

**Kauno technologijos universitetas  
Informatikos fakultetas**

# **PROGRAMAVIMO PAMOKĖLIŲ UŽDUOTYS 2017-2018 M.M.**

**Užduotis rengė: Andrius Šulskis, Arnas Nakrošis, Gerardas Karnila, Aleksas Golubevas, Monika Pociūtė, Andrius Petrauskis, Mindaugas Butkus, Matas Žilinskas, Justė Žilinskaitė, Otas Meironas, Edvinas Kilčiauskas, Gytis Vaičekauskas, Karolina Šarauskaitė, Deividas Andrikis, Viktorija Varnaitė, Arnas Valenta, Regijus Borodinas, Mangirdas Kazlauskas, Artūras Marcišauskas, Jurgis Gečys, Edvinas Šulčius, Simonas Baltulionis, Paulius Melaika, Skirmantas Navickas, Renata Burbaitė**

**6 skyrelio uždaviniai „Žiogas“, „Sultis gerti sveika“, „Varlių koncertas“ – iš Lietuvos mokinių informatikos olimpiadų archyvo**

**Kaunas  
2017-2018**

## Turinys

1.	TIESINIAI ALGORITMAI.....	5
1.	1. Gudruolis Tomukas.....	5
2.	2. Programuotojų sportinio ėjimo varžybos.....	5
3.	3. Skaičių konvertavimas iš dvejetainės į dešimtainę sistemą.....	6
4.	4. Rudens gėrybės.....	6
5.	5. Skanėstai.....	6
6.	6. Siuntos.....	7
7.	7. Kelio apskaičiavimas.....	7
8.	8. Pirato lobiai.....	7
9.	9. Minecraft namas.....	8
10.	10. Programuotojų iššūkiai.....	8
2.	ŠAKOTIEJI ALGORITMAI.....	9
1.	1. Triliemenių slibinų kovos.....	9
2.	2. Simbolis? Raidė? Skaičius?.....	9
3.	3. Temperatūros konvertavimas.....	10
4.	4. Programuotojų konkursas.....	10
5.	5. Kalėdiniai pirkiniai.....	11
6.	6. Dovanų skaičiavimas.....	11
7.	7. Mikroprocesoriaus funkcija.....	12
8.	8. Skaičių karas.....	12
9.	9. Kriptografija.....	13
3.	ŽINOMO KARTOJIMŲ SKAIČIAUS CIKLAS.....	13
1.	1. Paros energijos poreikis.....	13
2.	2. Keliamieji metai.....	13
3.	3. Poledinė žūklė.....	14
4.	4. Surikiuotas?.....	14
5.	5. Interneto svetainės populiarumas.....	15
6.	6. Atkurtos Lietuvos šimtmečio minėjimas.....	15
4.	NEŽINOMO KARTOJIMŲ SKAIČIAUS CIKLAS.....	16
1.	1. Programavimo būrelis.....	16
2.	2. Laimingieji skaičiai.....	17
3.	3. Optimaliausio duomenų tipo radimas.....	17
4.	4. Skaičiuotuvai.....	18
5.	5. Kvadratai iš degtukų.....	18
6.	6. Saldainių dalybos.....	19
5.	SUMOS, KIEKIO, VIDURKIO SKAIČIAVIMAS TAIKANT ŽINOMO IR NEŽINOMO KARTOJIMŲ SKAIČIAUS CIKLUS.....	19
1.	1. First LEGO League varžybų iššūkis.....	19
2.	2. Jono treniruotės.....	20
3.	3. Žūklės varžybos.....	21
4.	4. Orų stebėjimas.....	21
5.	5. Skęstančiųjų gelbėjimas.....	21
6.	6. Tobulasis skaičius.....	22
7.	7. Supergalingas skaičius.....	22
6.	SKAITYMAS IŠ FAILO, RAŠYMAS Į FAILĄ. CIKLAS CIKLE.....	23
1.	1. Elektronikos elementų ir komponentų rinkinys.....	23
2.	2. Lygiašonės trapecijos.....	23
3.	3. Skaniausi pietūs.....	23
4.	4. Žiogas.....	24
5.	5. Sultis gerti sveika.....	24
6.	6. Varlių koncertas.....	24

7.	FUNKCIJOS, GRAŽINANČIOS APSKAIČIUOTĄ REIKŠMĘ PER FUNKCIJOS VARDĄ .....	25
1.	Kiek yra tokių skaičių? .....	25
2.	Pažymių vidurkiai .....	25
3.	Mokytojos padėjėjas .....	26
4.	Išlaidos .....	27
5.	Smėlio dėžė .....	27
6.	Saldainiai .....	28
7.	Įdomūs pirminiai skaičiai.....	28
8.	Žaidimų kultūros parodos „GameOn“ eksperimentas .....	28
9.	Verslo žurnalo „Forbes“ rekomendacija .....	29
10.	Ilgiausia didėjančių skaičių seka .....	30
8.	FUNKCIJOS, GRAŽINANČIOS APSKAIČIUOTAS REIKŠMES PER PARAMETRUS.....	31
1.	Didžiausio ploto stačiakampiai iš degtukų.....	31
2.	Apranga.....	31
3.	Šachmatų figūrų komplektai .....	31
4.	Liūdna prognozė .....	33
5.	Juodasis penktadienis.....	33
6.	Kilometrai .....	34
7.	Kalėdinis karnavalas .....	35
8.	Mokytojo Galvočiaus užduotis .....	36
9.	Nenaudėlis moksleivis .....	37
10.	Strateginio žaidimo kovos simuliacija.....	37
9.	MASYVAI. SUMOS, KIEKIO, VIDURKIO SKAIČIAVIMAS.....	40
1.	Krepšinio talentas.....	40
2.	Naujametinis pažadas.....	41
3.	Šaldytuvų parduotuvė .....	42
4.	Maistingųjų medžiagų skaičiuoklė.....	42
5.	Trejetukų žaidimas.....	43
6.	Statistinių duomenų analizė .....	44
10.	MASYVAI. DIDŽIAUSIOS IR MAŽIAUSIOS REIKŠMĖS PAIEŠKA .....	45
1.	Dežės.....	45
2.	Karjeras .....	45
3.	Filmai .....	46
4.	Grojaraštis .....	46
5.	Kripto valiutos .....	47
6.	Stovykla .....	48
7.	Slidinėjimas .....	49
8.	Stalo žaidimų parduotuvė.....	50
11.	MASYVO ELEMENTŲ ŠALINIMAS. NAUJŲ ELEMENTŲ ĮTERPIMAS .....	51
1.	Staigna.....	51
2.	Juodųjų skylių spiečiaus virtimas viena skykle.....	51
3.	Iššūkis vienuoliktokams.....	52
4.	Monetų kolekcionieriai .....	53
12.	SIMBOLIAI. SIMBOLIŲ EILUTĖS.....	53
1.	Anagrama.....	53
2.	Stalo žaidimas „Informacijos matavimo vienetai“ .....	54
3.	ATBASH slaptaraštis .....	55
4.	Atblokavimas .....	56
5.	Teksto komentarų analizė .....	56
6.	IP adresai.....	57
7.	Straipsnių pavadinimai.....	57
8.	Pasaka .....	58
9.	Kiti formatai.....	59

13.	STRUKTŪROS, STRUKTŪRŲ MASYVAI .....	60
1.	Rudens dargana .....	60
2.	Krepšinio rungtynės .....	60
3.	Sėkmės žaidimas .....	61
4.	„Talento stipendijų“ konkursas .....	62
5.	Vaistinė .....	63
6.	Studentų maratonas .....	64
7.	Sūrio pjaustymas .....	65
8.	Automobiliai .....	65
14.	STRUKTŪRŲ MASYVAI. DIDŽIAUSIOS IR MAŽIAUSIOS REIKŠMĖS PAIEŠKA .....	66
1.	Automobilių ieškojimo varžybos .....	66
2.	Idėja Lietuvai .....	67
3.	Moksliniai tyrimai ir eksperimentinė plėtra (MTEP).....	68
4.	Kontrolinio darbo rezultatai .....	69
15.	STRUKTŪRŲ MASYVAI. DUOMENŲ ATRANKA, RIKIAVIMAS .....	70
1.	Krepšinio rungtynės .....	70
2.	Geriausias futbolininkas.....	70
3.	Giminės istorija.....	71
4.	Katinių slidinėjimo varžybos .....	72
5.	Pieštukai.....	73
16.	ELEMENTŲ ŠALINIMAS IR ĮTERPIMAS IŠĮ STRUKTŪRŲ MASYVĄ .....	74
1.	Permoka už elektros energiją .....	74
2.	FLL regioninių varžybų apibendrintas protokolai .....	75
3.	Prekių galiojimo laikas .....	75

# 1. TIESINIAI ALGORITMAI

## 1. Gudruolis Tomukas

Tomukas su močiute išėjo grybauti į mišką. Jie pasiėmė švilpukus ir sutarė, kad, jei vienas nuo kito atsiskirs, tai vienas sušvilps kitam. Tas, kuris sušvilpė pirmasis, eis į garso pusę, o kitas grybautojas ramiai lauks. Močiutė yra patyrusi grybautoja, o Tomukui tai bus pirmasis kartas. Bet Tomukas mokosi fiziką, ir žino, kad garso greitis ore priklauso nuo oro temperatūros. Yra viena bėda: Tomukas dar nemoka programuoti, todėl padėkite jam. Sukurkite programą, kuri leistų apskaičiuoti, kurioje vietoje jo lauks močiutė, jei Tomukas pasiklys miške.

Tomukas per fiziką sužinojo, kad garso greitis  $g$  ore priklauso nuo temperatūros ir apskaičiuojamas pagal formulę:

$$g = 331,5 + 0,6T$$

Čia  $T$  – oro temperatūra. Garso greitis matuojamas m/s ir yra realusis skaičius.

Tomukas užfiksuos laiką, kada jis sušvilpė ( $tv$  – valanda,  $tm$  – minutė,  $ts$  – sekundė, visi duomenys sveikąjo tipo) ir laiką, kada išgirdo močiutės švilpimą ( $mv$ ,  $mm$ ,  $ms$ , visi duomenys sveikąjo tipo). Tada, žinodamas oro temperatūrą  $T$  (realusis skaičius), įves jos reikšmę ir programa apskaičiuos, kur jo laukia močiutė. Spręsdami uždavinį laikykite, kad močiutė Tomukui sušvilpė, kai tik išgirdo Tomuko švilpuko garsą.

Pradiniai duomenys įvedami tokia tvarka:  $tv$ ,  $tm$ ,  $ts$ ,  $mv$ ,  $mm$ ,  $ms$ ,  $T$ .

Rezultatas – atstumas, kurį turės nueiti Tomukas iki močiutės, metrais – turi būti pateikiamas vieno skaitmens po kablelio tikslumu.

Pradiniai duomenys	Rezultatai
15 10 15 15 10 17 20	343.5

## 2. Programuotojų sportinio ėjimo varžybos

Viena pašėlusių programuotojų kompanija, paminėdama programuotojų dieną, nusprendė surengti neįprastas sportinio ėjimo varžybas. Kiekvienas varžybų dalyvis nueina  $p$  žingsnių į priekį ir grįžta atbulas  $a$  žingsnių atgal ( $p > a$ ). Taip dalyviai žingsniuoja  $k$  kartų. Parašykite programą, kuri apskaičiuotų:

1) kiek metrų  $m$  ir centimetrų  $c$  nužingsniauvo programuotojas Donatas, jei jo žingsnio ilgis einant į priekį lygus  $pz$ , o einant atgal žingsnio ilgis yra  $az$  centimetrų ( $pz > az$ );

2) koks Donato vidutinis greitis  $v$  m/s, jei distanciją jis įveikė per  $t$  sekundžių.

Pradiniai duomenys yra sveikieji skaičiai. Jie įvedami tokia tvarka:  $p$ ,  $pz$ ,  $a$ ,  $az$ ,  $k$ ,  $t$ .

Rezultatai  $m$  ir  $c$  taip pat sveikieji skaičiai,  $v$  – realiojo tipo. Jie išvedami vienoje eilutėje, atskiriant vieną nuo kito tarpais, tokia tvarka:  $m$ ,  $c$ ,  $v$ .  $v$  reikšmė išvedama dviejų ženklų po kablelio tikslumu.

