

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

**PROGRAMAVIMO PAMOKĖLIŲ UŽDUOTYS
2015-2016 M.M.**

Užduotis rengė: Albertas Dinda, Jūratė Gimbutienė, Irma Lažauskienė, Eimantas Mačius, Deimantas Makrickas, Deividas Andrikis, Arnas Guoba, Tomas Jankūnas, Darius Matulis, Edgaras Rancans, Jurgis Gečys, Mangirdas Kazlauskas, Lukas Skamarakas, Edvinas Šulčius, Beatričė Zalbaitė, Renata Burbaitė

**Kaunas
2015-2016**

Turinys

1.	TIESINIAI ALGORITMAI.....	4
1.	Apelsino tūris	4
2.	Vandens tūris.....	4
3.	Temperatūrų skalės	4
4.	Kontrolinis darbas	5
5.	Saldainiai.....	5
6.	Autobuso atvykimo laikas	5
2.	ŠAKOTIEJI ALGORITMAI	6
1.	Trikampis	6
2.	Dvi dėžutės.....	6
3.	Kelių remontas	6
4.	Bilietas į koncertą.....	7
5.	Triušiai	7
3.	CIKLINIAI ALGORITMAI: ŽINOMO KARTOJIMŲ SKAIČIAUS CIKLAS.....	8
1.	Mažylis ir Karlsonas.....	8
2.	Oro temperatūra.....	8
3.	Testas.....	9
4.	Grikių medus	9
5.	Liūtis	10
6.	Krepšinis	11
4.	CIKLINIAI ALGORITMAI: NEŽINOMO KARTOJIMŲ SKAIČIAUS CIKLAS	11
1.	Medus.....	11
2.	Egzaminas	12
3.	Gėrimų automatas	12
4.	Kredito kortelė.....	13
5.	Gobšumas ir klasta	14
5.	SKAITYMAS IŠ FAILO. RAŠYMAS Į FAILĄ. CIKLAS CIKLE.....	14
1.	Moksleivių pietūs	14
2.	Buitinės technikos gamykla.....	15
3.	Sudėties ar daugybės lentelė	16
4.	Lošimas kauliukais	16
5.	Daugiausia daliklių.....	17
6.	ĮVAIRŪS PIRMO LYGIO UŽDAVINIAI	17
1.	Keturženklių skaičių formavimas.....	17
2.	Amstrongo skaičiai.....	17
3.	Pirmokų žaidimas	18
4.	Gėlių parduotuvė.....	18
7.	FUNKCIJA, GRAŽINANTI APSKAIČIUOTĄ REIKŠMĘ PER FUNKCIJOS VARDĄ	18
1.	Tinkamas automobilis	18
2.	Kelionė	19
3.	Vėlavimas į pamoką	20
4.	Įvairiakraščiai trikapiai.....	21
8.	FUNKCIJA SU PARAMETRAIS-NUORODOMIS.....	21
1.	Siena.....	21
2.	Mėgstamiausi TV kanalai.....	22
3.	Tilto statyba.....	22
4.	Traukiniai	23
5.	Skalbykla.....	23
9.	MASYVAI. SUMOS, KIEKIO, VIDURKIO SKAIČIAVIMAS.....	24
1.	Knygų Kalėdos	24
2.	Pirkiniai	25
3.	Egzamino rezultatai	26
4.	Gyvulių registracija	26

10.	MASYVAI. DIDŽIAUSIOS IR MAŽIAUSIOS REIKŠMIŲ PAIEŠKA.....	27
1.	Skaitomiausios knygos	27
2.	Automobilių modeliukų kolekcija	28
3.	Virusų atakos.....	29
11.	MASYVAI. ELEMENTŲ ŠALINIMAS IR ĮTERPIMAS	30
1.	Knygų registravimas	30
2.	Obuolių pyragų receptai	31
3.	Stebuklingos kortelės	32
4.	Geriausių knygų rinkimai	33
12.	ĮVAIRŪS ANTRO LYGIO UŽDAVINIAI	34
1.	Dėlionės.....	34
2.	Šachmatų turnyras	35
3.	Ekologiški grikiai	36
13.	SIMBOLIAI. SIMBOLIŲ MASYVAI	37
1.	El. pašto dėžutės pavadinimas	37
2.	Slaptas bendravimas	37
3.	Šifravimas	38
4.	Sutrumpinimai	40
5.	SMS šifras	40
14.	SIMBOLIŲ EILUTĖS. SIMBOLIŲ EILUČIŲ MASYVAI	41
1.	Nuotaikos detektorius.....	41
2.	Teksto analizė.....	43
3.	Kryžiažodis	45
4.	Formatavimas	46
5.	Namų darbai	48
15.	STRUKTŪROS, STRUKTŪRŲ MASYVAI	49
1.	Laiko planavimas	49
2.	Grojaraštis	50
3.	Knygų parduotuvė	51
4.	Monopolio bankomatas	53
16.	VEIKSMAI SU STRUKTŪRŲ TIPO DUOMENIMIS IR STRUKTŪRŲ MASYVŲ ELEMENTAIS	55
1.	Knygų parduotuvės apskaita	55
2.	Informatikos konkursas	58
3.	Mokytojai ir mokiniai.....	59
4.	Tarpusavio pagalba	60
17.	STRUKTŪRŲ MASYVŲ RIKIAVIMAS, ELEMENTŲ ŠALINIMAS IR ĮTERPIMAS	62
1.	Prekių apskaita	62
2.	Geriausi mokiniai	64
3.	Prekių tvarkymas.....	65
	Reikalavimai programai	67
4.	Kortų žaidimas	67
5.	Vartotojo sąsaja	70
	Reikalavimai programai	70
18.	ĮVAIRŪS TREČIO LYGIO UŽDAVINIAI	71
1.	Atmosferos teršalai.....	71
2.	Spalvotųjų metalų laužo susidarymas ir panaudojimas	73
3.	Vidutinė temperatūra.....	74
4.	Juridinių asmenų gauta parama	76

1. TIESINIAI ALGORITMAI

1. Apelsino tūris

Rutulio formos apelsino skersmuo d cm. Apelsino žievelės storis yra a cm. Apskaičiuokite, kokį tūrį vam užima apelsino minkštumas. Pradiniai duomenys ir rezultatai yra realiojo (double) tipo. Rezultatai turi būti išvedami dviejų ženklų po kablelio tikslumu.

Duomenys	Rezultatas
įveskite apelsino skersmenį: 10 įveskite apelsino žievelės storį: 1	Apelsino minkštimo tūris: 268.08
įveskite apelsino skersmenį: 15 įveskite apelsino žievelės storį: 1.5	Apelsino minkštimo tūris: 904.78

2. Vandens tūris

Turime kubo formos dėžutę. Kubo briaunos ilgis yra a . Į jį yra įdedamas b skersmens rutulys. Yra žinoma, kad rutulio skersmuo neviršija kubo kraštinės ilgio. Dėžutė pripilama vandens iki viršaus. Parašykite programą, skaičiuojančią, koks yra vandens tūris. Į ekraną taip pat išveskite dėžutės ir rutulio tūrių reikšmes.

Duomenys	Rezultatai
įveskite kubo kraštinės ilgį a : 4 Įveskite rutulio skersmenį b : 4	Kubo tūris: 64.00 Rutulio tūris: 33.49 Vandens tūris: 30.51

3. Temperatūrų skalės

Buityje temperatūrą esame įpratę matuoti Celsijaus laipsniais. Šią skalę pasiūlė švedų astronomas Andersas Celsijus 1742 m. Fizikoje dažnai naudojama Kelvino temperatūrų skalė. Ją sugalvojo airių mokslininkas Viljamas Tomsonas baronas Kelvinas 1848 m.. Ryšys tarp šių skalių yra nesudėtingas, nes laipsniai abiejose skalėse yra vienodo dydžio, skiriasi tik atskaitos taškas.

$$^{\circ}\text{K} = ^{\circ}\text{C} - 273,16$$

Be šių skalių temperatūrai matuoti kai kuriose šalyse (pavyzdžiui JAV) yra naudojama vokiečių fiziko Farenheito skalė, kuri buvo pasiūlyta 1724 m. Čia sąryšis tarp skalių yra sudėtingesnis:

$$9 \cdot ^{\circ}\text{C} = 5 \cdot (^{\circ}\text{F} - 32)$$

Prancūzų mokslininkas Reomiūras 1730 m siūlė skalę, kurios nulinė padala atitiko vandens užšalimo temperatūrą, o vandens virimo temperatūra – 80° , t.y.

$$^{\circ}\text{R} = 0,8 \cdot ^{\circ}\text{C}$$

Parašykite programą, kuri temperatūrą, duotą Kelvino skalėje išreikštą Celsijaus, Farenheito ir Reomiūro skalių temperatūros vienetais.

Skaičius 273,16 ir 0,8 paskelbkite konstantomis.

Rezultatus spausdinkite su dviem skaitmenimis po kablelio, kiekvienam skaičiui išskirdami po 10 pozicijų.

Duomenys įvedami klaviatūra. Rezultatai rodomi ekrane.

Kontroliniai duomenys:

Temperatūrų skalės Įveskite temperatūrą Kelvino skalėje: 310.94 37.78 Celsijaus laipsnių 100.00 Farenheito laipsnių 30.22 Reomiūro laipsnių Programa darbą baigė. Ačiū, kad skaičiavote.

4. Kontrolinis darbas

Miško mokykloje mokosi daug mokinių. Kiekvienoje klasėje mokinių skaičius ne didesnis už 49. Mokykloje naudojama penkiabalė vertinimo sistema. Antrojų klasė parašė kontrolinį darbą. Už kontrolinį darbą $1/7$ klasės mokinių gavo penketus, $1/3$ – ketvertus, $1/2$ – trejetus. Visų kitų klasės mokinių darbai buvo įvertinti nepatenkinamai.

Parašykite programą mokinių skaičiui klasėje rasti, taip pat apskaičiuokite: kiek mokinių gavo: a) penketus, b) ketvertus, c) trejetus, d) nepatenkinamą pažymį.

Rezultatai
Klasėje mokosi 42 mokiniai Klasėje 6 mokiniai gavo penketus Klasėje 14 mokiniai gavo ketvertus Klasėje 21 mokiniai gavo trejetus Klasėje 1 mokinių gavo nepatenkinamą pažymį

5. Saldainiai

Saldainių maišelis kainuoja a eurų. Kiekviename maišelyje yra kuponas. Už tris kuponus galima gauti nemokamai vieną tokį saldainių maišelį. Saldainius už kuponus **galima pirkti 3 kartus**. Kiek daugiausia galima gauti saldainių maišelių už b eurų? Pradiniai duomenys yra sveikojo tipo.

Duomenys	Rezultatai
Įveskite saldainių maišelio kainą: 1 Įveskite pinigų sumą skirtą saldainiams pirkti: 15	22
Įveskite saldainių maišelio kainą: 1 Įveskite pinigų sumą skirtą saldainiams pirkti: 17	24
Įveskite saldainių maišelio kainą: 2 Įveskite pinigų sumą skirtą saldainiams pirkti: 26	18

6. Autobuso atvykimo laikas

Autobusas iš Vilniaus išvyko, kai laikrodis rodė vi valandą mi minučių. Iki Panevėžio autobusas važiavo vp valandų ir mp minučių. Panevėžyje autobusas stovėjo ms minučių. Iš Panevėžio į Rygą autobusas važiavo vr valandų ir mr minučių. Parašykite programą, skaičiuojančią, kuriuo laiku (valandomis v ir minutėmis m) autobusas atvyks į Rygą. Autobusas gali išvykti vakare, o į Rygą atvykti kitą parą. Pradiniai duomenys ir rezultatai yra sveikojo tipo.

Duomenys	Rezultatai
Kada autobusas išvyko iš Vilniaus? 22 30 Kiek laiko autobusas važiavo iš Vilniaus į Panevėžį? 2 5 Kelas minutes autobusas stovėjo Panevėžyje? 10 Kiek laiko autobusas važiavo iš Panevėžio į Rygą? 2 15	Autobusas į Rygą atvyks: 3 val. 0 min.
Kada autobusas išvyko iš Vilniaus? 14 30 Kiek laiko autobusas važiavo iš Vilniaus į Panevėžį? 2 5 Kelas minutes autobusas stovėjo Panevėžyje? 10 Kiek laiko autobusas važiavo iš Panevėžio į Rygą? 2 15	Autobusas į Rygą atvyks: 19 val. 0 min.

2. ŠAKOTIEJI ALGORITMAI

1. Trikampis

Duoti trijų atkarpų ilgiai a , b ir c . Parašykite programą, kuri išvestų į ekraną, kokį trikampį galima sudaryti iš duotų atkarpų: lygiašonį, lygiakraštį, statųjį, įvairiakraštį. Yra žinoma, kad trikampį galima sudaryti tik tada, kai **bet kurių dviejų kraštinių suma yra didesnė už trečiąją**. Stačiojo trikampio kraštinės tenkina Pitagoro teoremą: įžambinės (ilgiausios kraštinės) kvadratas yra lygus statinių (trumpesniųjų kraštinių) kvadratų sumai. Jei trikampio sudaryti negalima, į ekraną turi būti išvedamas pranešimas: „Trikampio sudaryti negalima“.

Pradiniai duomenys	Rezultatas
4 4 4	Trikampis yra lygiakraštis
3 4 5	Trikampis yra statusis
6 6 7	Trikampis yra lygiašonis
4 5 6	Trikampis yra įvairiakraštis
1 4 7	Trikampio sudaryti negalima

2. Dvi dėžutės

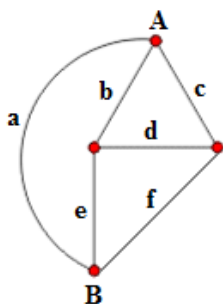
Jūsų rankose dvi dėžutės, kurių išoriniai matmenys yra $a1$, $b1$, $c1$ ir $a2$, $b2$, $c2$. Matmenys yra sveikieji skaičiai, neviršijantys 100. Viena dėžutė telpa į kitą, jeigu jos matmenys nors vienu vienetu yra mažesni už kitos dėžutės atitinkamus matmenis. Dėžutes galima vartyti. Galimos kelios situacijos: pirmoji telpa antrojeje, antroji telpa pirmojoje, abi vienodų matmenų, dėžutės nepalyginamos. Parašykite programą dviem dėžutėms palyginti.

Kontroliniai duomenys

Duomenys $a1, b1, c1, a2, b2, c2$	Rezultatas
5 6 7 7 5 6	Abi vienodų matmenų
3 4 10 8 2 3	Antroji telpa pirmojoje

3. Kelių remontas

Iš taško A į tašką B galima patekti keliais, kurie pažymėti mažosiomis raidėmis: a , b , c , d , e , f . Sudarykite programą, kuri pasakytų ar galima iš taško A patekti į tašką B, kai duota, kuriais keliais galima važiuoti (1), o kurie remontuojami (0).



Duomenys	Rezultatas
Ar galima važiuoti keliu a? 0 Ar galima važiuoti keliu b? 0 Ar galima važiuoti keliu c? 1 Ar galima važiuoti keliu d? 1 Ar galima važiuoti keliu e? 1 Ar galima važiuoti keliu f? 1	Iš taško A į tašką B patekti galima.
Ar galima važiuoti keliu a? 0 Ar galima važiuoti keliu b? 1 Ar galima važiuoti keliu c? 1 Ar galima važiuoti keliu d? 1 Ar galima važiuoti keliu e? 0 Ar galima važiuoti keliu f? 0	Iš taško A į tašką B patekti negalima.

4. Bilietas į koncertą

Dainiui labai patinka viena muzikos grupė. Apie šią grupę jis žino viską. Vieną dieną naujienų portale pamatė informaciją apie šios grupės gastroles Lietuvoje. Aišku, Dainius labai nori patekti į šį koncertą, bet nežino ar ar turės pakankamai pinigų bilietui. Dabar Dainius turi susitaupęs a eurų, be to kiekvieną mėnesį tėvai duoda smulkioms išlaidoms b eurų. Grupė koncertuos po 5 mėnesių, o bilietus pradės platinti po 3 mėnesių. Dainius nori sėdėti kuo arčiau, kad geriau matytų grupės muzikantus.

Užduotis. Nustatykite, ar galės Dainius nusipirkti bilietą, kuriame sektoriuje sėdėdamas Dainius galės klausytis koncerto?

Pradiniai duomenys. Pirmoje eilutėje pateikti Dainiaus turimi pinigai a ir tėvų duodami pinigai smulkioms išlaidoms b.

Antroje eilutėje pateiktos bilietų kainos sektoriuose KC, DJ, PA, EH, FG.

Rezultatai. Sektoriaus pavadinimas, kuriame Dainius stebės koncertą ir bilieto kaina. Jei Dainius negalės nusipirkti bilieto, išveskite žodį **NEGALĖS**.

Pradiniai duomenys	Rezultatas
30 10 50 40 60 35 30	Dainius nusipirks bilietą K arba C sektoriuje bilietas kainuos 50 eurų
15 7 50 40 60 35 30	NEGALĖS

5. Triušiai

Petraičių šeima persikėlė gyventi į sodybą. Norėdami efektyviau išnaudoti sodybos galimybes nutarė auginti triušius. Jie įsigijo tris poreles skirtingų veislių triušių: Belgijos milžinų, Vokietijos dėmėtųjų ir Prancūzijos avinų. Šios veislės tinka ir kailiams, ir mėsai. Belgijos milžinų patelės per metus atsiveda **bm** triušiukų vadas. Vokietijos dėmėtųjų – **vd** ir Prancūzijos avinų patelės gali atsivesti per metus **pa** vadas. Triušiukų skaičiaus vienoje vadoje vidurkis – **trk**. Trijų mėnesių triušiukai veisimui kainuoja **trkain** eurų. Belgijos milžinai užauga iki **svbm** kg, Vokietijos dėmėtieji iki **svvd** kg, Prancūzijos milžinai iki **pasv** kg.

Užduotis. Apskaičiuokite: kurią veislę augindami Petraičiai gaus didžiausią pelną jei triušiukus parduos veisimui; kurios veislės triušiukas užaugs didžiausias.

Pradiniai duomenys. Pirmoje eilutėje pateikti triušiukų vadų kiekis (Belgijos milžinų, Vokietijos dėmėtųjų ir Prancūzijos avinų).

Antroje eilutėje įrašyta suaugusių triušių masė kilogramais (Belgijos milžinų, Vokietijos dėmėtųjų ir Prancūzijos avinų).

Trečioje eilutėje triušiukų vados skaičiaus vidurkis, trijų mėnesių triušiuko kaina.

Rezultatai.

Pirmoje eilutėje: triušių veislė, kurią augindami ir veisdami Petraičiai gaus didžiausią pelną.

Antroje eilutėje: triušių veislė, kurios triušiukas užaugs didžiausias.

Pradiniai duomenys	Rezultatas
3 5 5 8 7 6 6 30	Auginant Prancūzijos avinus arba Vokietijos dėmėtuosius pelnas bus didžiausias Didžiausi užauga Belgijos milžinai

3. CIKLINIAI ALGORITMAI: ŽINOMO KARTOJIMŲ SKAIČIAUS CIKLAS

1. Mažylis ir Karlsonas

Mažylis, Karlsonas ir namų tvarkytoja Frekenbok labai mėgsta aviečių uogienę. Virtuvės spintelėje yra stiklainis su n šaukštelių uogienės. Visi trys smaližiai nepastebimai po vieną nueina į virtuvę ir suvalgo tam tikrą kiekį uogienės: Mažylis – 2 šaukštelių, Karlsonas – 5 šaukštelių, o Frekenbok 3 šaukštelių (žinoma, jei uogienės dar yra). Jų apsilankymus virtuvėje fiksuoja spintelėje esantis jutiklis.

Sudarykite programą, kuri apskaičiuotų uogienės likutį, ir kiek suvalgė kiekvienas smaližius: Mažylis, Karlsonas ir Frekenbok (šaukšteliais). Mažylį žymėkime skaičiumi 1, Karlsoną – 2, o Frekenbok – 3. Pirmiausia reikia įvesti skaičių n , paskui – apsilankymų virtuvėje skaičių m , o tada – skaičius, atitinkančius virtuvėje apsilankiusius asmenis.

Sąlygoje esančius skaičius apiforminkite konstantomis.

Kontroliniai duomenys
Įveskite turimą uogienės kiekį šaukšteliais 12 ir numatomų vizitų skaičių 5 Dabar įveskite visų lankytojų kodus ir paspauskite įvesties klavišą ENTER 1 2 3 2 3 Uogienės neliko Mažyliui teko 2 Karlsonui 7 Panelei Frekenbok 3 šaukšt. uogienės.
Įveskite turimą uogienės kiekį šaukšteliais 13 ir numatomų vizitų skaičių 4 Dabar įveskite visų lankytojų kodus ir paspauskite įvesties klavišą ENTER 1 2 1 1 Liko 2 šaukšt. uogienės Mažyliui teko 6 Karlsonui 5 Panelei Frekenbok 0 šaukšt. uogienės.

2. Oro temperatūra

Oro temperatūra per parą matuojama 7 kartus kas keturias valandas, pradedant nuo vidurnakčio (0 val.). Sudarykite programą, nurodančią, kurią paros valandą buvo aukščiausia oro temperatūra.

Duomenys	Rezultatas
Įveskite matavimus: 0 val. buvo 5 4 val. buvo 6 8 val. buvo 8.5 12 val. buvo 11 16 val. buvo 13 20 val. buvo 10 24 val. buvo 7	Aukščiausia temperatūra buvo 16 val.

3. Testas

Mokiniai rašė penkiolikos minučių testą. Pirmieji mokiniai testą baigė n-tąją minutę. Nuo šio laiko mokytojas kas minutę pildė lentelę, kurioje žymėjo kiek mokinių testą baigė.

Sudarykite programą, skaičiuojančią:

- kiek mokinių iš viso laikė testą,
- kiek mokinių testą baigė n-tąją ir paskutinę (15-tą) minutę,
- kiek vidutiniškai mokinių testą baigė per minutę (raskite aritmetinio vidurkio sveikąją dalį nuo n-tosios iki 15-tos minutės).

Duomenys	Paaiškinimas	Rezultatas
Pirmieji baigė: 8 8 min. baigė 2 9 min. baigė 4 10 min. baigė 3 11 min. baigė 10 12 min. baigė 15 13 min. baigė 6 14 min. baigė 4 15 min. baigė 1	Pirmas(-ieji) mokiniai baigė testą 8-tą minutę. Mokinių, kurie baigė testą 8-tą min., skaičius. Mokinių, kurie baigė testą 9-tą min., skaičius. Mokinių, kurie baigė testą 10-tą min., skaičius. Mokinių, kurie baigė testą 11-tą min., skaičius. Mokinių, kurie baigė testą 12-tą min., skaičius. Mokinių, kurie baigė testą 13-tą min., skaičius. Mokinių, kurie baigė testą 14-tą min., skaičius. Mokinių, kurie baigė testą 15-tą min., skaičius.	Testą rašė 45 mokiniai. 8-tą minutę baigė 2 mokiny(-iai). Paskutinę minutę baigė 1 mokiny(-iai). Vidutiniškai per minutę baigė 6 mokiniai.

4. Grikių medus

Medunešis iš grikių žiedų prasideda nuo liepos 10–15 dienos. Grikių medus pasižymi neįprastu, stipriu, aitriu skoniu. Jis laikomas viena sveikiausių medaus rūšių. Tyrimai rodo, kad iš hektaro grikių lauko bitės gali prinešti nuo 60 iki 120 kg medaus. Tiesa, grikiams turi gausiai žydėti, o oras būti pakankamai drėgnas ir šiltas. Jei įsivyrėja sausi orai, bitės žiedus lanko tik iki 10 valandos, kol jie dar drėgni nuo rasos, todėl pačios geriausios medunešiui drėgnos, bet šiltos dienos, kai ilgai nesisklaido migla.

Užduotis. Apskaičiuokite, kiek stipri bei didelė bičių šeima, kurioje yra 60-80 tūkstančių bičių prineš medaus per n dienų, lietingų dienų kiekį per šį laikotarpį, palankiausių medunešiui dienų kiekį, vidutinį bičių prinešto medaus kiekį per dieną, kai oro sąlygos gali būti:

- oras drėgnas ir šiltas (esant temperatūrai nuo $+21^{\circ}\text{C}$ iki $+25^{\circ}\text{C}$ dieną, o naktį nuo $+14^{\circ}\text{C}$ iki $+16^{\circ}\text{C}$ bitės neša medų visą dieną ir prineša per dieną 3 kilogramus medaus);
- sausi orai (kai temperatūra didesnė už 25°C dieną, o naktį didesnė už 16°C prineša per dieną 1 kilogramą medaus);
- vėsesni orai (kai temperatūra dieną mažesnė už 21°C , o naktį ne didesnė už 14°C prineša per dieną 2 kilogramus medaus);
- visą dieną lyja lietus ir bitės neskrenda rinkti nektaro.

