.

Pradiniai duomenys	Rezultatai
5	5445 9999 7887
5445	5445 8778 9999 4774 7887
8773	
9999	
4771	
7887	
0 9	
1 4	
2 7	
3 8	

4 1	
5 6	
6 5	
7 2	
8 3	
9 0	

Reikalavimai

- Naudokite sveikųjų skaičių masyvus.
- Duomenų įvedimo funkcija **void**.
- Rezultatų išvedimo funkcija **void**.
- Masyvo elemento šalinimo funkcija **void**.
- Elemento įterpimo į masyvą funkcija **void**.
- Funkcijos, gražinančios apskaičiuotą reikšmę per funkcijos vardą, reikalingos darbui su palindromais:
 - atbulojimo skaičiaus formavimo funkcija;
 - funkcija, nustatanti, ar skaičius yra palindromas;
 - funkcija, formuojanti naują bilieta numerį, jei jame buvo nekokybiškų skaitmenų.

12. KĄ SLEPIA MASYVO INDEKSAS?

1. Temperatūra

Nuobodžiaudamas per vasaros atostogas Juliukas sugalvojo kartą per dieną tuo pačiu paros metu matuoti oro temperatūrą n skaičių dienų ($1 \leq n \leq 30$). Rezultatus jis suvedinėjo į savo sąsiuvinį norėdamas išsiaiškinti, kokia temperatūra kartojosi dažniausiai ir kelias dienas temperatūra buvo aukštesnė už t ($0 \leq t \leq 30$) laipsnių.

Parenkite programą, kuri:

- Suskaičiuotų, kokių dažnumu kartojosi kiekviena temperatūra ir išvestų rezultatus į ekraną.
- Rastų, kelias dienas temperatūra buvo aukštesnė už t laipsnių ir išvestų į ekraną.
- Rastų, kokia temperatūra kartojosi dažniausiai ir išvestų į ekraną laipsnius ir dažnį.

Duomenys. Pirmoje eilutėje įvedamas dienų skaičius n ir temperatūros riba t . Antroje eilutėje yra įvedamos n temperatūrų.

Rezultatai. Pirmoje eilutėje yra išvedamos temperatūros ir jų dažniai. Antroje eilutėje išvedamas skaičius dienų, per kurias temperatūra buvo aukštesnė už t laipsnių. Trečioje eilutėje yra išvedama dažniausiai pasikartojanti temperatūra ir jos dažnis.

Pradiniai duomenys	Rezultatai
10 16 17 16 15 17 19 18 19 17 19 17	15 1 16 1 17 4 18 1 19 3 8 17 4

Reikalavimai

- Pradinių duomenų įvedimo funkcija **void**.
- Paieškos funkcija, grąžinanti didžiausią temperatūros dažnį per funkcijos vardą.
- Sumavimo funkcija, per funkcijos vardą grąžinanti skaičių dienų, per kurias temperatūra buvo didesnė už t laipsnių.
- Rezultatų išvedimo funkcija **void**.

2. Statistikos protokolas

Po krepšinio varžybų suvedamas statistikos protokolas. Reikia parašyti programą, kuri suskaičiuotų kiek kiekvienas žaidėjas žaidė laiko, ir kiek vidutiniškai laiko žaidė komandos nariai. Duomenys įvedami klaviatūra. Pirmoje eilutėje yra užrašytas žaidėjų skaičius n ($1 \leq n \leq 12$). Toliau eilutėmis pateikiamas kiekvieno žaidėjo numeris, išėjimų į aikštelę skaičius ir laiko intervalai, kada jis žaidė. Viena duomenų eilutė skirta vienam asmeniui. Pirmasis skaičius eilutėje reiškia to žaidėjo numerį nr ($4 \leq nr \leq 16$), antrasis rodo kiek kartų jis buvo aikštelėje. Toliau poromis surašyti duomenys apie laiką rungtynėse: pirmieji du skaičiai rodo kada žaidėjas išėjo į aikštę (minutė ir sekundė), o kiti du – kada atsisėdo pailsėti. Rezultatai. Ekrane pirmoje eilutėje parodyti, kiek vidutiniškai (sekundės tikslumu) žaidė kiekvienas žaidėjas. Kitose eilutėse rodyti žaidėjų numerius ir kiek laiko jie žaidė, kaip parodyta pavyzdyje. Žaidėjus vardinti jų numerių didėjimo tvarka.

Pradiniai duomenys	Rezultatai
3 14 2 1 15 2 25 3 16 7 15 7 1 10 55 19 14 9 2 22 10 30 25 33 16 37 19	Vidutiniškai žaista 8 min 35 sec 7 numeris žaidė 8 min 19 sec 9 numeris žaidė 12 min 18 sec 14 numeris žaidė 5 min 9 sec

Reikalavimai

- Pradinių duomenų įvedimo funkcija **void**.
- Vidurkio skaičiavimo funkcija, grąžinanti apskaičiuotą reikšmę per funkcijos vardą.
- Rezultatų išvedimo funkcija **void**.

3. Rezultatyviausias žaidėjas

Futbolo komanda po sėkmingai sužaistos sezono atkarpos iškopė į turnyrinės lentelės pirmąją poziciją. Komandos vadovybė nusprendė pinigine premija paskatinti per mėnesį didžiausią skaičių įvarčių per n ($1 \leq n \leq 10$) rungtynių įmušusius žaidėjus.

Reikia parašyti programą, kuri:

- suskaičiuotų, kiek iš viso įvarčių įmušė kiekvienas žaidėjas ir visų jų numerius su įmuštais įvarčiais išvestų į ekraną (žaidėjo numeris yra intervale $[0;99]$).
- Išrinktų rezultatyviausią žaidėją ir išvestų jo numerį į ekraną. Jei tokių žaidėjų yra daugiau, nei vienas – išvestų visus numerius.

Pradiniai duomenys. Duomenys įvedami klaviatūra. Pirmoje eilutėje įvedamas rungtynių skaičius n . Kitose n eilučių (vienerioms rungtynėms skiriama viena eilutė) pirmasis skaičius yra komandos įvarčių skaičius k , tolimesni – k žaidėjų numerių, tose rungtynėse įmušusių įvarčius.

Rezultatai. Į ekraną pirmoje eilutėje yra išvedami žaidėjų numeriai su skaičiumi per mėnesį įmuštų įvarčių. Antroje eilutėje yra išvedami visų rezultatyviausių žaidėjų numeriai.

Pradiniai duomenys	Rezultatai
5	
3 25 9 29	
2 9 25	9 4 10 2 11 1 17 1 25 3 29 2
4 11 17 9 9	9
1 25	
3 10 29 10	

Reikalavimai

- Pradinių duomenų įvedimo funkcija **void**.
- Paieškos funkcija, gražinanti didžiausią skaičių įmuštų įvarčių per funkcijos vardą.
- Rezultatų išvedimo funkcija **void**.

4. Pinigai

Turistų būrys nutarė visus turimus pinigus sudėti į bendrą kasą. Reikia parašyti programą, kuri suskaičiuotų, kiek pinigų ir kokiais banknotais bus kasoje. Duomenys įvedami klaviatūra.

Pirmoje eilutėje yra užrašytas turistų skaičius n ($1 < n < 100$). Toliau eilutėmis pateikiami kiekvieno turisto turimų banknotų vertės ir jų skaičius. Viena duomenų eilutė skirta vienam asmeniui. Pirmasis skaičius eilutėje reiškia to asmens turimų banknotų skaičių. Toliau poromis surašyti duomenys apie banknotus: pirmasis skaičius reiškia banknoto vertę, o antrasis – tos vertės banknotų skaičių. Rezultatai. Ekrane pirmoje eilutėje parodyti, kiek iš viso kasoje yra banknotų. Antroje eilutėje rodyti banknotų vertes, jų skaičių bei bendrą pinigų sumą, kaip parodyta pavyzdyje. Banknotus vardinti jų verčių didėjimo tvarka. Banknotų vertės gali būti bet kokios nuo 1 iki 50.

Duomenys

3

31 2 3 13 1 50 2 3 25

16 5 5 20 3 10 2 13 6

2 50 2

Rezultatai

49

$$2 * 3 + 3 * 25 + 5 * 5 + 10 * 2 + 13 * 7 + 20 * 3 + 50 * 4 = 477$$

Reikalavimai

- Pradinių duomenų įvedimo funkcija **void**.
- Sumos skaičiavimo funkcija, gražinanti apskaičiuotą reikšmę per funkcijos vardą.
- Rezultatų išvedimo funkcija **void**.

13. SIMBOLIAI. SIMBOLIŲ EILUTĖS

1. Heteroliteralai

Žodžių pora vadinama heteroliteralu, jei abu žodžiai neturi nė vienos bendros raidės. Parašykite programą, kuri nustatytų, ar duotoji žodžių pora yra heteroliteralas. Poros žodžiai bus pateikti tik didžiosiomis raidėmis.