Pradiniai duomenys	Rezultatai
5 50 3 45 3 5	3 45 2.31

### 3. Skaičių konvertavimas iš dvejetainės į dešimtainę sistemą

Sukurti dvejetainio skaičiaus konvertavimo į dešimtainį skaičių programą.

**Pradiniai duomenys:** vienas nuo kito atskirti tarpeliais įvedami 2 skilčių dvejetainis skaičius **d**, 3 skilčių dvejetainis skaičius **t**, 4 skilčių dvejetainis skaičius **k**, 5 skilčių dvejetainis skaičius **p**.

**Rezultatai:** vienas nuo kito tarpeliais atskirti dešimtainiai skaičiai.

Pradiniai duomenys	10 101 1011 10111
Rezultatai	2 5 11 23

### 4. Rudens gėrybės

Ūkininkas Antanas turi stačiakampį žemės plotą, kurio ilgis **a**, o plotis **b** metrų. Parašykite programą, skaičiuojančią:

a) Kiek hektarų žemės turi šis ūkininkas.

b) Paklauses kaimyno patarimo ūkininkas trečdalį ploto užsodina bulvėmis, dešimtadalį – runkeliais, o likusį plotą – morkomis. Apskaičiuokite, kokį plotą užsodino kiekviena kultūra.

c) Atėjus rudeniui derlius yra nuimamas. Ūkininkas pastebėjo, kad iš vieno aro prikasa **n** kilogramų bulvių, **m** – runkelių ir **k** – morkų. Visą nuimtą derlių Antanas nuveža į supirktuvę. Apskaičiuokite, kiek pelno iš viso gaus ūkininkas jei kilogramas bulvių kainuoja 0.20 €, runkelių 0.13 €, o morkų – 0.16 €.

Duomenys įvedami tokia tvarka: **a, b, n, m, k**.

Rezultatus išvesti į ekraną viena eilute, atskirdami vieną nuo kito tarpeliais dviejų skaičių po kablelio tikslumu.

Pradiniai duomenys	Rezultatai
150 198 120 150 100	2.97 0.99 0.30 1.68 5647.95

### 5. Skanėstai

Marytė turi daug draugų, nes pati yra labai draugiška. Taip pat Marytė turi tradiciją – kiekvieną mėnesį kiekvienam savo draugui padovanoti kažkokį skanėstą. Prieš kelias dienas ji gavo arbatpinigių, todėl sumastė, kad pats laikas nusipirkti skanėstų šiai „šventei“. Kadangi Marytei nelabai sekasi skaičiuoti, nuėjusi į parduotuvę ir pamačiusi kainas Marytė susirūpino, kiek visi skanėstai kainuos. Žinoma, kad Marytė turi **a** kiekį draugų, iš kurių **a – b** yra mergaitės (**a > b**). Marytė nori mergaitėms pirkti tokius skanėstus, kurių vieneto kaina **x** centų, o berniukams – **y** centų (**x > y**). Parašykite programą, skaičiuojančią:

1. Kiek procentų **prc** Marytės draugų sudaro mergaitės.

2. Padėkite Marytei apskaičiuoti, kiek pinigų jai prireiks perkant pasirinktus skanėstus.

Atsakymą **ats** pateikite eurais dviejų skaičių po kablelio tikslumu.

Pradiniai duomenys yra sveikieji skaičiai. Duomenys įvedami tokia tvarka: **a, b, x, y**.

Rezultatai **prc** ir **ats** yra realiojo tipo skaičiai. Jie išvedami vienoje eilutėje, atsikiriant vieną nuo kito tarpu. Rezultatai išvedami tokia tvarka: **prc, ats**. Reikšmės **prc** ir **ats** išvedamos dviejų skaičių po kablelio tikslumu.

Pradiniai duomenys	Rezultatai
7 5 61 23	28.57 2.37

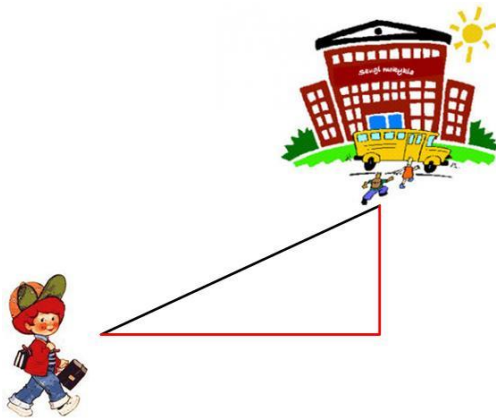
## 6. Siuntos

Tomas planuoja nusipirkti tris prekes iš užsienio. Kiekvienos prekės kaina nurodyta be PVM (21%) ir be bendros siuntimo kainos.

Sukurkite Tomui programą, kuri įvedus trijų prekių kainas ir siuntimo išlaidas apskaičiuotų bendrą kainą (visi duomenys realiojo tipo). Bendrą kainą išveskite dviejų skaitmenų po kablelio tikslumu.

Pradiniai duomenys	Rezultatai
20 20 20 10	82.60
30 20 15 10	88.65
41.9 23.3 18.5 10.3	111.58
32 15 65 4	139.52

## 7. Kelio apskaičiavimas



Petriukas yra pripratęs vaikščioti į mokyklą savo įprastu keliu (raudona spalva pavaizduotas kelias). Kadangi, suprato, kad tai nėra jo trumpiausias kelias, nutarė paskaičiuoti, koks yra atstumas einant trumpesniu keliu.

Mokinio kelionė senuoju keliu prasideda nuo to, kad išėjus iš namų eina tiesiai kelias minutes, vėliau pasuka  $90^\circ$  laipsniu kampu į kairę ir eina dvigubai ilgiau nei prieš tai.

Padėkite apskaičiuoti Petriukui, koks bus naujo kelio ilgis, jei jis eitų tiesesniu keliu.

Pradiniai duomenys: kelio ilgis iki posūkio.

Rezultatas: naujojo kelio ilgis, pateikiamas keturių ženklų po kablelio tikslumu.

Pradiniai duomenys	Rezultatai
5	11.1803
7	15.6525
10	22.3607

## 8. Pirato lobiai

Bobas vidutiniškai per metus užsidirba  $n$  auksinių. 65% jų išleidžia pragyvenimui, o likusius slepia skryniuose, į kurias telpa po  $z$  auksinių. Kai skrynja prisipildo, Bobas ją užkasa.

1. Suskaičiuokite  $m$ , kiek vidutiniškai per mėnesį Bobas išleidžia pragyvenimui.
2. Kiek skrynių  $s$  Bobas paslėpė per  $x$  metų? (neužpildytų skrynių jis neužkasa)

Pradiniai duomenys yra sveikieji skaičiai. Jie įvedami tokia tvarka:  $n, z, x$ .

Rezultatai  $m$  – realiojo tipo,  $s$  – sveikasis skaičius. Jie išvedami vienoje eilutėje, atskiriant vieną nuo kito tarpais, tokia tvarka:  $m, v, m$  reikšmė išvedama dviejų ženklų po kablelio tikslumu.

Pradiniai duomenys	Rezultatai
2800 500 5	151.67 9

## 9. Minecraft namas



Padėkite Tadei apskaičiuoti, kiek žaliavų reikės pastatyti namą Minecraft žaidime. Namo struktūra gan paprasta: grindys ir lubos iš medžio lentų, o sienos iš akmens, stačiakampio gretasienio forma. Namo ilgis bus  $x$ , plotis  $y$ , o aukštis –  $z$ . Aukštis  $z$  yra gyvenamosios ervės aukštis, neskaitant lubų ir grindų. Namas taip pat turės  $m$  durų ir  $n$  langų. Vienos durys užima du blokus sienoje, o vienas langas vieną bloką.

4 medžio lentoms pagaminti reikia 1 rąsto. 1 durims pagaminti reikia 6 medžio lentų. 1

stiklo bloką gauname iš 1 smėlio bloko.

Kad pastatytų namą Tadas turės iškasti atitinkamą kiekį akmens, smėlio, bei prisikirsti rąstų.

Parašykite programą, kuri apskaičiuotų, kiek Tadei reikės prikasti akmens, smėlio ir kiek prikirsti rąstų. Taip pat apskaičiuokite, kiek laiko Tadei reikės, kad gautų visus šiuos dalykus. Laikai, per kiek Tadas iškasa vieną akmens bloką (**a**), vieną smėlio bloką (**b**) ir nukerta vieną rąstą (**c**) įvedami klaviatūra. Akmenų, smėlio ir rąstų kiekius išveskite į ekraną. Laiką išveskite sekundėmis.

Jei žaliavų reikės tik dalies, tuomet suapvalinkite į didžiąją pusę (funkcija **ceil**). Pvz., vienoms durims reikia 6 lentų, kad gauti 6 lentas reikia pusantro rąsto, bet Tadas vis tiek turės nukirsti du rąstus. Nepanaudotas durims lentas Tadas galės panaudoti stogui arba grindims, arba tiesiog pasilikti vėlesniam laikui.

Pradiniai duomenys įvedami tokia tvarka:  $x y z m n a b c$

Rezultatai išvedami tokia tvarka: akmuo rąstai smėlis laikas

Visos reikšmės, tiek įvestyje, tiek išvestyje atskiriamos vienu tarpu.

Pradiniai duomenys	Rezultatai
3 3 3 1 2 1 1 1	20 6 2 28
8 5 3 2 7 3 2 1	55 23 7 202

## 10. Programuotojų iššūkiai

Išspręskite programuotojų „+“ „!“ mestus iššūkius:

- „+“ iššūkis: „sukeisti dviejų natūraliųjų skaičių  $x$  ir  $y$  reikšmes naudojant tik tuos du kintamuosius“.
- „!“ iššūkis: „sukurti tokį realųjį skaičių  $m$ , kurio sveikoji dalis būtų skaičius  $x$ , o liekana skaičius  $y$ “.

Pradiniai duomenys yra natūralieji skaičiai. Jie įvedami tokia tvarka:  $x, y$ .

Rezultatai  $v$  – realiojo tipo ir  $x, y$ . Jie išvedami vienoje eilutėje, atskiriant vieną nuo kito tarpais, tokia tvarka:  $x, y, v$  reikšmė išvedama dviejų ženklų po kablelio tikslumu.

Pradiniai duomenys	Rezultatai
5 2	2 5 5.02
11 99	99 11 11.99
69 96	96 69 69.96



## 2. ŠAKOTIEJI ALGORITMAI

### 1. Triliemenių slibinų kovos

Pasakose slibinas dažniausiai vieną liemenį ir daug galvų, o Vytukas sugalvojo sukurti žaidimą, kuriame kovoja du triliemeniai slibinai. Vytukas kurs žaidimo dizainą ir ieško programuotojo, kuris suprogramuotų slibinų kovos baigtį. Vytuko sumanymas toks. Slibinai turi po 3 liemenis: kairinį, vidurinį ir dešinį. Kiekvienas liemuo turi tam tikrą galvų skaičių (ant vieno liemens ne daugiau kaip 1000). Kai slibinai pradeda kovoti, pirmiausia kovoja abiejų slibinų kairiojo liemens galvos. Jei abiejų slibinų kairiojo liemens galvų skaičiai yra vienodi, tuomet kova baigiasi lygiosiomis. Jei ne – į kovą įsitraukia viduriniojo liemens galvos ir kovoja pagal tas pačias taisykles, kaip ir kairiojo liemens galvos. Jei viduriniojo liemens galvų skaičius vienodas, tuomet kova irgi baigiasi lygiosiomis, jei ne – į kovą įsitraukia dešiniojo liemens galvos. Jei dešiniųjų galvų skaičius yra vienodas, tuomet abiejų slibinų jėgos lygios. Jei ne – laimi tas slibinas, kuris turi daugiau dešiniojo liemens galvų.

Parašykite programą, kuri nustatytų slibiną laimėtoją.

Pradiniai duomenys įvedami klaviatūra tokia tvarka: pirmojo slibino kairiojo, viduriniojo ir dešiniojo liemens galvos, po to antrojo slibino kairiojo, dešiniojo ir viduriniojo liemens galvos.