Pradiniai duomenys. Pirmoje eilutėje pateiktas dienų kiekis n, kitose eilutėse pateikti trys skaičiai (oro sąlygos, dienos ir nakties temperatūra) arba vienas skaičius (oro sąlygos), kai lyja lietus.

Duomenų faile oro sąlygos žymimos: 1 – oras šiltas ir drėgnas arba vėsesni orai arba sausi orai;

0 – lyja lietus.

Rezultatai. Kiek bitės prinešė medaus per n dienų, lietingų dienų kiekis, palankiausių dienų medunešiui kiekis, bei kiek vidutiniškai bitės prineša medaus per dieną.

Duomenys.txt	Rezultatas.txt
10 1 21 15 1 20 13 1 19 14 1 18 14 1 17 10 0 0 1 21 14 1 25 15 1 26 17	Per 10 dienų bitės sunešė 18 kilogramų medaus Lietingų dienų kiekis 2 Medunešiui palankiausių dienų kiekis 3 Vidutiniškai bitės per dieną prinešė medaus 1.800 kg
7 1 25 16 1 24 16 1 24 15 1 22 14 1 21 13 1 19 10 0	Per 7 dienas bitės sunešė 14 kilogramų medaus Lietingų dienų kiekis 1 Medunešiui palankiausių dienų kiekis 4 Vidutiniškai bitės per dieną prinešė medaus 2.000 kg

5. Liūtis

Liūtis, tai reiškiny, kai per pakankamai trumpą laikotarpį iš kamuolinių debesų iškrinta labai didelis kiekis kritulių. Pavojingais krituliai laikomi, kai per parą iškrinta 30 mm ir daugiau kritulių. Tokie krituliai laikomi stipria liūtimi, o kai per 12 valandų ir trumpesnę laikotarpį iškrenta 50 mm kritulių, jau laikoma labai stipria liūtimi, - pastebi sinoptikai.

Kritulių kiekis – tai vandens sluoksnio storis (milimetrais), susidaręs ant horizontalaus paviršiaus iš lietaus, dulksnos, ištirpusio sniego ir kt. reiškinių per nurodytą laiko tarpą, kai nėra išgaravimo, nutekėjimo ar susigėrimo ($1 \text{ mm}=1 \text{ l/m}^2$). Kritulių kiekis matuojamas kritulmačiu.

Užduotis. Apskaičiuokite, kiek kritulių iškrito liūties metu, kurią liūties valandą iškrito daugiausia kritulių, kurią mažiausia.

Pradiniai duomenys. Pirmoje eilutėje pateiktas valandų skaičius (liūties trukmė) – n, kitose eilutėse pateikti per valandą iškritusių kritulių kiekiai mm.

Rezultatai. Kritulių kiekis liūties metu, kurią liūties valandą kritulių iškrito daugiausia, kurią valandą mažiausia.

Duomenys.txt	Rezultatas.txt
3 50 90 60	Liūties metu iškrito 200 mm kritulių Daugiausia kritulių iškrito 90 mm per 2 liūties valandą Mažiausia kritulių iškrito 50 mm per 1 liūties valandą Vidutiniškai per valandą iškrito 66.67 mm kritulių
2 50 65	Liūties metu iškrito 115 mm kritulių Daugiausia kritulių iškrito 65 mm per 2 liūties valandą Mažiausia kritulių iškrito 50 mm per 1 liūties valandą Vidutiniškai per valandą iškrito 57.50 mm kritulių

6. Krepšinis

Vyksta paauglių gatvės krepšinio varžybos. Kiekvienoje komandoje yra po penkis žaidėjus. Kaip žinoma, įmetus dvitaškį, yra skiriamas vienas taškas, įmetus tritaškį, 2 taškai. Parašykite programą, kuri:

- apskaičiuotų, kiek kiekviena komanda įmetė taškų;
- kiek vidutiniškai taškų pelnė kiekvienas komandos narys;
- kuri komanda laimėjo, jei surinko vienodai taškų - lygiosios;
- kuris žaidėjas iš visų buvo rezultatyviausias.

Duomenys:

(Komandos pavadinimas)

Duomenys apie žaidėjus:

Jonas 2 4

Antanas 4 1

(vardas) (dvitaškių skaičius) (tritaškių skaičius)

Duomenys	Rezultatas
Seneliai	Komanda Seneliai
Jonas 2 4	Įmetė 40 taškų
Antanas 4 1	Vidutiniškai žaidėjai pelnė: 8 taškų
Arnoldas 2 5	Komanda Greitis
Lebron 2 2	Įmetė 47 taškų
Karolis 4 1	Vidutiniškai žaidėjai pelnė: 9.4 taškų
Greitis	Žaidimas baigėsi lygiosiomis
Mažylis 5 2	Rezultatyviausias žaidėjas buvo: Arnoldas, pelnė
Strazdas 4 2	12 taškų
Plonas 4 4	
Rankelė 1 4	
Mantas 3 3	

4. CIKLINIAI ALGORITMAI: NEŽINOMO KARTOJIMŲ SKAIČIAUS CIKLAS

1. Medus

Bitininkui per medunešį bitės sunėšė a kilogramų medaus. Jis medų supylė į stiklainius. Savo geriausiems draugams padovanojo po vieną stiklainį medaus (po 1 kg). Likusį medų pasiliko, jis pats labai mėgsta arbatas su medumi ir kiekvieną mėnesį sunaudoja nemažai medaus. Kartais jis kepa medaus pyragą, o kur dar kalėdiniai meduoliai. Taigi, kai kuriais mėnesiais sunaudoto medaus kiekis smarkiai padidėja. Bitininkas nusprendė, jei kuris draugas susirgs, jis duos draugui 0,5 kilogramo medaus stiklainį. Iki naujo medunešio 9 mėnesiai.

Užduotis: Apskaičiuokite, kelis mėnesius turės medaus bitininkas, kiek medaus atidavė draugams ir kiek medaus jam liko.

Pradiniai duomenys: Pirmoje eilutėje medaus kiekis kilogramais, draugų kiekis, kuriems bitininkas davė medaus, susirgusių draugų kiekis, antroje eilutėse – bitininko per mėnesį sunaudotas medaus kiekis.

Rezultatas: Ar užteko bitininkui medaus per šį laikotarpį? Kiek medaus atidavė draugams? Kiek medaus suvalgė jis pats? Kiek medaus liko?

Duomenys	Rezultatai
18 5 3 1 1 1 2 2 1 2 1 1	Bitininkui užteks medaus 8 mėnesiams Bitininkas draugams davė 6.5 kg medaus Bitininkas sunaudojo 11.5 kg medaus Bitininkui medaus neliko
30 6 4 1 1 2 2 2 2 1 1 1	Bitininkui užteks medaus visam laikotarpiui Bitininkas draugams davė 8 kg medaus Bitininkas suvalgė 13 kg medaus Bitininkui medaus liko 9 kg

2. Egzaminas

Studentui pirmoji sesija. Jis ruošiasi laikyti pirmąjį egzaminą, todėl studentas sugalvoja ruošimosi egzaminui planą: kiekvieną dieną jis išmoks po a egzamino temų, o likusį laiką skirs egzamino medžiagos kartojimui.

Užduotis. Apskaičiuokite per kiek dienų studentas išmoks visas egzamino temas? Kiek dienų liks kartojimui? Ar šis planas tinkamas?

Planas laikomas tinkamu, jei kartojimui lieka 1 arba 2 dienos, arba bent jau spėja išmokti visas egzamino temas.

Pradiniai duomenys. Pirmoje eilutėje pateiktas egzamino temų skaičius, kurias studentas turi išmokti, antroje – temų skaičius, kurias ketina išmokti per dieną, trečioje – dienų skaičius iki egzamino.

Rezultatai. Pirmoje eilutėje pateikite per kiek dienų studentas išmoko egzamino temas, antroje – kiek dienų liko pakartojimui, trečioje – plano tinkamumą.

Duomenys	Rezultatai
75 20 5	visus klausimus išmoks per 4 dienas kartojimui liks 1 diena mokymosi planas geras
80 15 6	Visus klausimus išmoks per 6 dienas Kartojimui laiko neliks Studentas spės pasiruošti egzaminui, bet nespės pakartoti medžiagos
60 15 3	Visus klausimus išmoks per 4 dienas, tai užtruks ilgiau nei studentas turi laiko iki egzamino Kartojimui laiko neliks Mokymosi planą reikia keisti, nespės išmokti visų egzamino temų

3. Gėrimų automatas

Tarkime, jog kavos ir karštųjų gėrimų prekybos automatas pirma prašo įvesti už kokią sumą eurais perkamas gėrimas, o po to leidžia mesti monetas. Metamų monetų nominalai euro centais: 1, 2, 5, 10, 20, 50.

Sudarykite programą, kuri skaičiuotų metamų monetų sumą ir kiekį. Kiekvieną kart metant monetą, pasakytų kiek jų dar reikia įmesti. Jeigu įvedamas netinkamas monetas nominalas, tai programa teikia informaciją: „Atmesta“. Kai monetų užtenka, rašytų pranešimus:

„Gėrimas ruošiamas“; išvestų įmestų monetų kiekį ir sumą eurais bei gražą, jeigu suma viršija gėrimo kainą.

<i>Duomenys</i>	<i>Rezultatai</i>
Gėrimo kaina 0.45 Meskite monetą. 1 Trūksta 0.44 eur. Meskite monetą. 3 Atmesta. Meskite monetą. 10 Trūksta 0.34 eur. Meskite monetą. 5 Trūksta 0.29 eur. Meskite monetą. 50	Gėrimas ruošiamas. Jūs įmetėte 4 monetas. Iš viso 0.66 eur. Jūsų graža 0.21 eur.
Gėrimo kaina 0.33 Meskite monetą. 50	Gėrimas ruošiamas. Jūs įmetėte 1 monetas. Iš viso 0.50 eur. Jūsų graža 0.17 eur.

4. Kredito kortelė

Kredito kortelės numeruojamos vieningai – numerį sudaro 13–16 skaitmenų, iš kurių du ar trys pirmi nusako kortelės tipą, 6 skaičiai nusako kortelę išdavusį banką ar įstaigą (BIN), o likę skaičiai parenkami pagal tam tikras taisykles, leidžiančias lengvai atpažinti klaidingai įvestą kortelės numerį.

Luno algoritmas: kas antras skaičiaus skaitmuo (skaičiuojama iš kairės) dauginamas iš 2, jei daugybos rezultatas dviženklis skaičius, abu jo skaitmenys sudedami. Gauti nauji skaičiaus skaitmenys sudedami. Jeigu gauta skaitmenų suma baigiasi nuliu, tai įvestas numeris yra teisingas.

Pavyzdžiui, identifikacinis kortelės numeris 49927398716 yra teisingas, nes atlikus algoritmą, skaitmenų suma yra lygi 70.

Užduotis. Nustatykite ar įvestas identifikacinis kortelės numeris yra teisingas. Kortelės numeris gali būti bet kokio ilgio skaičius.

<i>Duomenys</i>	<i>Rezultatai</i>
Nr. 49927398716	Teisingas
Nr. 12121212	Neteisingas
Nr. 1907039	Teisingas
Nr. 32165	Neteisingas
Nr. 19	Teisingas

5. Gobšumas ir klasta

Kartą gobšus žmogus su nelabuoju sudarė sandorį. Žmogus troško pinigų. Sandoris vyko pirmadienį naktį. Pirmąją naktį žmogus panorėjo A auksinių ($1 \leq A \leq 1000000$), kitą pirmadienį – $2A$ auksinių, kiekvieną pirmadienį vis A auksinių daugiau. Nelabasis sutiko, bet tik su viena sąlyga, esą už pajamas reikės mokėti simbolinius mokesčius. Už pirmąją sumą – 1 auksinį, už antrąją – 2 auksinius, už trečiąją – 4 auksinius. Kaskart mokestis dvigubinamas. Gavęs atitinkamą sumą žmogus turi iš karto sumokėti mokesčius. Jei atsitiktų taip, kad žmogus negalėtų sumokėti mokesčių nelabajam – tas paimtų žmogaus sielą. Taip ir sutarė. Mes jau suprantame, kad žmogus vėlios buvo apgautas. Kelintą pirmadienio naktį nelabasis pasiims to gobšaus žmogelio sielą?

Kontroliniai duomenys
Gobšumas ir klasta Kiek auksinių pageidaujate pirmojo pirmadienio naktį? 3 7-ąjį pirmadienį pasiims sielą Dėkojame, kad skaičiavote. Iki kito karto!
Gobšumas ir klasta Kiek auksinių pageidaujate pirmojo pirmadienio naktį? 100 14-ąjį pirmadienį pasiims sielą Dėkojame, kad skaičiavote. Iki kito karto!

5. SKAITYMAS IŠ FAILO. RAŠYMAS Į FAILĄ. CIKLAS CIKLE

1. Moksleivių pietūs

Mokykloje veikia valgykla. Valgyklos maistas puikus, todėl daug moksleivių pietauja mokykloje. Moksleiviai dažnai užsisako kompleksinius pietus, arba atskirus patiekalus. Kompleksinius pietus sudaro trys patiekalai ir vaisių gėrimas.

Užduotis. Žinomas per dieną mokykloje pietavusių moksleivių skaičius, bei kiek ir kokių patiekalų užsisakė kiekvienas valgęs moksleivis. Apskaičiuokite, keli moksleiviai valgė kompleksinius pietus, keli gėrė gėrimą, keli valgė sriubą, keli antrąjį patiekalą ir keli moksleivių pirko desertą.

Pradiniai duomenys. Pirmoje failo Duomenys1.txt eilutėje pateiktas valgykloje pietavusių moksleivių skaičius n , kitose eilutėse pateikti sveikieji skaičiai: pirmasis skaičius parodo, kelis patiekalus valgė moksleivis, jei skaičius 0, tai moksleivis valgė kompleksinius pietus, toliau pateikti skaičiai rodo, kokius patiekalus valgė moksleiviai (1 – gėrimas, 2 – antras patiekalas, 3 – sriuba, 4 – desertas).

Rezultatai. Rezultatų faile Rezultatai1.txt įrašykite, keli moksleiviai valgė kompleksinius pietus, keli gėrė gėrimą, keli valgė sriubą, keli – antrąjį patiekalą ir keli moksleiviai pirko desertą.

Duomenų nuskaitymui ir rezultatų išvedimui naudokite tekstinius failus.

Duomenys1.txt	Rezultatai1.txt
10 0 4 1 2 3 4 3 1 2 4 2 2 3 1 2 3 2 3 4 0 0 1 2 3 1 2 3	3 moksleiviai valgė kompleksinius pietus 3 moksleiviai gėrė gėrimą 7 moksleiviai valgė antrąjį patiekalą 4 moksleiviai valgė sriubą 3 moksleiviai valgė desertą
7 3 1 2 4 4 1 2 3 4 2 1 3 1 4 3 1 3 2 0 0	2 moksleiviai valgė kompleksinius pietus 4 moksleiviai gėrė gėrimą 3 moksleiviai valgė antrąjį patiekalą 3 moksleiviai valgė sriubą 3 moksleiviai valgė desertą

2. Buitinės technikos gamykla

Gamykla „ViRGa“ (Virtuvės reikmenų gamykla) gamina smulkiają virtuvės techniką: pjaustykles, virdulius, mikserius ir plaktuves. Duomenų faile duomenys surašyti tokia tvarka: pirmojoje eilutėje yra duomenų eilučių skaičius n ($1 \leq n \leq 100$). Šiose n eilučių duomenys užrašyti tokia tvarka: gaminio numeris gn , pagamintų gaminių skaičius gs ($1 \leq gs \leq 100$). Toliau eilutėje yra gs skaičių, reiškiančių jų savikainas. Gaminiai koduojami skaičiais pjaustyklė – 1, virdulys – 2, mikseris – 3, plaktuvė – 4.

Apskaičiuokite, kokiai sumai pagaminta kiekvienos rūšies gaminių. Kurių gaminių pagaminta didžiausiai sumai. Visos gaminių savikainų sumos yra skirtingos. Visi skaičiai sveikieji.

Kontroliniai duomenys

duom.txt	rez.txt
6 1 3 125 45 17 2 2 45 71 3 2 56 78 4 1 120 2 1 45 3 1 56	Pjaustyklių: 187 Eur Virdulių: 161 Eur Mikserių: 190 Eur Plaktuvių: 120 Eur Mikserių pagaminta didžiausiai sumai

3. Sudėties ar daugybos lentelė

Sudarykite $n \times n$ dydžio sudėties arba daugybos lentelę.

Duomenų faile įvesti du skaičiai: pirmas nusako aritmetinį veiksmą (1 – sudėtis; 2– daugyba); antras – lentelės dydį.

Duomenų failas	Rezultatų failas
1 5	+ 1 2 3 4 5 <hr/> 1 2 3 4 5 6 2 3 4 5 6 7 3 4 5 6 7 8 4 5 6 7 8 9 5 6 7 8 9 10
2 4	* 1 2 3 4 <hr/> 1 1 2 3 4 2 2 4 6 8 3 3 6 9 12 4 4 8 12 16

4. Lošimas kauliukais

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

Žinoma, jeigu iškrenta lyginis kauliukų skaičius, žaidėjo žaidimas baigiamas, todėl toliau esami akučių skaičiai nesumuojami, nes tada yra sukčiaujama.

Pirmoje duomenų failo Duomenys4.txt eilutėje įrašytas žaidėjų skaičius n . Tolesnėse n eilučių parašyta, kiek kartų žaidėjas metė kauliuką ir atsivertusių akučių skaičiai.

Duomenys4.txt	Rezultatai4.txt
4 3 5 5 6 2 1 6 3 3 5 6 1 2	1 - Žaidėjas surinko 16 taškų 2 - Žaidėjas surinko 7 taškų 3 - Žaidėjas surinko 14 taškų 4 - Žaidėjas surinko 2 taškų Žaidimą laimėjo 1 žaidėjas, jis surinko 16 taškų
6 3 6 5 6 8 1 2 1 2 3 4 6 6 4 3 3 3 2 4 3 3 3 2 5 3 2 6 6 5 8 1 3 5 5 3 1 3 6	1 - Žaidėjas surinko 16 taškų 2 - Žaidėjas surinko 3 taškų 3 - Žaidėjas surinko 11 taškų 4 - Žaidėjas surinko 11 taškų 5 - Žaidėjas surinko 5 taškų 6 - Žaidėjas surinko 27 taškų Žaidimą laimėjo 6 žaidėjas, jis surinko 27 taškų

5. Daugiausia daliklių

Kiekvienas skaičius turi daliklių. Mažiausias, bet kurio skaičiaus, daliklių kiekis yra du (išskyrus vienetą), bet yra skaičių, kurie daliklių turi daug daugiau. Pvz.: skaičius 5 turi du daliklius 1 ir 5; skaičius 8 turi keturis daliklius 1, 2, 4 ir 8.

Užduotis. Žinomas skaičių intervalas nuo a iki b . Suraskite, kuris šio intervalo skaičius turi daugiausia daliklių ir visus jo daliklius.

Pradiniai duomenys. Tekstiniame faile Duomenys5.txt pirmoje eilutėje pateikti du skaičiai a ir b (intervalo pradžia ir pabaiga).

Rezultatai. Tekstiniame faile Rezultatai5.txt pirmoje eilutėje pateikite skaičių, kuris turi daugiausia daliklių ir daliklių skaičių.

Antroje eilutėje visus to skaičiaus daliklius.

Duomenys5.txt	Rezultatai5.txt
10 30	Skaičius 24 turi daugiausia daliklių 8 1 2 3 4 6 8 12 24
25 100	Skaičius 60 turi daugiausia daliklių 12 1 2 3 4 5 6 10 12 15 20 30 60

6. ĮVAIRŪS PIRMO LYGIO UŽDAVINIAI

1. Keturženklių skaičių formavimas

Pirmasis keturženklis skaičiaus skaitmuo įgyja reikšmes iš sveikųjų skaičių intervalo $[a; b]$. Antrasis skaičiaus skaitmuo gaunamas prie pirmojo skaitmens pridėjus vieną, trečiasis – prie antrojo skaitmens pridėjus vieną, ketvirtasis – prie trečiojo skaitmens pridėjus vieną. Jeigu formuojant naują skaitmenį gaunamas dviženklis skaičius, tuomet kaip skaitmuo imamas paskutinis skaitmuo ir kitas skaitmuo gaunamas pridėjus vienetą prie atskirtojo skaitmens.

Pvz., jei keturženklis skaičiaus pirmasis skaitmuo $sk_1 = 1$, tuomet sk_2 bus lygus 2, $sk_3 = 3$, $sk_4 = 4$. O pats keturženklis skaičius bus lygus 1234. Jei $sk_1 = 7$, $sk_2 = 8$, $sk_3 = 9$, $sk_4 = 0$, o pats keturženklis skaičius bus lygus 7890.

Pastaba: reikia spausdinti ne atskirus skaičiaus skaitmenis, o pilnai suformuotą keturženklį skaičių.

Pasitikrinkite: kai $a = 5$, $b = 9$, tuomet turi būti rodomi tokie keturženkliai skaičiai:

5678

6789

7890

8901

9012

2. Amstrongo skaičiai

Amstrongo skaičiais vadinami n -ženkliai natūralieji skaičiai, kurie yra lygūs savo skaitmenų, pakeltų n -tuoju laipsniu, sumai. Pvz., $371 = 3^3 + 7^3 + 1^3$ yra Amstrongo skaičius. Parašykite programą, randančią visus triženklis ir keturženklis Amstrongo skaičius. Skaičiai turi būti rodomi ekrane arba rezultatų faile.

Pasitikrinkite:

153

370

371

407

1634

8208

9474

3. Pirmokų žaidimas

Pirmokai žaidžia žaidimą: jie vienas po kito sako skaičius x iš sveikųjų skaičių intervalo $[1; 100]$, o žaidimo vedėjas skaičiuoja skaičių, kurie baigiasi skaitmeniu, ne didesniu už 5, sumą. Kai vaikai sugalvoja, kad žaisti nebeįdomu, pasako „nulis“ ir žaidimas baigiamas. Parašykite programą, skaičiuojančią: 1) kelis skaičius k pasakė pirmokai; 2) kokią sumą s suskaičiavo žaidimo vedėjas.

Pasitikrinkite: kai $x = 12, x = 10, x = 19, x = 14, x = 29, x = 0$, tuomet $k = 5, s = 36$

4. Gėlių parduotuvė

Pradinių duomenų failo *geles.txt* pirmoje eilutėje įrašytas gėlių parduotuvėje esančių vazoninių gėlių rūšių skaičius n . Tolesnėse n eilučių įrašyta po tris tarpais atskirtus skaičius: pirmas skaičius (sveikasis) r apibūdina gėlės rūšį, antrasis (realusis) s – tos rūšies gėlės vieneto kainą ir trečiasis (sveikasis) k – kiek vienetų tos rūšies gėlių yra parduotuvėje. Į rezultatų failą *gelesrez.txt* įrašykite:

1. pirmoje eilutėje, kiek iš viso gėlių kv yra parduotuvėje;
2. antroje eilutėje, kokią pinigų sumą sv kainuoja parduotuvėje parduodamos gėlės;
3. trečioje eilutėje, kurios rūšies gėlių yra daugiausia ir kiek jų yra;
4. ketvirtoje eilutėje, kurios rūšies gėlės, kainuoja mažiausiai ir kokią pinigų sumą.

<i>geles.txt</i>	<i>gelesrez.txt</i>
3	6
1 12.50 2	58.00
3 10.50 1	2 3
2 7.50 3	3 10.50

7. FUNKCIJA, GRAŽINANTI APSKAIČIUOTĄ REIKŠMĘ PER FUNKCIJOS VARDA

1. Tinkamas automobilis

Išlaikęs vairavimo egzaminą Tomas sugalvojo įsigyti automobilį, tačiau negali apsispręsti, kokiais degalais varomo automobilio norėtų. Norėdamas sužinoti, kurio automobilio eksploatacija kainuos mažiausiai (įvertinant ir automobilio kainą), jis į duomenų failą *Duomenys1.txt* surašė benzinu ir dyzelinu varomų automobilių vidutinės kuro sąnaudas 100 km. Žinant automobilių kainas (benziniu varomo – 2500 eur, dyzelinu – 3200 eur) bei tai, kad Tomas automobiliu važinės du metus ir per metus nuvažiuos 18000 km, apskaičiuokite, kurio automobilio eksploatacija per 2 metus kainuos pigiau. Benzino kaina – 1.12 eur/l, dyzelino – 0.99 eur/l. Kiekvieno automobilio eksploatacijos išlaidas įrašykite į rezultatų failą *Rezultatai1.txt*, po vieną eilutėje dviejų ženklų po kablelio tikslumu.