Pradiniai duomenys	Rezultatai
PIKTAS VILKAS	Ne heteroliteralas
PIKTOJI RAGANA	heteroliteralas

Reikalavimai

- Parašykite loginę funkciją, kuri patikrina, ar pirmajame žodyje yra atskirtas antrojo žodžio simbolis.

2. Mažiausiai kartų panaudoti simboliai

Pradiniai duomenys – ne ilgesnė kaip 100 simbolių eilutė. Parašykite programą, kuri išspausdintų eilutę ekrane, o jos apačioje žvaigždutėmis pažymėtų mažiausiai kartų pavartotus simbolius. Eilutėje bus tik didžiosios lotyniškos abėcėlės raidės. Pateiktame pavyzdyje žvaigždutės yra po raidėmis N, T, I

Pradiniai duomenys	Rezultatai
SENA TIESA	SENA TIESA * **

Reikalavimai

- Turi būti rezultatų išvedimo funkcija **void**, kuri ekrane išspausdina pradinę eilutę, o kitoje eilutėje išspausdinamos žvaigždutės tose vietose, kur yra mažiausiai kartų panaudoti simboliai.

Programą rekomenduojama struktūrizuoti, parašant daugiau funkcijų, pvz.:

- suformuojančią simbolių masyvą, kuris saugo, kiek kokių simbolių yra eilutėje;
- surandančią, kiek mažiausiai simbolių yra eilutėje;
- patikrinančią, ar nagrinėjamas simbolis yra panaudotas eilutėje mažiausiai kartų.

3. Valdovės

Valdovė yra galingiausia šachmatų figūra. Ji gali judėti visomis kryptimis (pirmyn, atgal, kairėn, dešinėn, įstrižai) neatsižvelgiant į peršokamų langelių skaičių.

Šachmatų lenta (8 x 8) sužymėta simboliais:

- '.' (taškas) – lentos elementas tuščias.
- '*' (žvaigždutė) – lentos elemento pozicijoje egzistuoja valdovė.

Kiekvienas šachmatų lentos langelis nuo greta esančio atskirtas '.' (tarpas) simboliu.

Raskite, kiek valdovių kerta kiekvieną langelį, ir išveskite rezultatą lentelėje.

Duomenys	Rezultatai
*	1 1 1 1 1 1 1 1
.....	1 1 0 0 0 0 0 0
.....	1 0 1 0 0 0 0 0
.....	1 0 0 1 0 0 0 0
.....	1 0 0 0 1 0 0 0
.....	1 0 0 0 0 1 0 0
.....	1 0 0 0 0 0 1 0
.....	1 0 0 0 0 0 0 1

Reikalavimai

- Parašykite funkciją **void**, kuri suranda valdovės užimamus langelius.

4. Drugio efektas

Ko gero, kiekvienas esate girdėjęs apie, o galbūt net ir matęs, vikšro virsmą drugiu - įstabus transformacijos pavyzdys, nelikęs nepastebėtas nei mokslininkų, nei menininkų: „*Drugio efektas*“ - matematikoje sutinkamos „*Chaos teorijos*“ metafora, kurios pėdsakus taip pat aptiksime ir amerikiečių rašytojo Rėjaus Bredberio (*Ray Douglas Bradbury*) kūrinyje „*Nudundėjo griaustinis*“.

Transformacijos galimos ir tarp žodžių, kuomet pakeisdami vieną simbolį, gauname visai kitą žodį. Kaip pavyzdžiui žodžių sekoje: *Lapas Latas Ratas Batas Baras Karas* - čia pradinis žodis *lapas* palaipsniui, pakeičiant tik vieną simbolį žodyje, galiausiai tampa žodžiu *karas*.

Pradiniai duomenys – atsitiktinių žodžių seka, kurią sudaro ne daugiau kaip 100 žodžių. Parašykite programą, kuri išspausdintų ilgiausią žodžių grandinėlą, sudarytą iš pradinių duomenų žodžių.

Pradiniai duomenys	Rezultatai
Lapas Baras Kuras Karas Blatas Latas Knatas Ratas Raktas Turas Batas	Lapas-Latas-Ratas-Batas-Baras-Karas-Kuras-Turas

Reikalavimai

- Kiekvienas grandinėlą sudarantis žodis nuo savo kaimyno gali skirtis tik vienu simboliu.
- Jei egzistuoja keletas vienodo ilgio žodžių grandinėlių, rezultatų faile išveskite pirmąją.
- Jei grandinėlės sudaryti neįmanoma, išveskite pirmąjį duomenų failo žodį.

5. Anagrama

Anagrama - tai toks vieno žodžio raidžių ar skiemenų sukeitimas, kai sudaromas kitas žodis ar prasminga frazė. Parašykite programą, kuri nustatytų, ar duotoji žodžių (frazijų) pora yra vienas kito anagrama.

Atkreipkite dėmesį į tai, kad viename žodyje ar frazėje esanti didžioji raidė kitame gali virsti mažąja ir atvirkščiai. Taip pat tarpo simboliai neturėtų daryti įtakos programos veikimui (žodis gali virsti į frazę, sudarytą iš daug tarpais atskirtų žodžių ir atvirkščiai). Nepamirškite, kad anagramos sudarymui turi būti panaudotos **visos** žodžio raidės.

Programos duomenys: 2 **kableliu** atskirti žodžiai (frazės), o rezultatai - jei žodžiai yra vienas kito anagrama, išvedama "Anagrama", priešingu atveju - "Ne anagrama".

Pradiniai duomenys	Rezultatai
Semti, Mesti	Anagrama
Semti, Mest	Ne anagrama
Dormitory, Dirty room	Anagrama

Reikalavimai

- Parašykite loginę funkciją, kuri grąžintų **true**, jei duotoji žodžių (frazių) pora yra vienas kito anagrama, priešingu atveju – grąžintų **false**.

14. STRUKTŪROS, STRUKTŪRŲ MASYVAI

1. KMI

Programuotojų mokyklos administracija susirūpino savo moksleivių sveikata, todėl nusprendė atlikti tyrimą, kurio metu bus nustatytas tam tikro amžiaus moksleivių kūno masės indekso (KMI) vidurkis. Parašykite programą, kuri padėtų greitai ir efektyviai atlikti šį tyrimą.

Programos duomenys: **pirmoje eilutėje** - du tarpu atskirti skaičiai - amžiaus intervalas moksleivių, dalyvausiančių mokyklos moksleivių KMI vidurkio nustatymo tyrime. Moksleivis yra tinkamas tyrimui, jei jo amžius yra duotajame intervale, t.y., minimalus tyrimo amžius \leq moksleivio amžius \leq maksimalus tyrimo amžius. **Antroje eilutėje** - moksleivių skaičius (n). **Kitose n eilučių** - duomenys apie kiekvieną moksleivį: moksleivio ūgis, svoris, amžius.

Pastaba: minimalus tyrimo amžius visada bus mažesnis maksimalų tyrimo amžių (papildomo tikrinimo nereikia).

Kūno masės indeksas apskaičiuojamas pagal šią formulę:

$$KMI = \frac{\text{masė (kg)}}{(\text{ūgis(m)})^2}$$

Pradiniai duomenys	Rezultatai
12 18 2 1.95 88 19 1.50 49 10	0.00
12 18 2 1.95 88 18 1.50 49 12	22.46
12 18 3 1.95 88 19 1.77 72 18 1.67 78 17	25.47

Reikalavimai

- Aprašyti moksleivio struktūrą.
- Užbaigti rašyti duomenų skaitymo funkciją.
- Parašyti KMI vidurkio nustatymo funkciją, gražinančią tyrime dalyvavusių moksleivių kūno masės indekso vidurkį.

2. Nauji mokesčiai

Laplandijos vyriausybei prieš Kalėdas nebeužtenka pinigų dovanų pirkimui, todėl valdžia nusprendė įvesti naujus mokesčius šio regiono gyventojams. Parašykite programą, kuri nustatytų, kiek papildomų pajamų po naujų mokesčių įvedimo surinks regiono valdžia.

Įvesti tokie nauji mokesčiai:

- Regiono gyventojams, kuriems yra **6 ir mažiau** metų ir kuriems yra **65 ir daugiau** metų, reikės mokėti mokestį už vardą, t.y., jie mokės **po 5** Laplandijos dolerius už kiekvieną pirmosios raidės nutolimą (vienos raidės atstumą) nuo abėcėles pradžios, pvz., jei gyventojas vardas prasideda raide A, jis mokės 5 Laplandijos dolerius, jei B - 10, jei E - 25 dolerius ir pan.
- Regiono gyventojams, kuriems yra **daugiau nei 6** metai ir kuriems yra **mažiau nei 65**, reikės mokėti mokestį už dantis, t.y., jie mokės **po 5** Laplandijos dolerius už kiekvieną turimą dantį, pvz., gyventojas, turintis 32 dantis, mokės 160 Laplandijos dolerių.

Programos duomenys: **pirmoje eilutėje** - gyventojų skaičius n. **Kitose n eilučių** - duomenys apie kiekvieną gyventoją: vardas, amžius, turimų dantų skaičius.