Rezultatai išvedami į ekraną, nurodant slibiną laimėtoją. Jei laimėjo pirmasis slibinas turi būti išvedamas 1, jei antrasis turi būti išvedamas 2, jei abu – turi būti išvedama 1 2.

Pradiniai duomenys	Rezultatai
3 5 7 2 5 8	1

### 2. Simbolis? Raidė? Skaičius?

Į programą yra įvedamas simbolis. Reikia patikrinti, ar įvestas simbolis yra lotyniškos abėcėlės raidė, ar skaitmuo, ar kitas simbolis.

Rezultatai atitinkamai turi būti: "skaitmuo", "raide", "kitas simbolis".

**Iššūkis:** atlikti užduotį sąlyginiame sakinyje netikrinant kiekvienos lotyniškos raidės ir skaitmenų atskirai.

Papildomi simboliai įvedami laikant paspaustą mygtuką ALT ir rašant skaičius. Pvz.: ALT+43

Kodas	Simbolis
251	√
174	«
175	»
168	ı
236	∞
241	±

Įvedamas simbolis	Rezultatas
!	kitas simbolis
-	kitas simbolis
Æ	kitas simbolis
∞	kitas simbolis
2	skaitmuo
8	skaitmuo
i	raide
N	raide

### 3. Temperatūros konvertavimas

Mokslininkai tirdami cheminius ryšius, ar bandydami surasti naujas medžiagas, fiksuoja temperatūrą.

Šios užduoties tikslas – išmokti konvertuoti temperatūras iš vienos skalės į kitą. Toks konvertavimas plačiai taikomas fizikoje ir chemijoje.

Įvedus (1) temperatūros skalę (C – Celsijaus skalė, K – Kelvino skalė, F – Farenheito skalė), (2) skaičių, nurodantį, kokią tos skalės temperatūrą reikia konvertuoti, ir (3) tipą, į kurį norime konvertuoti, turi būti atliekamas konvertavimas ir išvedamas rezultatas.

Atkreipkite dėmesį į tai, kad skalės įvedamos didžiosiomis raidėmis ir jei nurodyta, kad skalė, iš kurios konvertuojama, ir skalė, į kurią konvertuojama sutampa, tuomet konvertuoti nereikia.

Geriausia skaičiuoti be apvalinimų, matematiškai, nors atliekant realius eksperimentus išmatuotos ir konvertuotos į kitą skalę temperatūros turi būti pateikiamos vienodu tikslumu.

Pradiniai duomenys	Rezultatai
K 290.8 C	17.65
K 270.1 F	26.51
C -90.4 K	182.75
C 56.4 F	133.52
F 290.8 K	416.928
F -5.0 K	252.594
F 0.1 C	-17.7222

### 4. Programuotojų konkursas

Programuotojų konkurse reikia išspręsti penkias užduotis.

Konkurso taisyklės yra sudėtingos ir nebūtinai sąžiningos:

Pirmas uždavinys vertas 10 taškų, antras uždavinys – 20 taškų trečias uždavinys – 30 taškų, ketvirtas uždavinys – 40 taškų, penktas uždavinys – 50 taškų.

Jei dalyvis išsprendžia visus uždavinius, papildomai gauna 50 taškų. Tai iš viso sudarys 200 taškų.

Jei dalyvis išsprendžia pirmą ir paskutinį uždavinius, gauna papildomai 20 taškų. Tai iš viso sudarys 80 taškų.

Jei dalyvis išsprendžia trečią ir ketvirtą uždavinius, bet neišsprendžia pirmo arba antro, tuomet gauna tik 50 taškų.

Jei dalyvis neišsprendžia pirmo, antro ir trečio uždavinio, tuomet už likusius du uždavinius gauna 50 procentų mažiau taškų. Visais kitais atvejais dalyvis gauna 0 taškų.

Ar atlikta užduotis, ar ne rodo skaičiukai: 1–reiškia, kad uždavinys atliktas, 0 – uždavinys neatliktas ir vertinamas 0 taškų.

Programai pateikiami dviejų programuotojų duomenys, kuriuos uždavinius išsprendė, kurių nesugebėjo atlikti.

Reikia nustatyti, kuris programuotojas laimėjo, pirmas ar antras. Jei abu programuotojai surinko vienodai taškų, tuomet skelbiamos lygiosios.

Pradiniai duomenys	Rezultatai
1 1 1 1 1 1 0 0 1 0	laimejo pirmas programuotojas surinkes 200 tasku antro taskai: 0
1 0 0 0 1 1 0 1 1 0	laimejo pirmas programuotojas surinkes 80 tasku antro taskai: 60
1 1 0 0 1 0 1 1 1 0	antras programuotojas surinkes 70 tasku pirmo taskai: 0

## 5. Kalėdiniai pirkiniai

Artėjant didžiosioms metų šventėms Petriukas nusprendė nudžiuginti savo šeimą, nupirkdamas jiems po dovaną. Pirkinių krepšeliui šiais metais jis žada skirti  $n$  eurų. Todėl apsilankė elektroninėje parduotuvėje, norėdamas sužinoti, ar jam užteks pinigų. Ieškodamas optimaliausio sprendimo, jis išvydo, kad norimos prekės kainuoja **a**, **b** ir **c** eurų (kainos yra sveikieji skaičiai). Parašykite programą, kuri:

a) patikrintų, ar Petriukui užtektų pinigų, jei jis pirktų kiekvienos prekės po vieną vienetą;  
b) jei Petriukas pamato, kad perkant tik tris dovanas jam dar lieka pinigų, jis nusprendžia nudžiuginti ir savo draugus. Tuomet patikrina kiekvienos prekės kainą dar kartą ir jeigu pirmosios prekės kaina yra mažesnė už 10 eurų ir dalijasi iš trijų Petriukas šių prekių nusiperka papildomai tris. Jei antrosios prekės kaina dalijasi iš 2 ir 5 be liekanos, jis nusiperka papildomai dvi tas prekes. Trečiąją prekę perka vieną, kaip ir prieš tai.

Duomenys įvedami tokia tvarka: **n a b c**.

Į ekraną išveskite „Taip“ – jei pinigų užteko perkant po vieną prekę, arba „Ne“ – jei pinigų pritrūko.

Taip pat patikrinkite, ar padidinus pirkinių krepšelį Petriukui užteks pinigų. Jeigu biudžetas yra pakankamai didelis, išveskite pinigų likutį formatu "liko: x" (x – likusių pinigų suma), o jei pinigų pritrūko – išveskite papildyto krepšelio kainą.

Pradiniai duomenys	Rezultatai
50 1 3 6	Taip liko: 40

## 6. Dovanų skaičiavimas

Kalėdų Senelis atsižvelgs į Nojaus gerus ir blogus darbus ir duos jam tam tikrą skaičių dovanų. Nojus šiemet padarė X gerų ir Y blogų darbų. Vienas blogas darbas anuliuoja du gerus darbus. Kalėdų Senelis Nojui duos dovanų atitinkamai pagal tai, kiek Nojus padarė gerų darbų:

1-3 geri darbai – 1 dovana;

3-5 geri darbai – 2 dovanos;

5-10 gerų darbų – 3 dovanos;

10 ir daugiau gerų darbų – 4 dovanos.

Jei Nojus gerų darbų nepadarė, arba padarė gerų darbų dvigubai daugiau, nei blogų, jis dovanų iš Kalėdų Senelio negaus. Tačiau Nojaus tėvai ir seneliai jį labai myli, tai bet koku atveju padovanos jam po dovaną. Nojus gaus dovaną iš savo tėvų ir po dovaną iš senelių, bei dovanas iš Kalėdų Senelio. Padėkite suskaičiuoti, kiek dovanų iš viso gaus Nojus.

Nykštukai sugalvojo papokštauti ir vietomis neesant gerų ar blogų darbų vietoje nilių parašė minusinį skaičių. Esant minusiniam darbų kiekiui, laikykite jį 0.

Pradiniai duomenys įvedami klaviatūra: **x** ir **y**, Nojaus atitinkamai gerų ir blogų darbų skaičius.

Rezultatas – dovanų skaičius – išvedamas į ekraną.

Duomenų pavyzdys:

Pradiniai duomenys	Rezultatai
5 2	4
0 0	3
10 0	7

## 7. Mikroprocesoriaus funkcija

Kuriant naują mikroprocesorių, komanda sugalvojo pridėti naują funkciją, kuri galėtų konvertuoti bet kurio 8 bitų registro reikšmę į kitą skaičiavimo sistemą.

Komanda pasirinko parašyti funkciją, konvertuojančią 8 bitų registre saugomą dvejetainį kodą į dešimtainę sistemą.

Pradiniai duomenys bus bet kurio 8 bitų registro duomenys, tai yra bitai b7, b6, b5, b4, b3, b2, b1, b0. b7 bitas yra ženklo skiltis, kuri nurodo kokio ženklo skaičius saugojamas registre. Jeigu b7 lygus 0 tai reiškia kad registre saugomas teigiamas skaičius **tiesioginiame** kode. Jeigu b7 bitas lygus 1, tai registre saugoma neigiama reikšmė ir ji saugoma **papildomame** kode.

Dvejetainio kodo konvertavimas į dešimtainį vyks pagal tokį algoritmą:

1) jei reikia, registro duomenys iš papildomo kodo paverčiami į tiesioginį kodą. Bitai b6, b5, b4, b3, b2, b1 ir b0 invertuojami (jei b6 buvo 1, invertavę jį gausime 0 ir su kitais bitais tas pats mechanizmas). Gauti bitai padauginami iš reikšmės, kurią saugo:  $b6 * 64$ ,  $b5 * 32$ ,  $b4 * 16$ ,  $b3 * 8$ ,  $b2 * 4$ ,  $b1 * 2$ ,  $b0 * 1$ . Gautos reikšmės sudedamos ir prie galutinės reikšmės pridodamas 1;

2) tiesioginio kodo bitai padauginami iš reikšmės, kurią saugo. Gautos reikšmės sudedamos.

Rezultatas – dešimtainis skaičius (int tipo).

Pradiniai duomenys	Rezultatai
1 1 1 0 1 1 0 0	-20
0 1 1 1 1 0 0 1	121
1 0 1 0 0 0 1 0	-94

## 8. Skaičių karas

Skaičių šalyje gyvena vienetai ir nuliai. Kartą vienetai susipyko su nuliais ir norėdami įrodyti, kad yra geresni už nulius, nusprendė nesutarimus išspręsti kariaudami. Žinoma, kad iš viso, sudėjus nulių ir vienetų kariuomenes, į karą buvo pasiūstas **a** kiekis kareivių. Kadangi nulių populiacija kiek didesnė nei vienetų, todėl jie į karą išsiuntė daugiau karių nei vienetai, tačiau visiems yra žinoma, jog vienetų kariai yra 2 kartus stipresni už nulių. Kariuomenėms susitikus mūšio lauke, tapo žinoma, kiek karių sudaro nulių kariuomenę, šis skaičius lygus **nul** ( $nul > 0$ ).

Parašykite programą, skaičiuojančią:

1. Kokia yra visos vienetų kariuomenės karinė galia **vGalia**, kai yra žinoma, jog vieno nulių kario galia yra **b**.
2. Padėkite pasauliui sužinoti, kas laimės šį karą – "NULIAI", "VIENETAI" ar "NIEKAS". Raskite rezultatą **rez**.

Pradiniai duomenys **a**, **b** ir **nul** yra sveikieji skaičiai. Duomenys įvedami tokia tvarka: **a**, **b**, **nul**.

Rezultatas **vGalia** yra sveikasis skaičius, o **rez** – string tipo tekstas. Jie išvedami vienoje eilutėje, atsikiriant vieną nuo kito tarpu. Atsakymas **rez** priklausomai nuo to, kas laimės karą yra lygus "NULIAI", "VIENETAI" arba "NIEKAS". Rezultatai išvedami tokia tvarka: **vGalia**, **rez**.