Paskutinėje eilutėje įrašykite, kurį automobilį (benzinu ar dyzeline varomą) apsimoka įsigyti Tomui.

Duomenų failo pavyzdys	Rezultatų failo pavyzdys
5.8	4838.56
8.2	6122.48 Geriau apsimoka pirkti benzinu varomą automobilį.

Reikalavimai programai

1. funkciją, grąžinančią automobilio eksploatacijos išlaidas per funkcijos vardą;
2. funkciją, grąžinančią išvadą, kurį automobilį apsimoka įsigyti (funkcija turi grąžinti string tipo rezultatą).

2. Kelionė

Iš miesto A į miestą B galima nuvažiuoti dviem keliais. Skiriasi abiejų kelių ilgiai ir maksimalus leistinas greitis. Automobilis pilnu baku gali nuvažiuoti ribotą skaičių kilometrų. Kai baigiasi kuras sustojama prisipilti naujo. Kiekvienas sustojimas pridedamas prie bendro kelionės laiko. Vienas sustojimas visada trunka 5 minutes. Automobilis pradeda kelionę su pilnu baku ir kiekvieną kartą sustojus taip pat pripilamas pilnas bakas. Parašyti programą, kuri apskaičiuotų, kuriuo keliu greičiausiai galima nuvažiuoti iš miesto A į miestą B, visą laiką važiuojant didžiausiu maksimaliu greičiu, įskaičiuojant visus sustojimus kurui pilti.

Pastaba: reikia įvertinti ir tai, kad prieš pradedant kelionę bakui pripildyti sugaištamoms 5 minutėms.

Pradinių duomenų failo *Duomenys2.txt* pirmoje eilutėje pateiktas atstumas *b*, kurį automobilis gali nuvažiuoti su pilnu baku. Antroje eilutėje: *s1* – koks atstumas bus nuvažiuotas važiuojant pirmu keliu, *v1* – maksimalus greitis važiuojant pirmu keliu. Trečioje eilutėje: *s2* – koks atstumas bus nuvažiuotas važiuojant antru keliu, *v2* – maksimalus greitis važiuojant antru keliu.

Rezultatų faile *Rezultatai2.txt* reikia išvesti kiek laiko (valandomis ir minutėmis) užtruks kelionė pirmu ir antru keliu, bei kuriuo keliu važiuojant kelionės tikslas bus pasiektas greičiau.

Pradinių duomenų pavyzdžiai	Rezultatai
60 110 90 180 110	Pirmu keliu: 1 h 23 min Antru keliu: 1 h 53 min Pirmu keliu greičiau
210 552 90 612 110	Pirmu keliu: 6 h 23 min Antru keliu: 5 h 49 min Antru keliu greičiau
150 90 90 100 100	Pirmu keliu: 1 h 5 min Antru keliu: 1 h 5 min Abiem keliais laikas nesiskiria

Reikalavimai programai

- 1) funkciją, suskaičiuojančią, kiek laiko (minutėmis) tęsis kelionė važiuojant maksimaliu greičiu;
- 2) funkciją, suskaičiuojančią kiek kartų reiks stoti įsipilti kuro.

3. Vėlavimas į pamoką

Mokiniai per chemijos pamoką nespėjo atlikti chemijos laboratorinio darbo todėl dalis jų pavėlavo į informatikos pamoką. Pradinių duomenų failo *Duomenys3.txt* pirmoje eilutėje įrašytas mokinių, atėjusių į informatikos pamoką, skaičius n ir informatikos pamokos pradžios laikas (valanda ir minutė). Tolesnėse n eilučių įrašyta po 2 tarpais atskirtus sveikuosius skaičius, nurodančius, kada kiekvienas mokinys atėjo į pamoką (prieš skambutį atėjusių mokinių nėra, su skambučiu atėję mokiniai laikomi nevėluojančiais). Parašykite programą, skaičiuojančią:

- kelias minutes į pamoką vėlavo kiekvienas mokinys;
- keli mokiniai į pamoką pavėlavo ir keli atėjo laiku;
- koks vidutinis vėlavimo laikas minutėmis;
- išvadą, ar galima mokytojui aiškinti naują temą. Jei į pamoką vėlavo mažiau, negu ketvirtadalis visų mokinių, tai mokytojui naują temą aiškinti galima, jei daugiau – negalima.

Duomenų failo pavyzdys	Rezultatų failo pavyzdys
7 11 05 11 10 11 05 11 07 11 12 11 06 11 05 11 05	5 0 2 7 1 0 0 4 3 2.1 NEGALIMA
7 11 05 11 05 11 05 11 07 11 05 11 05 11 05 11 05	0 0 2 0 0 0 0 1 6 0.3 GALIMA

Reikalavimai programai

1. funkciją, skaičiuojančią vieno mokinio vėlavimo trukmę minutėmis;
2. funkciją, grąžinančią išvadą, ar mokytoja gali aiškinti naują temą (string tipo rezultatą grąžinanti funkcija).

4. Įvairiakraščiai trikampiai

Iš sveikųjų skaičių intervalo $[a; b]$ reikia išrinkti ir išspausdinti visų galimų įvairiakraščių trikampių kraštinių ilgius ir plotus.

Pradiniai duomenys a ir b įvedami klaviatūra, rezultatai turi būti pateikiami faile *Rezultatai4.txt*.

Pradiniai duomenys	Rezultatai
5 8	5 6 7 14.70 5 6 8 14.98 5 7 8 17.32 6 7 8 20.33

Reikalavimai programai

Reikia sukurti:

1. funkciją, patikrinančią, ar galima sudaryti trikampį (funkcija grąžina reikšmę true, jei trikampį sudaryti galima, priešingu atveju – false);
2. funkciją, patikrinančią, ar trikampis įvairiakraštis (funkcija grąžina reikšmę true, jei trikampis įvairiakraštis, priešingu atveju – false);
3. funkciją, grąžinančią apskaičiuotą trikampio plotą.

8. FUNKCIJA SU PARAMETRAIS-NUORODOMIS

1. Siena

Statybininkas mūrija sieną. Sienoje planuojama padaryti langą. Statybininkas žino sienos matmenis, plytų, iš kurių statoma siena, matmenis ir lango matmenis. Plytos parduodamos paletėmis. Apskaičiuoti kiek plytų reikės sienos statybai ir kiek palečių su plytomis statybininkui reiks pirkti (visais atvejais apvalinti į didesnę pusę).

Duomenys: pirmoje eilutėje – plytos ilgis ir plotis metrais, antroje eilutėje – sienos ilgis ir plotis metrais, trečioje – lango ilgis ir plotis metrais, ketvirtoje – plytų skaičius paletėje.

Rezultatai: kiek plytų ir kiek palečių reikės sienai sumūryti.

Duomenys1.txt	Rezultatai1.txt
0.1 0.06 5 8 1 2 120	6334 53

Reikalavimai programai

Reikia parašyti:

- Funkciją, gražinančią apskaičiuotą reikšmę per funkcijos vardą, kuri skaičiuotų plotą. Funkciją naudoti plytos, sienos ir lango ploto skaičiavimui.
- Funkciją `void` tipo, kuri suskaičiuotų kiek plytų ir palečių reikės sienai pastatyti.

2. Mėgstamiausi TV kanalai

Rokas žiūrėjo televizorių, išjungęs vieną iš mėgstamų kanalų. Berniukas perjungia kanalus mygtukų **D** (pirmyn) ir **K** (atgal) paspaudimais. Vienas paspaudimas – vienas kanalas. Pirmoje `Duomenys2.txt` failo eilutėje nurodyta kiek yra mėgstamų kanalų, bei kuris kanalas įjungtas dabar. Tolesnėse failo eilutėse surašyti likę kanalai, kuriuos norės įsijungti Rokas.

Rezultatų failo `Rezultatai2.txt` pirmoje eilutėje pateiktas mygtukų paspaudimų skaičius, antroje - mygtukų paspaudimų seka.

Duomenys2.txt	Rezultatai2.txt
3 3 5 7 2	9 DDDDKKKKK

Reikalavimai programai

- Parašykite `void` tipo funkciją apskaičiuojančią, kiek kartų reikės paspausti mygtuką **D** arba **K** (**D** – pirmyn, **K** – atgal), bei išspausdinkite raidžių **D** ir **K** seką, nurodančią mygtukų paspaudimus.

3. Tilto statyba

Reikia pastatyti y metrų ilgio tiltą. Kiekvienam tilto metrui reikia 8 polių. Kiekviena sekli vieta papildomai reikalauja keturių polių. Parašykite programą, skaičiuojančią, ar n polių užteks tiltui pastatyti.

Pradinių duomenų faile `Duomenys3.txt` įrašyti 3 sveikieji skaičiai: polių skaičius, tilto ilgis ir seklumų skaičius.

Rezultatų faile `Rezultatai3.txt` turi būti įrašyta, ar užteks polių tilto statybai ir keli poliai liks nepanaudoti (teigiamas skaičius), pritrūks (neigiamas skaičius), arba bus panaudoti visi poliai.

Duomenys3.txt	Rezultatai3.txt
45 4 2	Polių užteks tiltui pastatyti. 5
40 4 2	Polių užteks tiltui pastatyti. 0
38 4 2	Polių neužteks tiltui pastatyti. -2

Reikalavimai programai

- Parašykite funkciją per parametrus grąžinančią:
 1. ar užteks polių tiltui pastatyti (string tipo kintamasis);
 2. polių skaičius (int tipo kintamasis).

4. Traukiniai

Mažoji Maška pradėjo mokytis fizikos, ir jau žino, kad kūno padėčiai erdvėje nusakyti yra reikalingas atskaitos taškas, kurio atžvilgiu kūnas judės.

Kad visa tai geriau įsisavintų, ji nušlepsėjo prie bėgių esančių netoli jos namų. Savo pačios nuostabai, Maška išvydo du, vieną priešais kitą skirtingais bėgiais judančius traukinius. Mergaitė iškart nusprendė, jog atskaitos tašku ji nori pasirinkti traukinių prasilenkimo vietą (prasilenkimu laikomas traukinių priekų susitikimas). Kai įvyksta prasilenkimas, ji aktyvuoja laikmatį ir stebi kokį atstumą metrais traukiniai nuvažiuoja iki sustabdant laikmatį. Remdamasi matavimų duomenimis mergaitė apskaičiuoja abiejų traukinių greičius.

Parašykite programą, kuri padėtų rasti Maškai abiejų traukinių greičius.

Pradiniai duomenys: matavimo laikas t (sekundėmis) ir abiejų traukinių nuvažiuoti atstumai a_1 ir a_2

(metrais) įvedami klaviatūra. Pradiniai duomenys yra sveikieji skaičiai.

Rezultatus – abiejų traukinių greičius km / val – pateikite ekrane dviejų ženklų po kablelio tikslumu.

Duomenys	Rezultatai
5 20 25	Pirmo traukinio greitis: 14.40 Antro traukinio greitis: 18.00

Reikalavimai programai

- Traukinių greičių skaičiavimui parašykite `void` tipo funkciją.

5. Skalbykla

Studentų bendrabutyje įrengta savitarnos skalbykla, kuri dirba 12 valandų per parą ir kurioje yra 3 skalbimo mašinos. Skalbykloje esančios skalbimo mašinos skalbia dviejų rūšių skalbinius: medvilninius ir visus likusius. Vienu skalbimu galima skalbti iki 5kg skalbinių. Skalbimo mašinos darbo vadove rašoma, kad medvilniniai skalbiniai vienu skalbimu skalbiami 60 min, o visi likę – 50 min. Tačiau, studentai pastebėjo, kad vieno skalbimo trukmė iš tikrųjų svyruoja iki 5 min. Pvz. medvilniai skalbiniai gali būti skalbiami 55 - 65 min., o visi likę 45 - 55 min.

Apskaičiuokite kiek minučių daugiausiai ir mažiausiai užtruks kiekvieno studento skalbinių skalbimas, taip pat kaip kokia bus minimali ir maksimali skalbimo kaina, laikant, kad skalbimas apmokestinamas pagal valandas. Taip pat suskaičiuokite, kiek studentų suspės išskalbti skalbinius per vieną dieną iki skalbyklos darbo laiko pabaigos.

Pradinių duomenų faile *Duomenys5.txt* pirmoje eilutėje duotas skalbinius skalbiančių studentų skaičius ir vienos valandos skalbimo kaina. Sekančiose eilutėse – duomenys apie kiekvieno studento skalbinius: medvilninių skalbinių svoris ir visų likusių skalbinių svoris.

Rezultatų faile *Rezultatai5.txt* reikia išvesti kiekvieno studento minimalią ir maksimalią skalbinių skalbimo trukmę, minimalią ir maksimalią skalbimo kainą, o taip pat kiek studentų spės išskalbti skalbinius per vieną dieną.

Pradinių duomenų pavyzdys	Rezultatai
9 3.2	220-260 11.73-13.87
6 7	330-390 17.60-20.80
11.6 12.8	165-195 8.80-10.40
8 3.2	110-130 5.87-6.93
4 2.4	220-260 11.73-13.87
6.5 7	495-585 26.40-31.20
20 23	220-260 11.73-13.87
9 10	165-195 8.80-10.40
3 7	165-195 8.80-10.40
4 9	Rūbus suspės išskalbti: 7 studentai

Reikalavimai programai

Reikia sukurti dvi funkcijas:

1. Funkciją su parametrais-nuorodomis, suskaičiuojančią minimalią ir maksimalią vieno studento skalbimo trukmę.
2. Funkciją su parametrais-nuorodomis, suskaičiuojančią minimalią ir maksimalią vieno studento skalbimo kainą.

9. MASYVAI. SUMOS, KIEKIO, VIDURKIO SKAIČIAVIMAS

1. Knygų Kalėdos

Prieš kiekvienas Kalėdas skelbiama akcija „Knygų Kalėdos“. Šios akcijos metu mokyklų bibliotekoms dovanojamos knygos. Šiais metais mokykla gavo n skirtingų pavadinimų knygų. Kai kurių knygų buvo ne po vieną vienetą. Bibliotekininkė turi įtraukti knygas į apskaitą. Ji paprašė programuojančių mokinių pagalbos, nes reikia apskaičiuoti:

- už kokią pinigų sumą buvo padovanota kiekvieno pavadinimo knygų;
- už kokią pinigų sumą buvo padovanota knygų iš viso;
- kelių pavadinimų knygų buvo padovanota daugiau kaip po 5 vienetus.

Mokiniai sutiko padėti ir paprašė bibliotekininkės suformuoti pradinių duomenų failą *Duomenys1.txt*, kurio pirmoje eilutėje įrašyta, kiek skirtingų pavadinimų knygų buvo padovanota, o likusiose n eilučių – informacija apie kiekvieno pavadinimo knygą – knygos vieneto kaina (realusis skaičius) ir to pavadinimo knygų kiekis (sveikasis skaičius).

Rezultatų faile *Rezultatai1.txt* turi būti:

- pirmosiose n eilučių – vieno pavadinimo knygos vieneto kaina, knygų skaičius, visų to pavadinimo knygų kaina;
- po to išvedama pinigų suma, už kurią padovanota knygų;
- kelių pavadinimų knygų buvo padovanota daugiau kaip po 5 vienetus.

Duomenys1.txt	Rezultatai1.txt
3	5.65 2 11.30
5.65 2	4.92 12 59.04
4.92 12	8.71 3 26.13
8.71 3	96.47
	1

Reikalavimai programai

- Naudokite sveikųjų ir realiųjų skaičių masyvus: sveikųjų – kiekiams, realiųjų – knygos vieneto kainai ir pilnai kainai saugoti.
- Sukurkite pradinių duomenų skaitymo funkciją `void`. Joje į masyvus sudėkite knygų kiekius ir knygos vieneto kainas.
- Sukurkite funkciją `void`, kuri suformuotų knygų pilnų kainų masyvą.
- Sukurkite visų knygų kainų sumos skaičiavimo funkciją, grąžinančią apskaičiuotą sumą per funkcijos vardą.
- Sukurkite kiekio skaičiavimo funkciją, grąžinančią apskaičiuotą kiekį per funkcijos vardą.
- Rezultatų išvedimui į failą parašykite funkciją `void`.

2. Pirkiniai

Pirkėjas turi s pinigų sumą (s sveikasis skaičius). Parduotuvėje yra n prekių, kurių kiekvieną aprašo prekės kodas (sveikasis šešiaženklis skaičius) ir kaina (sveikasis skaičius). Reikia parašyti programą, randančią, kurias iš parduotuvėje parduodamų prekių gali įsigyti pirkėjas. Pirkėjas prekes renkasi iš eilės. Jei prekei įsigyti trūksta pinigų, pirkėjas toliau prekių nebesirenka.

Pirmoje pradinių duomenų failo `Duomenys2.txt` eilutėje duota kiek pirkėjas turi pinigų. Antroje eilutėje nurodyta kiek yra prekių. Sekančiose n eilutėse duotas prekės kodas ir kaina. Rezultatų faile `Rezultatai2.txt` reikia išvesti kiekvienos prekės kodą ir atsakymą, ar pirkėjas galės prekę nusipirkti. Paskutinėje failo eilutėje išvesti, kiek daugiausia prekių pirkėjas gali nusipirkti už turimą pinigų sumą.

Duomenys2.txt	Rezultatai2.txt
45 7 156864 5 154684 12 148463 25 185867 45 156884 50 189755 69 186417 132	156864 TAIP 154684 TAIP 148463 TAIP 185867 TAIP 156884 NE 189755 NE 186417 NE Galima nusipirkt 3 prekes
56 8 156864 1 154684 3 148463 15 185867 16 156884 69 189755 75 154789 76 186417 398	156864 TAIP 154684 TAIP 148463 TAIP 185867 TAIP 156884 NE 189755 NE 154789 NE 186417 NE Galima nusipirkt 4 prekes

Reikalavimai programai

- Naudokite sveikųjų skaičių ir eilutės tipo (string) masyvus.
- Parašykite pradinių duomenų skaitymo funkciją `void`.
- Parašykite funkciją `void`, atrenkančią, kurias prekes galima nusipirkti (sudaryti masyvą su string tipo reikšmėmis TAIP ir NE).
- Parašykite funkciją, grąžinančią apskaičiuotą reikšmę per funkcijos vardą, kuri suskaičiuotų kiek prekių galima nusipirkti.
- Parašykite rezultatų įrašymo į failą funkciją `void`.

3. Egzamino rezultatai

Sausio mėnesį vyksta egzaminų sesija. Studentai laikė pirmąjį egzaminą. Parašykite programą, kuri apskaičiuotų kiekvienos grupės vidurkį.

Sausio mėnesį vyksta egzaminų sesija. Studentai laikė pirmąjį egzaminą. Parašykite programą, kuri apskaičiuotų kiekvienos grupės pažymių vidurkį.

Pirmoje pradinių duomenų failo Duomenys3.txt eilutėje pateikta, kelios grupės n laikė egzaminą. Kitose n eilučių pateiktas grupės pavadinimas, grupės studentų skaičius ir jų gauti pažymiai.

Rezultatų faile Rezultatai3.txt turi būti n eilučių, kuriose turi būti: grupės pavadinimas ir vidurkis, dviejų ženklų po kablelio tikslumu.

Duomenys3.txt:	Rezultatai3.txt
4	IF4/8 7.80
IF4/8 5 5 7 8 9 10	IF4/9 6.80
IF4/9 5 8 5 10 5 6	IF4/10 6.50
IF4/10 4 5 5 7 9	IF3/5 7.5
IF3/5 4 6 7 9 8	

Reikalavimai programai

- Naudokite sveikųjų (vienos grupės studentų pažymiams saugoti), realiųjų (vidurkiams saugoti) skaičių ir eilučių (string) (grupių pavadinimams saugoti) masyvus.
- Parašykite vidurkio skaičiavimo funkciją, grąžinančią apskaičiuotą reikšmę per funkcijos vardą, ir ją panaudokite kiekvienos grupės vidurkio skaičiavimui.
- Pradinius duomenis skaitykite ir apdorokite gabaliukais (po vieną grupę). Sukurkite `void` funkciją.
- Parašykite rezultatų įrašymo į failą funkciją `void`.

4. Gyvulių registracija

Stasys užsiima gyvulininkyste - auginama avis bei karves. Kadangi jis privalo registruoti savo gyvulius, viskas yra fiksuojama duomenų bazėje. Tačiau viskas buvo rašoma viename faile, kuriame informacija, ūkiui besiplečiant, pradėjo painiotis, taigi buvo nuspręsta duomenis išskirti į atskirus failus.

Užduotis: parašykite programą, kuri iš duomenų failo Duomenys4.txt nuskaitytų visus duomenis ir išvestų juos į atskirus rezultatų failus, priklausomai nuo gyvulio (karvės surašomos į failą RezultataiK.txt, avys – į failą RezultataiA.txt). Taip pat realizuokite funkciją, kuri pagal įvestą registracijos numerį nustatytų, tai yra karvė ar avis. Registracijos numeris įvedamas klaviatūra, rezultatas išvedamas į ekraną.

Duomenų faile Duomenys4.txt duomenų kiekis yra nežinomas. Kiekvienoje eilutėje pirma raidė nurodo gyvulio tipą (A - avis; K - karvė), o toliau seka registracijos numeris.

Rezultatai: gyvulių registracijos numeriai išvedami į du skirtingus rezultatų failus (rezultataiA.txt ir rezultataiK.txt)

Duomenys4.txt	RezultataiA.txt	RezultataiK.txt
A AV1538	Avių kodai:	Karvių kodai:
A AV9981	AV1538	KA8136
K KA8136	AV9981	KA9882
A AV4687	AV4687	KA2136
K KA9882	AV5136	KA8792
K KA2136	AV8662	KA5633
A AV5136	AV1368	
A AV8662	AV6549	
K KA8792	AV7823	
A AV1368		
K KA5633		
A AV6549		
A AV7823		

Reikalavimai programai

- Naudokite simbolių eilučių masyvus.
- Sukurkite pradinių duomenų skaitymo funkciją void. Joje gali būti formuojami karvių ir avių registracijos numerių masyvai.
- Sukurkite rezultatų rašymo į failą funkciją void. Į funkciją kreipkitės 2 kartus: 1) rašydami avių kodus; 2) rašydami karvių kodus.

10. MASYVAI. DIDŽIAUSIOS IR MAŽIAUSIOS REIKŠMIŲ PAIEŠKA

1. Skaitomiausios knygos

Leidyklos „Alma Littera“ n ($n < 21$) knygų vertina m ($m < 11$) skaitytojų. Kiekvienas skaitytojas kiekvieną knygą vertina balais nuo 0 iki 10.

Vertinimai yra sveikieji skaičiai. Pradinių duomenų failo *knygos.txt* pirmoje eilutėje įrašyti skaičiai n ir m . Tolesnėse n eilučių įrašytas knygos kodas k (sveikasis skaičius iš intervalo $[1; 20]$) ir po m sveikųjų skaičių, vienas nuo kito atskirtų tarpais – kiekvieno vertintojo knygai skirti balai. Skaičiuojant galutinį knygos įvertinimą, **sumuojami didžiausias ir mažiausias įvertinimai** (jei didžiausių ir mažiausių įvertinimų yra daugiau, negu po vieną, tuomet sumuojama tik po vieną didžiausią ir mažiausią įvertinimą). Apskaičiuota suma yra galutinis knygos įvertinimas. Parašykite programą, surandančią ir rezultatų faile *knygosrez.txt* spausdinančią:

Koks yra galutinis kiekvienos knygos įvertinimas. Rezultatus surašykite faile vienai knygai skirdami vieną eilutę: knygos kodas ir įvertinimas, atskirti tarpais. Sąrašas išdėstytas knygų kodų didėjimo tvarka

Kuri knyga įvertinta blogiausiai. Jei kelios knygos įvertintos blogiausiai, jų visų kodai turi būti spausdinami vienoje eilutėje didėjimo tvarka, atskiriant juos vieną nuo kito tarpais.