Pastabos:

- Laplandijos valdžia taip pat priėmė įstatymą, jog nuo šiol šio regiono oficiali kalba - anglų, todėl šio uždavinio sprendimui naudokite lotynišką abėcėlę, o vardai taip pat gali susidėti tik iš lotyniškų raidžių.
- Atkreipkite dėmesį į tai, jog funkcijoje *mokesciai* jau pateikta angliška abėcėlė didžiosiomis raidėmis, tačiau Laplandijoje vardai gali prasidėti ir iš mažosios raidės, todėl gali būti, jog reikės naudoti vieną iš funkcijų *toupper()* arba *tolower()*.

Pradiniai duomenys	Rezultatai
1 Senelis 50 32	160
2 Senelis 50 32 Jaunelis 3 7	210
2 senelis 3 32 jaunelis 50 7	130

Reikalavimai

- Aprašyti gyventojų struktūrą.
- Užbaigti rašyti duomenų skaitymo funkciją.
- Parašyti visos papildomos pinigų sumos, gautos po naujų mokesčių įvedimo gyventojams, apskaičiavimo funkciją.

3. Juodasis penktadienis

Pastarosiomis dienomis daug kur sutinkamas „Juodojo penktadienio“ terminas, žymintis kalėdinio apsipirkimo laikotarpio pradžią, atsirado 1960-aisiais Jungtinėse Amerikos Valstijose. Kodėl būtent juodasis? Atsakymas į pastarąjį klausimą yra pakankamai paprastas: minėtuoju laikotarpiu buhalterinės apskaitos knygos dažniausiai buvo pildomos ranka; tam kad aiškiai atskirti pelną ir nuostolius šie buvo žymimi skirtingomis spalvomis (pelnas - juoda spalva, o nuostoliai - raudona). Būtent dėl kalėdinio apsipirkimo sezono atnešamo pelno, šio laikotarpio pradžia ir gavo „Juodojo penktadienio“ vardą.

Panašią apskaitą pamėginsite atlikti ir Jūs.

Pradiniai duomenys - detali užsakymų informacija: pirmojoje eilutėje pateiktas užsakymų skaičius n ($n \leq 100$), sekančioje eilutėje pateikiama konkretaus užsakymo informacija: užsakymo numeris, užsakymo data, sąskaitos numeris, mokėjimo numeris, sąskaitos apmokėjimo data; toliau pateikiamas skaičius prekių įeinančių į užsakymo pirkinių krepšelį; po to seka pačios užsakymo pirkinių krepšelyje esančios prekės: prekės pavadinimas, prekės tikroji (**reguliari**) kaina, tos prekės nusipirktų vienetų skaičius ir nuolaidos pritaikomos kiekvienam įsigytos prekės vienetui dydis procentais. Pasibaigus užsakymo

pirkinių krepšelio prekių duomenims, pateikiami ir pirkėjo duomenys: pirkėjo vardas ir pavardė. Pačioje paskutinėje duomenų failo eilutėje pateikiamas laiko intervalas, pagal kurį ir bus sudaryta užsakymų apskaita.

Parašykite programą kuri į rezultatų failą išvestų apskaitai naudojamą laiko intervalą, visų, nurodytu laiko intervalu sėkmingai įvykdytų užsakymų bendrą mokėtiną, reguliarią, vidutinę mokėtiną ir sutaupytas sumas bei nupirktų prekių kiekį.

Rezultatų duomenys išsaugojami *Ataskaitos* duomenų struktūroje.

*Reguliari prekės kaina - prekės kaina nepritaikius nuolaidos;

*Mokėtina prekės kaina - prekės kaina pritaikius nuolaidą.

*Sutaupyta suma už pirkinius - reguliarios pirkinių sumos ir mokėtinios pirkinių sumos skirtumas.

*Vidutinė mokėtina užsakymų suma - mokėtinios užsakymų sumos ir užsakymų kiekio santykis.

*Užsakymas yra įvykdytas sėkmingai, jei užsakymo *Mokėjimo apmokėjimoData*, yra nelygi "0000-00-00"

*Prekių krepšelyje saugomas prekių kiekis atspindi tik skirtingų prekių skaičių krepšelyje skaičių. Tačiau jei pirkėjas įsigija kelis tos pačios prekės vienetus, šie kiekiai bus saugomi *Prekės* duomenų struktūros lauke *kiekis*.

Pradiniai duomenys	Rezultatai
6	2016-11-24 2016-11-27 6662.02 7705
209 2016-11-27 123414554 1544 2016-11-27	1332.4 1042.98 15
3	
Kompiuteris 600 1 10	
Kompiuterio-stalas 160 1 15	
Biuro-kėdė 100 1 25	
Petras Petraitis	
210 2016-11-25 123414555 1545 2016-11-25	
4	
Dviratis 699 1 16	
Kavos-aparatas 200 1 12	
Biuro-kėdė 100 2 25	
Paspirtukas 489 1 20	
Jonas Jonaitis	
211 2016-11-24 123414553 1543 2016-11-24	
2	
Namu-kino-sistema 1254 1 15	
LCD-monitorius 200 2 12	

<p>Tomas Miskinis 212 2016-11-26 123414558 1548 2016-11-27 2 Dviratis 699 1 16 LCD-monitorius 200 1 12 Ignas Andriulis 211 2016-11-25 123414556 1546 0000-00-00 1 Baldu-komplektas 2544 1 10 Audrius Naujokaitis 211 2016-11-26 123414557 1547 2016-11-27 2 Baldu-komplektas 2544 1 10 Kompiuterio-stalas 160 1 15 Modestas Kalnaitis 2016-11-24 2016-11-27</p>	
---	--

Reikalavimai

- Skaitymo funkcija, kuri nuskaitytų pradinį duomenis ir išsaugotų juos **Užsakymo** struktūrų masyve.
- Funkcija **MoketinaSumaUzKrepseli**, kuri grąžina mokėtiną pirminių krepšelio sumą.
- Funkcija **ReguliariSumaUzKrepseli**, kuri grąžina reguliarią pirminių krepšelio sumą.
- Funkcija **UzsakymuMoketinaSuma**, kuri grąžina nurodytu laikotarpiu sėkmingai įvykdytų užsakymų mokėtiną sumą.
- Funkcija **UzsakymuReguliariSuma**, kuri grąžina nurodytu laikotarpiu sėkmingai įvykdytų užsakymų reguliarią sumą.
- Funkcija **UzsakymuVidutineMoketinaSuma**, kuri grąžina nurodytu laikotarpiu sėkmingai įvykdytų užsakymų vidutinę mokėtiną sumą.
- Funkcija **KrepselioPrekiuKiekis**, kuri grąžina visų krepšelyje esančių prekių kiekį.
- Funkcija **UzsakymuPrekiuKiekis**, kuri grąžina nurodytu laikotarpiu sėkmingai įvykdytų užsakymų prekių kiekį.

15. DIDŽIAUSIOS IR MAŽIAUSIOS REIKŠMIŲ PAIEŠKA STRUKTŪRŲ MASYVE

1. Akimirkos tyrimas

Dirbant su dideliais duomenų kiekiais, kuomet nežinome aiškaus duomenų reikšmių svyravimo intervalo ir norint išlaikyti kiek įmanoma didesnę darbo efektyvumą, negalime iš anksto numatyti koks bus duomenų konteinerio dydis. Todėl dažnu atveju konteinerio dydžio reikšmė yra parenkama dinamiškai, algoritmo vykdymo metu nustatant duomenų imčių maksimalias ir minimalias reikšmes. Tai ypač aktualu taikant „Counting sort“ rikiavimo algoritmą (https://en.wikipedia.org/wiki/Counting_sort).

Pabandydysime ir mes surasti maksimalias ir minimalias pateiktų duomenų reikšmes.

Pradiniai duomenys - konkretūs laiko momentai, nusakomi metais, mėnesiais, dienomis, valandomis, minutėmis ir sekundėmis. Kiekvienas laiko momentas duomenų faile yra saugomas atskiroje eilutėje. Rezultatų faile išsaugomi dviejų laiko momentų įrašai. Pirmasis įrašas - vėliausias laiko momentas iš pateiktųjų, antrasis - anksčiausias.

Pradiniai duomenys	Rezultatai
2001 1 15 1 54 23	-----
2004 4 25 24 52 54	Metai Mėnesis Diena Valandos Minutes Sekundes
2001 6 24 5 41 16	-----
2001 11 12 15 2 28	2012 1 15 1 52 53
2012 1 15 1 52 53	-----

	Metai Mėnesis Diena Valandos Minutes Sekundes

	2001 1 15 1 54 23

Reikalavimai

- Funkcija **VėliausiasLaikoMomentas**, grąžinanti vėliausią laiko momentą iš pateiktųjų.
- Funkcija **AnksčiausiasLaikoMomentas**, grąžinanti anksčiausią laiko momentą iš pateiktųjų.