Pradiniai duomenys	Rezultatai
8 2 5	12 VIENETAI

## 9. Kriptografija

Programuotojas užkodavo dviejų raidžių **r1**, **r2** kodą. Tos dvi raidės gali būti "X", "Y" arba "Z". "X" kodas yra 0, "Y" – 1 0, "Z" – 1 1.

Įvedant duomenis pirma užrašomas pirmosios raidės **r1** kodas, tada antros **r2** (atkreipkite dėmesį, kad įvedamų skaičių kiekis gali kisti priklausomai nuo raidžių).

Išveskite atkurtą kodą.

r1, r2 – char tipo kintamieji, jie išvedami vienas po kito.

Pradiniai duomenys	Rezultatai
0 1 0	XY

## 3. ŽINOMO KARTOJIMŲ SKAIČIAUS CIKLAS

### 1. Paros energijos poreikis

Paros energijos poreikis (PEP) apskaičiuojamas padauginus pagrindinę energijos apykaitą (PEA) iš fizinio aktyvumo koeficiento (K).

$$PEP = PEA \times K$$

Paros energijos apykaitos PEA formulės:

- PEA (vyrams) =  $66,47 + 13,75 \times (\text{kūno svoris}) + 5,00 \times (\text{ūgis}) - 6,76 \times (\text{amžius})$
- PEA (moterims) =  $655,10 + 9,46 \times (\text{kūno svoris}) + 1,85 \times (\text{ūgis}) - 4,68 \times (\text{amžius})$

Fizinio aktyvumo koeficientas (K):

Fizinio aktyvumo grupė --Atliekamas fizinis darbas -- Fizinio aktyvumo koeficientas (K)

1 --Labai lengvas darbas --1,2

2 --Lengvas darbas --1,3

3 --Vidutinio sunkumo darbas --1,5

4 --Sunkus darbas --2,0

5 --Labai sunkus darbas --2,7

Pavyzdžiui, 40 metų moteriai, kurios ūgis 170 cm, kuri sveria 70 kg ir kuri dirba lengvą fizinį darbą, per dieną reikia suvartoti:

$$(655,10 + 9,46 \times 70 + 1,85 \times 170 - 4,68 \times 40) \times 1,3 = 1880 \text{ kcal.}$$

Parašykite programą, kuri apskaičiuotų energijos poreikį pagal įvedamus duomenis. Pirmiausia programoje reikia įvesti žmonių kiekį **n**, vėliau žmogaus amžių **a**, fizinio aktyvumo koeficientą **k**: svorį **s**, lytį **l** ir ūgį **u**. Programa turėdama šiuos duomenis pagal pateiktas duomenis suskaičiuos energijos poreikį **pep**, jei žmonių kiekis didesnis už 1 tai atspausdinus vieno asmens duomenis leis įvesti naujus duomenis, priešingu atveju programa baigs darbą. Atspausdinti reikia tik skaičių kilokalorijų. Duomenys ir rezultatai turi būti rašomi taip pat kaip pavyzdyje, kitu atveju nepraeis visų testų.

Pradiniai duomenys	1 40 1,3 70,0 moteris 170
Rezultatai	1880

### 2. Keliamieji metai

Parašykite programą, kuri išvestų į ekraną visus keliamuosius metus nuo **n** iki **m** (imtinai) remiantis **Grigaliaus kalendoriumi**.

- **Julijaus kalendoriuje** keliamieji metai yra tie, kurie be liekanos dalijasi iš 4, pavyzdžiui 1504, 1500, 1600.

- **Grigaliaus kalendoriuje** keliamieji metai skaičiuojami kaip Julijaus kalendoriuje, tačiau paskutiniai amžiaus metai (dalijasi iš 100 be liekanos) yra keliamieji tik tada, jei be liekanos dalijasi iš 400, pavyzdžiui 1504, 1600.

Pradiniai duomenys	Rezultatai
1596 1608	1596 1600 1604 1608
1997 2018	2000 2004 2008 2012 2016
1775 1801	1776 1780 1784 1788 1792

### 3. Poledinė žūklė

Tradiciskai poledinės žūklės konkurse dalyvauja ir Saulius. Užsiregistravęs į konkursą jis sužinojo apie teikiamus prizus.

Šiais metais, renginio organizatoriai nusprendė apdovanojimus dalinti pagal surinktų taškų skaičių.

Prizų sąrašas:

1. Žvejo kuprinę galima gauti už 25 ir daugiau taškų.
2. Žūklės priemonių rinkinį galima gauti surinkus nuo 20 iki 25 taškų. (Surinkus 25 taškus gaunama žvejo kuprinė).
3. Masalų rinkinį galima gauti surinkus nuo 15 iki 20 taškų. (Surinkus 20 taškų gaunamas žūklės priemonių rinkinys).

P.s. Surinkus mažiau nei 15 taškų, prizas nėra suteikiamas.

Taškai yra skiriami taip:

1. Jeigu žvejo pagauta žuvis yra lengvesnė nei kilogramas – jis ją paleidžia ir už tai nėra skiriami taškai.
2. Jeigu žvejo pagauta žuvis sveria nuo 1 ir 2 kilogramų, žvejys gauna 2 taškus.
3. Jeigu žvejo pagauta žuvis sveria nuo 2 ir 3 kilogramų, žvejys gauna 3 taškus.
4. Jeigu žvejo pagauta žuvis yra sunkesnė nei 3 kilogramai, žvejys gauna 5 taškus.

Užduotis:

Parašykite programą, kuri apskaičiuotų, kokį prizą gavo Saulius.

Pirmiausia klaviatūra yra įvedamas Sauliaus pagautų žuvų kiekis  $n$ . Tolimesni  $n$  skaičiai yra kiekvienos žuvies svoris gramais.

Į ekraną išvedamas tik prizo numeris (1 2 3) arba "Prizo negavo".

Pradiniai duomenys	Rezultatai
6 5000 4500 4321 400 800 1500	3

### 4. Surikiuotas?

Jūsų užduotis yra patikrinti, ar įvesti skaičiai yra surašyti didėjimo tvarka. Taip pat sąrašas gali būti nesurikiuotas. Jei dviejų elementų reikšmės yra vienodos, tai rikiuotame sąraše jos būtų rašomos viena šalia kitos.

Pirmas įvestas skaičius  $n$  rodo, kiek duomenų bus įvesta.

Po to iš eilės klaviatūra įvedami duomenys (sveikieji skaičiai).

Įvedus visus duomenis į ekraną išvedamas atsakymas: jei sąrašas surikiuotas didėjimo tvarka, išvedamas žodis "surikiuotas", kitu atveju išvedama "nesurikiuotas".

Patarimas: sąrašo elementams saugoti naudokite du kintamuosius.

Pradiniai duomenys	Rezultatai
5 1 4 4 7 9	surikiuotas

## Reikalavimai

- Nenaudoti masyvų arba bool tipo kintamųjų.

### 5. Interneto svetainės populiarumas

Antanas susikūrė interneto svetainę, įsidedė `hey.lt` skaitiklį ir seka svetainės populiarumą. Svetainėje `hey.lt` pateikiama informacija, kiek svetainės puslapių buvo aplankyta ir keli unikalūs lankytojai apsilankė svetainėje kiekvieną dieną.

Antanas nori įsitikinti, ar jo svetainės populiarumas nemažėja, t.y., ar kiekvieną dieną didėja vieno lankytojo atverstų puslapių skaičius, kuris gaunamas visų tą dieną aplankyto puslapių skaičių dalinant iš svetainės lankytojų skaičiaus. Gautas dalmuo yra realusis skaičius.

Pradiniai duomenys įvedami klaviatūra. Pirmiausia įvedamas stebėjimo dienų skaičius  $d$  (dienos numeruojamos nuo 1 iki  $d$ ,  $1 < d \leq 50$ ). Toliau įvedama  $d$  duomenų rinkinių. Rinkinį sudaro tą dieną iš viso atvertų puslapių skaičius ir unikalių lankytojų skaičius.

Parašykite programą, kuri patikrintų, ar svetainės populiarumas nemažėja. Svetainės populiarumas nemažėja tuomet, jei kiekvieną kitą dieną vieno lankytojo atverstų puslapių skaičius yra ne mažesnis už prieš tai buvusią dieną. Tokiu atveju į ekraną turi būti išvedamas žodis NE. Jei atsiranda diena, kai svetainės populiarumas tampa mažesnis, negu prieš tai buvusią dieną, tuomet turi būti išvedamas dienos, kai svetainės lankomumo populiarumas sumažėja, numeris ir toliau nebetikrinama.

Pradiniai duomenys	Rezultatai
3 10 3 11 3 12 3	NE
3 10 3 11 3 12 4	3

## Reikalavimai

- Nenaudokite masyvų.

### 6. Atkurtos Lietuvos šimtmečio minėjimas

2018 m. – atkurtos Lietuvos šimtmečio metai. Robotikos būrelio nariai sugalvojo projektą, skirtą šiai progai.

Vaikai sugalvojo sukonstruoti  $n$ -kampę aikštę ( $2 < n < 10$ ) su  $n$  arkinių vartų (arkiniais vadinami vartai, kurių viršutinė dalis aprašoma parabole  $y = ax^2 + b$ ), išdėstyto ant kiekvienos aikštės kraštinės, kurie sudaro įvažiavimus į aikštę. Pro kiekvienus arkinis vartus įvažiuos po vieną robotą, ant kurių bus įtvirtintos Lietuvos valstybinės vėliavos.

Vaikai pasiskirstė į 2 komandas: konstruktorių ir programuotojų. Konstruktoriai pirmiausia sukstravo  $n$  ( $n > 2$ ) robotų, kurių pločiai ir aukščiai skiriasi. Ant kiekvieno roboto konstruktoriai įtvirtino po vėliavėlę. Vėliavėlių stiebo aukščiai yra vienodi.

Programuotojų uždavinys – apskaičiuoti, kokie turi būti parabolės koeficientai  $a$  ir  $b$ , kad per sukonstruotus arkinis vartus įvažiuotų pačių didžiausių matmenų robotas. Visi matmenys yra sveikieji skaičiai, išreikšti centimetrais. Taip pat apskaičiuoti, koks turi būti vartų pagrindinės dalies (iki arkos) plotis ir aukštis.

Pagal programuotojų skaičiavimų rezultatus konstruktoriai sukstruos arkinis vartus.

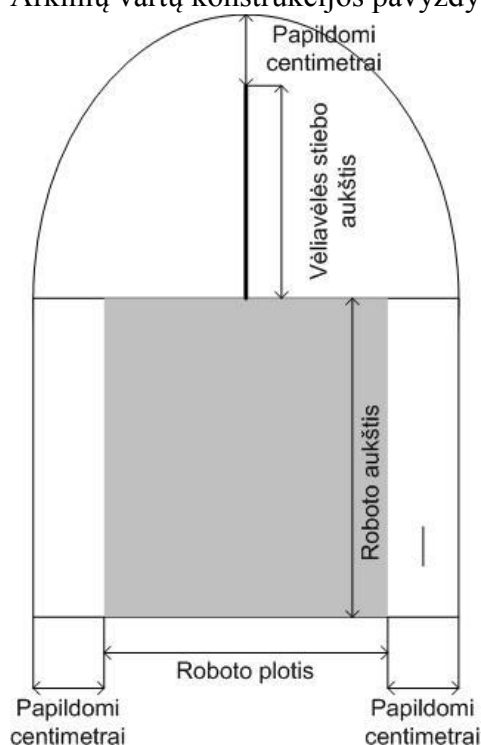
Pradiniai duomenys įvedami klaviatūra tokia tvarka: sukstruotų robotų skaičius  $n$ , vėliavėlės stiebo aukštis, kiek papildomų centimetrų  $pc$  konstruojant arkinis vartus palikti, kad robotas

galėtų įvažiuoti ir neužkliūti (paliekama iš abiejų pusių po pc ir iš viršaus pc). Toliau įvedama n sveikųjų skaičių dvejetų – kiekvieno roboto plotis ir aukštis be vėliavėlės.

Parašykite programą, kuri apskaičiuotų ir į ekraną išvestų pagrindinės vartų dalies plotį ir aukštį (sveikieji skaičiai), bei parabolės koeficientus a ir b (realieji skaičiai) ir juos išvestų šimtųjų tikslumu. Išvedami rezultatai vienas nuo kito atskiriami tarpo simboliu.