Kuri knyga įvertinta geriausiai. Jei kelios knygos įvertintos geriausiai, jų visų numeriai turi būti spausdinami vienoje eilutėje mažėjimo tvarka.

<i>knygos.txt</i>	<i>knygosrez.txt</i>
5 6 2 2 5 6 6 5 2 1 3 8 8 7 7 7 4 4 5 5 7 7 6 3 5 5 5 5 3 3 5 7 8 8 7 2 6	Knygų įvertinimai: 1 11 2 8 3 8 4 11 5 10 Blogiausiai įvertintos knygos: 2 3 Geriausiai įvertintos knygos: 4 1

Reikalavimai programai

- Naudokite sveikųjų skaičių masyvus.
- Sukurkite pradinių duomenų skaitymo funkciją `void`.
- Sukurkite mažiausios ir didžiausios reikšmių paieškos funkcijas, grąžinančias apskaičiuotą reikšmę per funkcijos vardą.
- Sukurkite `void` funkciją, faile spausdinančią blogiausiai įvertintas knygas.
- Sukurkite `void` funkciją, faile spausdinančią geriausiai įvertintas knygas.
- Sukurkite `void` funkciją, faile spausdinančią knygų kodus ir įvertinimą.

2. Automobilių modeliukų kolekcija

Keli draugai nusprendė surinkti automobilių modeliukų kolekciją ir padaryti parodą. Jie susitarė, kad parodą bus galima daryti gimnazijos skaitykloje. Kad eksponatai būtų matomi ir neužgožtų vienas kito, buvo nuspręsta juos suskirstyti į g grupių pagal dydį. Lentynose skirtingo dydžio modeliukų telpa skirtingas skaičius – pačių didžiausių mažiausiai, o mažiausių – daugiausiai.

Pradinių duomenų failo *auto.txt* pirmoje eilutėje įrašyti du sveikieji skaičiai – kolekcininkų skaičius n ($3 \leq n \leq 10$) ir į kelias grupes g ($3 \leq g \leq 6$) skirstomi modeliukai. Antroje failo eilutėje įrašyta g sveikųjų skaičių – kiek kiekvieno dydžio modeliukų telpa lentynoje. Tolesnėse n eilučių surašyta po g sveikųjų skaičių, vienas nuo kito atskirtų tarpais – kiek kokio dydžio modeliukų atnešė mokinys. Jei kurio nors dydžio modeliuko mokinys į parodą neatnešė, toje vietoje rašomas nulis.

- Rezultatų failo *autorez.txt* pirmosiose g eilučių turi būti įrašyta, kiek yra kiekvieno dydžio modeliukų ir kiek lentynų reikės kiekvieno dydžio automobilių modeliukams eksponuoti (paskutinėse kiekvieno dydžio modeliukų lentynose gali likti tuščios vietos, skirtingų dydžių modeliukai į vieną lentyną nestatomi).
- Tolesnėje eilutėje rašoma, kiek iš viso lentynų reikės norint eksponuoti kolekciją.
- Tolesnėje eilutėje rašoma, kurio dydžio modeliukams reikės daugiausiai ir kurio dydžio - mažiausiai lentynų.

<i>auto.txt</i>	<i>autorez.txt</i>
3 4 2 3 4 6 1 2 0 1 5 0 3 1 0 3 1 5	6 3 5 2 4 1 7 2 8 1 3

Reikalavimai programai

- Rašydami programą naudokite sveikųjų skaičių masyvus.
- Parašykite funkciją, skaičiuojančią, kiek kiekvieno dydžio modeliukų yra iš viso.
- Parašykite funkciją, skaičiuojančią, kiek lentynų reikės kiekvieno dydžio modeliukams.
- Parašykite funkciją, skaičiuojančią, kiek iš viso lentynų reikės modeliukų ekspozicijai.
- Parašykite funkciją, randančią, kurio dydžio modeliukams eksponuoti reikės daugiausiai lentynų.
- Parašykite funkciją, randančią, kurio dydžio modeliukams eksponuoti reikės mažiausiai lentynų.

3. Virusų atakos

Pastaruoju metu labai suaktyvėjo virusų atakos. n programuotojų nusprendė atlikti virusų aktyvumo analizę. Jie d mėnesio dienų stebėjo v virusų rūšių atakų aktyvumą.

Pradinių duomenų failo *virusai.txt* pirmoje eilutėje įrašyti trys sveikieji skaičiai, atskirti tarpais: programuotojų skaičius n ($3 \leq n \leq 10$), dienų, kai buvo stebėtos virusų atakos, skaičius d ($1 \leq d \leq 31$), stebėtų virusų rūšių skaičius v ($3 \leq v \leq 7$). Antroje failo eilutėje įrašyta d sveikųjų skaičių – dienų, kada buvo stebėti virusai, numeriai. Tolesnėse $n \times v$ eilučių įrašyta po d sveikųjų skaičių – kiekvieno programuotojo kiekvienos rūšies virusų stebėtų atakų skaičius – vienos rūšies virusui skiriama viena eilutė. Jei kurią nors dieną viruso ataka nestebėta, toje vietoje rašomas nulis.

Rezultatų faile *virusairez.txt* turi būti įrašyta:

- pirmosiose n eilučių – kiek virusų atakų iš viso užregistravo kiekvienas programuotojas;
- tolesnėse d eilučių – kiek virusų atakų iš viso užregistruota kiekvieną stebėjimo dieną. Spausdinamas dienos numeris ir tarpu atskirtas tos dienos atakų skaičius;
- tolesnėse v eilučių – kiekvieno viruso atakų skaičius per visą stebėjimo laiką;
- tolesnėje eilutėje – kiek iš viso virusų atakų užregistruota per visą stebėjimo laiką;
- tolesnėje eilutėje – kurią dieną užregistruota mažiausiai virusų atakų. Spausdinamas dienos numeris ir atakų skaičius. Jei yra kelios tokios dienos, tuomet spausdinti tą dieną, kurios numeris didžiausias;
- tolesnėje eilutėje – kurios rūšies virusų atakų užregistruota daugiausiai. Jei yra kelios tokios virusų rūšys, tai spausdinti tos numerį, kuris yra mažiausias;
- tolesnėje eilutėje – kiek vidutiniškai virusų atakų užregistravo programuotojas per visą stebėjimo laiką. Rezultatą spausdinkite vieno skaitmens po kablelio tikslumu.

virusai.txt	virusairez.txt
3 3 4	18
1 3 8	12
1 1 1	17
1 0 0	1 12
2 5 4	3 22
0 1 2	8 13
2 2 0	11
2 0 1	10
1 0 2	18
0 1 1	8
1 2 1	47
1 4 1	1 12
1 3 0	3
0 3 0	15.7

Reikalavimai programai

- Turi būti naudojami sveikųjų skaičių masyvai.
- Turi būti funkcija, skaičiuojanti kiekvienos virusų rūšies atakas.
- Turi būti funkcija, skaičiuojanti, kiek iš viso virusų atakų buvo užfiksuota per visą stebėjimo laiką.
- Turi būti funkcija, randanti kurią dieną užregistruota mažiausiai virusų atakų.
- Turi būti funkcija, randanti kurios rūšies virusų atakų užregistruota daugiausiai.
- Turi būti funkcija, randanti kiek vidutiniškai virusų atakų užregistravo programuotojas per visą stebėjimo laiką.

11. MASYVAI. ELEMENTŲ ŠALINIMAS IR ĮTERPIMAS

1. Knygų registravimas

Bibliotekos duomenų bazėje yra n knygų, kurioms priskirtas bet koks numeris. Visi numeriai surikiuoti nuo mažiausio iki didžiausio. Kiekvieną dieną knygos paimamos ir gražinamos į biblioteką. Kai knyga paimama, jos numeris ištrinamas iš duomenų bazės, kai knyga gražinama – jos numeris įvedamas atgal į atitinkamą vietą. Parašyti programą, kuri išvestų knygų sąrašą po paėmimo ir knygų sąrašą po paėmimo ir pridėjimo.

Duomenys: pirmoje eilutėje - kiek iš viso yra knygų bibliotekos duomenų bazėje (n), antroje eilutėje yra paimtų knygų kiekis ir numeriai, trečioje eilutėje – kiek gražinta knygų ir tų knygų numeriai. Tolesnėse eilutėse – bibliotekoje esančių knygų numeriai.

Duomenys	Rezultatai
8 3 4 10 13 2 2 11 1 3 4 7 9 10 13 14	knygų sąrašas po išėmimo: Nr. 1 Nr. 3 Nr. 7 Nr. 9 Nr. 14 knygų sąrašas po išėmimo ir pridėjimo: Nr. 1 Nr. 2 Nr. 3 Nr. 7 Nr. 9 Nr. 11 Nr. 14
8 4 1 9 10 14 1 5 1 3 4 7 9 10 13 14	knygų sąrašas po išėmimo: Nr. 3 Nr. 4 Nr. 7 Nr. 13 knygų sąrašas po išėmimo ir pridėjimo: Nr. 3 Nr. 4 Nr. 5 Nr. 7 Nr. 13

Reikalavimai programai

- Naudokite sveikųjų skaičių masyvus.
- Masyvo elementų šalinimo funkcija.
- Masyvo elementų įterpimo funkcija.

2. obuolių pyragų receptai

Naujai išleistoje pyragų receptų knygoje gausu obuolių pyragų receptų, kurių sudedamosios dalys labai panašios. Jurgita iš knygos išrinko n receptų ($n < 30$) ir surašė kiekvieno pyrago sudedamąsias dalis bei gaminimo ypatumus: kelių obuolių, kiaušinių reikia pyragui; kiek gramų miltų, cukraus sunaudojama, kiek laiko pyragas kepamas ir kokioje temperatūroje.

Pradinių duomenų failo *pyragai.txt* pirmoje eilutėje įrašytas pyragų skaičius n . Tolesnėse n eilučių įrašyta po 2 sveikuosius skaičius: obuolių ir kiaušinių skaičius, bei po 4 realiuosius skaičius: miltų, cukraus kiekiai gramais, kepimo laikas valandomis, kepimo temperatūra. Parašykite programą, skaičiuojančią:

- kelių obuolių ir kelių kiaušinių reikės norint iškepti visus sąraše išvardintus pyragus;
- kiek gramų miltų ir cukraus vidutiniškai reikia vienam pyragui;
- keli buvo pyragai, kurie kepa ne mažiau kaip v valandų;
- keli buvo pyragai, kurie keps ne mažesnėje kaip t laipsnių temperatūroje;

Pastaba: v ir t reikšmės įvedamos klaviatūra pagrindinėje funkcijoje `main()`.

- pašalinkite iš pradinio sąrašo receptus, pagal kuriuos pagaminti pyragai keps ne mažesnėje kaip 200 laipsnių temperatūroje ir išspausdinkite sąrašą be pašalintų elementų.

pyragai.txt	pyragairez.txt
3	14 7
7 4 250.0 75.0 1.5 180.0	166.7 50.0
3 1 100.0 25.0 0.75 210.0	2
4 2 150.0 50.0 1.0 200.0	3
$v = 1.0, t = 180.0$	7 4 250.0 75.0 1.5 180.0

Reikalavimai programai

- Turi būti naudojami sveikųjų ir realiųjų skaičių masyvai.
- Turi būti pradinių duomenų skaitymo funkcija `void`.
- Turi būti sumos skaičiavimo funkcija `Suma()`, grąžinanti apskaičiuotą reikšmę per funkcijos vardą. Į funkciją reikia kreiptis 2 kartus: su obuolių ir kiaušinių masyvais.
- Turi būti vidurkio skaičiavimo funkcija `Vidurkis()`, grąžinanti apskaičiuotą reikšmę per funkcijos vardą. Į funkciją reikia kreiptis 2 kartus: su miltų ir cukraus masyvais.
- Turi būti kiekio (pyragų skaičiaus) skaičiavimo funkcija `Keli()`, kuri grąžina apskaičiuotą pyragų skaičių. Į funkciją reikia kreiptis 2 kartus: su kepimo laiko bei kepimo temperatūros masyvais.
- Turi būti masyvo elementų šalinimo funkcija.
- Turi būti rezultatų išvedimo į failą funkcija `void`.

3. Stebuklingos kortelės

Stalo žaidimą „Stebuklingos kortelės“ žaidžia n žaidėjų ($n < 21$), kurių numeriai yra sveikieji skaičiai iš intervalo $[1; 20]$. Kiekvienas žaidėjas, laikydamasis žaidimo taisyklių, padaro ėjimą ir užsirašo, kelis taškus gauna (sveikasis skaičius). Pradiniai duomenys surašyti tekstiniam faile *zaidimas1.txt*. Pirmoje failo eilutėje įrašytas ėjimų skaičius m ($m < 100$). Tolesnėse m failo eilučių įrašyta po du sveikuosius skaičius – žaidėjo numeris ir taškų skaičius. Parašykite programą, surandančią ir rezultatų failę *zaidimas1rez.txt* spausdinančią:

- Kokia galutinė kiekvieno žaidėjo surinktų taškų suma. Surinktais laikomi tik teigiami sveikieji skaičiai. Rezultatus surašykite faile vienam žaidėjui skirdami vieną eilutę: žaidėjo numeris ir surinktų taškų suma, atskirti tarpais.
- Kuris žaidėjas surinko mažiausiai taškų. Jei keli žaidėjai surinko po vienodai taškų, jų visų numeriai turi būti spausdinami vienoje eilutėje mažėjimo tvarka, atskiriant juos vieną nuo kito tarpais.
- Kuris žaidėjas surinko daugiausiai taškų. Jei keli žaidėjai surinko po vienodai taškų, jų visų numeriai turi būti spausdinami vienoje eilutėje didėjimo tvarka.
- Prieš mažiausiai taškų surinkusį žaidėją, kurio numeris yra didžiausias, įterpkite žaidėją, kuris surinko daugiausiai taškų ir kurio numeris yra mažiausias.
- Prieš daugiausiai taškų surinkusį žaidėją, kurio numeris yra mažiausias, įterpkite žaidėją, surinkusį mažiausiai taškų, kurio numeris yra didžiausias.
- Įterpę aukščiau nurodytus elementus, rezultatų faile išspausdinkite žaidėjų numerius ir surinktų taškų skaičių – vienam žaidėjui skiriama viena eilutė.

<i>zaidimas1.txt</i>	<i>zaidimas1rez.txt</i>
16	Žaidėjai ir jų surinkti taškai:
1 5	1 6
2 4	2 5
3 3	3 6
4 0	4 5
1 0	Mažiausiai taškų surinko:
2 -2	4 2
3 2	Daugiausiai taškų surinko:
4 1	1 3
1 1	Po įterpimo:
2 0	4 5
3 1	1 6
4 -1	2 5
1 0	3 6
2 1	1 6
3 0	4 5
4 4	

Reikalavimai programai

- Naudokite sveikųjų skaičių masyvus.
- Sukurkite pradinių duomenų skaitymo funkciją void.
- Sukurkite mažiausios ir didžiausios reikšmių paieškos funkcijas, grąžinančias apskaičiuotą reikšmę per funkcijos vardą.
- Sukurkite masyvo elemento įterpimo funkciją void.
- Sukurkite void funkciją, faile spausdinančią mažiausiai taškų surinkusius žaidėjus.
- Sukurkite void funkciją, faile spausdinančią daugiausiai taškų surinkusius žaidėjus.
- Sukurkite void funkciją, faile spausdinančią žaidėjų numerius ir surinktų taškų skaičių.

4. Geriausių knygų rinkimai

Leidyklos „Alma Littera“ n ($n < 21$) knygų vertina m ($m < 11$) skaitytojų. Kiekvienas skaitytojas kiekvieną knygą vertina balais nuo 0 iki 10. Vertinimai yra sveikieji skaičiai. Pradinių duomenų failo *knygos1.txt* pirmoje eilutėje įrašyti skaičiai n ir m . Tolesnėse neilučių įrašytas knygos kodas k (sveikasis skaičius iš intervalo $[1; 20]$) ir po m sveikųjų skaičių, vienas nuo kito atskirtų tarpais – kiekvieno vertintojo knygai skirti balai. Skaičiuojant galutinį knygos įvertinimą, **sumuojami didžiausias ir mažiausias įvertinimai** (jei didžiausių ir mažiausių įvertinimų yra daugiau, negu po vieną, tuomet sumuojama tik po vieną didžiausią ir mažiausią įvertinimą). Apskaičiuota suma yra galutinis knygos įvertinimas. Parašykite programą, surandančią ir rezultatų failę *knygos1rez.txt* spausdinančią:

- Koks yra galutinis kiekvienos knygos įvertinimas. Rezultatus surašykite faile vienai knygai skirdami vieną eilutę: knygos kodas ir įvertinimas, atskirti tarpais. Sąrašas išdėstytas knygų kodų didėjimo tvarka
- Kuri knyga įvertinta blogiausiai. Jei kelios knygos įvertintos blogiausiai, jų visų kodai turi būti spausdinami vienoje eilutėje mažėjimo tvarka, atskiriant juos vieną nuo kito tarpais.
- Kuri knyga įvertinta geriausiai. Jei kelios knygos įvertintos geriausiai, jų visų numeriai turi būti spausdinami vienoje eilutėje didėjimo tvarka.
- Sukeiskite geriausiai įvertintą knygą su blogiausiai įvertinta knyga sąrašė vietomis. Jei yra ne po vieną geriausiai/blogiausiai įvertintą knygą, tuomet vietomis turi būti sukeičiamos geriausia ir blogiausia knygos, kurių kodai yra mažiausi.
- Rezultatų faile išspausdinkite naują knygų sąrašą, kuriame knygos sukeistos vietomis: kodus ir įvertinimus – vienai knygai skiriama viena eilutė.

<i>knygos1.txt</i>	<i>knygos1rez.txt</i>
5 6 2 2 5 6 6 5 2 1 3 8 8 7 7 7 4 4 5 5 7 7 6 3 5 5 5 5 3 3 5 7 8 8 7 2 6	Knygų įvertinimai: 1 11 2 8 3 8 4 11 5 10 Blogiausiai įvertintos knygos: 3 2 Geriausiai įvertintos knygos: 1 4 Naujas sąrašas: 2 8 1 11 3 8 4 11 5 10

Reikalavimai programai

- Naudokite sveikųjų skaičių masyvus.
- Sukurkite pradinių duomenų skaitymo funkciją void.
- Sukurkite mažiausios ir didžiausios reikšmių paieškos funkcijas, grąžinančias apskaičiuotą reikšmę per funkcijos vardą.
- Sukurkite masyvo elementų sukeitimo funkciją void.
- Sukurkite void funkciją, faile spausdinančią blogiausiai įvertintas knygas.
- Sukurkite void funkciją, faile spausdinančią geriausiai įvertintas knygas.
- Sukurkite void funkciją, faile spausdinančią knygų kodus ir įvertinimą.

12. ĮVAIRŪS ANTRO LYGIO UŽDAVINIAI

1. Dėlionės

Fizikos, matematikos ir informatikos olimpiadų dalyviai išsprendę uždavinius laisvalaikiu žaidė „Galvosūkių pasaulio“ loginius žaidimus. Vieni iš populiariausių žaidimų buvo dėlionės, kurias sudaro 4 rūšių skirtingi paveikslėliai. Pilnoje dėlionėje yra po 2 pirmos ir antros rūšies paveikslėlius ir po 4 trečios ir ketvirtos rūšies paveikslėlius.

Dėlionėms keliaujant iš vienos olimpiados į kitą dalis paveikslėlių pasimetė.

Informatikos olimpiada buvo paskutinė, todėl informatikai turėjo tenkintis dėlionėmis, kuriose trūko daugiausiai paveikslėlių.

„Galvosūkių pasaulis“ į informatikos olimpiadą atvežė n ($5 \leq n \leq 15$) dėlionių, tačiau kai kuriose iš jų trūksta paveikslėlių. Pradinių duomenų failo U1.txt pirmoje eilutėje įrašytas į olimpiadą atvežtų dėlionių skaičius n . Tolesnėse n eilučių įrašyta po 4 sveikuosius skaičius, vienas nuo kito atskirtus tarpais – kiekvienos rūšies paveikslėlių skaičius dėlionėje.

Rezultatų faile U1rez.txt reikia spausdinti:

- pirmoje eilutėje – kiek pilnų dėlionių galima sukomplektuoti iš turimų skirtingų rūšių paveikslėlių;
- tolesnėse keturiuose eilutėse – po kiek kiekvienos rūšies paveikslėlių liko nepanaudotų komplektuojant dėliones.

U1.txt	Paaiškinimai
5 1 2 1 3 2 0 2 2 2 2 3 3 2 2 2 4 2 2 4 4	Iš viso 5 dėlionės Pirmoje dėlionėje: pirmos rūšies 1, antros – 2, trečios – 1, ketvirtos – 3 paveikslėliai. Antroje dėlionėje: pirmos rūšies 2, antros – 0, trečios – 2, ketvirtos – 2 paveikslėliai. Trečioje dėlionėje: pirmos rūšies 2, antros – 2, trečios – 3, ketvirtos – 3 paveikslėliai. Ketvirtoje dėlionėje: pirmos rūšies 2, antros – 2, trečios – 2, ketvirtos – 4 paveikslėliai. Penktoje dėlionėje: pirmos rūšies 2, antros – 2, trečios – 4, ketvirtos – 4 paveikslėliai.
U1rez.txt	
3 3 2 0 4	Susidaro 3 dėlionių komplektai. Lieka nepanaudoti 3 pirmos rūšies paveikslėliai. Lieka nepanaudoti 2 antros rūšies paveikslėliai. Trečios rūšies paveikslėlių nelieka. Lieka nepanaudoti 4 ketvirtos rūšies paveikslėliai.

Reikalavimai programai

- turi būti naudojami sveikųjų skaičių masyvai;
- duomenų skaitymo iš failo funkcija `void`;
- funkcija, grąžinanti apskaičiuotą reikšmę per funkcijos vardą: skaičiuojanti, kiek dëlionių galima sukomplektuoti iš turimų skirtingų rûšių paveikslëlių;
- funkcija `void`, kuri į rezultatų failą įrašo **TIK** komplektuojant dëliones nepanaudotų paveikslëlių kiekius.

2. Šachmatų turnyras

Šachmatų turnyre „Panevëžys Open“ dalyvavo n ($n < 101$) šachmatininkų. Po turnyro Šiaulių miesto komandos treneris į failą *sachmatai2.txt* surašë kiekvieno sportininko rezultatus: prie kurios lentos žaidë sportininkas (sveikasis skaičius iš intervalo $[1; 10]$), kelias partijas sportininkas laimëjo, kelias baigë lygiosiomis ir kelias pralaimëjo. Už laimëtą partiją sportininkas gauna 1 tašką, už lygiasias – 0,5 taško, už pralaimëtą partiją taškai neskiriami. Vienam sportininkui skiriama viena eilutë. Parašykite programą, surandančią ir rezultatų failę *sachmatai2rez.txt* spausdinančią:

- Koks yra galutinis kiekvieno turnyre dalyvavusio sportininko surinktų taškų skaičius. Rezultatus surašykite failë vienam sportininkui skirdami vieną eilutę: lentos numeris ir sportininko surinktų taškų skaičius vieno ženklo po kablelio tikslumu, išdëstant sportininkus pagal tai, prie kurios lentos kuris žaidë.
- Kuris sportininkas surinko mažiausiai taškų. Jei keli sportininkai surinko po mažiausiai taškų, jų lentų numeriai turi būti spausdinami vienoje eilutėje didëjimo tvarka, atskiriant jas vieną nuo kitos tarpais.
- Kuris sportininkas surinko daugiausiai taškų. Jei keli sportininkai surinko po daugiausiai taškų, jų lentų numeriai turi būti spausdinami vienoje eilutėje mažëjimo tvarka, atskiriant jas vieną nuo kitos tarpais.
- Surikiuokite sportininkus nuo surinkusio daugiausiai taškų iki surinkusio mažiausiai taškų. Jei surinktų taškų sumos sutampa, tai tuomet dar rikiuokite lentos numerio mažëjimo tvarka.
- Rezultatų failë išspausdinkite surikiuotą sportininkų sąrašą: lentos numerį ir sportininko surinktų taškų skaičių vieno ženklo po kablelio tikslumu – vienam sportininkui skiriama viena eilutë.