2. Grojaraštis

Ko gero kiekvieno mediatekoje per metus susikaupė nemažas skaičius įvairiausių muzikos įrašų, kuriuos derėtų tvarkingai suskirstyti į atitinkamus grojaraščius. Naujų metų pradžia yra puikiausias metas minėtiems darbams nuveikti, o kad darbas nepabostų, pamėginsime šį procesą automatizuoti.

Jūsų užduotis yra suformuoti **dešimties** daugiausia perklausų surinkusių įrašų grojaraštį, kuriame įrašai išdėstyti nuo klausomiausio iki mažiausiai klausomo.

Pradiniai duomenys - muzikos įrašų informacija: įrašo pavadinimas, įrašo trukmė (minutės:sekundės), įrašo perklausų skaičius.

Rezultatai - išspausdintas suformuotas populiariausių įrašų grojaraštis.

Pradiniai duomenys
Justin_Timberlake-Can't_Stop_The_Feeling 3:45 10
Calvin_Harris-This_Is_What_You_Came_For 3:56 15
Sia-Cheap_Thrills 4:23 10
James_Bay-Let_It_Go 3:15 25
Beyonce-Sorry 6:45 32
Shawn_Mendes-Stitches 5:34 12
Zara_Larsson_&_MNEK-Never_Forget_You 3:24 0
Adele-Hello 4:25 32
The_Weeknd-The_Hills 4:15 44
Troye_Sivan-Youth 2:56 14
Alessia_Cara-Here 4:35 38
Adele-When_We_Were_Young 4:48 27
The_Weeknd-Can't_Feel_My_Face 3:35 13
Coldplay-Adventureof_a_Lifetime 4:10 17
X_Ambassadors-Renegades 3:28 29
Ed_Sheeran-Photograph 3:55 23
The_Lumineers-Ophelia 4:16 41
Coldplay-Hymn_for_The_Weekend 3:33 37
Sia-Alive 5:28 3

Rezultatai		
Daina	Trukme	Perklausu Skaicius
The_Weeknd-The_Hills	4:15	44
The_Lumineers-Ophelia	4:16	41
Alessia_Cara-Here	4:35	38
Coldplay-Hymn_for_The_Weekend	3:33	37
Beyonce-Sorry	6:45	32
Adele-Hello	4:25	32
X_Ambassadors-Renegades	3:28	29
Adele-When_We_Were_Young	4:48	27
James_Bay-Let_It_Go	3:15	25
Ed_Sheeran-Photograph	3:55	23

Reikalavimai

- Duomenų struktūra **Irasas** įrašo informacijai saugoti.
- Duomenų struktūra **Grojarastis** grojaraštį sudarančių įrašų informacijai saugoti.
- Pradinių duomenų skaitymo funkcija **Skaityti**.
- Funkcija **SalintiIrasa** įrašo duomenims pašalinti iš bendro įrašų masyvo **irasai**;
- Klausomiausio įrašo paieškos bendrame įrašų masyve funkcija **IeskotiKlausomiausio**;
- Grojaraščio formavimo funkcija **FormuotiGrojarasti**;
- Rezultatų spausdinimo funkcija **Spausdinti**.

3. Mokinių sąrašas

Informiko organizuojamose programavimo pamokėlėse kiekvienais metais formuojami mokinių sąrašai, kurie vėliau apdorojami, išsiaiškinant, kaip sekėsi programavimo pamokėlių dalyviams.

Parašykite programą, surandančią kiekvienų mokslo metų dalyvį, kuriam programuoti sekėsi geriausiai, ir dalyvį, kuriam programuoti sekėsi sunkiausiai.

Pradiniai duomenys: pirmoje eilutėje nurodytas mokslo metų skaičius, toliau – mokymosi metų sąrašas. Tolesnėje eilutėje nurodytas mokinių skaičius n. Tolesnėse n eilučių įrašyta informacija apie kiekvieną mokinį: mokinio vardas, mokymosi metai, surinktų taškų skaičius.

Pradiniai duomenys	Rezultatai
4	2016 Geriausias Martynas
2016	2016 Blogiausias Arturas
2015	2015 Geriausias Sonata
2014	2015 Blogiausias Antanas
2013	2014 Geriausias Vacys
10	2014 Blogiausias Kazys
Arturas 2016 3100	2013 Geriausias Robertas
Antanas 2015 2590	2013 Blogiausias Arturas
Petras 2015 2900	
Sonata 2015 3200	
Martynas 2016 3300	
Kazys 2014 1500	
Jonas 2014 2350	
Arturas 2013 0	
Vacys 2014 3000	
Robertas 2013 3010	

Reikalavimai

- Geriausio mokinio nurodytais mokslo metais radimo funkcija, gražinanti mokinio indeksą masyve.
- Blogiausio mokinio nurodytais mokslo metais radimo funkcija, gražinanti mokinio indeksą masyve.

4. Žaidimas „Laimingas ūgis“

Programuotojų mokyklos moksleiviai nusprendė pažaisti žaidimą, kurio laimėtojai gaus prizų. Padėkite nustatyti, kiek moksleivių turi „laimingą ūgį“.

Žaidimo taisyklės labai paprastos: moksleiviai pirmiausia sustoja į eilę visiškai atsitiktine tvarka. Pasižymima, koks moksleivis kurioje eilės vietoje stovi bei užrašomas jo ūgis. Tada moksleiviai į eilę sustatomi tokia tvarka: pirmiausia atsistoja aukščiausias moksleivis, tada žemiausias, tada antras pagal aukščiausią ūgį, tada antras pagal žemiausią ūgį ir t.t. Jei yra du tokio paties ūgio moksleiviai, tai į eilę pirmiau pastatomas tas, kurio vardas abėcėlėje yra pirmesnis (pvz., Gabrielius abėcėlėje eina pirmiau nei Haroldas). Galiausiai suskaičiuojama, kiek moksleivių liko stovėti toje pačioje eilės vietoje ir skelbiama, kad jų ūgis yra laimingas bei gauna prizų.

Programos duomenys: **pirmoje eilutėje** - moksleivių skaičius (n), nurodantis, kiek moksleivių stovi eilėje ir dalyvauja žaidime. Kitose **n** eilučių - moksleivių duomenys: vardas ir ūgis (sveikasis skaičius, centimetrais).

Programos rezultatai: „laimingą ūgį“ turinčių moksleivių skaičius.

PASTABA: darome prielaidą, kad nėra dviejų moksleivių, turinčių ir tokį patį ūgį, ir tokį patį vardą.

Pradiniai duomenys	Rezultatai
1 Petras 199	1
4 Petras 199 Simas 180 Kazys 171 Aidas 176	2
5 Petras 199 Simas 180 Kazys 171 Aidas 176 Petras 174	1

Reikalavimai

- Aprašyti moksleivio struktūrą.
- Užbaigti rašyti duomenų skaitymo funkciją.
- Parašyti funkciją minMax, kuri sudaro naują moksleivių eilę pagal ūgius.
- Užbaigti rašyti funkciją sutampa, kuri nustato, kiek moksleivių turi „laimingą ūgį“.

16. STRUKTŪRŲ MASYVŲ ELEMENTŲ RIKIAVIMAS

1. Asmens kodas

Universiteto administracija nusprendė sudaryti studentų sąrašą, kurį nori išrikiuoti pagal lytį bei studentų amžių mažėjančia tvarka (nuo vyriausio iki jauniausio), tačiau studentai identifikuojami pagal vardą bei asmens kodą, pagal kurį labai nepatogu atlikti tokį sąrašo rikiavimą. Parašykite programą, kuri padėtų universiteto administracijai išrikiuoti studentus norima tvarka.

Remiantis šiuo (https://lt.wikipedia.org/wiki/Asmens_kodas) straipsniu, asmens kodą sudaro 11 skaitmenų, iš kurių:

- pirmasis rodo gimimo šimtmetį ir asmens lytį (1 – XIX a. gimęs vyras, 2 – XIX a. gimusi moteris, 3 – XX a. gimęs vyras, 4 – XX a. gimusi moteris, 5 – XXI a. gimęs vyras, 6 – XXI a. gimusi moteris);
- tolesni šeši – **asmens gimimo metų du paskutiniai skaitmenys**, mėnuo (du skaitmenys), diena (du skaitmenys);
- dar tolesni trys skaitmenys – tą dieną gimusių asmenų eilės numeris;
- paskutinis – iš kitų skaitmenų išvedamas kontrolinis skaičius.

Administracija nori išrikiuoti studentus tokia tvarka: pirmiausia studentai rikiuojami pagal **lytį** (aišku, **moteriškos lyties atstovės sąrašė yra pirmiau** už vaikus), o vėliau rikiuojama pagal **amžių (nuo vyriausio iki jauniausio)**.

Programos duomenys: **pirmoje eilutėje** - studentų skaičius (n), nurodantis, kiek studentų yra sąrašė. Kitose **n** eilučių - studentų duomenys: vardas ir asmens kodas.

Programos rezultatai: studentų sąrašas (studento asmens kodas ir vardas), išrikiuotas reikiama tvarka.