Pradiniai duomenys	Rezultatai
3 15 5 12 20 17 15 15 20	27 20 -1.48 20.00

Arkinių vartų konstrukcijos pavyzdys.



### Reikalavimai

- Nenaudokite masyvų.

## 4. NEŽINOMO KARTOJIMŲ SKAIČIAUS CIKLAS

### 1. Programavimo būrelis

Kasparas pernai lankė programavimo būrelį ir vedė lankomumo statistiką. Atsižvelgdami į pernai metų lankymo statistiką, padėkite Kasparui nuspręsti kurį planą labiausiai verta pasirinkti. Programavimo būrelio nariai moka skirtingą mokestį, priklausantį nuo to, kelias dienas per savaitę lanko būrelį. Padėkite Kasparui pasirinkti labiausiai tinkantį mokėjimo planą.

Kiekvienas papildomas užsiėmimas kainuoja 5 €.



Planas	Dienų per savaitę	Kaina per mėn.
P1	2	25 €
P2	3	37 €
P3	4	45 €
P4	5	50 €

Duomenys įvedami tokiu formatu:

Pirmoje eilutėje nurodomas planas, kurį Kasparas buvo pasirinkęs pernai metais, pvz P3.

Antroje eilutėje nurodomas Kasparo lankomumas (9 mėn.), pvz BNB NBSSBPBNB..., čia **B** reiškia buvo būrelyje, **N** - nebuvo, **S** - savaitgalis, **P** buvo papildomam užsiėmimui, už kurį mokėjo 5 €.

Išveskite kiek iš viso Kasparas mokėjo už programavimo būrelį (tiek už planą, tiek už papildomą lankymą), kiek mokėjo už papildomą lankymą, ir kokį planą Kasparui labiausiai verta pasirinkti.

Pavyzdžiai testavimui:

Įvestis	Išvestis
P2 NBSSBPBNBSSBNBPSSBPBNBSSBPBNBSSBPBPSSBNBNBSSBNBNBSSBPBNBSSBNBNB SSBNBNBSSBNBNBSSBNBNBSSBNBNBSSBNBPSSBPBNBSSBNBNBSSBNBNBSSBNBPBS SBNBPSSBNBPSSBNBNBSSBNBNBSSBNBPSSBNBNBSSBNBNBSSBNBNBSSBNBPBS BNBNBSSBNBNBSSBNBNBSSBNBNBSSBNBNBSSBNBNBSSBNBNBSSBNBNBSSBNBNBSS BNBNBSSBNBPSSBNB	P3 408 75
P2 NNSSBNBPSSBNBNSSBNBNSSBNBNSSBNBNSSBNBNSSBNBNSSBNBNSSBNBNSSBNBN NPSSBNBNSSBNBNPSSBNBNSSBNBNSSBNBNSSBNBNSSBNBNSSBNBNSSBNBNSSBNBN NNSSBNBNSSBNBNSSBNBNSSBNBNSSBNBNSSBNBNSSBNBNSSBNBNSSBNBNSSBN BNNSSBNBNSSBNBNSSBNBNSSBNBNSSBNBNSSBNBNSSBNBNSSBNBNSSBNBNSSBN NBNNSSBNBNSSBNBNSSBNBN	P1 348 15

## 2. Laimingieji skaičiai

Moksleivis Antanukas nori dalyvauti loterijoje ir laimėti pagrindinį prizą. Kad galėtų tai padaryti, jam reikia atspėti 4 laimingus skaičius  $a_1, a_2, a_3, a_4$ . Loterijos skaičių intervalas nuo 0 iki 100. Kad būtų užtikrintas savo sėkme, Antanukas kreipėsi į būrėją, kuri jam išpranašavo tuos 4 laimingus skaičius, tačiau jos pranašystė buvo miglota ir Antanukas gavo tik užuominą, už kelių skaitmenų pasikartojimų, pradedant skaičiuoti nuo nulio, yra jo laimingieji skaičiai. Padėkite Antanukui sužinoti laiminguosius skaičius

Pradiniai duomenys:

Natūralieji skaičiai  $a_1, b_1, a_2, b_2, a_3, b_3, a_4, b_4$ , kur  $a_1, a_2, a_3, a_4$  pasikartojantieji skaitmenys, o  $b_1, b_2, b_3, b_4$  skaitmenų pasikartojimo kiekis. Pvz.: skaičius 11 turi du vienetus taigi tariame, kad vienetas pasikartoja du kartus. Tarkime  $a_1$  ir  $b_1$  yra 1 ir 10, tai rezultato pirmasis skaičius bus 17 (nes **1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17**)

Rezultatai pateikiami vienoje eilutėje, juos atskiriant tarpo simboliu.

Pradiniai duomenys	Rezultatai
1 10 0 1 2 3 9 5	17 10 20 49

## 3. Optimaliausio duomenų tipo radimas

Kuriant programas, kurios apdoroja didelius natūraliuosius skaičius, svarbu optimizuoti kiekvieną programos dalį. Pavyzdžiui, surasti optimaliausią duomenų saugojimo tipą. Tai galima

atlikti tokiu būdu: surandamas didžiausias duomenų narys ir patikrinama, kurį tipą geriausia naudoti. Nors šitas būdas laiko nesutaupo, tačiau sutaupo atminties resursų.

Parašykite programą, kuri surastų, kiek didžiausias **natūrinis** skaičius **a** užims bitų **n** ir pasirinkti optimaliausią duomenų tipą. Visus gautus rezultatus išspausdinti ekrane.

Pagrindiniai sveikųjų skaičių tipai:

1. **short int**, kuris saugo nuo  $-32768 (-2^{15})$  iki  $32767 (2^{15} - 1)$  ir atmintyje užima **du baitus** arba **šešiolika bitų** (16 bitas skirtas skaičiaus ženklui nurodyti, o 15 bitų yra skirti saugoti pačiam skaičiui).
2. **long int**, kuris saugo nuo  $-2147483648 (-2^{31})$  iki  $2147483647 (2^{31} - 1)$  ir atmintyje užima **keturis baitus** (arba 32 bitus).
3. **long long int**, kuris saugo nuo  $-2^{63}$  iki  $2^{63} - 1$  ir atmintyje užima **aštuonis baitus** (arba 64 bitus).

Norint parinkti optimaliausią duomenų tipą reikia:

1. Surasti, kiek **natūrinis** skaičius **a** užims **bitų n**. Patarimas **n** skaičiavimui - dvejetainis laipsnis ( $2^m$ ) **m** sutampa su bitų skaičiumi **n** (žiūrint į teigiamus skaičius reikia atimti 1, tai yra  $2^m - 1$ ). Arba kitais žodžiais surasti mažiausią skaičių  $2^m$ , kuris būtų didesnis už **a** ( $2^m > a$ ) ir **m = n**.
2. Pagal **n** reikšmę sprendžiama, kuris duomenų tipas yra optimaliausias, tai yra mažiausiai užimantis duomenų tipas į kurį telpa skaičius **a**.

Pradiniai duomenys	Rezultatai
65	7 short int
32768	16 long int
32767	15 short int
2147483700	32 long long int

### Reikalavimai

- Skaičiavimų rezultatus išvesti į ekraną be papildomų žodžių arba simbolių.
- Skaičiavimams panaudoti nežinomo kartojimų skaičiaus ciklą **while**.
- **a** > 0, a - long long int tipo.
- **n** > 0.

## 4. Skaičiuotuvai

Šiuolaikiniai skaičiuotuvai naudojami tiek kasdiniame gyvenime, tiek sprendžiant sunkias matematinės problemas. O kodėl nepamėginus patiems sukurti paprastą skaičiuotuvą.

Skaičiuotuvai gali atlikti pagrindinius veiksmus: sudėtį (+), atimtį (-), daugybą (\*) ir dalybą (/). Tačiau nėra žinoma, kiek veiksmų bus atliekama.

Duomenys yra vedami taip: skaičius, simbolis, skaičius, simbolis, ... Jeigu simbolis yra lygybė (=) tai reiškia, kad skaičiavimai baigti ir yra išvedamas atsakymas. Veiksmai yra atliekami iš eilės, t.y. jei pirma įvesta sudėtis, o po to daugyba, tai pirmi du skaičiai sudedami, o po to rezultatas padauginamas iš trečio skaičiaus.

Pradiniai duomenys	Rezultatai
2.5 + 2.5 =	5
3 + 4 * 5 =	35
3 * 4 - 2 / 5 + 2.4 =	4.4

### Reikalavimai

- Visi įvesti skaičiai ir rezultatas yra realieji skaičiai.

## 5. Kvadratai iš degtukų

Mantas turi **n** degtukų ir iš jų dėlioja kvadratus. Pirmojo kvadrato kraštinės **a** ilgis lygus vienam degtukui, antrojo - dviem, trečiojo - trim ir t.t. Parenkite programą, skaičiuojančią kelis

kvadratus **k** sudėlios Mantas iš turimų degtukų ir keli degtukai **d** liks nepanaudoti ir koks bus didžiausio sudėlioto kvadrato kraštinės ilgis **a**.

Pradinis duomuo - degtukų skaičius **n** įvedamas klaviatūra.

Rezultatai - kvadratų skaičius **k**, likusių nepanaudotų degtukų skaičius **d** ir didžiausio kvadrato kraštinės ilgis **a** - išvedami į ekraną, atskiriant juos vieną nuo kito tarpais.

Pradiniai duomenys	Rezultatai
20	2 8 2

### Reikalavimai

- Naudokite nežinomo kartojimų skaičiaus ciklą WHILE.

## 6. Saldainių dalybos

Mokykloje paskelbta Saldainių diena. Kiekvienas mokinys atsineša dviejų rūšių saldainių ir keičiasi saldainiais su savo draugais. Justas atsinešė **k** saldainių „Karvutė“ ir **n** saldainių „Nomeda“.

Parašykite programą, skaičiuojančią, su keliais draugais **d** Justas galės keistis saldainiais, jei kiekvienas draugas turi gauti po vienodai kiekvienos rūšies saldainių. Kiekvienos rūšies saldainių gali likti. Likusių saldainių skaičius taip pat reikia išvesti į ekraną.

Saldainiai dalinami pagal tokį algoritmą:

Skaičiuojamas **k** ir **n** didžiausias bendras daliklis (DBD). Jei didžiausias bendras daliklis bus didesnis už 1, tuomet jis ir bus draugų skaičius. Jei DBD bus lygus 1, tuomet laikykite, kad kiekvienas draugas gauna po 1 kiekvienos rūšies saldainį.

Pavyzdžiui, jei Justas atsinešė 36 „Karvutes“ ir 24 „Nomedas“, tuomet jis gali keistis saldainiais su 12 draugų. Kiekvienas draugas gaus po 3 „Karvutes“ ir po 2 „Nomedas“. Nei vienos rūšies saldainių neliks.

Jei Justas atsinešė 7 „Karvutes“ ir 5 „Nomedas“, tuomet jis gali keistis saldainiais su 5 draugais ir jam liks 2 „Karvutės“ saldainiai.

Pradiniai duomenys **k** ir **n** įvedami klaviatūra.

Rezultatai - draugų skaičius **d** ir likusių saldainių „Karvutė“ ir „Nomeda“ skaičiai išvedami į ekraną vienoje eilutėje juos atskiriant vieną nuo kito tarpais.

Pradiniai duomenys	Rezultatai
6 9	3 0 0

### Reikalavimai

- Didžiausiam bendram dalikliui surasti naudokite nežinomo kartojimų skaičiaus ciklą WHILE.

## 5. SUMOS, KIEKIO, VIDURKIO SKAIČIAVIMAS TAIKANT ŽINOMO IR NEŽINOMO KARTOJIMŲ SKAIČIAUS CIKLUS

### 1. First LEGO League varžybų iššūkis

Jaunieji robotų konstruktoriai ir programuotojai rengiasi First LEGO League (FLL) varžyboms. Jie sukonstravo robotą ir sukūrė roboto valdymo programas.