<i>sachmatai2.txt</i>	<i>sachmatai2rez.txt</i>
5 3 1 1 4 2 3 2 1 4 4 0 2 1 2 2 2 5 0 3 3	Sportininkų rezultatai: 1 3.0 2 4.0 3 1.5 4 4.0 5 1.5 Mažiausiai taškų surinkę sportininkai: 3 5 Daugiausiai taškų surinkę sportininkai: 4 2 Surikiuotas sąrašas: 4 4.0 2 4.0 1 3.0 5 1.5 3 1.5

Reikalavimai programai

- Naudokite sveikųjų ir realiųjų skaičių masyvus.
- Sukurkite pradinių duomenų skaitymo funkciją void.
- Sukurkite mažiausios ir didžiausios reikšmių paieškos funkcijas, grąžinančias apskaičiuotą reikšmę per funkcijos vardą.
- Sukurkite masyvo elementų rikiavimo funkciją void.
- Sukurkite void funkciją, faile spausdinančią mažiausiai taškų surinkusius sportininkus.
- Sukurkite void funkciją, faile spausdinančią daugiausiai taškų surinkusius sportininkus.
- Sukurkite void funkciją, faile spausdinančią sportininkų lentų numerius ir surinktų taškų skaičių.

3. Ekologiški griekiai

Ūkininkas prekiauja ekologiškais griekiais, supakuotais po vieną ir po du kilogramus. Kiekvienas pirkėjas perka ne mažiau kaip po 1 kg ir ne daugiau kaip 50 kg griekių. Ūkininkas iš pradžių nori parduoti didesnes pakuotes. Pavyzdžiui, jei pirkėjas perka 5 kg griekių, tai ūkininkas jam duoda 2 pakuotes po 2 kg ir 1 vieno kilogramo pakuotę. Kai 2 kg pakuotės baigiasi, ūkininkas pardavinėja 1 kg pakuotes.

Parenkite programą, kuri apskaičiuotų:

- kiek 1 kg ir kiek 2 kg griekių pakuočių pardavė ūkininkas;
- keli pirkėjai nusipirko bent po vieną griekių pakuotę;
- kiek kilogramų griekių nusipirko paskutinis pirkėjas, gavęs bent vieną griekių pakuotę.

Pradinių duomenų failo *griekiai.txt* pirmoje eilutėje nurodyta, kiek griekių pakuočių atvežė ūkininkas: n_1 ($1 \leq n_1 \leq 100$) – 1 kg ir n_2 ($1 \leq n_2 \leq 100$) – 2 kg. Antroje failo eilutėje įrašytas pirkėjų skaičius n ($1 \leq n \leq 100$). Tolesnėse n eilučių nurodyta, kiek kg griekių pageidauja pirkti kiekvienas pirkėjas.

Rezultatų failo *griekiairez.txt*:

- pirmoje eilutėje nurodykite, kiek 1 kg ir kiek 2 kg pakuočių nupirkta (jei kurios nors rūšies pakuočių nėra parduota, rašykite nulį);
- antroje failo eilutėje pateikite, kelis kilogramus griekių nusipirko paskutinis pirkėjas, dar gavęs bent vieną griekių pakuotę.

<i>griekiai.txt</i>	<i>griekiairez.txt</i>
1 2	1 1
2	2
1	2
3	

Reikalavimai programai

- Pirkėjų pageidavimams atmintinėje laikyti naudokite vienmatį sveikųjų skaičių masyvą.
- Sukurkite funkciją void, kuri apskaičiuotų, kiek pirkėjų nusipirko bent vieną griekių pakuotę ir kiek kilogramų griekių nusipirko paskutinis pirkėjas, gavęs dar bent vieną pakuotę.

13. SIMBOLIAI. SIMBOLIŲ MASYVAI

1. El. pašto dėžutės pavadinimas

Faile *email.txt* pateikti duomenys: minimalus pašto dėžutės pavadinimo ilgis, minimalus domeno ilgis, el. pašto adresas. Reikia parašyti programą, kuri patikrintų, ar el. pašto adresas yra teisingas pagal tokius kriterijus:

1. Adresą sudaro tik lotyniškos raidės, '@' ir '.' simboliai.
2. Adreso pašto dėžutės pavadinimas turi būti netrumpesnis nei nurodyta faile.
3. Adreso domeno ilgis turi būti netrumpesnis nei nurodyta faile.

Patikrinimo rezultatus išveskite į ekraną.

Pradiniai duomenys	Rezultatai
5 3 informikas@gmail.com	Email teisingas.
5 2 informikas@one.lt	Email neteisingas. Per trumpas pašto dėžutės pavadinimas.
2 3 informikas@one.lt	Email neteisingas. Per trumpas domeno pavadinimas.
4 2 info\$#ikas@yahoo.com	Email neteisingas. Yra neleidžiamų simbolių.

Reikalavimai programai

- Naudoti simbolių masyvus.
- Duomenų skaitymo funkcija.
- Tikrinimo funkcija.

2. Slaptas bendravimas

Du draugai, Antanas ir Petras, susirašinėja slaptomis žinutėmis. Antanas užkoduoja ir perduoda Petrai pirminį šifravimo raktą ir užšifruotą žinutę, kurią sudaro tik mažosios lotyniškos raidės. Petras turi visas užšifruotos žinutės raides perstumti atgal per tiek pozicijų kiek nurodo šifravimo raktas, tada peržiūrėti naujai gautą žinutę ir rasti pirmąją 'x' raidę, kuri nurodo, kad baigėsi žodis, taip buvo iššifruotas vienas žodis. Rastą 'x', jis pasikeičia į tarpelį, suskaičiuoja kiek gautas žodis turi raidžių – tai naujas šifravimo raktas, ir kartoja šifravimą nuo raidės sekančios už rastos 'x' raidės tokiu pačiu principu, kartoja tiek kartų kol neberanda žinutėje nei vienos 'x' raidės, tada žinutė būna iššifruota pilnai.

Sudarykite programą, kuri padėtų Petrai iššifruoti žinutę.

Duomenų faile *Duomenys1.txt* pirmoje eilutėje įrašytas pirminis raktas, o kitoje failo eilutėje užšifruotos žinutės tekstas (tik mažosios lotyniškos raidės, neatskirtos tarpais).

Rezultatų faile *Rezultatai1.txt* įrašykite pilnai iššifruotos žinutės tekstą.

Duomenys1.txt	Rezultatai1.txt
1 tmbqubygpubal	Iššifruota žinutė: slapta zinute
3 vdxqxrolvaueeurdmhmujpaeojcwe	Iššifruota žinutė: saunuolis issifravai teisingai

Reikalavimai programai

- Duomenims saugoti naudokite simbolių masyvą.
- Sukurkite funkciją, kuri atliktų simbolių masyvo reikiamų elementų perstūmimą atgal per tiek pozicijų, kiek nurodo šifravimo raktas.

3. Šifravimas

Gimnazistė Urtė tikriausiai pamirštų, kur pasidėjusi savo galvą, jei ji nebūtų saugiai pritvirtinta, tačiau būdama klasės seniūne, ji privalo elgtis atsakingai ir pareiškinti. Štai dėl ko Urtė paprašė savo klasioko Deivido, kuris yra susižavėjęs programavimu, pagalbos, kad užšifruotų visus jos gaunamus elektroninius laiškus, tam, kad pamiršus atsijungti nuo el. pašto dėžutės jai nereikėtų bijoti, jog kažkas paskleis ar pasinaudos jos turima informacija. Deividas sumąstė faile *Sifras.txt* išsaugoti lotyniškos abėcėlės didžiąsias raides ir kiekvienai jų priskirti skirtingus simbolius. Jei savaitės diena yra pirmadienis, Urtės gautų žinučių simboliai yra tiesiog sukeičiami su juos atitinkančiais simboliais faile *Sifras.txt*, jei savaitės diena yra ne pirmadienis - žinučių simboliai yra sukeičiami su juos atitinkančiais simboliais faile *Sifras.txt*, bet perstumtais per tiek pozicijų koks yra tos savaitės dienos faktorialas.

Duomenys: Failo *Zinutes.txt* pirmoje eilutėje yra pateiktas žinučių skaičius ($0 < n < 10$), kitoje eilutėje yra pateikiami duomenys apie pirmąją žinutę: vienoje eilutėje yra nurodyta savaitės dienos numeris ir siuntėjo vardas, kitoje - žinutės tekstas). Visa informacija pateikiama didžiosiomis lotyniškomis raidėmis. Faile *Sifras.txt* yra saugomas 25 lotyniškų simbolių šifras: vienoje eilutėje yra nurodytas vienas lotyniškas simbolis ir jo šifras. Faile *Rezultatai2.txt* turi būti nurodytas kiekvienos žinutės siuntėjas, siuntimo diena (žodžiais), užšifruota ir iššifruota žinutės.

<i>Zinute.txt</i>
2 1 IGNAS LABAS NORIU PRANESTI JOG KLASIU SENIUNU SUSITIKIMAS ATIDETAS 3 LAURA SVEIKI SALE YRA PARUOSTA RENGINIUI LAUKIAM VISU NORINCIU SUDALYVAUTI SVENTEJE
<i>Sifras.txt</i>
A ~ B ` C ! D @ E #

F \$
 Y %
 G ^
 H &
 J *
 K (
 L)
 M _
 N -
 O =
 P +
 R [
 S]
 T {
 I }
 U \
 V |
 W :
 Q >
 Z <

Rezultatai1.txt

 Siuntejas: IGNAS

Gauta: Pirmadienis

Sifruota zinute:)~`~] -=[] +[~-#]{} *=-^ ()~}]#-]-]}]{}({}_~] ~-[]@#{~]

Issifruota zinute: LABAS NORIU PRANESTI JOG KLASIU SENIUNU SUSITIKIMAS
 ATIDETAS

 Siuntejas: LAURA

Gauta: Treciadienis

Sifruota zinute: c,)f.f c,/) +a,, 1,,abCc□ ,, a)3df3fbf /,,b.f,,e ,fcb 3Caf3'fb cbF,,/+,,,b□ f
 c,)3□)0)

Issifruota zinute: SVEIKI SALE YRA PARUOSTA RENGINIUI LAUKIAM VISU
 NORINCIU SUDALYVAUTI SVENTEJE

Reikalavimai programai

- Funkcija skaitanti duomenis iš failo *Zinutes.txt*.
- Funkcija skaitanti duomenis iš failo *Sifras.txt*.
- Faktorialo skaičiavimo funkcija.
- Šifravimo funkcija.
- Iššifravimo funkcija.
- Duomenų spausdinimo funkcija.
- Duomenys apie kiekvieną žinutę (siuntėjo vardas, siuntimo diena, žinutės tekstas) saugomi struktūrų masyve.

- Duomenys apie šifrą (lotyniški simboliai, jų šifrai) saugomi struktūroje.
- Įvykdžius šifravimą, žinutė, kuri buvo šifruojama, turi būti pakeista šifruota žinute.

4. Sutrumpinimai

Tekste yra naudojami sutrumpinimai. Jie užrašyti laužtiniuose skliaustuose didžiosiomis raidėmis. Vietoje sutrumpinimų reikia įterpti pilną tekstą ir rezultatą išspausdinti rezultatų faile *Rezultatai3.txt*. Sutrumpinime negalima naudoti kitų simbolių, išskyrus didžiąsias raides. Maksimalus sutrumpinimo ilgis – 10 raidžių.

Pirmoje pradinių duomenų failo *Duomenys3.txt* eilutėje užrašyta, kiek sutrumpinimų iš viso yra naudojama. Kitose eilutėse surašyti sutrumpinimai ir tarpu atskirta pilna sutrumpinimo versija. Po sutrumpinimais parašytas tekstas, kuriame tie sutrumpinimai naudojami.

<i>Duomenys3.txt</i>
2 VARDAS Petras Petrauskas DATA 2015 01 19 Sveiki, Mano vardas yra [VARDAS]. Sita programa parasiau [DATA].
<i>Rezultatai3.txt</i>
Sveiki, Mano vardas yra Petras Petrauskas. Sita programa parasiau 2015 01 19.

Reikalavimai programai

- Tekstui saugoti naudokite simbolių masyvą (char[]).
- Parašykite funkciją, kuri patikrintų, ar duotas masyvo intervalas atitinka sutrumpinimo sąlygą.
- Parašykite funkciją, kuri sukurtų naują masyvą su sutrumpinimų vietoje įterptu pilnu tekstu.

5. SMS šifras

Informatikai Paskalis ir Šarpis kiekvieną dieną tarpusavyje susirašinėja SMS žinutėmis. Paskaliui sugedo jo išmanusis telefonas, todėl šiuo metu jis naudojami telefonu su mygtukine klaviatūra.

Pamatęs šį telefoną, Šarpis sugalvojo genialų būdą, kaip užšifruoti siunčiamas SMS žinutes, kad Paskalis, naudodamasis savo telefonu, galėtų nesunkiai jas iššifruoti. Žinutės šifruojamos taip:

- Kiekvieną lotynišką raidę atitinka skaičių seka, kuri nurodo mygtuko paspaudimų skaičių, reikalingą raidei įvesti, pvz., raidę „a“ atitinka 2, raidę „s“ – 7777 ir t.t.;
- Po kiekvienos raidės į skaičių seką įterpiamas tarpas (0);
- Po kiekvieno žodžio į skaičių seką įterpiami du tarpai (00);
- Užšifruota žinutė baigiasi, kai skaičių sekos gale yra 1 – taško ženklas.

Tačiau Paskalis yra didelis tinginys ir labai nenoriai iššifruoja Šarpio siųstas žinutes. Parašykite programą, kuri padėtų Paskaliui iššifruoti skaičių seką. Pradinių duomenų failo *Duomenys4.txt* pirmoje eilutėje įrašytas sveikasis skaičius n ($0 < n < 1000$) – skaičių sekoje esančių skaitmenų kiekis. Tolesnėse eilutėse yra nurodytas užšifruotas tekstas – skaičių seka.

Rezultatų failo „Rezultatai4.txt” pirmoje eilutėje išvedamas iššifruotos skaičių sekos tekstas, pasibaigiantis tašku. Kiekvienas žinutės žodis atskirtas vienu tarpo simboliu, visos raidės mažosios, lotyniškos.

Duomenys4.txt	Rezultatai4.txt
1 1	.
55 4440660333066607770604440550 666002055020303306044405021	informiko akademija.
98 7020777705502055504440666006 0330409999080444066044407777 0055502022020444003044403033 05550444077771	paskalio megztinis labai didelis.

Reikalavimai programai

- Pats iššifravimo raktas turi būti saugomas struktūros duomenų tipe, pvz., struktūroje saugomas dvimatis masyvas, kurio atskirose eilutėse yra saugomi kiekvieno mygtuko simboliai.
- Funkcija duomenų skaitymui iš failo į char duomenų tipo masyvą.
- Duomenų skaitymo funkcija negali išsaugoti n (sekos skaitmenų kiekio) reikšmės.
- Funkcija, iššifruojanti skaičių seką. Funkcijoje, kuri iššifruoja skaičių seką, galite panaudoti tik while ciklus, for ciklo naudoti negalima (duomenų skaityme ir rezultatų įrašyme for ciklus naudoti galima).
- Rezultatų įrašymo į rezultatų failą funkcija.

14. SIMBOLIŲ EILUTĖS. SIMBOLIŲ EILUČIŲ MASYVAI

1. Nuotaikos detektorius

Paskalis internete perskaitė straipsnį, kuriame rašoma, jog žmogaus nuotaiką galima nustatyti iš to, kokius žodžius jis vartoja kalbėdamas. Žmogaus nuotaika iš jo vartojamų žodžių atpažįstama taip:

- Jei žmogus vartoja daug „pakilių“ žodžių, jo nuotaika yra linksma;
 - Jei žmogus savo kalboje vartoja daug palindromų, jo nuotaika yra liūdna;
 - Visi kiti žodžiai laikomi neutraliais ir teigiama, kad jie įtakos nuotaikai neturi;
 - Žmogaus nuotaika yra tokia, kokių žodžių jis vartoja daugiau („pakilių“ ar palindromų);
 - Jei žmogus vartoja po lygiai „pakilių“ žodžių ir palindromų, jo nuotaika yra neutrali.
- „Pakilus“ žodis – tai toks žodis, kurio kiekviena raidė abėcėliškai yra toliau už prieš tai buvusią, pvz., „pakilus“ žodis yra *ačiū* (raidė a yra pirmoji abėcėlėje, č – 5, i – 12, ū – 29) arba *ABC* (A – 1, B – 3, C – 4) ir pan.

Žodis palindromas – tai toks žodis, kuris ir iš vieno, ir iš kito žodžio galo yra skaitomas vienodai, yra simetriškas raidžių išsidėstymo atžvilgiu, pvz., *SOS*, *Apapa*, *AKA*, *DPD*, *AA* ir pan.

Padėkite Paskaliui parašyti programą, kuri nustatytų žmogaus nuotaiką iš jo vartojamų žodžių.

Pradinių duomenų failo „Duomenys.txt“ pirmoje failo eilutė įrašytas sveikasis skaičius n ($0 < n < 1000$) – užfiksuotas žmogaus vartojamų žodžių skaičius. Tolimesnėse n eilučių yra pateikiami žodžiai (ne ilgesni nei 100 simbolių ir turintys bent vieną raidę), po vieną kiekvienoje eilutėje.

Rezultatų failo „Rezultatai.txt“ pirmoje eilutėje išvedama žmogaus nuotaika:

- Išvedamas simbolis „:|“, kai žmogaus nuotaika linksma;
- Išvedamas simbolis „:(“, kai žmogaus nuotaika liūdna;
- Išvedamas simbolis „:|“, kai žmogaus nuotaika yra neutrali.

Pastabos:

- Nuotaikos nustatymui naudojama lietuviška 32 raidžių abėcėlė;
- Vienos raidės žodis, pvz., jaustukas *O*, laikomas ir „pakiliu“ žodžiu, ir palindromu;
- Didžiosios ir mažosios raidės vertinamos vienodai, t. y., jų vieta abėcėlėje yra ta pati.

Duomenys.txt	Rezultatai.txt	Paaiškinimas
1 K	:	„Pakilių“ žodžių: 1 Palindromų: 1 Raidė yra ir „pakilus“ žodis, ir palindromas, todėl nuotaika yra neutrali.
5 Ačiū uŽ JūSŪ KanTRYBę M	:)	„Pakilių“ žodžių: 3 Palindromų: 1 „Pakilūs“: Ačiū, uŽ, M Palindromai: M Nuotaika: linksma
7 mAn šakės SOS HELPLEH eI Kur Jūs	:(„Pakilių“ žodžių: 1 Palindromų: 2 „Pakilūs“: eI Palindromai: SOS, HELPLEH Nuotaika: liūdna

Reikalavimai programai

- Struktūros duomenų tipe saugomi 3 kintamieji: visų žodžių masyvas, žodžių skaičius ir string eilutė, kurioje saugoma lietuviška abėcėlė;
- Funkcija duomenų skaitymui;
- Funkcija rezultatų įrašymui;
- Funkcija, kuri per savo vardą grąžina TRUE, jei žodis yra „pakilus“;
- Funkcija, kuri per savo vardą grąžina TRUE, jei žodis yra palindromas;
- Funkcija, kuri skaičiuoja „pakilių“ bei žodžių palindromų skaičių.

2. Teksto analizė

Petriukas labai mėgsta analizuoti tekstus, perskaitęs tekstą jis visada susirašo kiek kokių žodžių jame buvo. Bet daryti tai rankomis labai varginantis ir ilgas darbas, dėl to padėkite Petriukui ir parašykite programą, kuri tai darytų už jį.

Duomenų faile Duomenys.txt yra tik viena eilutė, kurioje yra tekstas sudarytas tik iš didžiųjų raidžių ir simbolių ".,;:-!?)", kurie skiria žodžius (tarp žodžių gali būti ir daugiau nei vienas skyriklis).

Rezultatų faile Rezultatai.txt išspausdinkite, kiek iš viso yra skirtingų žodžių ir visus juos, bei kiek kartų kiekvienas pasikartoja tekste.

Duomenys.txt	Rezultatai.txt
<p>PETRIUKAS LABAI MĖGSTA ANALIZUOTI TEKSTUS, PERSKAITĖS TEKSTĄ JIS VISADA SUSIRAŠO KIEK KOKIŲ ŽODŽIŲ JAME BUVO. BET DARYTI TAI RANKOMIS LABAI VARGINANTIS IR ILGAS DARBAS, DĖL TO PADĖKITE PETRIUKUI IR PARAŠYKITE PROGRAMĄ, KURI TAI DARYTŲ UŽ JĮ.</p>	<p>Iš viso skirtingų žodžių 33</p> <ol style="list-style-type: none"> 1. PETRIUKAS 1 2. LABAI 2 3. MĖGSTA 1 4. ANALIZUOTI 1 5. TEKSTUS 1 6. PERSKAITĖS 1 7. TEKSTĄ 1 8. JIS 1 9. VISADA 1 10. SUSIRAŠO 1 11. KIEK 1 12. KOKIŲ 1 13. ŽODŽIŲ 1 14. JAME 1 15. BUVO 1 16. BET 1 17. DARYTI 1 18. TAI 2 19. RANKOMIS 1 20. VARGINANTIS 1 21. IR 2 22. ILGAS 1 23. DARBAS 1 24. DĖL 1 25. TO 1 26. PADĖKITE 1 27. PETRIUKUI 1 28. PARAŠYKITE 1 29. PROGRAMĄ 1 30. KURI 1 31. DARYTŲ 1 32. UŽ 1 33. JĮ 1
<p>PETRIUKAS LABAI MĖGSTA ANALIZUOTI TEKSTUS, PERSKAITĖS TEKSTĄ JIS VISADA SUSIRAŠO KIEK KOKIŲ ŽODŽIŲ JAME BUVO. BET DARYTI TAI RANKOMIS LABAI VARGINANTIS IR ILGAS DARBAS, DĖL TO PADĖKITE PETRIUKUI IR PARAŠYKITE PROGRAMĄ, KURI TAI DARYTŲ UŽ JĮ. O PETRIUKAS YRA LABAI LABAI DRAUGIŠKAS VISIEMS IR VISADA LABAI STENGIASI PADĖTI, NORS KARTAIS PADĖTI NEIŠEINA, O IŠEINA TIK DĖTI. KARTAIS PETRIUKAS SKAITO</p>	<p>Iš viso skirtingų žodžių 52</p> <ol style="list-style-type: none"> 1. PETRIUKAS 3 2. LABAI 7 3. MĖGSTA 1 4. ANALIZUOTI 1 5. TEKSTUS 2

Duomenys.txt	Rezultatai.txt
<p>LABAI LABAI ILGUS TEKSTUS, DĖL TO JAM REIKIA, KAD JŪS JAM PADĖTUMĖTE!</p>	<p>6. PERSKAITĖS 1 7. TEKSTĄ 1 8. JIS 1 9. VISADA 2 10. SUSIRAŠO 1 11. KIEK 1 12. KOKIŲ 1 13. ŽODŽIŲ 1 14. JAME 1 15. BUVO 1 16. BET 1 17. DARYTI 1 18. TAI 2 19. RANKOMIS 1 20. VARGINANTIS 1 21. IR 3 22. ILGAS 1 23. DARBAS 1 24. DĖL 2 25. TO 2 26. PADĖKITE 1 27. PETRIUKUI 1 28. PARAŠYKITE 1 29. PROGRAMĄ 1 30. KURI 1 31. DARYTŲ 1 32. UŽ 1 33. JĮ 1 34. O 2 35. YRA 1 36. DRAUGIŠKAS 1 37. VISIEMS 1 38. STENGIASI 1 39. PADĖTI 2 40. NORS 1 41. KARTAIS 2 42. NEIŠEINA 1 43. IŠEINA 1 44. TIK 1 45. DĖTI 1 46. SKAITO 1 47. ILGUS 1 48. JAM 2 49. REIKIA 1 50. KAD 1 51. JŪS 1 52. PADĖTUMĖTE 1</p>

Reikalavimai programai

- Parašykite skaitymo funkciją, kuri perskaito visą tekstinio failo eilutę į simbolių eilutę (string).
- Žodžiams ir jų pasikartojimams saugoti naudokite struktūrą.
- Parašykite funkciją, kuri patikrintų ar žodis jau yra žodžių (ir jų pasikartojimų) struktūrų masyve.

- Parašykite funkciją, kuri nustatytų kiek kartų nurodytas žodis kartojasi tekste.
- Naudokite string bibliotekos funkcijas, tokias kaip find, find_first_of ir kitas.

3. Kryžiažodis

Petriukas mėgsta spręsti kryžiažodžius, tačiau ne įprastus, o „surask žodį“ tipo. Deja, jam ne visada pavyksta išspręsti, tad jis nori pasitikrinti sprendimą.