PASTABA (PATARIMAS): sąrašo rikiavimui iš asmens kodo reikės išskirti pirmąjį skaitmenį (lytį) ir 2-3 skaitmenis (gimimo metų paskutiniai 2 skaitmenys), todėl patogu asmens kodą saugoti ne simbolių eilutės, bet skaičiaus formatu, nes skaitmenys gali būti išskiriami naudojant dalybos bei dalybos su liekana veiksmus!

Pradiniai duomenys	Rezultatai
2 Mangirdas 39606156783 Goda 49809064356	49809064356 Goda 39606156783 Mangirdas
4 Steponas 50207133465 Mantas 39107142363 Mangirdas 39606156783 Goda 49809064356	49809064356 Goda 39107142363 Mantas 39606156783 Mangirdas 50207133465 Steponas

5	49809064356 Goda
Mangirdas 39606156783	60008013343 Inga
Mantas 39107142363	39107142363 Mantas
Goda 49809064356	39606156783 Mangirdas
Steponas 50207133465	50207133465 Steponas
Inga 60008013343	

Reikalavimai

- Aprašyti studento struktūrą.
- Užbaigti rašyti duomenų skaitymo funkciją.
- Užbaigti rašyti studentų sąrašo spausdinimo funkciją.
- Realizuoti studentų sąrašo rikiavimo algoritimą, rikiuojantį pagal lytį ir amžių.

2. Pažangumas

Baigiantis seniesiems metams bei prasidedant naujiesiems, ko gero, tarp dešimties populiariausių pažadų, suplanuotų įvykdyti per naujus metus studentų bei mokinių tarpe, aptiktume ir pažadą geriau mokytis.

Tam, kad galėtume efektyviau sekti savo pažangumą bei lengviau ištesėti duotus pažadus, sukursime studentų rikiavimo, pagal turimų įvertinimų vidurkį, algoritimą.

Pradiniai programos duomenys - studentai bei jų turimi įvertinimai. Pirmojoje duomenų failo eilutėje yra pateiktas studentų skaičius, kuris negali būti didesnis kaip 100. Tolesnėse duomenų failo eilutėse yra pateikta kiekvieno studento informacija: studento vardas, pavardė, įvertinimų skaičius (ne didesnis nei 20) bei įvertinimai.

Rezultatų faile yra išvedamas studentų sąrašas (studento vardas, pavardė ir įvertinimų vidurkis, pateiktas šimtųjų tikslumu), surikiuotas didėjančiai pagal įvertinimų vidurkį.

Pradiniai duomenys	Rezultatai
11	Jonas Jonaitis 6.43
Tomas Petrauskas 12 5 6 8 9 10 4 7 9 10 4 10 9	Motiejus Zdanaitis 7.00
Ignas Siaurusaitis 11 9 10 4 6 5 8 10 9 7 4 10	Martynas Karalius 7.22
Petras Petraitis 10 9 2 5 8 7 10 9 6 10 10	Ignas Siaurusaitis 7.45
Martynas Karalius 9 8 2 10 10 9 9 3 6 8	Jurgis Niekas 7.50
Lukas Sakalauskas 8 10 10 3 8 9 9 8 7	Tomas Petrauskas 7.58
Jonas Jonaitis 7 5 5 5 6 9 7 8	Petras Petraitis 7.60
Simonas Arminas 6 8 8 9 10 2 10	Simonas Arminas 7.83
Paulius Kairaitis 5 10 9 8 10 8	Lukas Sakalauskas 8.00
Ignas Paulauskas 4 10 9 8 5	Ignas Paulauskas 8.00
	Paulius Kairaitis 9.00

Motiejus Zdanaitis 3 10 2 9

Jurgis Niekas 2 10 5

Reikalavimai

- Funkcija *Rikiuoti*, surikiuojanti studentus pagal jų turimą įvertinimų vidurkį didėjančiai (jei studentų vidurkiai sutampa, išlaikykite jų eilės tvarką tokią, kokia yra pateikta pradiniam duomenų rinkinyje);
- funkcija *GrazintiStudentoVidurki*, apskaičiuojanti ir grąžinanti studento vidurkį;
- duomenų struktūra *Studentas*, sauganti konkretaus studento duomenis (vardas, pavardė, įvertinimų kiekis bei įvertinimai).

3. Pasiklydę skaičiai

Skaičiai yra universali, konkreti kalba, padedanti užrašyti tai, ką žodžiais dažnai yra sunku nusakyti. Tačiau skaičiai konkretūs ir vienareikšmiai išlieka į juos žvelgiant tik iš tam tikros, vienos skaičiavimo sistemos. Kadangi egzistuoja ne viena skaičiavimo sistema, tas pats skaitmens užrašymas gali įgyti visiškai skirtingas vertes.

Pamėginsime sukurti programą, kuri pateiktus dvejetainės, aštuntainės, dešimtainės, šešiolyktainės skaičiavimo sistemų skaičius surikiuos didėjančiai dešimtainėje skaičiavimo sistemoje.

Pateikiama nuoroda į darbo su skirtingomis skaičiavimo sistemomis įvadą:

<http://www.mif.vu.lt/~ragaisis/InfIvadas/SkSistemas.htm>

Kiekvienoje pradinių duomenų failo eilutėje turime pateiktą sveikąjį skaičių - skaičiavimo sistemos tipą (2 - dvejetainė skaičiavimo sistema, 8 - aštuntainė skaičiavimo sistema, 10 - dešimtainė skaičiavimo sistema, 16 - šešiolyktainė skaičiavimo sistema) ir skaitmenį, užrašytą nurodytoje skaičiavimo sistemoje, kurį nagrinėsime kaip simbolių eilutę.

Rezultatų faile didėjančiai dešimtainėje skaičiavimo sistemoje yra išvedami pradinių duomenų faile esantys skaičiai (originali skaičiaus išraiška, dešimtainė skaičiaus išraiška, bei originalios skaičiaus išraiškos skaičiavimo sistemos tipas).

Pradiniai duomenys	Rezultatai
10 1544	A 10 16
16 EEEEF	F 15 16
2 11111111	55 55 10
8 76455	77 63 8
16 BCDF654	01010101 85 2
10 99	99 99 10
16 A	11111111 511 2
8 77	1544 1544 10
16 F	76455 32045 8
10 55	EEEEF 978671 16
2 01010101	BCDF654 198047316 16

Reikalavimai

- Funkcija *Rikiuoti*, surikiuojanti pradinių duomenų faile pateiktus skaičius didėjančiai dešimtainėje skaičiavimo sistemoje (jei dešimtainėje skaičiavimo sistemoje skaičių vertės sutampa, išlaikykite jų eilės tvarką tokią, kokia yra pateikta pradiniam duomenų rinkinyje).
- Funkcija *Konvertuoti*, verčianti nurodytos skaičiavimo sistemos skaičius į dešimtainės sistemos sveikuosius skaitmenis.
- Duomenų struktūra *Skaičius*, sauganti konkretaus skaitmens duomenis (skaičiaus išraišką nurodytoje skaičiavimo sistemoje, bei skaičiavimo sistemos, kurioje yra išreikštas skaitmuo, tipą).
- Nagrinėjami tik sveikieji, teigiami skaičiai.

4. Biblioteka

Informiko bibliotekoje knygos žymimos tam tikrais kodais ir laikomos tam tikra tvarka. Tačiau Informikas nuvertė lentyną ir visos knygos susimaišė.

Parašykite programą, kuri surikiuotų knygas pagal šalies pavadinimą, leidyklos pavadinimą ir knygos identifikacinį numerį.

Paaiškinimas:

Kodą sudaro trys dalys: šalies kodas, leidyklos kodas ir knygos identifikacinis numeris

Pvz: 150 1596 0010 reiškia: 150 – Lietuva, 1596 - Alma Littera, 10 - knygos identifikacinis numeris

Duomenys:

- Kodų skaičius ir kodai.
- Šalių kodų skaičius ir duomenys: šalies kodas ir pavadinimas.
- Leidyklos kodų skaičius ir duomenys: leidyklos kodas ir pavadinimas.

Rezultatai: ekrane turi būti parodytas surikiuotas knygų sąrašas pagal šalies pavadinimą, leidyklos pavadinimą ir knygos identifikacinį numerį.

Duomenys	Rezultatai
7	Anglija British_Publisher 123
150 1596 10	Lietuva Alma_Littera 10
150 1596 124	Lietuva Alma_Littera 124
150 1596 536	Lietuva Alma_Littera 536
039 4532 123	Rusija Rasputnik 2359
123 1145 2359	Rusija Rasputnik 3596
123 1015 2345	Rusija Sag_Za_Sagom 2345
123 1145 3596	
3	
150 Lietuva	

039 Anglija	
123 Rusija	
4	
1596 Alma_Littera	
4532 British_Publisher	
1145 Rasputnik	
1015 Sag_Za_Sagom	

Reikalavimai

- Turi būti naudojami struktūrų masyvai.
- Pradinių duomenų įvedimo funkcija **void Skaityti()**.
- Rezultatus saugančio struktūrų masyvo sudarymo funkcija **void Sudaryti()**.
- Rikiavimo funkcija **void Rikiuoti()**.
- Rezultatų išvedimo **funkcija Spausdinti()**.