Robotas FLL varžybų lauke turi įveikti  $n$  ( $n \leq 20$ ) misijų (misija - tam tikra užduotis, kurią turi atlikti robotas). Robotas testuojamas taip: iš bazės jis paleidžiamas įveikti misijos ir fiksuojamas laikas, per kurį robotas įvykdo misiją ir grįžta atgal į bazę. Jei misijos robotui įveikti nepavyko, tuomet laikoma, kad laiko reikšmė yra neigiama. Prieš varžybas robotų konstruktoriams ir programuotojams brangi kiekviena minutė, todėl pagelbėkite jiems ir sukurkite programą, skaičiuojančią:

- kelių misijų robotui nepavyko įveikti;
- kiek laiko robotas sugaišo įveikdamas misijas sėkmingai;
- kiek laiko robotas vidutiniškai sugaišo įveikdamas vieną misiją.

Pradiniai duomenys, misijų skaičius  $n$  ir roboto sugaištas laikas sekundėmis įveikiant kiekvieną iš misijų (sveikasis skaičius), įvedami klaviatūra.

Rezultatai išvedami į ekraną po vieną eilutėje:

- pirmoje eilutėje – nesėkmingų misijų skaičius;
- antroje – laikas, sugaištas įveikiant misijas sėkmingai;
- trečioje – vidutinis laikas, per kurį robotas įveikia vieną misiją vieno skaitmens po kablelio tikslumu.

Pradiniai duomenys	Rezultatai
5	
14	
21	1
-25	74
23	19.8
16	

## 2. Jono treniruotės

Informatikos fakulteto pirmakursis Jonas nusprendė pagerinti savo fizinę formą ir pradėjo kiekvieną dieną iš bendrabučio keletą sustojimų eiti pėstute, o likusią kelio dalį iki fakulteto važiuoti visuomeniniu transportu. Jonas registruoja kiekvieną dieną nueitų žingsnių skaičių, einant sudegintų kilokalorijų kiekį (sveikasis skaičius) ir pasižymi nuovargio lygį (0 - nepavargęs, 1 - truputį pavargęs, 2 - vidutiniškai pavargęs, 3 - labai pavargęs).

Parašykite programą, skaičiuojančią:

- kiek iš viso kilometrų nuėjo Jonas per  $n$  dienų, kai Jono žingsnio ilgis yra lygus  $z$  centimetrų. Skaičiuodami atskirkite tik sveikąjį kilometrų skaičių.
- kelias dienas Jonas buvo: a) nepavargęs; b) truputį pavargęs; c) vidutiniškai pavargęs; d) labai pavargęs;
- kiek iš viso kcal sudegino Jonas vaikščiodamas pėstute;
- kiek gramų riebalų neteko Jonas pasirinkęs vaikščiojimą pėstute. 1 gramas riebalų atitinka 9 kilokalorijas. Skaičiuodami atskirkite tik sveikąjį gramų skaičių.

Pradiniai duomenys įvedami klaviatūra. Pirmiausia įvedamas dienų skaičius  $n$  ( $n \leq 30$ ), po to Jono žingsnio ilgis  $z$ . Tolesnėse  $n$  eilučių įvedamas Jono kiekvieną dieną nueitų žingsnių skaičius, sudegintų kilokalorijų kiekis ir nuovargio lygis.

Rezultatai išvedami ekrane vienoje eilutėje atskiriant vieną nuo kito tarpais tokia tvarka: Jono iš viso nueitų kilometrų skaičius, kelias dienas Jonas buvo nepavargęs, truputį pavargęs, vidutiniškai pavargęs ir labai pavargęs, kiek kcal Jonas sudegino, kiek gramų riebalų neteko Jonas.

Pradiniai duomenys	Rezultatai
5 65	
3250 76 3	
3190 82 3	
2850 52 3	9 0 0 2 3 317 35
2150 47 2	
2521 60 2	

### 3. Žūklės varžybos

Kiekvienais metais žvejų mėgėjų draugija organizuoja žūklės varžybas. Kiekvieno žvejo sugautos žuvys suskaičiuojamos, pasveriamos (gramai, sveikasis skaičius), išmatuojamas ir sudedamas jų ilgis (centimetrai, sveikasis skaičius). Galutinis kiekvieno žvejo įvertinimas skaičiuojamas pagal taisyklę: jei žuvų svoris mažesnis už 100 gramų, skiriamas vienas taškas, jei 100-199 gramai - 2 taškai, 200 ir daugiau gramų - 3 taškai; jei žuvų ilgis mažesnis už 50 cm, skiriamas vienas taškas, jei 50-99 - 2 taškai, jei 100 ir daugiau - 3 taškai. Už svorį ir ilgį gauti taškai sumuojami ir padauginami iš sugautų žuvų skaičiaus.

Parašykite programą, skaičiuojančią kiekvieno žvejo galutinį įvertinimą ir kiek taškų daugiausiai buvo surinkta per varžybas.

Pradiniai duomenys įvedami klaviatūra. Pirmiausia įvedamas varžybose dalyvavusių žvejų skaičius  $n$  ( $n \leq 50$ ). Tolesnėse  $n$  eilučių įvedama informacija apie kiekvieno žvejo sugautas žuvis: žuvų skaičius, svoris ir ilgis.

Rezultatai išvedami ekrane. Pirmosiose  $n$  eilučių išvedami visų žvejų taškai. Vienam žvejui skiriama viena eilutė. Tolesnėje eilutėje - geriausias varžybų rezultatas taškais.

Pradiniai duomenys	Rezultatai
5	60
15 150 75	72
12 210 115	30
5 315 145	6
3 75 25	12
6 89 30	72

### 4. Orų stebėjimas

Per gamtos pažinimo pamoką mokytoja antrojų paprašė stebėti orus ir užsirašyti kiekvienos dienos oro temperatūrą. Stebėjimai nutraukiami, kai dienos temperatūra tampa didesnė už nulį. Vaikai turėjo surasti, kokia buvo žemiausia ir aukščiausia temperatūra stebėjimų dienomis, bei kokia buvo vidutinė dienos temperatūra per stebėjimų laikotarpį.

Pradiniai duomenys - temperatūros reikšmės (sveikieji skaičiai) įvedami klaviatūra. Įvedimas nutraukiamas tuomet, kai įvedama teigiama, arba nuliui lygi temperatūros reikšmė).

Rezultatai išvedami į ekraną vienoje eilutėje reikšmes vieną nuo kitos atskiriant tarpais: žemiausia temperatūra, aukščiausia temperatūra ir vidutinė temperatūra dviejų skaitmenų po kablelio tikslumu. Tuo atveju, kai pirmoji temperatūros reikšmė yra didesnė arba lygi nuliui, tuomet išvedami nuliai laikantis aukščiau nurodytą formatavimo taisyklių.

Pradiniai duomenys	Rezultatai
-15	
-19	
-12	-19 -7 -13.25
-7	
4	

### 5. Skęstančiųjų gelbėjimas

Vienas labai išmintingas matematikos mokytojas sugalvojo labai gerą metodą, kuris padeda mokiniams suprasti ir pamėgti matematiką. Jis kiekvieną klasės mokinį, kuris gerai supranta matematiką, priskyrė kaip pagalbininką mokiniui, kuris matematikos nesupranta. Mokytojas kartą per savaitę patikrina, kiek pagerėjo su pagalbininkais dirbančių mokinių gebėjimai. Kai globojamas mokinys be klaidų išsprendžia  $k$  uždavinių, tuomet jam pagalbininko nebereikia. Vienas iš mokytojo pagalbininkų savo globotiniui pasiūlė spręsti papildomai uždavinių laikantis

tokio susitarimo: jei globotinis per savaitę išsprendžia mažiau kaip 3 uždavinius, tuomet jis turi papildomai išspręsti 3 pagalbininko skirtus uždavinius, jei 3-5 uždavinius - tuomet papildomai sprendžia 2 pagalbininko skirtus uždavinius, jei daugiau kaip 5, tai sprendžia 1 pagalbininko skirtą uždavinį.

Parašykite programą, skaičiuojančią:

- per kelias savaites mokiniui, kuriam sunkiai sekasi matematika, nebereiks pagalbininko, jei jis spręs mokytojo ir pagalbininko skirtus uždavinius;
- kelis pagalbininko skirtus uždavinius mokinys išsprendė.

Pradiniai duomenys įvedami klaviatūra. Pirmiausia įvedamas mokytojo skirtų uždavinių skaičius, po to įvedami mokinio per savaitę išsprendžiamų uždavinių skaičiai. Mokinio išspręstų uždavinių nebereikia įvesti tuomet, kai bus išspręsti visi mokytojo ir pagalbininko skirti uždaviniai.

Rezultatai išvedami į ekraną vienoje eilutėje atskiriant tarpais: savaitių skaičius ir pagalbininko skirtų uždavinių skaičius.

Pradiniai duomenys	Rezultatai
25	7 12
4	
2	
5	
6	
5	
9	
15	

## 6. Tobulasis skaičius

Tobuluoju vadinamas skaičius, kuris yra lygus visų savo daliklių, mažesnių už jį patį, sumai. Pavyzdžiui,  $28 = 1 + 2 + 4 + 7 + 14$ . Keletas tobulųjų skaičių: 6, 28, 496, 8128.

Senovės graikai žinojo tik pirmuosius keturis tobuluosius skaičius. Žymus graikų filosofas ir matematikas Nikomachas Gerazietis (I a.) rašė: „Tobulieji skaičiai gražūs. Tačiau žinoma, kad gražūs daiktai reti, netikusių yra pilna visur“. Tobulieji skaičiai buvo nepaprastai vertinami. Neveltui Biblijoje sakoma, jog pasaulis buvo sukurtas per 6 dienas – juk tai pirmas tobulasis skaičius! Tobulieji skaičiai dar nėra visiškai ištirti: nežinoma, ar jų baigtinis ar begalinis skaičius, kol kas nerasta nė vieno nelyginio tobulojo skaičiaus ir neįrodyta, kad jis neegzistuoja.

Parašykite programą, kuri nustatytų, ar klaviatūra įvestas natūralusis skaičius  $x$  yra tobulasis. Jei skaičius yra tobulasis, ekrane turi būti rodomas žodis TAIP, jei ne – žodis NE.

## 7. Supergalingas skaičius

Mantui labai patinka eksperimentuoti su natūraliaisiais skaičiais. Jis sugalvojo, kad supergalingais skaičiais būtų galima vadinti tokius skaičius, kurių daliklių kiekis, padidintas 25 kartus, sutampa su pačiu skaičiumi.

Parašykite programą, kuri patikrintų, ar klaviatūra įvestas natūralusis skaičius yra supergalingas. Jei skaičius supergalingas, ekrane turi būti rodomas žodis TAIP, jei ne – žodis NE.

Pradiniai duomenys	Rezultatai
225	TAIP
50	NE

## 6. SKAITYMAS IŠ FAILO, RAŠYMAS Į FAILĄ. CIKLAS CIKLE

### 1. Elektronikos elementų ir komponentų rinkinys

Elektrinėse grandinėse naudojami įvairūs puslaidininkiniai elementai, integriniai grandynai (mikroschemos), elektroniniai komponentai bei elektriniai prietaisai, kitos dalys. Visi jie turi savo paskirtį ir atlieka grandinėse atitinkamas funkcijas.

Mokyklos 7-10 klasių mokiniai gali mokytis jungti elektrines grandines per technologijų pamokas. Mokytojui reikia suformuoti užsakymą elektrinių grandinių komponentams.

Mokytojas tekstinio failo **grandines.txt** pirmoje eilutėje įrašė, kelių tipų **t** komponentus reikia užsakyti. Toliau seka **t** eilučių, kurių kiekvienoje įrašyta, kiek skirtingų vieno tipo komponentų **s** reikia užsakyti ir po **s** skaičių dvejetų. Pirmasis dvejeta skaičius rodo komponentų skaičių (sveikasis skaičius) **k** ir komponento vieneto kainą **vk** (realusis skaičius).

Parašykite programą, kuri tekstiname faile **grandinesrez.txt** išvestų kiekvieno tipo bendrą komponentų skaičių ir jų kainą dviejų skaitmenų po kablelio tikslumu. Vienam komponentui skiriama viena eilutė. Paskutinėje eilutėje turi būti išvestas bendras įsigytų komponentų skaičius ir jų kaina dviejų skaitmenų po kablelio tikslumu.