Parašykite programą, kuri išspręstų kryžiažodį ir išvestų sprendimą į rezultatų failą Rezultatai.txt.

Kryžiažodžio pvz:

```
P D B I Q V S S
D R N I K J U T
G F O F I I A A
O U Š G C F I L
W V U I R G S A
X D A U Q A X S
C K Z S O K M K
S L I E T U V A
```

Kryžiažodis sprendžiamas ieškant žodžių horizontaliai, vertikaliam ir įstrižai ir juos išbraukiant.

Žodžiai gali būti įrašyti iš dešinės į kairę, pvz 4 eilutėje žodis ŠUO.

Kryžiažodžius galite patys susigeneruoti čia:

<http://puzzlemaker.discoveryeducation.com/WordSearchSetupForm.asp>

Pradiniai duomenys:

Kryziazodis.txt pirmoje eilutėje nurodytas kryžiažodžio dydis n, o tolimesnėse eilutėse pats kryžiažodis.

- Zodziai.txt pirmoje eilutėje nurodytas žodžių skaičius k, o tolimesnėse eilutėse surašyti žodžiai.

Duomenys	Rezultatai
Kryziazodis.txt	Rezultatai.txt
10 SUIROTKEVA EASHEBLDKS AUKWBSGIFI HMRIAITOPP KWAPMATNHA EGMRMRNECL GOQEGNOUMS KMTXKOQFCU BABPRCRRNP MKATERDPXI	SUIROTKEVA +A+++++KS A+K+BS+I+I +M+IAIT++P ++APMAT++A ++MRMR+E+L +O+EG+O++S K+T++O+F+U +A++++R+NP MKATE++P+I
Zodziai.txt	
8 BITE INFORMIKAS KATE KOMPAS	

Duomenys	Rezultatai
MATEMATIKA PROGRAMA PUSLAPIS VEKTORIUS	
Duomenys	Rezultatai
Kryziazodis.txt	Rezultatas.txt
8 T I L O W R M C F C I Z F L W P G M E F R N R O N P T M Q O I Z G L U D G W Z O Q S V R U E B U S K A I C I U S N M S A L A T S	++L O++++ ++I+F++P ++E++NR+ ++T++OI+ ++U+G++O ++VR+++U S K A I C I U S +M S A L A T S
Zodziai.txt	
6 INFO LIETUVA PROGRAM SKAICIUS STALAS SUO	

Reikalavimai programai

- Skaitymo, spausdinimo ir sprendimo funkcijos.
- Sprendžiant uždavinį panaudoti string duomenų tipą.

4. Formatavimas

Petriukas pastebėjo, kad kai kurie jo klasės draugai nesugeba suformatuoti savo kodo informatikos pamokoje, todėl jis nusprendė padėti jiems sukurdamas programą, kuri padėtų jiems programuoti tvarkingai.

Pradiniai duomenys:

- Duom.txt – neformatuotas kodas

Rekomenduotina naudoti šiuos std::string metodus:

- find()
<http://www.cplusplus.com/reference/string/string/find/>
- find_first_of()
http://www.cplusplus.com/reference/string/string/find_first_of/
- find_first_not_of()
http://www.cplusplus.com/reference/string/string/find_first_not_of/

- insert()
<http://www.cplusplus.com/reference/string/string/insert/>

Reikalavimai programai:

- Nuskaityti ir suformatuoti kodą pagal reikalavimus:
- Prieš kiekvieną kodo eilutę turi būti palikta $x*4$ tarpų (x – priklauso nuo $\{, \{, ir \}$ skliaustelių kiekio kode)
- Suformatuotą kodą išspausdinti rezultatų faile.
- Simboliai $\{, \{, ir \}$ turi būti atskirose eilutėse, pvz:

```
for(...){
    ... }
Arba for(...){ ... }
Pakeisti i:
for(...)
{
    ...
}
```

(į ciklus ir if-us be skliaustelių neatsižvelgti)

Pradiniai duomenys

```
//-----
----
/* Nuskaityti duomenis iš duomenų failo @CDuom*/
void SkaitytiDuomenis(){
ifstream fi(CDuom);
if (!fi.is_open()) { return; }
int nA;
fi >> nA;
N.detiAukstus(nA);
int nr, pl, sk, telnr;
float kaina;
for (int i = 0; i < nA; i++){
Aukstas aTemp;
for (int j = 0; j < 3; j++){
{fi >> nr >> pl >> sk >> kaina >> telnr;
Butas temp = Butas(nr, pl, sk, telnr, kaina);
aTemp.detiButa(temp, j);}
N.detiAuksta(aTemp, i);
}
fi.close();
}
```

Rezultatai

```
//-----
/* Nuskaityti duomenis iš duomenų failo @CDuom*/
void SkaitytiDuomenis()
{
ifstream fi(CDuom);
if (!fi.is_open())
```

Rezultatai

```
{
    return;
}
int nA;
fi >> nA;
N.detiAukstus(nA);
int nr, pl, sk, telnr;
float kaina;
for (int i = 0; i < nA; i++)
{
    Aukstas aTemp;
    for (int j = 0; j < 3; j++)
    {
        fi >> nr >> pl >> sk >> kaina >> telnr;
        Butas temp = Butas(nr, pl, sk, telnr, kaina);
        aTemp.detiButa(temp, j);
    }
    N.detiAuksta(aTemp, i);
}
fi.close();
}
```

Reikalavimai programai

- Prieš kiekvieną kodo eilutę turi būti palikta $x*4$ tarpų (x – priklauso nuo „{, ir „}“ skliaustelių kiekio kode)
- Simboliai „{, ir „}“ turi būti atskirose eilutėse
- Suformatuotą kodą išspausdinti rezultatų faile.

5. Namų darbai

Martynas, kaip ir daugelis mokinių, amžinai pasilieka pernelyg daug darbų paskutiniam vakarui prieš atsiskaitymus. Šis vakaras taip pat ne išimtis, o darbų ant nosies kabo tiek, kad nežinia ko imtis.

Išgelbėtumėte Martyną, padėdami jam išspręsti jo informatikos namų darbų užduotį ir taip sutaupydami jo sparčiai senkanti laiką. Uždavinys skamba taip: n žaidėjų sustoja į eilę, kiekvienas žaidėjas rankose laiko kortelę, kurioje yra užrašyta viena raidė ir du skaičiai, reiškiantys kitų dviejų žaidėjų pozicijas eilėje. Žaidėjui atvertus kortelę, joje įrašyta raidė ir žaidėjo vardas yra užrašomi ant lentos. Žaidėjai korteles privalo atversti pagal tokias taisykles: pirmasis kortelę atverčia tas, kurio antras skaičius kortelėje yra lygus 0, o pozicijos numeris eilėje yra mažiausias. Toliau kortelę atverčia tas žaidėjas, kurio pozicija eilėje sutampa su vienu iš atverstos kortelės skaitmenų. Jei kortelėje pirmasis skaitmuo nurodo žaidėją, kuris jau atvertęs savo kortelę, kortelę atverčia žaidėjas, kurį nurodo antrasis skaitmuo, tačiau jei ir jis yra atvertęs savo rankose laikomą kortelę, tuomet privalome sugrįžti per viena „žingsnį“ atgal t.y. grįžtame pas tą žaidėją, kurio kortelės skaičius nurodė esamo žaidėjo poziciją. Visa tai kartojame tol, kol nelieka neatverstų kortelių. Nulinė pozicija eilėje neegzistuoja, todėl bet kuris skaičiukas, nurodantis poziciją 0 yra ignoruojamas.

Pradinių duomenų failo „Žaidėjai.txt“ pirmoje failo eilutė įrašytas sveikasis skaičius n ($0 < n < 10$) – žaidėjų skaičius. Kiekvienoje sekančioje eilutėje yra įrašytas žaidėjo vardas, jo kortelėje esanti raidė ir du skaitmenys.

Rezultatų failo „Rezultatai.txt“ pirmose n eilučių yra surašyti žaidėjų vardai, tokia tvarka, kokia buvo atverčiamos kortelės. Paskutinėje eilutėje pateikiamas iš raidžių sudėliotas žodis.

Pastabos:

- Duomenų faile žaidėjai pateikti jų stovėjimo eilėje tvarka;
- Eilėje žaidėjai indeksuojami nuo 1.

Žaidėjai.txt	Rezultatai.txt
9	Linas
Ignas A 2 4	Ignas
Linas P 1 0	Kristina
Simas S 5 0	Jokūbas
Kristina S 1 6	Tautvydas
Paulius I 3 7	Andrėja
Jokūbas L 4 9	Karina
Karina T 5 8	Paulius
Andrėja P 7 9	Simas
Tautvydas A 6 8	Žodis: PASLAPTIS

Reikalavimai programai

- Duomenų skaitymo funkcija.
- Duomenų struktūra, kurioje yra saugomas žaidėjo vardas ir kortelėje esantys duomenys.
- Ieškomą žodį sudėliojanti funkcija.
- Rezultatų išvedimo funkcija.

15. STRUKTŪROS, STRUKTŪRŲ MASYVAI

1. Laiko planavimas

Petriukas nori geriau išmokti planuoti savo laiką, bei suprasti kiek laiko jam reikia skirti tam tikriems tikslams, dėl to jis pradėjo stebėti savo laiką: susiplanuoja užduotį, skiria jai laiko (kiek jam atrodo, kad užtruks atlikinėdamas tą užduotį) ir užsirašo kiek užtruko tai atlikti (jei užduoties nevykdė įrašo 0). Stebėti savo laiką Petriukui nėra sunku, tačiau jam sunkiai sekasi pačiam analizuoti savo surinktus duomenis.

Parašykite programą, kuri išanalizuotų Petriuko suvestus duomenis ir apskaičiuotų: 1) kiek jis turėjo užtrukti, 2) kiek užtruko, 3) kiek vidutiniškai skyrė laiko užduotims (neskaičiuoti nevykdytų užduočių).

Duomenų faile “Duomenys1.txt” pateikta sąrašas tikslų: užduoties pavadinimas iki simbolio ‘,’ , planuotas laikas atlikti užduotį, užtruktas laikas atlikti užduotį.

Rezultatų faile “Rezultatai1.txt” turi būti pateikta kiek buvo suplanuota užtrukti, kiek užtruko, kiek užduočių vykdyta ir kiek joms vidutiniškai laiko skirta.

Duomenys1.txt	Rezultatai1.txt
Skaityti knygą, 60 30 Išplauti grindis, 10 0 Pasimokyti programuoti, 110 150 Sportuoti, 60 130 Atlikti namų darbus, 120 50 Padėti Jonukui, 60 73 Pavedžioti šuniuką, 20 0	Iš viso Petriukas buvo suplanavęs užtrukti 440 minutes Iš viso Petriukas užtrukto 433 minutes Petriukas vykdė 5 užduotis, joms vidutiniškai skyrė po 86.6 minutes
Skaityti knygą, 60 30 Išplauti grindis, 10 0 Pasimokyti programuoti, 110 150 Sportuoti, 60 70 Atlikti namų darbus, 120 50 Padėti Jonukui, 60 73 Pavedžioti šuniuką, 20 0 Pažaisti lauke, 120 180 Aplankyti močiutę, 120 150 Pažaisti su šuniuku, 60 30 Padėti mamai, 20 40	Iš viso Petriukas buvo suplanavęs užtrukti 760 minutes Iš viso Petriukas užtrukto 773 minutes Petriukas vykdė 9 užduotis, joms vidutiniškai skyrė po 85.8889 minutes

Reikalavimai programai

- Naudoti struktūrų masyvą duomenims saugoti.
- Sukurti pradinių duomenų skaitymo funkciją void.
- Sukurti funkciją, kuri apskaičiuotų kiek laiko turėjo užtrukti. Funkcija grąžina apskaičiuotą reikšmę per funkcijos vardą.
- Sukurti funkciją, kuri apskaičiuotų kiek laiko užtrukto. Funkcija grąžina apskaičiuotą reikšmę per funkcijos vardą.
- Sukurti funkciją, kuri apskaičiuotų kiek vidutiniškai Petriukas skyrė laiko vykdytoms užduotims. Funkcija grąžina apskaičiuotą reikšmę per funkcijos vardą.
- Sukurti rezultatų įrašymo į failą funkciją void.

2. Grojaraštis

Tomas turi nuostabią kolekciją dainų, kurias norėtų su draugais paklausyti. Bėda ta, kad Tomas tiksliai nežino, kiek per dieną galėtų perklausyti, todėl jis norėtų, kad sukurtumėte programą, kuri rastų kiek duoto autoriaus dainų jis turi, ir kiek laiko užtruktų jas perklausyti.

Duomenų faile "*Dainos.txt*" yra duomenys apie Tomo grojaraštį, kuriame pirmoje eilutėje nurodyta kiek dainų jis turi, o sekančiose eilutėse dainos autorius iki '-' simbolio, dainos pavadinimas iki ',' simbolio ir dainos trukmė minutėmis ir sekundėmis atskirta tarpo simboliu.

Paleidus programą, ieškomą atlikėją įvesti klaviatūra.

Reikia rasti:

1. nurodyto autoriaus dainų kiekį;
2. kiek laiko užtruks perklausyti to autoriaus dainas, jei tarp dainų būna 5 sekundžių pertrauka (pabaigoje pertraukos nebūna);
3. rasti viso grojaraščio dainų trukmės vidurkį.

Dainos.txt
9 Arctic Monkeys-Do I Wanna Know?, 4 32 AC/DC-Thunderstruck, 4 52 AC/DC-You Shook Me All Night Long, 3 30 The Kooks-Naive, 3 23 Arctic Monkeys-Why'd You Only Call Me When You're High?, 2 41 The Kooks-She Moves In Her Own Way, 2 49 AC/DC-Highway to Hell, 3 28 AC/DC-If You Want Blood, 4 33 Arctic Monkeys-R U Mine, 3 22
Rezultatai.txt
Atlikejo AC/DC dainų kiekis: 4 Jas perklausyti užtruks 16m 38s Visų dainų trukmės vidurkis: 3m 41s

Reikalavimai programai

- Naudokite struktūras ir struktūrų masyvus.
- Pradinių duomenų skaitymo funkcija void.
- Rezultatų rašymo į failą funkcija void.
- Visų dainų vidutinės trukmės skaičiavimo funkcija, grąžinanti apskaičiuotą reikšmę per funkcijos vardą.
- Funkcija void likusiems rezultatams skaičiuoti.

3. Knygų parduotuvė

Informikas, mokydamasis programavimo, įsigijo ir perskaitė begalę knygų ir vadovėlių, kurios šiuo metu tiesiog netelpa lentynose. Kaip šios problemos sprendimą smalsusis programuotojas sugalvojo sukurti savo internetinę knygų parduotuvę. Padėkite Informikui įgyvendinti savo idėją, sukurdami knygų pirkimo programą.

Pradinių duomenų failo „Knygos.txt“ pirmoje failo eilutė įrašytas sveikasis skaičius n ($0 < n < 100$) – parduodamų knygų skaičius. Kiekvienoje sekančioje eilutėje yra įrašytas knygos kodas, autorius, pavadinimas, išleidimo metai ir kaina. Duomenų failo „Pirkejai.txt“ pirmojoje eilutėje yra nurodytas sveikasis skaičius p ($0 < p < 10$) – pirkėjų skaičius. Sekančiose eilutėse yra pateikti pirkėjo duomenys: vardas, santaupos, norimų įsigyti knygų skaičius k ($0 < k < 10$), ir šių knygų kodai.

Rezultatų faile „Rezultatai.txt“ yra pateikti pirkėjų vardai ir kiekvieno pirkėjo įsigytų knygų duomenys (kodas, autorius, pavadinimas, išleidimo metai, kaina).

Pastaba: Knygas pirkėjai gali įsigyti tokia tvarka, kokia jas yra pasirinkę.

Knygos.txt
10 1001A J.Kazlauskas Informatikos_Pradmenys 2003 10.05 1025B I.Informikas Informiko_Akademija 2015 15.50 152AB P.Petrauskas Programavimo_Pagrindai 2004 11.25 1111A L.Ilgauskas C++_Kalbos_Sintaksė 2008 9.05 1091C T.Petraitis Kompiuteriu_Architektura 2006 12.45 109SC P.Svajoklis Duomenu_Strukturos 2007 16.15 1189L S.Jankauskas Skaitmenine_Logika 2009 20.00 1002K A.Sakalauskas Tiesine_Algebra 2001 11.55 259AL H.Alkevicius Matematikos_Pagrindai 2012 13.05 1098O R.Girdauskas Kompiuterine_Grafika 2014 20.50
Pirkejai.txt
5 Tomas 25.50 2 1001A 1111A Ignas 15.25 3 1025B 152AB 1091C Viktorija 100.65 4 1098O 1001A 1002K 1189L Klaudija 55.50 1 1111A Linas 5.65 0
Rezultatai.txt
Tomas ----- 1001A J.Kazlauskas Informatikos_Pradmenys 2003 10.05 1111A L.Ilgauskas C++_Kalbos_Sintaksė 2008 9.05 ----- Ignas ----- 152AB P.Petrauskas Programavimo_Pagrindai 2004 11.25 ----- Viktorija -----

1098O R.Girdauskas Kompiuterine_Grafika 2014 20.5
1002K A.Sakalauskas Tiesine_Algebra 2001 11.55
1189L S.Jankauskas Skaitmenine_Logika 2009 20

Klaudija

Linas

Reikalavimai programai

- Struktūra, sauganti knygos duomenis.
- Struktūra, sauganti knygų masyvą ir masyvo elementų skaičių.
- Struktūra, sauganti kliento duomenis.
- Struktūra, sauganti klientų masyvą ir masyvo elementų skaičių.
- Knygų skaitymo iš failo „Knygos.txt“ funkcija.
- Pirkėjų skaitymo iš failo „Pirkejai.txt“ funkcija.
- Knygų pirkimo funkcija.
- Funkcija, šalinanti parduotas knygas iš knygų struktūros.
- Funkcija, tikrinanti, ar norima įsigyti knyga egzistuoja.
- Funkcija, tikrinanti, ar klientas išgalės įsigyti norimą knygą
- Funkcija, gražinanti knygą pagal nurodytą knygos kodą.
- Rezultatų įrašymo į failą funkcija.

4. Monopolio bankomatas

Paskalis per Kalėdas gavo žaidimą „Monopolis“, todėl per atostogas jis kas dieną susitikdavo su draugais, kad galėtų kartu pažaisti. Tačiau jam labiausiai patikdavo būti bankininku, kuris dalindavo visiems pinigus, juos skaičiuodavo. Deja, per savo neatidumą Paskalis dažnai klysdavo – tai atiduodavo per didelę pinigų sumą, tai per mažą, tai pritrūkdavo tam tikro nominalo kupiūrų. Tai pastebėjęs jo draugas Šarpis nusprendė parašyti programą, kuri tarsi bankomatas išvestų, kokiomis pinigų kupiūromis Paskalis turi atiduoti reikiamą pinigų sumą, nurodytą, ar išvis tai padaryti yra įmanoma, ar pinigų banke užtenka ir žaidimo gale taip pat išvestų bendrą visų likusių kupiūrų skaičių ir bendrą likusią pinigų sumą banke. Padėkite Šarpiui parašyti šią programą.

Pradinių duomenų failo „Duomenys.txt“ pirmoje eilutėje yra pateikti 7 skaičiai: 500 100 50 20 10 5 1 – „Monopolio“ kupiūrų piniginės vertės. Antroje eilutėje taip pat pateikti 7 skaičiai – turimi atitinkamo nominalo kupiūrų kiekiai banke. 3 eilutėje pateiktas žaidėjų skaičius n ($1 < n < 10$). Tolimesnėse n eilučių yra pateikiama žaidėjo informacija – vardas ir sveikasis skaičius, nurodantis, kokia pinigų suma turi būti atiduota tam žaidėjui.

Rezultatų failo „Rezultatai.txt“ pirmose n eilučių turi būti pateikti žaidėjų vardai ir šalia jų 7 skaičiai, nurodantys, kiek atitinkamų (kairiausias skaičius – 500 nominalo kupiūra, dešiniausias – 1, kaip ir pateikta duomenų faile) nominalo kupiūrų turi būti atiduota tam žaidėjui, kad būtų padengta jam skirta pinigų suma. Jei pinigų žaidėjui atiduoti nėra galimybių (suma per didelė, neužtenka kupiūrų ar pan.), šalia žaidėjo vietoj skaičių turi būti išvestas žodis „NEGALIMA“. Paskutinėje rezultatų failo eilutėje pateikti 2 skaičiai: pirmasis – bendras banke likęs visų kupiūrų skaičius, antrasis – bendra likusi pinigų suma banke.

Pastaba: darome prielaidą, kad pinigai žaidėjams yra atiduodami tokia tvarka, kokia žaidėjai yra surašyti duomenų faile.

Duomenys.txt	Rezultatai.txt
500 100 50 20 10 5 1 20 20 20 20 20 20 20 9 Daivaras 1169 Jovita 1250 Egidijus 1721 Algis 1694 Šarpis 2111 Jonas 2564 Petras 960 Gardevutis 2120 Smygolas 1203	Daivaras 2 1 1 0 1 1 4 Jovita 2 2 1 0 0 0 0 Egidijus 3 2 0 1 0 0 1 Algis 3 1 1 2 0 0 4 Šarpis 4 1 0 0 1 0 1 Jonas 5 0 1 0 1 0 4 Petras 1 4 1 0 1 0 0 Gardevutis 0 9 15 17 13 0 0 Smygolas NEGALIMA 28 131
500 100 50 20 10 5 1 1 1 1 1 1 1 1 9 Daivaras 1169 Jovita 1250 Egidijus 1721 Algis 1694 Šarpis 2111 Jonas 2564 Petras 960 Gardevutis 2120 Smygolas 1203	Daivaras NEGALIMA Jovita NEGALIMA Egidijus NEGALIMA Algis NEGALIMA Šarpis NEGALIMA Jonas NEGALIMA Petras NEGALIMA Gardevutis NEGALIMA Smygolas NEGALIMA 7 686
500 100 50 20 10 5 1 500 100 50 20 10 5 1 5 Daivaras 1169 Jovita 1250 Egidijus 1721 Algis 1694 Smygolas 1203	Daivaras NEGALIMA Jovita 2 2 1 0 0 0 0 Egidijus 3 2 0 1 0 0 1 Algis NEGALIMA Smygolas NEGALIMA 674 260055
500 100 50 20 10 5 1 500 100 50 20 10 5 1 5 Daivaras 7 Jovita 6 Egidijus 501 Algis 50000 Smygolas 6900	Daivaras NEGALIMA Jovita 0 0 0 0 0 1 1 Egidijus NEGALIMA Algis 1000 0 0 0 0 0 Smygolas 13 4 0 0 0 0 0 567 206120
500 100 50 20 10 5 1 1 1 1 1 1 1 1 1 Daivaras 686	Daivaras 1 1 1 1 1 1 1 0 0

500 100 50 20 10 5 1 1 1 1 1 1 1 1 1 Daivaras 687	Daivaras NEGALIMA 7 686
--	----------------------------

Reikalavimai programai

- Struktūros duomenų tipas, saugantis informaciją apie vieną kupiūrą: nominalo vertė ir kiekis banke.
- Struktūros duomenų tipas, saugantis informaciją apie vieną žaidėją: vardas ir pinigų suma, kurią reikia tam žaidėjui atiduoti.
- Duomenų skaitymo funkcija void.
- Rezultatų rašymo funkcija void.
- Funkcija, per vardą gražinanti reikšmę TRUE, jei yra įmanoma žaidėjui atiduoti pinigų, FALSE – jei to padaryti nėra įmanoma.
- Funkcija, skaičiuojanti, kiek atitinkamo nominalo kupiūrų reikia atiduoti žaidėjui.

16. VEIKSMAI SU STRUKTŪRŲ TIPO DUOMENIMIS IR STRUKTŪRŲ MASYVŲ ELEMENTAIS

1. Knygų parduotuvės apskaita

Trečiojoje pamokėlėje Informikas, padedamas jūsų, sėkmingai įgyvendino savo internetinės knygų parduotuvės idėją. Šį kartą smalsusis verslininkas nusprendė susisteminti savo parduotuvės duomenis, susijusius su knygų pardavimais ir juos atvaizduoti grafiškai.

Sukurkite programą, kuri pagal pateiktus knygų pardavimų duomenis atspausdintų pelningas mėnesio dienas lentele ir grafiškai atvaizduotų parduotuvės gautas pajamas.