17. STRUKTŪRŲ MASYVŲ ELEMENTŲ ŠALINIMAS IR ĮTERPIMAS

1. Krepšinio komanda

Krepšinio komandos treneris padarė minutės pertraukėlę, norėdamas pakeisti žaidėjus. Klaviatūra įvedami aikštelėje esančių žaidėjų duomenys (krepšinio aštuoliai tikrai žino, kad aikštelėje yra 5 žaidėjai): vardas, nuovargio įvertinimas, pelnyti taškai. Toliau įvedama informacija apie atsarginius žaidėjus: atsarginių žaidėjų skaičius ir informacija apie kiekvieną atsarginį žaidėją: vardas, nuovargio įvertinimas, pelnyti taškai. Parašykite programą, kuri aikštelėje esančius žaidėjus, kurių nuovargio įvertis yra daugiau negu 5, pašalintų iš aikštelės ir įterptų į likusių žaidėjų sąrašą žaidėjus nuo atsarginių suolelio, kad aikštelėje vėl būtų 5 žaidėjai, išrikiuoti pagal abėcėlę. Pradiniuose duomenyse pateikti sąrašai pagal abėcėlę surikiuoti.

Pradiniai duomenys	Rezultatai
Adas 6 13	Antanas
Domantas 3 5	Domantas
Jonas 8 16	Mantas
Mantas 3 1	Mindaugas
Renaldas 4 1	Renaldas
4	
Antanas 0 0	
Marius 9 13	
Mindaugas 2 1	
Paulius 1 4	

Reikalavimai

- Naudojami struktūrų masyvai.
- Pradinių duomenų įvedimo funkcija **void**. Į funkciją kreiptis 2 kartus: įvedant aikštelėje esančių krepšininukų duomenis ir atsarginių krepšininukų duomenis.
- Žaidėjų šalinimo funkcija **void**.
- Žaidėjų įterpimo funkcija **void**.
- Rezultatų išvedimo funkcija **void**.

2. Informacijos saugojimas

Pabandysime sukurti programą, kuri valdytų informacijos išsaugojimo standžiajame diske procesą. Tarkime, jog standusis diskas - cilindras, sudarytas iš 4 magnetinių plokštelių. Kiekviena magnetinė plokštelė yra sudaryta iš 8 takelių, o kiekvienas takelis - iš 32 vienodo dydžio sektorių. Priimkime sąlygą, jog bet kuriame sektoriuje galime išsaugoti tik vieną pradinių duomenų įvesties elementą. Pradinių duomenų įvesties elementų skaičius ≤ 100 ; Informacija diske saugoma nuosekliai, t.y. magnetinės plokštelės takelis užpildomas tuomet, kai šiame takelyje nebelieka laisvų sektorių, magnetinė plokštelė yra užpildoma, kai šioje plokštelėje nebelieka neužpildytų takelių ir pan.

Pradiniai duomenys - vienodo ilgio simbolių eilutės. Simbolių eilučių skaičius yra nurodomas pradinių duomenų failo pradžioje.

Pradiniai duomenys
40 KEWY WERN OxjT sEWN yzmU yMPU dqsy jjAY AEir koxT BDbL PKmW EDNS bVHG qspV xsBX rVbK hbtF RtEJ gWvA DVWR KrmO oqTi skMh EIOP EiaE dqDo vhbQ ciqB CAuM PMST JjCO MKYU LsQY IgSr vobH Vfjv ybNO BNwj zdRC

Rezultatai:

Rezultatų faile pateikiama informacija apie diske saugomus duomenis: standžiojo disko plokštelės, takelio, sektoriaus numeriai bei saugoma informacija.

Rezultatai							
Plokštelės nr:	1	Takelio nr:	1	Sektoriaus nr:	1	Duomenys:	KEWY
Plokštelės nr:	1	Takelio nr:	1	Sektoriaus nr:	2	Duomenys:	WERN
Plokštelės nr:	1	Takelio nr:	1	Sektoriaus nr:	3	Duomenys:	OxjT
Plokštelės nr:	1	Takelio nr:	1	Sektoriaus nr:	4	Duomenys:	sEWN
Plokštelės nr:	1	Takelio nr:	1	Sektoriaus nr:	5	Duomenys:	yzmU
Plokštelės nr:	1	Takelio nr:	1	Sektoriaus nr:	6	Duomenys:	yMPU
Plokštelės nr:	1	Takelio nr:	1	Sektoriaus nr:	7	Duomenys:	dqsy

Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	8	Duomenys:	jjAY
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	9	Duomenys:	AEir
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	10	Duomenys:	koxT
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	11	Duomenys:	BDbL
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	12	Duomenys:	PKmW
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	13	Duomenys:	EDNS
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	14	Duomenys:	bVHg
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	15	Duomenys:	qspV
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	16	Duomenys:	xsBX
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	17	Duomenys:	rVbK
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	18	Duomenys:	hbtF
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	19	Duomenys:	RtEJ
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	20	Duomenys:	gWvA
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	21	Duomenys:	DVWR
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	22	Duomenys:	Krmo
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	23	Duomenys:	oqTi
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	24	Duomenys:	skMh
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	25	Duomenys:	ELOp
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	26	Duomenys:	EiaE
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	27	Duomenys:	dqDo
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	28	Duomenys:	vhbQ
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	29	Duomenys:	ciqB
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	30	Duomenys:	CAuM
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	31	Duomenys:	PMST
Ploksteles nr:	1	Takelio nr:	1	Sektoriaus nr:	32	Duomenys:	JJeO
Ploksteles nr:	1	Takelio nr:	2	Sektoriaus nr:	1	Duomenys:	MKYU
Ploksteles nr:	1	Takelio nr:	2	Sektoriaus nr:	2	Duomenys:	LsQY
Ploksteles nr:	1	Takelio nr:	2	Sektoriaus nr:	3	Duomenys:	IgSr
Ploksteles nr:	1	Takelio nr:	2	Sektoriaus nr:	4	Duomenys:	vobH
Ploksteles nr:	1	Takelio nr:	2	Sektoriaus nr:	5	Duomenys:	Vfjv
Ploksteles nr:	1	Takelio nr:	2	Sektoriaus nr:	6	Duomenys:	ybNO
Ploksteles nr:	1	Takelio nr:	2	Sektoriaus nr:	7	Duomenys:	BNwj
Ploksteles nr:	1	Takelio nr:	2	Sektoriaus nr:	8	Duomenys:	zdRC

Reikalavimai

- Pradinių duomenų skaitymo funkcija: **SkaitytiPradiniusDuomenis**.
- Informacijos rašymo į standųjį diską funkcija: **IraSytiDuomenis**.
- Rezultatų išvedimo funkcija: **Spausdinti**.

3. Mokinių ekskursija

Artėjant pavasario atostogoms, mokiniai nusprendė važiuoti į ekskursiją. Mokytoja ėmėsi sudarinėti į ekskursiją važiuojančių mokinių sąrašą. Po kiek laiko kai kurie mokiniai atsisakė važiuoti, o prie važiuojančių prisijungė kiti. Parašykite programą, kuri iš pradinio mokinių sąrašo pašalintų nebevažiuojančius mokinius ir išlaikant sąrašo surikiavimą pagal mokinio sąrašo numerį įtrauktų naujus važiuojančius mokinius.

Pradiniai duomenys: pirmoje eilutėje – sąrašo dydis N, po to nurodomas pradinis sąrašas (surikiuotas pagal sąrašo numerį): N eilučių su mokinio vardu ir sąrašo numeriu. Po to nurodomas į ekskursiją nebevažiuojančių mokinių skaičius ir K eilučių su mokinio vardu ir sąrašo numeriu. Po to nurodomas naujai užsirašiusių mokinių skaičius ir M eilučių su mokinio vardu ir sąrašo numeriu.

Rezultatai: į ekraną išvedami galutiniam sąrašo esančių mokinių vardai.

Pradiniai duomenys	Rezultatai
7	Antanas
Antanas 1	Marijonas
Aldona 2	Martynas
Ilona 3	Audrius
Marijonas 4	Daiva
Martynas 5	Giedre
Tadas 8	Saulius
Saulius 10	Evelina
3	
Aldona 2	
Ilona 3	
Tadas 8	
4	
Audrius 6	
Daiva 7	
Giedre 9	
Evelina 11	

Reikalavimai

- Šalinimo funkcija.
- Pridėjimo į surikiuotą sąrašą funkcija.
- Šalinimo vietos radimo funkcija.
- Pridėjimo vietos radimo funkcija.

4. Studentų sąrašas

Universitete įvyko keista programų sistemų studijų programos reforma, kurios metu studentai galėjo patys pasirinkti, kurioje grupėje norėjo mokytis. Dėl šios priežasties daugybė studentų keitė savo grupes, o tai universiteto administracijai tapo dideliu galvos skausmu, nes teko labai daug kartų koreguoti studentų grupių sąrašus. Parašykite programą, kuri pagelbėtų universiteto administracijai ir automatiškai koreguotų studentų grupės sąrašą.