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

### 2. Lygiašonės trapecijos

Trapecija, kurios šoninės kraštinės yra vienodo ilgio, vadinama **lygiašone**.

Parašykite programą, kuri iš sveikųjų skaičių intervalo **[a;b]** skaičių suformuotų visas lygiašones trapecijas, kurių pagrindų ilgiai skiriasi **k** vienetų (**k** - sveikasis skaičius).

Pradiniai duomenys, intervalo pradžia **a**, pabaiga **b** ir pagrindų ilgių skirtumas **k**, įrašyti **faile trapecija.txt**. Faile **trapecijarez.txt** išveskite visas galimas trapecijas: šoninės kraštinės ilgis, trumpesniojo pagrindo ilgis, ilgesniojo pagrindo ilgis. Šoninės kraštinės ilgis turi būti išvedamas didėjimo tvarka. Jei šoninės kraštinės ilgiai sutampa, tuomet pagrindų ilgiai taip pat turi būti išvedami didėjimo tvarka.

trapecija.txt	trapecijarez.txt
2 7 4	3 2 6
	3 3 7
	4 2 6
	4 3 7
	5 2 6
	5 3 7
	6 2 6
	6 3 7
	7 2 6
	7 3 7

### 3. Skaniausi pietūs

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

geriausias ir blogiausias įvertinimai sutampa, tuomet iš sumos atimamos dvi vienodos vertės. Šou laimi daugiausia taškų surinkęs dalyvis.

Pirmoje pradinių duomenų **pietus.txt** eilutėje įrašytas šou dalyvių skaičius  $n$  ( $n > 3$ ). Tolesnėse  $n$  eilučių įrašyta po  $n-1$  sveikųjų skaičių - kaip kažkurį dalyvį vertina likusieji šou dalyviai. Dalyviai numeruojami nuo **1** iki  $n$ . Jei keli nors šou dalyviai surinko po vienodą taškų skaičių, tuomet nugalėtojas yra tas, kuris pietus gamino anksčiau (turi mažesnę numerį).

Rezultatų faile **pietusrez.txt** turi būti įrašyti du sveikieji skaičiai, vienas nuo kito atskirti tarpu: šou laimėtojo numeris ir jo galutinis įvertinimas.

pietus.txt	pietusrez.txt
5	
5 4 4 5	
5 4 4 4	4 10
5 5 4 4	
5 5 5 4	
4 4 4 5	

#### 4. Žiogas

Žiogas tupi ant horizontaliai ištemptos virvutės, kairiajame gale. Virvutės ilgis yra  $s$  sprindžių. Žiogo šuolio į priekį ilgis  $a$  sprindžių, atgal -  $b$  sprindžių ( $a > b$ ). Žiogui reikia patekti ant virvutėje užmegzto mazgo, kuris nuo pradinės žiogo padėties nutolęs per  $c$  sprindžių ( $c > a$ ;  $c < s$ ).

Pradiniai duomenys (sveikieji skaičiai)  $s$ ,  $a$ ,  $b$ ,  $c$  įrašyti tekstiniame faile **ziogas.txt**

Parašykite programą, skaičiuojančią, kiek mažiausiai šuolių turi padaryti žiogas, kad pasiektų mazgą. Jei mazgo pasiekti negalima, tuomet rezultatų faile **ziogasrez.txt** turi būti įrašytas žodis **NEGALIMA**.

ziogas.txt	ziogasrez.txt
10 3 2 6	2
20 4 2 5	NEGALIMA

#### 5. Sultis gerti sveika

Kiekvienos savaitės pirmadienio rytą Jonas gauna  $k$  centų kišenpinigių. Vienas butelis sulčių kainuoja  $s$  centų, o tušti buteliai superkami po  $b$  centų.  $i$ -tąją savaitės dieną Jonas, turėdamas  $n$  centų, nusprendė, pirkti sulčių už visus turimus pinigus (pradedant  $i$ -tąja diena). Pinigai, gauti pardavus butelius, bus panaudojami kitą dieną sultims pirkti. Šitaip Jonas darys tol, kol įstengs nusipirkti bent vieną butelį sulčių.

Parašykite programą, skaičiuojančią, kiek butelių sulčių išgers Jonas.

Pradinių duomenų faile **sultys.txt** įrašytos kintamųjų  $k$ ,  $s$ ,  $b$ ,  $n$ ,  $i$  ( $1 \leq i \leq 7$ ) reikšmės. Jei Jonas nusprendžia sultis pirkti pirmadienį, tai iš karto gaunami kišenpinigiai (pasipildo turima pinigų suma). Už pirmadienį gautus kišenpinigius sultys perkamos pirmadienį.

Rezultatų faile **sultysrez.txt** įrašomas Jono išgertų sulčių butelių kiekis.

sultys.txt	sultysrez.txt
25 10 3 2 15	1

#### 6. Varlių koncertas

Kartą vienoje kūdroje gyveno daug varlių, ir ne bet kokių, o dresuotų. Kiekviena varlė sugebėdavo iššokti iš vandens ir sukvarksėti jai būdingais tiksliais laiko momentais. Pavyzdžiui,



jei varlės kvarksėjimo periodas lygus 5 ir ji sukvarksėjo pirmą minutę, tai antrą kartą ji sukvarksės po 5 minučių, t.y. šeštą minutę, trečią kartą - vienuoliktą minutę ir t.t.

Patekęs saulei visos varlės iššoko iš vandens ir sukvarksėjo. Parašykite programą, kuri nustatytų, po kelių valandų ir minučių įvyks antrasis varlių koncertas, t.y. visos varlės vėl vienu metu iššoks ir sukvarksės.

Pradiniai duomenys įrašyti tekstiniam failui **varles.txt**. Pirmoje failo eilutėje įrašytas varlių skaičius, antroje - kiekvienos varlės kvarkimo periodas. Periodai vienas nuo kito atskiriami tarpais. Varlių skaičius kūdroje neviršys dešimties.

Rezultatų failui **varlesrez.txt** turi būti įrašyta, po kelių valandų ir minučių įvyks antrasis varlių koncertas.

varles.txt	varlesrez.txt
5 18 7 3 2 4	4 12

## 7. FUNKCIJOS, GRAŽINANČIOS APSKAIČIUOTĄ REIKŠMĘ PER FUNKCIJOS VARDĄ

### 1. Kiek yra tokių skaičių?

Parašykite programą, kuri surastų natūraliųjų skaičių intervalo  $[m; n]$  ( $m > 2518$ ;  $n > m$ ) skaičius, kuriuos padalijus iš 2 gaunama liekana 1, padalijus iš 3 — liekana 2, padalijus iš 4 — liekana 3, padalijus iš 5 — liekana 4, padalijus iš 6 — liekana 5, padalijus iš 7 — liekana 6, padalijus iš 8 — liekana 7 ir padalijus iš 9 — liekana 8.

Pradiniai duomenys:  $m$  ir  $n$  reikšmės įrašytos tekstiniam failui intervalas.txt. Rezultatai saugomi tekstiniam failui skaiciai.txt, vienam skaičiui skiriama viena eilutė.

intervalas.txt	skaiciai.txt
3000 10000	5039 7559

### Reikalavimai

- $m$  ir  $n$  yra long long tipo.
- Sukurkite loginę funkciją, kuri tikrina, ar skaičius  $x$  iš intervalo dalinasi iš skaičiaus  $y$  su liekana lygia  $y - 1$ .
- Sukurtą funkciją pritaikykite tiek kartų, kiek sąlygų reikia patikrinti.

### 2. Pažymių vidurkiai

Prieš pusmečio pabaigą Dominykas labai susirūpino savo pažymiais ir nusprendė apskaičiuoti svarbiausių dalykų pažymių vidurkį.

Duomenų failui Pazymiai.txt yra penkios eilutės, kuriose įrašyti penkių dalykų (matematikos, lietuvių kalbos, anglų kalbos, fizikos ir informacinių technologijų) pažymiai. Iš kiekvieno dalyko yra gauti penki pažymiai. Parašykite programą, kuri į failą Vidurkiai.txt išspausdintų vidurkius (vienam vidurkiui skirta viena eilutė) dviejų ženklų po kablelio tikslumu.

Pazymiai.txt	Vidurkiai.txt
3 6 6 8 2	5.00
8 8 9 5 10	8.00
2 3 4 5 6	4.00
3 6 6 8 2	5.00
8 8 9 5 10	8.00

### Reikalavimai

- Sukurti double tipo funkciją, gražinančią apskaičiuotą vieno dalyko vidurkį per funkcijos vardą.

### 3. Mokytojos padėjėjas

Parašykite programą, kuri apskaičiuotų funkcijos  $f(a,b,c,d) = (a+b)/(-c^2) + d^{1/2}$  reikšmes su skirtingais argumentais bei rastų didžiausią šios funkcijos reikšmę.

Pradiniai duomenys: tekstiniame faile "Argumentai.txt" pateikti pradiniai duomenys. Pirmoje eilutėje nurodytas skirtingų funkcijos argumentų rinkinių skaičius n. Tolimesnėse n eilučių pateikti argumentai a, b, c ir d (atskirti tarpo simboliu). Argumentai yra sveikieji skaičiai.

Rezultatai saugomi tekstiniame faile "Rezultatai.txt". n eilutėse pateikiamas apskaičiuotas funkcijos rezultatas arba klaidos žinutė, jei duota funkcija neegzistuoja su įvestais argumentais. Paskutinėje eilutėje išspausdinama didžiausia funkcijos reikšmė, jei tokios nėra (kai visos pateiktos funkcijos neegzistuoja) – išvedamas 0. Atkreipkite dėmesį, kad didžiausia funkcijos reikšmė gali būti ir neigiama!

Pavyzdys:

Argumentai.txt	Rezultatai.txt
5	0.00
2 2 2 1	KLAIDA! c negali būti lygus nuliui!
4 0 0 1	KLAIDA! d negali būti neigiamas!
19 12 2 -10	3.06
7 8 4 16	1.88
20 8 5 9	Didžiausia funkcijos reikšmė: 3.06
Argumentai.txt	Rezultatai.txt
4	KLAIDA! d negali būti neigiamas!
10 0 1 -25	KLAIDA! c negali būti lygus nuliui!
4 3 0 225	KLAIDA! d negali būti neigiamas!
-5 7 2 -3	KLAIDA! d negali būti neigiamas!
-9 -2 4 -1	Didžiausia funkcijos reikšmė: 0.00

### Reikalavimai

- n, a, b, c, d – int tipo kintamieji;
- funkcijos rezultatas – double tipo;
- Sukurkite funkciją, kuri per savo vardą gražina apskaičiuotą funkcijos  $f(a,b,c,d)$  reikšmę: double Rezultatas(int a, int b, int c, int d);
- Sukurkite "string" tipo funkciją, gražinančią atitinkamą klaidos žinutę, jei argumentas c yra lygus 0 arba d yra neigiamas;
- Sukurkite funkciją, gražinančią didžiausią  $f(a,b,c,d)$  reikšmę per funkcijos vardą;
- Rezultatus pateikite 2 skaičių po kablelio tikslumu ir įrašykite į tekstinį failą "Rezultatai.txt";
- Nenaudokite masyvų!

#### 4. Išlaidos

Nuo pirmadienio iki penktadienio Pranas užsirašinėja savo išlaidas. Atėjus šeštadieniui jis susiskaičiuoja, kiek pinigų išleido iš tų, kuriuos davė mama pirmadienį. Tačiau tai užtrunka laiko, todėl Pranas sugalvojo, kad šį procesą galima automatizuoti, tad kreipėsi į jus pagalbos.

Tekstiniame faile "Islaidos.txt" pateikti pradiniai duomenys. Pirmoje eilutėje yra dienpinigiai n, sekančiose eilutėse surašytos išlaidos m. n ir m double tipo.

Rezultatai saugomi tekstiniame faile "Ataskaita.txt". Pirmoje eilutėje turi būti išvedama, kiek pinigų liko iš dienpinigių, antroje eilutėje turi būti didžiausia suma, kurią išleido Pranas, ir trečioje išlaidų vidurkis.