Pastabos:

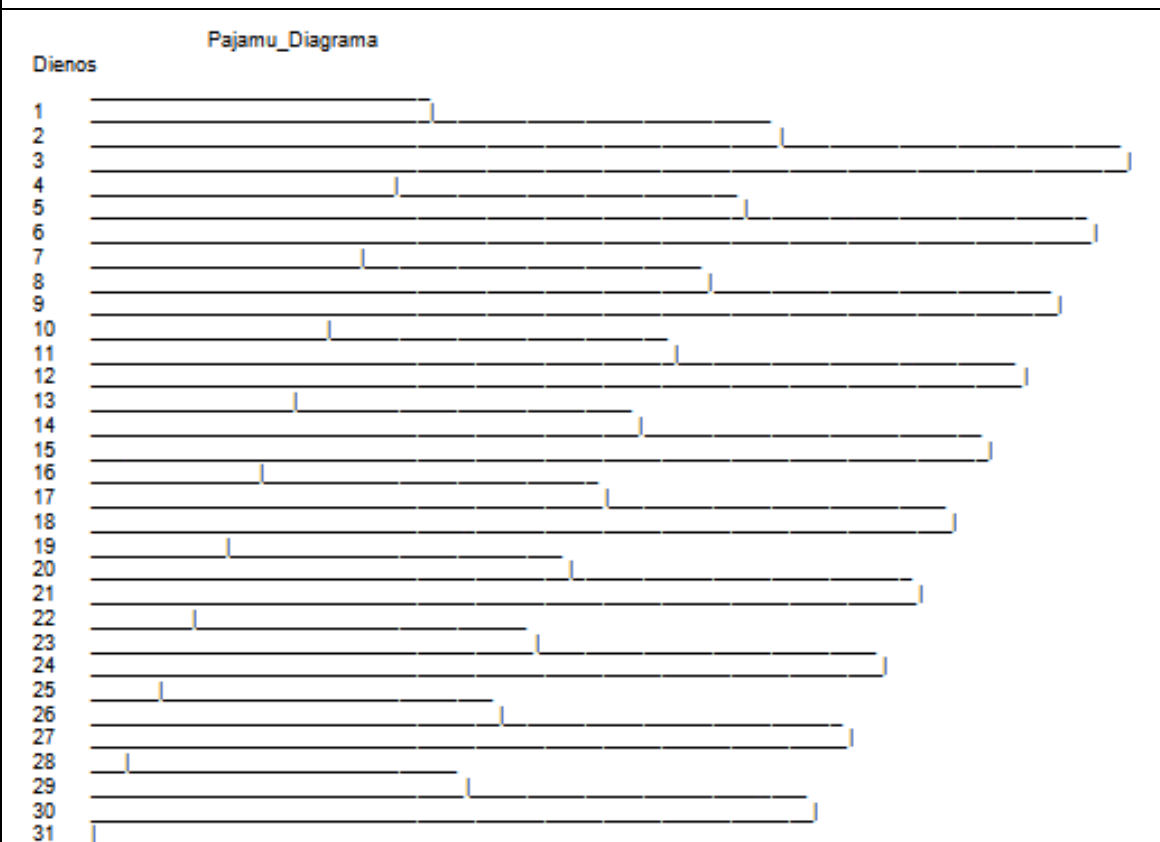
- Diena yra pelninga, jei tą dieną gautos pajamos yra ne mažesnės nei pinigų suma, kuri yra apskaičiuojama pagal formulę: (pelningiausios mėnesio dienos pajamos + mažiausiai pelningos mėnesio dienos pajamos) / 2;
 - Pelningiausios ir mažiausiai pelningos dienos ieškomos pagal du kriterijus: tą dieną gautas pajamas ir parduotų knygų skaičių;
 - Vaizduojant pajamas grafiškai, vienas simbolis ‘_’ atitinka vieną piniginių vienetą;
- Duomenų failo „**Duomenys1.txt**“ kiekvienoje eilutėje yra pateikta mėnesio diena, tą dieną parduotų knygų skaičius ir gautos pajamos.

Duomenys1.txt		
1	10	30
2	20	60
3	30	90
4	9	27
5	19	57
6	29	87
7	8	24
8	18	54
9	28	84

10	7	21
11	17	51
12	27	81
13	6	18
14	16	48
15	26	78
16	5	15
17	15	45
18	25	75
19	4	12
20	14	42
21	24	72
22	3	9
23	13	39
24	23	69
25	2	6
26	12	36
27	22	66
28	1	3
29	11	33
30	21	63
31	0	0

Rezultatų faile „Rezultatai1.txt“ lentelėje pateikiamos pelningiausios mėnesio dienos, ir pajamų diagrama.

Rezultatai1.txt



Pelningos_Dienos

Diena	Knygos	Pajamos
2	20	60
3	30	90
5	19	57
6	29	87
8	18	54
9	28	84
11	17	51
12	27	81
14	16	48
15	26	78
17	15	45
18	25	75
21	24	72
24	23	69
27	22	66
30	21	63

Reikalavimai programai

- Duomenys apie kiekvieną pardavimų dieną saugomi duomenų struktūroje.
- Duomenų skaitymo funkcija.
- Funkcija, gražinanti pelningiausią mėnesio dieną.
- Funkcija, gražinanti mažiausiai pelningą mėnesio dieną.
- Funkcija, suformuojanti pelningų dienų masyvą.
- Funkcija, atvaizduojanti pajamas diagrama.
- Funkcija, atvaizduojanti pelningas dienas lentele.

2. Informatikos konkursas

Informatikos mokytoja Paskaliui, kaip gambiausiam mokiniui, paskyrė užduotį surengti informatikos konkursą mokykloje. Konkurso taisyklės labai paprastos:

- Konkurso metu dalyviams pateikiamos 5 užduotys.
- Kiekvieno uždavinio programa turi įveikti 100 testinių duomenų rinkinių.
- Už kiekvieną įveiktą duomenų rinkinį skiriama po vieną tašką.
- Sakoma, kad uždavinys išspręstas tik tada, kai programa veikia su visais testiniais duomenų rinkiniais (100 taškų).
- Konkurso prizininkais skelbiami 3 dalyviai, išsprendę daugiausia uždavinių.
- Taip pat skelbiamas ir mažiausiai uždavinių išsprendęs dalyvis.
- Jei keli dalyviai išsprendė po tiek pat uždavinių, tai aukštesnę vietą užima tas, kuris surinko daugiau taškų.

Pradinių duomenų failo „Duomenys2.txt“ pirmoje eilutėje pateiktas sveikasis skaičius n ($4 < n < 20$) – konkurso dalyvių skaičius. Tolimesnėse n eilučių pateikta informacija apie kiekvieną dalyvį: vardas, pavardė bei tarpais atskirti 5 sveikieji skaičiai, nurodantys, kokių taškų skaičiumi įvertintas kiekvienas uždavinys.

Rezultatų failo „Rezultatai2.txt“ pirmose 3 eilutėse pateikti konkurso prizininkai (geriausias viršuje): vardas, pavardė, surinktų taškų skaičius bei išspręstų uždavinių kiekis. Kitoje eilutėje pateiktas prasčiausiai pasirodęs dalyvis: vardas, pavardė, surinktų taškų skaičius bei išspręstų uždavinių kiekis.

Duomenys2.txt	Rezultatai2.txt
5 Kazys Saja 10 9 100 100 99 Profesorius Protinguolis 100 100 100 0 0 Informikas Informikas 100 100 1 100 0 Nieko Nemoku 0 0 0 0 0 Vidutiniskai Moku 50 50 50 50 50	Konkurso prizininkai: Informikas Informikas 301 3 Profesorius Protinguolis 300 3 Kazys Saja 318 2 Prasčiausiai pasirodė: Nieko Nemoku 0 0
5 Kazys Saja 10 9 100 100 99 Profesorius Protinguolis 100 100 100 0 0 Informikas Informikas 100 100 1 100 0 Nieko Nemoku 100 0 0 0 0 Vidutiniskai Moku 50 50 50 50 50	Konkurso prizininkai: Informikas Informikas 301 3 Profesorius Protinguolis 300 3 Kazys Saja 318 2 Prasčiausiai pasirodė: Vidutiniskai Moku 250 0
5 Kazys Saja 10 9 100 100 100 Profesorius Protinguolis 100 100 100 0 2 Informikas Informikas 100 100 1 100 0 Nieko Nemoku 100 0 0 0 0 Vidutiniskai Moku 50 50 50 50 50	Konkurso prizininkai: Kazys Saja 319 3 Profesorius Protinguolis 302 3 Informikas Informikas 301 3 Prasčiausiai pasirodė: Vidutiniskai Moku 250 0

Reikalavimai programai

- Panaudoti struktūros duomenų tipą dalyvių informacijai saugoti.
- Skaičiuojant rezultatus sudaryti naują 4 vietų struktūros duomenų tipo masyvą, kurio pirmose 3 vietose saugomi konkurso prizininkų rezultatai, o 4-toje vietoje – prasčiausiai pasirodžiusio dalyvio informacija;
- Duomenų skaitymo funkcija.
- Rezultatų spausdinimo funkcija.
- Rezultatų masyvo sudarymo funkcija.
- Prasčiausiai pasirodžiusio dalyvio suradimo funkcija.

3. Mokytojai ir mokiniai

Petriukas renka duomenis apie mokytojus ir mokinius. Iš surinktų duomenų jis nori sužinoti:

- kiek kiekvienas mokytojas turi mokinių;
- mokytoją, kuris turi daugiausia mokinių;
- mokytoją, kurio mokiniai mokosi geriausiai (dalyko vidurkis).

Yra du duomenų failai.

Faile “Mokytojai.txt” pateikti duomenys apie mokytojus:

- pirmoje eilutėje – mokytojų skaičius;
- kitose eilutėse – vardas, pavardė, dalykas.

Faile “Mokiniai.txt” pateikti duomenys apie mokinius:

- pirmoje eilutėje – mokinių skaičius;
- kitose eilutėse – vardas, pavardė, dalykas, pažymys.

Mokytojai.txt	Rezultatai.txt
5 Vardenis Petrauskas Geografija Apolonijus Pavardenis Matematika Informikas Jonaitis Informatika Valentinas Gudrauskas Chemija Jadvyga Keturakė Fizika	Petrauskas 1 Pavardenis 1 Jonaitis 2 Gudrauskas 3 Keturakė 4 Daugiausia mokinių turi: Jadvyga Keturakė Geriausiai mokiniai mokosi pas: Informikas Jonaitis
Mokiniai.txt	
11 Jonas Karčiauskas Geografija 7 Petras Petraitis Matematika 5 Alfonsas Studentauskas Informatika 8 Rita Statinaitienė Chemija 9 Laima Kėdžiuvienė Fizika 2 Simona Padvelskytė Informatika 10 Raminta Masiulytė Chemija 3 Deividas Stankus Fizika 4 Jonas Kurtinaitis Chemija 8 Valdas Šarkauskas Fizika 7 Justas Pavardauskas Fizika 7	

Reikalavimai programai

- Naudoti struktūras ir struktūrų masyvus.
- Duomenų skaitymo funkcija.
- Duomenų apdorojimo funkcijos.
- Rezultatų rašymo funkcija.

4. Tarpusavio pagalba

Petriukas pastebėjo, kad jam ir jo klasės draugams sunkiau suprasti vienus dalykus, bet taip pat jiems kai kurie dalykai puikiai sekasi. Pavyzdžiui: mokiniui X puikiai sekasi matematika ir informatika, tačiau jam labai sunku mokytis istoriją, anglų bei lietuvių kalbas. Petriukas sugalvojo užregistruoti visus tuos duomenis. Kiekvienas vaikas susirašė kokius dalykus jam sekasi, o kokius nesiseka. Žiūrint į tuos duomenis, Petriukui šovė puiki idėja - kodėl gi nesuskirsčius visų duomenų pagal dalykus? Tada būtų galima lengvai matyti visus mokinius, kurie puikiai supranta ir gali padėti silpnesniems mokiniams, bei visus silpnesnius mokinius, kuriems reikia pagalbos.

Padėkite Petriukui ir sukurkite tokią programą.

Duomenų failo klase.txt pirmoje eilutėje įrašytas mokinių skaičius n, toliau seka n blokų apie kiekvieną mokinį. Informacija apie mokinį atrodo taip: mokinio vardas, kitoje eilutėje skaičius m, rodantis, kelis dalykus šis mokinys supranta labai gerai, tolimesnėse m eilučių išvardinti tie dalykai, naujoje eilutėje skaičius k, rodantis su kokiais dalykais mokinys turi problemų, ir tolimesnėse k eilučių išvardinti tie dalykai. Kad būtų lengviau suprasti duomenų failą, kiekvienas blokas yra atskirtas papildoma tuščia eilute.

7
Petriukas
2
Matematika
Informatika
3
Lietuvių kalba
Istorija
Anglų kalba
Silvija
1
Lietuvių kalba
2
Matematika
Istorija
Ingrida
1
Anglų kalba
1
Informatika
Paulius
2
Matematika
Anglų kalba
1
Biologija
Vytautas
3
Informatika
Istorija
Anglų kalba
1
Geografija

Monika
 2
 Geografija
 Biologija
 1
 Istorija
 Jonas Paulius
 3
 Anglu kalba
 Istorija
 Biologija
 2
 Rusu kalba
 Matematika

Rezultatų faile pamokos.txt Petriukas nori matyti pradinį duomenų lentelę, dalykų lentelę. Po lentelėmis turi būti pateikta: dalykas(-ai), kuriuos supranta daugiausia mokinių, dalykas(-ai), kuriuos supranta mažiausiai mokinių, ir dalykas(-ai), kurį (-iuos) suprantančių ir nesuprantančių mokinių skaičius skiriasi daugiausiai.

Mokinys	Puikiai moka	Nemoka
Petriukas	Matematika Informatika	Lietuviu kalba Istorija Anglu kalba
Silvija	Lietuviu kalba	Matematika Istorija
Ingrida	Anglu kalba	Informatika
Paulius	Matematika Anglu kalba	Biologija
Vytautas	Informatika Istorija Anglu kalba	Geografija
Monika	Geografija Biologija	Istorija
Jonas Paulius	Anglu kalba Istorija Biologija	Rusu kalba Matematika

Pavadinimas	Mokantys mokiniai	Nemokantys mokiniai
Matematika	Petriukas Paulius	Silvija Jonas Paulius
Informatika	Petriukas Vytautas	Ingrida
Lietuviu kalba	Silvija	Petriukas
Istorija	Vytautas Jonas Paulius	Petriukas Silvija Monika
Anglu kalba	Ingrida Paulius Vytautas Jonas Paulius	Petriukas
Biologija	Monika Jonas Paulius	Paulius
Geografija	Monika	Vytautas
Rusu kalba		Jonas Paulius

4 mokiniai supranta Anglu kalba.

3 mokiniai nesupranta Istorija.

Didžiausias skirtumas yra: Anglu kalba; skirtumas 3 mokiniai.

Reikalavimai programai

- Pradinius duomenis saugoti struktūros *Mokinys* masyve.
- Parašyti funkciją, kuri performuotų duomenis iš struktūros *Mokinys* masyvo į struktūros *Pamoka* masyvą.
- Parašyti funkciją, kuri per parametą grąžintų dalyko (arba dalykų, jei yra daugiau nei vienas), kurį supranta daugiausia moksleivių, pavadinimą ir moksleivių skaičių.
- Parašyti funkciją, kuri per parametą grąžintų dalyko (arba dalykų, jei yra daugiau nei vienas), kurio nesupranta daugiausia moksleivių, pavadinimą ir moksleivių skaičių.
- Parašyti funkciją, kuri per parametą grąžintų dalyko (arba dalykų, jei yra daugiau nei vienas), kur susidaro didžiausias suprantančių ir nesuprantančių skirtumas (laikoma, kad skirtumas visada teigiamas), pavadinimą ir skirtumą.
- Lentelėms gražiai suformuoti, naudoti <iomanip> bibliotekos funkciją setw.

17. STRUKTŪRŲ MASYVŲ RIKIAVIMAS, ELEMENTŲ ŠALINIMAS IR ĮTERPIMAS

1. Prekių apskaita

Petriukas sugalvojo registruoti parduotuvėje išleidžiamus pinigus. Dėl to į *prekes.txt* duomenų failą surašė visas perkamas prekes: prekės pavadinimas ir kaina eurais, pavadinimas nuo kainos atskirtas „,-“. Vienos prekės pavyzdys: Arbata - 0.99

Kuo daugiau Petriukas rašė prekių, tuo jose sunkiau susigaudyti, todėl jam būtų žymiai patogiau, jei prekės būtų surikiuotos nuo brangiausios iki pigiausios (jei prekių kaina ta pati, turėtų būti pateiktos abėcėlės tvarka).

Bet tada Petriukas suprato, kad nori matyti sąrašą ir po pakeitimų, dėl to sudarė dar vieną duomenų failą *pakeitimai.txt*, kuriame yra informacija apie prekes, kurias reikia pridėti arba pašalinti, prieš kiekvieną prekę yra „+“ arba „-“ ženklas, kuris nurodo, ar reikia pridėti prekę, ar ją pašalinti (visa kita informacija apie prekes pateikiama taip pat, kaip ir *prekes.txt* faile).

Padėkite Petriukui. Sukurkite programą, kuri pagal jo pateiktus duomenų failus ir reikalavimus sutvarkytų prekių sąrašą ir rezultatų faile *rezultatai.txt* pateiktų dvi lenteles: surikiuotų pradinių duomenų prekių sąrašą ir prekių sąrašą po pakeitimų.

prekes.txt	pakeitimai.txt	rezultatai.txt
Arbata - 0.99 Vaisvandeniai - 1.45 Paukstiena - 2.55 Jogurtas - 0.87 Sokoladas - 1.01 Pomidorai - 0.88 Svogunai - 0.20 Cesnakas - 0.35	- Arbata - 0.99 + Duona - 0.72 + Saldainiu dezute - 3.00 + Guma - 0.03 - Svogunai - 0.20 + Agurkas - 0.88	Surikiuoti pradiniai duomenys ----- Pavadinimas Kaina ----- Paukstiena 2.55 ----- Vaisvandeniai 1.45 ----- Sokoladas 1.01 ----- Arbata 0.99 ----- Pienas 0.88 ----- Pomidorai 0.88 ----- Jogurtas 0.87 ----- Cesnakas 0.35 ----- Svogunai 0.20 ----- Po pakeitimų ----- Pavadinimas Kaina ----- Saldainiu dezute 3.00 ----- Paukstiena 2.55 ----- Vaisvandeniai 1.45 ----- Sokoladas 1.01 ----- Agurkas 0.88 ----- Pienas 0.88 ----- Pomidorai 0.88 ----- Jogurtas 0.87 ----- Duona 0.72 ----- Cesnakas 0.35 ----- Guma 0.03 -----

Reikalavimai programai

- Rikiavimo funkcija, kurioje realizuotumėte pasirinktą rikiavimo algoritmą (turite nurodyti kokį algoritmą realizavote).
- Funkcija, kuri prekę įterpia į reikiamą vietą sąrašė (nebus įskaityta, jei įdės bet kur ir surikiuos).
- Funkcija, kuri pašalina prekę iš sąrašo, ir perstumia kitas, taip kad sąrašas liktų surikiuotas, netaikant rikiavimo funkcijos.

2. Geriausi mokiniai

Petriuko mokykloje norima apdovanoti geriausius 10-ų klasių mokinius. Padėkite Petriukui parašyti programą, kuri iš kiekvienos klasės sąrašo pašalintų mokinius, kurių vidurkis mažesnis nei visos klasės vidurkis, bei sudarytų geriausių mokinių sąrašą (kurių vidurkis aukštesnis, nei atrinktųjų bendras vidurkis).

Pateikti 4 duomenų failai, kuriuose yra N eilučių su mokinių informacija - vardas ir vidurkis.

1.txt	2.txt	3.txt	4.txt
Edvinas 9.9 Tomas 9.5 Antanas 4.3 Nerijus 7.1 Samanta 10 Gabrielė 2.9 Eugenijus 5.75	Ieva 8.2 Erika 5.9 Gabija 3.5 Agnė 9.1 Linus 1.5 Simonas 7.1 Aurimas 8.8	Lukas 9.2 Mantas 6.7 Tautvydas 7.1 Raminta 8.5 Reda 4.7 Diana 3.8	Emilija 10 Austėja 9.9 Viltė 7.2 Jokūbas 8.5 Nojus 3.9 Jonas 5.7 Kajus 6.7 Dovydas 6.8
rezultatai.txt			
Klasė 1: ----- Vardas Vidurkis ----- Samanta 10.00 Edvinas 9.90 Tomas 9.50 Nerijus 7.10 -----			
Klasė 2: ----- Vardas Vidurkis ----- Agnė 9.10 Aurimas 8.80 Ieva 8.20 Simonas 7.10 -----			
Klasė 3: ----- Vardas Vidurkis ----- Lukas 9.20 Raminta 8.50 Tautvydas 7.10 Mantas 6.70 -----			

Klasė 4:

| Vardas | Vidurkis |

Emilija	10.00
Austėja	9.90
Jokūbas	8.50

Rezultatas:

| Vardas | Vidurkis |

Emilija	10.00
Samanta	10.00
Austėja	9.90
Edvinas	9.90
Tomas	9.50
Lukas	9.20
Agnė	9.10
Aurimas	8.80

Reikalavimai programai

- Rikiavimo funkcija.
- Šalinimo funkcija (šalinti po vieną mokinį iš sąrašo ir perstumti kitus, taip, kad sąrašas liktų surikiuotas, netaikant rikiavimo funkcijos).
- Sujungimo funkcija.
- Išspausdinti kiekvienos klasės mokinius, kurių vidurkis aukštesnis už jų klasės vidurkį.
- Iš atrinktų kiekvienos klasės mokinių Išspausdinti geriausius mokinius, kurių vidurkis aukštesnis už visų atrinktųjų vidurkį.

3. Prekių tvarkymas

Petriukas gavo užduotį sukurti labai paprastą programą, kuri sektų prekes parduotuvėje. Pradinių duomenų failai “duom” ir “parduota” yra išsaugoti .csv formatu (Kableliais atskirtos reikšmės, angl. *comma - separated values*). Šiuos failus galima atidaryti ir teksto redaktoriumi, ir Microsoft Excel programa.

- “duom” faile yra duomenys apie parduotuvės prekes: Prekės kodas, pavadinimas, galiojimo data “unix time” formatu ([plačiau](#)), kiekis.
- “parduota” faile yra duomenys apie parduotas prekes: Prekės kodas, kiekis.

Užduotis:

Iš prekių sąrašo išbraukti parduotas prekes. Tuomet sukurti naują sąrašą be prekių, kurių galiojimo laikas baigėsi ir jas išvesti į ekraną. Likusias prekes surikiuoti pagal galiojimo datą didėjimo tvarka ir išvesti į rezultatų failą .csv formatu.

Dabartinį laiką unix formatu galima gauti taip:

long dabartinisLaikas = std::time(nullptr);

Pradiniai duomenys:

duom.csv
<p>16</p> <p>PR987456,Duona,1458432000,21</p> <p>PR365894,Coca-Cola,1493769600,43</p> <p>PR255668,Pepsi,1505001600,34</p> <p>PR877230,Makaronai,1478736000,16</p> <p>PR149879,Tualetinis Popierius,1601856000,40</p> <p>PR212035,Ryziai,1527811200,14</p> <p>PR274191,Kava,1496275200,20</p> <p>PR336347,Daktariska desra,1464825600,8</p> <p>PR398504,Aliejus,1474675200,17</p> <p>PR460660,Suris,1460246400,9</p> <p>PR522816,Skalbimo milteliai,1517616000,32</p> <p>PR584972,Koldunai,1483228800,11</p> <p>PR647128,Arbata,1613606400,10</p> <p>PR709285,Sureliai,1454630400,5</p> <p>PR771441,Jogurtas,1456272000,3</p> <p>PR833597,Pienas,1456790400,2</p>
parduota.csv
<p>13</p> <p>PR987456,2</p> <p>PR877230,1</p> <p>PR274191,5</p> <p>PR398504,2</p> <p>PR365894,4</p> <p>PR365894,2</p> <p>PR255668,6</p> <p>PR336347,4</p> <p>PR336347,2</p> <p>PR336347,2</p> <p>PR987456,5</p> <p>PR877230,8</p> <p>PR877230,7</p>
rez.csv
<p>11</p> <p>PR987456, Duona, 1458432000, 14</p> <p>PR460660, Suris, 1460246400, 9</p> <p>PR398504, Aliejus, 1474675200, 15</p> <p>PR584972, Koldunai, 1483228800, 11</p> <p>PR365894, Coca-Cola, 1493769600, 37</p> <p>PR274191, Kava, 1496275200, 15</p> <p>PR255668, Pepsi, 1505001600, 28</p> <p>PR522816, Skalbimo milteliai, 1517616000, 32</p> <p>PR212035, Ryziai, 1527811200, 14</p> <p>PR149879, Tualetinis Popierius, 1601856000, 40</p> <p>PR647128, Arbata, 1613606400, 10</p>

Reikalavimai programai

- Duomenų saugojimui naudoti struktūros duomenų tipą.
- Duomenų skaitymo funkcija.
- Parduotų prekių pašalinimo funkcija.
- Funkcija gerų prekių sąrašui sudaryti
- Rikiavimo funkcija.
- Rezultatų spausdinimo funkcija.