Administracijos turimi grupių studentų sąrašai yra pirmiausia išrikiuoti pagal pavardę, o jei pavardė sutampa - pagal vardą **abėcėlės tvarka** bei šalia studento sąrašė pateiktas jo asmens kodas. Studentui keičiant grupę, jis iš sąrašo pašalinamas, o studentui prisijungus prie grupės, jis yra įterpiamas į reikiamą vietą sąrašė pagal pavardę bei vardą.

Programos duomenys: **pirmoje eilutėje** - studentų skaičius (n), nurodantis, kiek studentų yra sąrašė šiuo metu. Kitose n eilučių - studentų duomenys: pavardė, vardas ir asmens kodas. Po šiuo sąrašė duotas skaičius (m), nurodantis, kiek kartų keitėsi sąrašė studentams keičiant grupes. Kitose m eilučių pateikta informacija apie studentus ir jų veiksmus (ar studentas iš grupės išėjo, ar studentas atėjo į grupę): "+" arba "-" (jei "+" - studentas į grupę atėjo, jei "-" - studentas iš grupės išėjo) bei studento pavardė, vardas ir asmens kodas.

Programos rezultatai: grupės studentų sąrašė po visų atliktų veiksmų: pavardė, vardas ir asmens kodas.

PASTABOS:

- Kadangi studentų asmens kodai yra skirtingi, patogiausia studentą iš sąrašo šalinti būtent pagal jį.
- Darome prielaidą, kad sąrašė niekada nebus daugiau nei 30 studentų.
- Nepamirškite, kad keičiantis sąrašė, keičiasi ir n reikšmė.

Pradiniai duomenys	Rezultatai
6 Armonas Steponas 50207133465 Kazakevicius Mantas 39107142363 Kazlauskas Arunas 39604035366 Kazlauskas Mangirdas 39606156783 Miklute Inga 60008013343 Uzeviciute Goda 49809064356 2 + Beinorius Algis 39604302223 + Zakarevicius Vytis 39901012323	Armonas Steponas 50207133465 Beinorius Algis 39604302223 Kazakevicius Mantas 39107142363 Kazlauskas Arunas 39604035366 Kazlauskas Mangirdas 39606156783 Miklute Inga 60008013343 Uzeviciute Goda 49809064356 Zakarevicius Vytis 39901012323
6 Armonas Steponas 50207133465 Kazakevicius Mantas 39107142363 Kazlauskas Arunas 39604035366 Kazlauskas Mangirdas 39606156783 Miklute Inga 60008013343 Uzeviciute Goda 49809064356 3 + Beinorius Algis 39604302223 - Uzeviciute Goda 49809064356 + Zakarevicius Vytis 39901012323	Armonas Steponas 50207133465 Beinorius Algis 39604302223 Kazakevicius Mantas 39107142363 Kazlauskas Arunas 39604035366 Kazlauskas Mangirdas 39606156783 Miklute Inga 60008013343 Zakarevicius Vytis 39901012323
6 Armonas Steponas 50207133465 Kazakevicius Mantas 39107142363 Kazlauskas Arunas 39604035366 Kazlauskas Mangirdas 39606156783 Miklute Inga 60008013343 Uzeviciute Goda 49809064356 5 + Beinorius Algis 39604302223 - Uzeviciute Goda 49809064356 - Miklute Inga 60008013343 - Beinorius Algis 39604302223 - Armonas Steponas 50207133465	Kazakevicius Mantas 39107142363 Kazlauskas Arunas 39604035366 Kazlauskas Mangirdas 39606156783

Reikalavimai

- Aprašyti studento struktūrą.
- Užbaigti rašyti duomenų skaitymo funkciją.
- Užbaigti rašyti studentų sąrašo spausdinimo funkciją.
- Užbaigti rašyti sąrašo koregavimo funkciją.
- Realizuoti studento šalinimo iš sąrašo funkciją.
- Realizuoti studento įterpimo į sąrašą funkciją.

18. ĮVAIRŪS STRUKTŪRŲ MASYVŲ UŽDAVINIAI

1. Perkamiausios prekės

Mažo miestelio parduotuvės savininkas nusprendė išsiaiškinti, kokia yra bendra visų pirkėjų apsipirkimų parduotuvėje per dieną išleista pinigų suma, bei kokios prekės jo parduotuvėje yra pačios perkamiausios, kad jas galėtų padėti arčiau parduotuvės įėjimo, ir pirkėjai jas lengviau pasiektų. Padėkite savininkui ir parašykite programą, kuri nustatytų, kokios prekės parduotuvėje yra pačios perkamiausios bei kokia yra bendra pirkėjų išleista pinigų suma parduotuvėje per dieną.

Perkamiausių prekių sąrašą sudaro 3 per dieną daugiausia kartų nupirktos prekės. Jei parduotuvėje yra mažiau nei 3 prekės - rodoma tiek prekių, kiek yra parduotuvėje. Prekės sąrašė nėra rodoma, jei ji nebuvo nupirkta nė vieno karto. Jei kelios prekės buvo nupirktos po tiek pat kartų, tai prekės išrikiuojamos abėcėlės tvarka pagal pavadinimą ir imamos prekės nuo išrikiuoto sąrašo viršaus.

Programos duomenys: **pirmoje eilutėje** - visų prekių, esančių parduotuvėje, skaičius (n) bei per dieną pirkėjų atliktų apsipirkimų parduotuvėje skaičius (m). Kitose **n** eilučių išvardintos prekės ir jų kainos. Kitose **m** eilučių - informacija apie pirkėjų apsipirkimus. Kiekvienoje eilutėje **pirmasis skaičius (kiekis)** nurodo, kiek apsipirkimo metu buvo įsigyta prekių, o toliau esantys **skaičiai (skaitmenų skaičius = kiekis)** - prekių indeksai, nurodantys pirktas prekes, prekes numeruojant nuo 0, pvz., skaitmuo 1 nurodys 2-ąją prekę, esančią duotajame prekių sąrašė ir pan.

Programos rezultatai: pirmoje eilutėje nurodoma bendra per dieną parduotuvėje pirkėjų išleista pinigų suma. O kitose eilutėse - perkamiausių prekių sąrašas, kur kiekvienoje eilutėje pateikiama tokia informacija: nupirktos prekės kiekis per dieną, prekės pavadinimas bei prekės kaina (tarp visų išvardintų duomenų - tarpo simboliai).

Pradiniai duomenys	Rezultatai
5 5 Pienas 0.78 Kakava 1.55 Sviestas 0.75 Duona 1.56 Makaronai 1.75 4 0 3 1 2 1 3 6 1 1 0 2 3 4 5 4 3 2 2 0 1 1	Bendra suma: 21 4 Duona 1.56 4 Kakava 1.55 4 Sviestas 0.75
2 3 Pienas 1.1 Kakava 0.99 7 0 1 1 0 0 1 0 3 0 0 0 3 0 1 0	Bendra suma: 13 9 Pienas 1.1 4 Kakava 0.99
1 0 Pienas 1.1	Bendra suma: 0

Reikalavimai

- Aprašyti prekės struktūrą.
- Aprašyti pirkėjo struktūrą.
- Užbaigti rašyti duomenų skaitymo funkciją.
- Užbaigti rašyti studentų sąrašo spausdinimo funkciją.
- Realizuoti bendros apsipirkimų sumos skaičiavimo funkciją.
- Realizuoti nupirktų prekių kiekių skaičiavimo funkciją.
- Realizuoti perkamiausių prekių nustatymo funkciją.

2. Savaitgalis

Aktyvi moksleivių grupė nusprendė savo savaitgalius skirti senelių kompiuterinio raštingumo ugdymui. Tačiau pirmiausia, tam, kad šią programą būtų galima įgyvendinti praktiškai, jiems reikia susidaryti ugdymo programą bei susidėti gaires, kurios konkrečiai apibrėžtų, ką norima pasiekti šios kilnios veiklos metu. Vienas pirmųjų ir svarbiausių žingsnių programos planavime yra nuspręsti programos vykdymo laikotarpį, kuris padėtų orientotis galvojant užsiėmimų temas bei kuriam pasibaigus būtų galima išvysti realius veiklos rezultatus. Pamėginsime palengvinti šį planavimo procesą moksleiviams.

Parašykite programą, kuri apskaičiuotų tikslų mėnesio savaitgalio dienų skaičių, žinant visų mėnesio dienų skaičių bei savaitės dieną, kuria menuo prasideda.