4. Kortų žaidimas

Kitą savaitę vyks Paskalio gimtadienis, susirinks visi geriausi jo draugai, todėl šiuo metu jis laužo galvą, ką reikės veikti gimtadienio metu. Kiek pagalvojęs, jis sugalvojo kortų žaidimą, kurio taisyklės tokios:

- Pradžioje visi žaidėjai turi po 100 taškų;
- Ant stalo padedama kortų kaladė, o ant kiekvienos kortelės nupieštas vienas simbolis: „V“, „P“, „-“, „+“ arba „~“;
- Kortelės „V“ ir „P“ reiškia, kad žaidėjų sąrašas išrikiuojamas atitinkamai pagal vardą arba pavardę abėcėlės tvarka;
- Kortelė „-“ reiškia, kad neiškritusių žaidėjų sąrašė tuo žaidimo metu esantis paskutinis (ne taškų atžvilgiu paskutinis, bet tiesiog esantis sąrašo apačioje) žaidėjas iškrenta iš žaidimo;
- Kortelė „+“ reiškia, kad į žaidimą grįžta vėliausiai į iškritusių žaidėjų sąrašą įtrauktas žaidėjas su tiek taškų, kiek jų turėjo prieš iškrisdamas;
- Kortelė „~“ reiškia, kad neiškritusių žaidėjų sąrašas išrikiuojamas tokia tvarka: pirmasis sąrašė - daugiausia taškų turintis žaidėjas, antrasis - mažiausiai taškų turintis, tada iš likusių dalyvių daugiausiai taškų turintis, tada iš likusių dalyvių mažiausiai taškų turintis ir t.t.;
- Po kiekvieno kortelės ištraukimo pirmiausia įvykdomas veiksmas, kurį nurodo kortelė, o po to žaidėjai gauna taškus: pirmasis sąrašė – $n * 10$, antrasis – $(n-1) * 10$, trečiasis – $(n - 2) * 10$ ir t.t., kur n – tuo žaidimo metu neiškritusių žaidėjų sąrašė esantis žaidėjų skaičius;
- Žaidimas baigiasi, kai kaladėje nebelieka kortelių, o laimėtoju skelbiamas tas žaidėjas, kuris turi daugiausia taškų.

Parašykite programą, kuri nustatytų Paskalio sukurto žaidimo laimėtoją.

Pradinių duomenų failo „Duomenys.txt“ pirmoje eilutėje pateiktas sveikasis skaičius n ($3 < n < 20$) – žaidėjų skaičius. Tolimesnėse n eilučių pateikta informacija apie kiekvieną žaidėją: vardas ir pavardė. Paskutinėje duomenų failo eilutėje pateikta simbolių eilutė – žaidimo eiga, t.y., kokia eilės tvarka buvo traukiamos kortelės žaidimo metu.

Rezultatų failo „Laimetojas.txt“ pirmoje eilutėje pateikta informacija apie žaidimo laimėtoją: vardas, pavardė ir surinktų taškų kiekis.

Duomenys.txt	Laimetojas.txt	Žaidimo eiga
<p>6 Blezas Paskalis Petras Klaustukas Balys Sruoga Sarpis Idomutis Informikas Baltutis Konstitucija Balsute V-P</p>	<p>Balys Sruoga 220</p>	<p>Kortelė: V</p> <p>-----</p> <p>Šarašas prieš veiksma: Blezas Paskalis 100 Petras Klaustukas 100 Balys Sruoga 100 Sarpis Idomutis 100 Informikas Baltutis 100 Konstitucija Balsute 100</p> <p>-----</p> <p>Šarašas po veiksmo: Balys Sruoga 100 Blezas Paskalis 100 Informikas Baltutis 100 Konstitucija Balsute 100 Petras Klaustukas 100 Sarpis Idomutis 100</p> <p>-----</p> <p>Šarašas po taškų pridėjimo: Balys Sruoga 160 Blezas Paskalis 150 Informikas Baltutis 140 Konstitucija Balsute 130 Petras Klaustukas 120 Sarpis Idomutis 110</p> <p>-----</p> <p>Kortelė: -</p> <p>-----</p> <p>Šarašas prieš veiksma: Balys Sruoga 160 Blezas Paskalis 150 Informikas Baltutis 140 Konstitucija Balsute 130 Petras Klaustukas 120 Sarpis Idomutis 110</p> <p>-----</p> <p>Šarašas po veiksmo: Balys Sruoga 160 Blezas Paskalis 150 Informikas Baltutis 140 Konstitucija Balsute 130 Petras Klaustukas 120</p> <p>-----</p> <p>Šarašas po taškų pridėjimo: Balys Sruoga 210 Blezas Paskalis 190 Informikas Baltutis 170</p>

		Konstitucija Balsute 150 Petras Klaustukas 130 ----- Kortelė: P ----- Sąrašas prieš veiksmą: Balys Sruoga 210 Blezas Paskalis 190 Informikas Baltutis 170 Konstitucija Balsute 150 Petras Klaustukas 130 ----- Sąrašas po veiksmo: Konstitucija Balsute 150 Informikas Baltutis 170 Petras Klaustukas 130 Blezas Paskalis 190 Balys Sruoga 210 ----- Sąrašas po taškų pridėjimo: Konstitucija Balsute 200 Informikas Baltutis 210 Petras Klaustukas 160 Blezas Paskalis 210 Balys Sruoga 220 -----
--	--	---

Reikalavimai programai

- Naudoti struktūros duomenų tipą žaidėjų informacijai saugoti: vardas, pavardė, taškų kiekis.
- Duomenų skaitymo funkcija.
- Rezultatų spausdinimo funkcija.
- Funkcija, kuri seka žaidimo eigą ir kurioje atliekami skaičiavimai.
- Funkcija, rikiuojanti sąrašą pagal vardą.
- Funkcija, rikiuojanti sąrašą pagal pavardę.
- Funkcija, rikiuojanti sąrašą pagal taškus: max, min, max, min... (kaip apibrėžta užduotyje).
- Funkcija, kuri pašalina žaidėją iš sąrašo.
- Funkcija, kuri prideda iškritusį žaidėją į sąrašą.
- Taškų pridėjimo funkcija.
- Funkcija, per savo vardą grąžinanti laimėtoją.

Pastabos:

- Patarčiau pasidomėti ir naudoti vector konteinerį (kintančio ilgio masyvą), nes taip bus patogiau šalinti, pridėti į sąrašą žaidėjus.
- Patarčiau iškritusius žaidėjus saugoti būtent vektoriuje, bet galima išsisukti ir naudojant masyvus.

- D priedaidą, kad „-“ kortelių kaladėje visada bus mažiau už bendrą žaidėjų skaičių, t.y. neiškritusių žaidėjų sąraše visada bus likęs bent vienas žaidėjas.
- Darome priedaidą, kad pradinis neiškritusių žaidėjų sąrašas yra toks, koks pateiktas duomenų faile.
- Jei keli žaidėjai pabaigoje yra surinkę vienodą taškų skaičių, tai laimėtoju tampa tas, kuris sąraše yra aukščiau.
- Gali būti, kad bus ištraukta kortelė „+“, bet nebus nė vieno iškritusio žaidėjo, tad reikalingas tikrinimas!

5. Vartotojo sąsaja

Regis, Informiko knygų parduotuvės sumanymas Jūsų dėka sėkmingai įgyvendintas. Beliko viena nedidelė smulkmena – sukurti konsolinę vartotojo sąsają greitam ir nesudėtingam duomenų apie knygas valdymui. Sąsajoje turi būti realizuoti: pagrindinis meniu su duomenų skaitymo iš failo (reikia nurodyti failo vardą), duomenų išvedimo į ekraną lentelę, duomenų rikiavimo (mažėjančiai ir didėjančiai), duomenų šalinimo (reikia nurodyti šalinamos knygos ID) bei duomenų įterpimo (reikia nurodyti knygos pavadinimą, id, autorių, kainą) parinktimis.

Duomenų failo struktūra: Duomenų failo pirmoje failo eilutė įrašytas sveikasis skaičius n ($0 < n < 100$) – parduodamų knygų skaičius. Kiekvienoje sekančioje eilutėje yra įrašytas knygos kodas, autorius, pavadinimas, išleidimo metai ir kaina.

Duomenys.txt

```
10
1001A J.Kazlauskas Informatikos_Pradmenys 2003 10.05
1025B I.Informikas Informiko_Akademija 2015 15.50
152AB P.Petrauskas Programavimo_Pagrindai 2004 11.25
1111A L.Ilgauskas C++_Kalbos_Sintaksė 2008 9.05
1091C T.Petraitis Kompiuteriu_Architektura 2006 12.45
109SC P.Svajoklis Duomenu_Strukturos 2007 16.15
1189L S.Jankauskas Skaitmenine_Logika 2009 20.00
1002K A.Sakalauskas Tiesine_Algebra 2001 11.55
259AL H.Alkevicus Matematikos_Pagrindai 2012 13.05
1098O R.Girdauskas Kompiuterine_Grafika 2014 20.50
```

Reikalavimai programai

- Struktūra, sauganti knygos duomenis.
- Struktūra, sauganti knygų masyvą ir masyvo elementų skaičių.
- Knygų skaitymo funkcija.
- Knygos šalinimo iš knygų masyvo pagal nurodytą knygos kodą funkcija.
- Knygų masyvo papildymo nauja knyga funkcija.
- Knygų masyvo rikiavimo didėjančiai pagal du kriterijus (knygos kainą ir pavadinimą) funkcija.
- Knygų masyvo rikiavimo mažėjančiai pagal du kriterijus (knygos kainą ir pavadinimą) funkcija.
- Rikiavimui naudoti vieną iš
- https://lt.wikipedia.org/wiki/Rikiavimo_algorithmas#Algoritmas.C5.B3_sud.C4.97tingum.C5.B3_lentel.C4.97_nurodytu_metodu
- Duomenų apie knygas išvedimo į ekraną lentelę funkcija.

Vartotojo sąsajos pavyzdys

```
Skaityti duomenis is failo(skaityti)
Rikiuoti duomenis(rikuiuoti)
Iterpti nauja irasa(Iterpti)
Salinti irasa(salinti)
Baigti(baigti)
Iveskite komanda: skaityti

Naudoti nauja duomenu faila (Y/N): Y

Iveskite duomenu failo varda: Knygos.txt
```

Kodas	Autorius	Pavadinimas	Metai	Kaina
1001A	J.Kazlauskas	Informatikos_Pradmenys	2003	10.05
1025B	I.Informikas	Informiko_Akademija	2015	15.5
152AB	P.Petrauskas	Programavimo_Pagrindai	2004	11.25
1111A	L.Ilgauskas	C++_Kalbos_Sintaks]	2008	9.05
1091C	T.Petraitis	Kompiuteriu_Architektura	2006	12.45
1095C	P.Svajoklis	Duomenu_Strukturos	2007	16.15
1189L	S.Jankauskas	Skaitmenine_Logika	2009	20
1002K	A.Sakalauskas	Tiesine_Algebra	2001	11.55
259AL	H.Alkevicius	Matematikos_Pagrindai	2012	13.05
10980	R.Girdauskas	Kompiuterine_Grafika	2014	20.5

```
Naudoti nauja duomenu faila (Y/N): N
Skaityti duomenis is failo(skaityti)
Rikiuoti duomenis(rikuiuoti)
Iterpti nauja irasa(Iterpti)
Salinti irasa(salinti)
```

18. ĮVAIRŪS TREČIO LYGIO UŽDAVINIAI

1. Atmosferos teršalai

Lietuvos Respublikos statistikos departamentas (<http://osp.stat.gov.lt/>) pateikia informaciją apie atmosferos teršalus.

Pirmoje pradinį duomenų failo *tersalai.txt* eilutėje įrašyti trys sveikieji skaičiai: teršalų rūšių skaičius r , pradiniai analizės metai $prmetai$, kelis metus m buvo atliekama atmosferos teršalų analizė. Tolesnėse r eilučių surašyta informacija apie kiekvieną teršalų rūšį: pavadinimas (skiriama 30 pozicijų) ir m realiųjų skaičių – kiekvienais tyrimo metais užregistruotų teršalų kiekis tonomis, pradedant nuo $prmetai$.

Parašykite programą, kuri analizuoja pateiktus statistinius duomenis:

- kuriais metais kiekvienos rūšies teršalų buvo užfiksuota daugiausia. Jei buvo keli vienodi metai, kai teršalų buvo užfiksuota daugiausia, tai turi būti išvesti anksčiausi metai, kai teršalų užfiksuota daugiausiai;
- kuriais metais kiekvienos rūšies teršalų buvo užfiksuota mažiausiai. Jei buvo keli vienodi metai, kai teršalų buvo užfiksuota mažiausiai, tai turi būti išvesti vėliausi metai, kai teršalų užfiksuota mažiausiai;
- kiek tonų teršalų buvo užregistruota kiekvienais metais iš viso;
- koks vidutinis teršalų kiekis užfiksuotas per metus (imti visų metų bendrą teršalų kiekį ir dalinti iš metų skaičiaus);
- surikiuoti teršalus pagal bendrą kiekį didėjimo tvarka, o jei kiekiai sutampa – pagal abėcėlę nuo Z iki A;
- pašalinti iš sąrašo teršalus, kurių užregistruotas kiekis yra didžiausias. Jei yra kelios rūšys, atitinkančios šią sąlygą, pašalinti tuos, kurie pagal abėcėlę yra žemiau.

Pradiniai duomenys	
5 2007 8	
Sieros dioksidas	18620.3 16804.8 16577.3 15303.8 16235.6 14201.9 11840.3 11047.3
Azoto oksidai	10998.1 11383.4 9850.2 10232.3 9556.8 9389.9 8117.3 7808.9
Anglies monoksidas	21106.7 19172.6 17015.2 19034.1 19343.1 18957.1 18615.2 16794.3
Lakūs organiniai junginiai	16924.4 18816.7 16432.7 14642.6 14776.7 14454.8 14655.7 10448.4
Fluoras ir kiti teršalai	781 895.1 1106.6 1244 3835.3 4023.2 3816.5 6906.8
Rezultatai	
Sieros dioksidas	2007 2014
Azoto oksidai	2008 2014
Anglies monoksidas	2007 2014
Lakūs organiniai junginiai	2008 2014
Fluoras ir kiti teršalai	2014 2007

2007	68430.5
2008	67072.6
2009	60982.0
2010	60456.8
2011	63747.5
2012	61026.9
2013	57045.0
2014	53005.7

	61470.9

Fluoras ir kiti teršalai	22608.5
Azoto oksidai	77336.9
Sieros dioksidas	120631.3
Lakūs organiniai junginiai	121152.0
Anglies monoksidas	150038.3

Fluoras ir kiti teršalai	22608.5
Azoto oksidai	77336.9
Sieros dioksidas	120631.3
Lakūs organiniai junginiai	121152.0

Reikalavimai programai

- 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 išsaugojimo faile funkcija void.
- Rezultatai rezultatų faile *tersalaires.txt* turi būti išsaugoti taip, kaip pateikta pavyzdyje.

2. Spalvotųjų metalų laužo susidarymas ir panaudojimas

Lietuvos Respublikos statistikos departamentas (<http://osp.stat.gov.lt/>) pateikia informaciją apie spalvotųjų metalų laužo susidarymą ir panaudojimą apskrityse.

Pirmoje pradinių duomenų failo *lauzas.txt* eilutėje įrašyti sveikasis skaičius: apskričių skaičius *a*. Tolesnėse *a* eilučių išvardinti apskričių pavadinimai po vieną pavadinimą eilutėje. Toliau yra dvi eilutės, kuriose įrašyta po *a* realiųjų skaičių ta tvarka, kuria išvardintos apskritys: pirmoje eilutėje spalvotųjų metalų laužo susidarymas kiekvienoje apskrityje, antroje – spalvotųjų metalų laužo panaudojimas kiekvienoje apskrityje. Laužo matavimo vienetas – tūkst. tonų. Atkreipkite dėmesį, kad laužo gali būti sunaudota daugiau, negu surinkta tais metais (gali būti naudojamas ankstesniais metais sukauptas laužas).

Parašykite programą, kuri analizuoja pateiktus statistinius duomenis:

- kiek iš viso laužo susidarė;
- kiek iš viso laužo buvo panaudota;
- kurioje apskrityje susidarė daugiausia spalvotųjų metalų laužo. Jei yra kelios tokios apskritys, tuomet reikia išvesti tą, kuri pradiniam sąrašė yra aukščiausiai;
- kurioje apskrityje sunaudota mažiausiai spalvotųjų metalų laužo. Jei yra kelios tokios apskritys, tuomet reikia išvesti tą, kuri pradiniam sąrašė yra žemiausiai;
- koks vidutinis susidariusio ir panaudoto spalvotųjų metalų laužo skirtumo modulis (reikia apskaičiuoti skirtumų modulius kiekvienoje apskrityje ir rasti jų vidurkį);
- surikiuokite apskritis sunaudoto spalvotųjų metalų laužo mažėjimo tvarka. Jei kiekiai sutampa, tuomet pagal abėcėlę nuo Z iki A;
- įterpkite į surikiuotą sąrašą po vidurinio nario tą apskritį, kurioje sunaudota mažiausiai spalvotųjų metalų laužo.

Pradiniai duomenys
10 Alytaus Kauno Klaipėdos Marijampolės Panevėžio Šiaulių Tauragės Telšių Utenos Vilniaus 100.7 16150.7 2516.6 127.5 1552.5 220.6 151.4 76.3 427.7 9631.4 1446.5 2006.7 6.0 0.3 1.9 25.2 0.0 0.4 0.0 56.0
Rezultatai
30955.4 ----- 3543.0 ----- Kauno 16150.7 ----- Utenos 0.0 ----- 3010.4 ----- Kauno 2006.7

Alytaus 1446.5
Vilniaus 56.0
Šiaulių 25.2
Klaipėdos 6.0
Panevėžio 1.9
Telšių 0.4
Marijampolės 0.3
Utenos 0.0
Tauragės 0.0

Kauno 2006.7
Alytaus 1446.5
Vilniaus 56.0
Šiaulių 25.2
Klaipėdos 6.0
Utenos 0.0
Panevėžio 1.9
Telšių 0.4
Marijampolės 0.3
Utenos 0.0
Tauragės 0.0

Reikalavimai programai

- 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 įterpimo funkcija void.
- Rezultatų išsaugojimo faile funkcija void.
- Rezultatai rezultatų faile *lauzasrez.txt* turi būti išsaugoti taip, kaip pateikta pavyzdyje.

3. Vidutinė temperatūra

Lietuvos Respublikos statistikos departamentas (<http://osp.stat.gov.lt/>) pateikia informaciją apie vidutinę temperatūrą, išmatuotą šalies teritorijoje esančiose meteorologijos stotyse. Pirmoje pradinių duomenų failo *temperatura.txt* eilutėje įrašyti trys sveikieji skaičiai: meteorologijos stočių skaičius *s*, pradiniai analizės metai *prmetai*, galutiniai analizės metai *gmetai*.

Tolesnėse *s* eilučių surašyti meteorologijos stočių pavadinimai, po vieną pavadinimą eilutėje. Pavadinimą sudaro vienas žodis.

Tolesnėse *s* eilučių surašytos kiekvienos meteorologijos stoties vidutinės metinės temperatūros nuo *prmetai* iki *gmetai*.

Parašykite programą, kuri analizuoja pateiktus statistinius duomenis:

- kurioje meteorologijos stotyje užfiksuota žemiausia metinė temperatūra. Jei yra kelios tokios stotys, turi būti įrašyta ta stotis, kuri pradiniame sąraše yra aukščiausiai;
- kurioje meteorologijos stotyje užfiksuota aukščiausia metinė temperatūra. Jei yra kelios tokios stotys, turi būti įrašyta ta stotis, kuri pradiniame sąraše yra žemiausiai;
- kokia kiekvienos meteorologijos stoties vidutinė metinė temperatūra yra nagrinėjamame metų intervale;

- surikiuokite meteorologijos stotis vidutinių metinių temperatūrų didėjimo tvarka. Jei vidutinės metinės temperatūros sutampa, tuomet reikia rikiuoti meteorologijos stotis pagal abėcėlę nuo A iki Z;
- iš surikiuoto sąrašo pašalinkite meteorologijos stotį, kurioje buvo užfiksuota aukščiausia metinė temperatūra.

Pradiniai duomenys
5 2007 2010 Biržai Laukuva Kaunas Klaipėda Šiauliai 7.5 8.1 6.9 6.1 7.1 7.6 6.4 5.4 8.1 8.5 7.2 6.5 8.8 9.1 8.1 6.6 7.6 8.0 7.0 6.0
Rezultatai
Laukuva 5.4 2010 ----- Klaipėda 9.1 2008 ----- Biržai 7.2 Laukuva 6.6 Kaunas 7.6 Klaipėda 8.2 Šiauliai 7.2 ----- Laukuva 6.6 Biržai 7.2 Šiauliai 7.2 Kaunas 7.6 Klaipėda 8.2 ----- Laukuva 6.6 Biržai 7.2 Šiauliai 7.2 Kaunas 7.6 -----

Reikalavimai programai

- 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 išsaugojimo faile funkcija void.
- Rezultatai rezultatų faile *temperaturarez.txt* turi būti išsaugoti taip, kaip pateikta pavyzdyje.

4. Juridinių asmenų gauta parama

Lietuvos Respublikos statistikos departamentas (<http://osp.stat.gov.lt/>) pateikia informaciją apie juridinius asmenis, gavusius paramą.

Pirmoje pradinių duomenų failo *parama.txt* eilutėje įrašyti du sveikieji skaičiai: paramos teikėjų tipų skaičius *p* ir sričių, kurioms skirta parama, skaičius *s*. Toliau seka *p* eilučių, kuriose išvardinti paramos teikėjų tipai. Tolesnėse *s* eilučių įrašyti tokie duomenys: srities pavadinimas (skiriama 20 pozicijų) ir *p* realiųjų skaičių – kokio dydžio paramą tūkst. eurų suteikė atitinkami teikėjai tai sričiai.

Parašykite programą, kuri analizuoja pateiktus statistinius duomenis:

- kurio tipo teikėjai suteikė daugiausia paramos. Jei kelių tipų yra vienodas skaičius, tuomet turi būti išvedamas tas tipas, kuris pradiniam sąraše yra žemiausiai;
- kuriai sričiai paramos skirta mažiausiai. Jei buvo kelios sritys, kurioms paramos buvo skirta mažiausiai, tai turi būti išvesta aukščiausiai pradiniam sąraše esanti sritis;
- koks bendras paramos dydis;
- koks vidutinis paramos dydis sričiai (imti visų sričių paramą ir dalinti iš sričių skaičiaus);
- surikiuoti sritis pagal bendrą paramą mažėjimo tvarka, o jei skaičiai sutampa – pagal abėcėlę nuo A iki Z;
- pašalinti iš sąrašo sritį, kurioje buvo užfiksuota bendra mažiausia gauta parama.

Pradiniai duomenys	
3	4
Užsienio_juridiniai_asmenys	
LR_juridiniai_asmenys	
Anoniminiai_fiziniai_asmenys	
Kultūra	2581.5 8556.8 1263.6
Švietimas	2937.3 6316.3 1310.5
Sportas	1815.3 35478.2 881.6
Religija	4715.8 3937.3 1944.6
Rezultatai	
LR_juridiniai_asmenys 54288.6	

Švietimas 10564.1	

71738.8	

17934.7	

Sportas	38175.1
Kultūra	12401.9
Religija	10597.7
Švietimas	10564.1

Sportas	38175.1
Kultūra	12401.9
Religija	10597.7

Reikalavimai programai

- 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 išsaugojimo faile funkcija void.
- Rezultatai rezultatų faile *paramairez.txt* turi būti išsaugoti taip, kaip pateikta pavyzdyje.