Pradiniai duomenys:

Pradiniai duomenys		
JAN	31	SUN
FEB	28	WED
MAR	31	WED
APR	30	SAT
MAY	31	MON
JUN	30	THU
JUL	31	SAT
AUG	31	TUE
SEP	30	FRI
OCT	31	SUN
NOV	30	THU
DEC	31	FRI

Metų mėnesių informacija - n eilučių ($n \leq 12$), kur kiekvienoje eilutėje yra pateiktas sutrupintas mėnesio pavadinimas (sausis (JAN), vasaris (FEB), kovas (MAR), balandis (APR), gegužė (MAY), birželis (JUN), liepa (JUL), rugpjūtis (AUG), rugsėjis (SEP), spalio (OCT), lapkritis (NOV), gruodis (DEC)), mėnesio dienų skaičius, bei pirmosios mėnesio savaitės dienos sutrumpintas pavadinimas (sekmadienis (SUN), pirmadienis (MON), antradienis (TUE), trečiadienis (WED), ketvirtadienis (THU), penktadienis (FRI) šeštadienis (SAT)).

Rezultatai:

n eilučių, kur kiekvienoje eilutėje pateikiamas sutrumpintas mėnesio pavadinimas bei to mėnesio savaitgalio dienų skaičius.

Rezultatai	
JAN	9
FEB	8
MAR	8
APR	10
MAY	8
JUN	8
JUL	10
AUG	8
SEP	9
OCT	9
NOV	8
DEC	10

Reikalavimai

- Duomenų struktūra **Diena**, sauganti savaitės dienos sutrumpintą pavadinimą.
- Duomenų struktūra **Mėnesis**, sauganti mėnesio sutrumpintą pavadinimą, mėnesio dienų skaičių bei pirmą mėnesio savaitės dieną.
- Duomenų struktūra **Metai**, sauganti metų mėnesių duomenis.
- Duomenų struktūra **Kalendorius**, sauganti metų duomenis.
- Pradinių duomenų nuskaitymo metodas **Skaityti**.
- Metodas, grąžinantis nurodyto mėnesio savaitgalio dienų skaičių **GrazintiMėnesioSavaitgalioDienųSkaiciu**.
- Rezultatų išvedimo metodas **Spausdinti**.

3. Tarptautinė migracija

Lietuvos Respublikos statistikos departamentas (<http://osp.stat.gov.lt/>) pateikia informaciją apie tarptautinę migraciją. Pirmiausia įvedami trys sveikieji skaičiai: savivaldybių skaičius s , pradiniai analizės metai $prmetai$, galutiniai analizės metai $gmetai$. Tolesnėse s eilučių surašyti savivaldybių pavadinimai ir kiekvienais tyrimo metais (nuo $prmetai$ iki $gmetai$) į savivaldybę atvykusių imigrantų skaičius. Parašykite programą, kuri analizuoja pateiktus statistinius duomenis:

- kuriais metais kiekvienoje savivaldybėje imigrantų buvo užfiksuota daugiausia. Jei buvo keli vienodi metai, kai imigrantų daugiausia, tai turi būti išvesti vėliausi metai, kai imigrantų užfiksuota daugiausiai;
- kuriais metais kiekvienoje savivaldybėje imigrantų buvo užfiksuota mažiausiai. Jei buvo keli vienodi metai, kai imigrantų buvo užfiksuota mažiausiai, tai turi būti išvesti anksčiausi metai, kai imigrantų užfiksuota mažiausiai;
- kiek imigrantų užregistruota kiekvienais metais iš viso;
- koks vidutinis imigrantų skaičius užfiksuotas per metus (imti visų metų bendrą imigrantų skaičių ir dalinti iš metų skaičiaus);
- surikiuoti savivaldybes pagal bendrą imigrantų skaičių mažėjimo tvarka, o jei skaičiai sutampa – pagal abėcėlę nuo Z iki A;
- įterpti į sąrašo pradžią savivaldybę, kurioje buvo užfiksuotas bendras mažiausias imigrantų skaičius.

Pradiniai duomenys:

6 2010 2014

Biržų 38 95 98 117 144

Kupiškio 16 81 107 96 79

Panevėžio_m 159 495 544 533 510

Panevėžio_r 18 133 181 152 197

Pasvalio 19 99 116 134 141

Rokiškio 37 119 132 161 210

Rezultatai:

Biržų 2014 2010
Kupiškio 2012 2010
Panevėžio_m 2012 2010
Panevėžio_r 2014 2010
Pasvalio 2014 2010
Rokiškio 2014 2010
2010 287
2011 1022
2012 1178
2013 1193
2014 1281
Panevėžio_m 2241
Panevėžio_r 681
Rokiškio 659
Pasvalio 509
Biržų 492
Kupiškio 379
Kupiškio 379
Panevėžio_m 2241
Panevėžio_r 681
Rokiškio 659
Pasvalio 509
Biržų 492
Kupiškio 379

Reikalavimai

- Uždaviniui spręsti reikalingas duomenų struktūras pasirinkite savo nuožiūra.
- Pradinių duomenų įvedimo funkcija **void**.
- Didžiausios ir mažiausios reikšmės paieškos funkcijos, grąžinančios apskaičiuotas reikšmes per funkcijos vardą.
- Sumos skaičiavimo funkcija, grąžinanti apskaičiuotą reikšmę per funkcijos vardą.
- Rikiavimo funkcija **void**.
- Vieno elemento įterpimo funkcija **void**.
- Rezultatų išvedimo funkcija **void**. Rezultatai turi būti išvedami taip, kaip pateikta pavyzdyje.

4. Sportininkų ir sporto trenerių skaičius

Lietuvos Respublikos statistikos departamentas (<http://osp.stat.gov.lt/>) pateikia informaciją apie atskirų sporto šakų sportininkų ir trenerių skaičius įvairiais metais. Pirmiausia įvedamas sporto šakų skaičius s . Toliau nurodyti du sveikieji skaičiai – pradiniai analizės metai p metai ir galutiniai analizės metai g metai. Tolesnėse s eilučių nurodomas sporto šakos pavadinimas ir tiek sveikųjų skaičių dvejetų, kiek buvo analizės metų. Pirmasis skaičius nurodo tos sporto šakos sportininkų skaičių, antrasis – trenerių skaičių. Parašykite programą, kuri analizuoja pateiktus statistinius duomenis:

- kurios sporto šakos trenerių skaičius per analizės laikotarpį pasikeitė mažiausiai (lyginti pirmųjų ir paskutiniųjų analizės metų duomenis). Jei yra kelios tokios sporto šakos, tuomet turi būti išvedama ta sporto šaka, kuri pradiniam sąraše yra žemiau;
- kurios sporto šakos sportininkų skaičius per analizės laikotarpį pasikeitė daugiausiai (lyginti pirmųjų ir paskutiniųjų analizės metų duomenis). Jei yra kelios tokios sporto šakos, tai turi būti išvedama ta, kuri pradiniam sąraše yra aukščiau;
- surikiuokite sporto šakas pagal vidutinį trenerių skaičių mažėjančiai. Jei vidutinis trenerių skaičius sutampa, tuomet reikia rikiuoti sporto šakas pagal abėcėlę nuo A iki Z;
- koks vidutinis trenerių skaičius yra kiekvienoje sporto šakoje nagrinėjamame metų intervale;
- iš surikiuoto sąrašo pašalinkite sporto šaką, kuri yra sąrašo viduryje (jei sporto šakų skaičius lyginis, tai šalinama taip: pvz., jei $s = 6$, tai šalinama 3-čia sporto šaka; jei nelyginis, pvz., $s = 7$, tuomet šalinama 4-ta sporto šaka).

Pradiniai duomenys:

5

2010 2014

Badmintonas 12 906 11 817 11 940 20 2714 15 2234

Dziudo_imtynės 87 4034 88 3814 85 3726 89 3343 82 2964

Futbolas 265 13947 316 18824 285 18329 288 18280 305 19116

Krepšinis 347 22447 373 23151 376 24455 386 23551 436 31273

Plaukimas 130 5573 134 5918 132 5825 138 6261 146 6146

Rezultatai:

```
Badmintonas 3
```

```
-----
```

```
Krepšinis 8826
```

```
-----
```

```
Badmintonas 14
```

```
Dziudo_imtynės 86
```

```
Futbolas 292
```

```
Krepšinis 384
```

```
Plaukimas 136
```

```
-----
```

```
Badmintonas 14
```

```
Dziudo_imtynės 86
```

```
Plaukimas 136
```

```
Futbolas 292
```

```
Krepšinis 384
```

```
-----
```

```
Badmintonas 14
```

```
Dziudo_imtynės 86
```

```
Futbolas 292
```

```
Krepšinis 384
```

Reikalavimai

- Uždaviniui spręsti reikalingas duomenų struktūras pasirinkite savo nuožiūra.
- Pradinių duomenų skaitymo funkcija **void**.
- Didžiausios ir mažiausios reikšmės paieškos funkcijos, grąžinančios apskaičiuotas reikšmes per funkcijos vardą.
- Sumos skaičiavimo funkcija, grąžinanti apskaičiuotą reikšmę per funkcijos vardą.
- Rikiavimo funkcija **void**.
- Vieno elemento šalinimo funkcija **void**.
- Rezultatų spausdinimo ekrane funkcija **void**. Rezultatai turi būti rodomi taip, kaip pateikta pavyzdyje.