Islaidos.txt	Ataskaita.txt
10	
0	
3.01	3.99
1.10	3.01
0.90	1.20
1.00	

#### Reikalavimai

- n ir m yra double tipo.
- Sukurti funkciją, kuri per funkcijos vardą grąžintų likusius dienpinigius.
- Sukurti funkciją, kuri tikrintų ar tas skaičius didžiausias, ir grąžintų didžiausią vertę tam tikrą dieną.
- Sukurti funkciją, kuri apskaičiuotų vidurkį ir grąžintų apskaičiuotą reikšmę per funkcijos vardą.
- Tekstiniame faile „Ataskaita.txt“ rezultatus išvesti dviejų skaičių po kablelio tikslumu.
- Nenaudokite masyvų!

#### 5. Smėlio dėžė

Vaikai sugalvojo pasidaryti kvadratinę smėlio dėžę. Prieš pradėdami jie prisikasė daug smėlio. Berniukai pamiršo pasiimti pjūklą ir dabar jie susidūrė su problema. Kokio ilgio ir pločio dėžės reikia, kad joje tilptų visas berniukų prikastas smėlis, jei jie turi tik 1 m ilgio ir 30 cm pločio lentas?

Parašykite programą, kuri apskaičiuotų, kokio tūrio turi būti smėlio dėžė, kad į ją sutilptų visas berniukų smėlis. Smėlio kiekis (kubiniais metrais) pateiktas faile smelis.txt, dėžės aukštis – 30 cm, sienos ilgis – sveikas skaičius. Sienos ilgį įrašyti į failą deze.txt. Dėžutė negali būti sklidina! (Jos tūris turi būti didesnis už smėlio tūrį).

smelis.txt	deze.txt
0.25	1

#### Reikalavimai

- Smėlio kiekis – double tipo.
- Sienos ilgis – int tipo.
- Sukurti funkciją, kuri apskaičiuotų dėžės tūrį. Apskaičiuotas tūris turi būti grąžinamas per funkcijos vardą.

## 6. Saldainiai

Antanas grįžta iš atostogų Paryžiuje. Savo klasės draugams jis nori parvežti tiek saldainių, kad būtų įmanoma juos padalinti visiems po lygiai. Tačiau Antanas nežino, kiek jo klasės draugų ateis į mokyklą. Padėkite Antanui rasti visus būdus, kuriais jis galėtų padalinti saldainius.

Sukurkite programą, kuri parodytų:

1. Kokį saldainių kiekį galima po lygiai padalinti daugiausia skirtingų būdų, t.y. koks skaičius nurodytame intervale turi daugiausiai daliklių.
2. Kiek skirtingų būdų yra padalinti tokį saldainių skaičių (kiek yra daliklių).
3. Saldainių kiekius, tenkančius vienam draugui kiekvienu atveju.

Pradiniai duomenys: min ir max intervalo reikšmės įrašytos tekstiniame faile intervalas.txt. Rezultatai yra saugomi tekstiniame faile rezultatai.txt. Pirmoje eilutėje įrašoma daugiausiai daliklių turintis skaičius x intervale  $\min \leq x \leq \max$  ir skaičiaus x daliklių kiekis.

Antroje eilutėje didėjimo tvarka surašomi visi skaičiaus x dalikliai.

Duomenų ir rezultatų failų pavyzdžiai:

intervalas.txt	rezultatai.txt
10 30	24 8 1 2 3 4 6 8 12 24
40 100	60 12 1 2 3 4 5 6 10 12 15 20 30 60
150 350	240 20 1 2 3 4 5 6 8 10 12 15 16 20 24 30 40 48 60 80 120 240

### Reikalavimai

- min ir max – teigiami int tipo kintamieji.
- Sukurti int tipo funkciją, kuri per funkcijos vardą grąžintų skaičiaus daliklių kiekį.

## 7. Įdomūs pirminiai skaičiai

Parašykite programą, kuri surastų visus įdomius n-ženklus pirminius skaičius, ( $1 < n < 5$ ) ir suskaičiuotų, kiek jų yra.

Priminimas: Pirminiais skaičiais yra vadinami natūralieji skaičiai, kurie dalinasi tik iš savęs ir iš vieneto. Įdomiais pirminiais skaičiais vadinami tokie skaičiai, kai pats skaičius yra pirminis ir jo skaitmenų suma taip pat yra pirminis skaičius.

Pradiniai duomenys: n reikšmė, įrašyta tekstiniame faile n.txt. Rezultatas – įdomių pirminių skaičių kiekis – yra saugomas tekstiniame faile pirminiai.txt.

n.txt	pirminiai.txt
2	10

### Reikalavimai

- Sukurti int tipo funkcijas, grąžinančias pradinę (galutinę) intervalo reikšmes.
- Sukurti bool tipo funkciją, kuri patikrina, ar skaičius yra pirminis ir per funkcijos vardą grąžina reikšmę true – jei skaičius yra pirminis, arba reikšmę false – jei ne pirminis.
- Sukurti int tipo funkciją, kuri grąžina apskaičiuotą natūraliojo skaičiaus skaitmenų sumą.

## 8. Žaidimų kultūros parodos „GameOn“ eksperimentas

Žaidimų kultūros parodoje „GameOn“ buvo pastatytas gaiviųjų gėrimų pardavimo aparatas. Jis skaičiavo nupirktus gėrimus, Kad būtų galima įvertinti, kiek jaunų žmonių renkasi sveikus gėrimus ir kiek saldžiuosius.

1 buvo žymimas Coca Cola, 2 – Sprite, 3 – vandens buteliukas. Pradinių duomenų failo Duomenys.txt pirmoje eilutėje įrašytas parodos lankytojų skaičius, antroje surašyti lankytojų įsigyti gėrimai. Rezultatų faile Rezultatai.txt iš eilės po vieną eilutėje surašykite:

a) kuri dalis lankytojų pirko Coca Cola, Sprite ir vandens buteliuką. Rezultatus išveskite dviejų skaitmenų po kablelio tikslumu.

b) kiek procentų žmonių pirko vandens buteliukus. Rezultatą išveskite 5 skaitmenų po kablelio tikslumu.

c) kiek nepilnamečių pirko Coca Cola gėrimą, jei žinoma, kad kultūros parodoje dalyvavo 40 procentų nepilnamečių. Skaičiuodami atskirkite sveikąją dalį.

Duomenys.txt	Rezultatai.txt
23 1 1 1 2 2 2 3 1 2 3 1 3 1 2 3 2 2 1 1 3 3 3 3	0.35 0.30 0.35 34.78261 3
12 1 1 1 1 2 3 3 3 1 3 1 2	0.50 0.17 0.33 33.33333 2
17 1 2 3 2 2 1 1 3 3 3 3 3 1 2 1 3 2	0.29 0.29 0.41 41.17647 2

### Reikalavimai

Parašykite funkcijas, grąžinančias apskaičiuotas reikšmes per funkcijos vardą:

- double tipo, skaičiuojančią, kuri dalis lankytojų pirko vienos rūšies gėrimą.
- double tipo, skaičiuojančią, kiek procentų lankytojų pirko vandens buteliukus.
- int tipo, kuri skaičiuoja, kiek nepilnamečių rinkosi Coca Cola gėrimą.

### 9. Verslo žurnalo „Forbes“ rekomendacija

Informikas, perskaitęs verslo žurnalo „Forbes“ rekomendaciją, nusprendė sekti savo išlaidas ir patikrinti, ar pajamos pasiskirstė pagal rekomenduojamą 50-20-30 taisyklę.

Parašykite programą, kuri apskaičiuotų, kiek iš viso buvo skirta tam tikrai sričiai ir įvertintų, ar Informikas sugebėjo tinkamai kontroliuoti savo išlaidas.

Pradiniai duomenys: tekstiniame faile „Duomenys.txt“ pateikti pradiniai duomenys. Pirmoje eilutėje nurodytas savaičių kiekis n. Tolimesnėse n eilučių pateikta, kiek Informikas skyrė tam tikrai sričiai: 1 skaičius – pinigai skirti kasdienėms išlaidoms, 2 – atsidėti pinigai, 3 – pinigai promogoms. Kadangi informikas nori tiksliai atitikti 50-20-30 taisyklę, pinigai – sveikieji skaičiai.

Rezultatai: tekstiniame faile „Rezultatai.txt“ pateikiamas įvertinimas. Jeigu išlaidos atitiko 50-20-30 taisyklę, vadinasi: Informikui pavyko tinkamai kontroliuoti savo išlaidas, kitu atveju - nepavyko.

Pavyzdys:

Duomenys.txt	Rezultatai.txt
4 19 0 10 12 10 10 10 5 10 9 5 0	Pavyko
Duomenys.txt	Rezultatai.txt
4 20 0 30 21 10 10 0 10 10 11 5 0	Nepavyko

### Reikalavimai

- n, kiekvienos srities sumos ir bendra suma – int tipo kintamieji.
- Sukurkite funkciją, kuri per savo vardą grąžina bendrą visų sričių sumą: int Bendrasuma(int a, int b, int c).
- Sukurkite funkciją, grąžinančią įvertinimą, ar Informikui pavyko tinkamai kontroliuoti savo išlaidas: string Patikrinimas(int bendrasuma, int a, int b, int c).
- main() funkcijoje patikrinkite įvertinimą ir į tekstinį failą „Rezultatai.txt“ įrašykite verdiktą.
- Nenaudokite masyvų!

### 10. Ilgiausia didėjančių skaičių seka

Faile pateikta įvairių skaičių seka. Nauja eilutė – naujas skaičių sekos elementas. Skaitant kiekvieno įvesties failo eilutę, nustatyti ilgiausią didėjančių skaičių seką ir tos sekos dydį išsaugoti išvesties faile.

Įvesties failo pavadinimas: datain.txt

Išvesties failo pavadinimas: dataout.txt

Pavyzdžiai

Įvestis	Išvestis
14 2 10 0	2
0 0 0	1
0 -1 -2 -3 -4 -5 1 2	3

Paryškinti skaičiai žymi ilgiausią didėjančią skaičių seką.

### Reikalavimai

- Parašykite bent vieną funkciją, grąžinančią apskaičiuotą reikšmę per funkcijos vardą.
- Nenaudokite masyvų.

## 8. FUNKCIJOS, GRAŽINANČIOS APSKAIČIUOTAS REIKŠMES PER PARAMETRUS

### 1. Didžiausio ploto stačiakampiai iš degtukų

Pradinėse (ir ne tik) klasėse sprendžiami uždaviniai, kuriuose prašoma sudėlioti įvairias figūras iš degtukų. Vienas uždavinių tipas susijęs su didžiausio ploto figūrų dėliojimu.

Mokytoja paruošė tiek degtukų rinkinių, kiek klasėje yra mokinių ( $2 < \text{mokinių skaičius} < 35$ ). Kiekviename rinkinyje yra tam tikras skaičius degtukų ( $1 < \text{degtukų skaičius} < 100$ ). Kiekvienas mokinys gavo užduotį: iš degtukų sudėlioti didžiausio ploto stačiakampį, surasti stačiakampio kraštinių ilgius ir kiek degtukų liko nepanaudota. Degtukų laužyti negalima.

Kad būtų lengviau tikrinti mokinių atliktas užduotis mokytoja sugalvojo paprašyti programuotojų pagalbos. Sukurkite programą šiam uždaviniui spręsti.

Pirmoje pradinių duomenų failo **degtukai.txt** eilutėje įrašytas degtukų rinkinių skaičius. Antroje eilutėje įrašyta, po kiek degtukų yra kiekviename rinkinyje.

Rezultatų faile **sprendimai.txt** turi būti tiek eilučių, kiek buvo rinkinių. Vieno rinkinio rezultatai išvedami tokia tvarka: stačiakampio plotas, trumpesniosios stačiakampio kraštinės ilgis, ilgesniosios stačiakampio kraštinės ilgis, likusių nepanaudotų degtukų skaičius.

Pradiniai duomenys	Rezultatai
3	0 0 0 3
3 5 6	1 1 1 1 2 1 2 0

#### Reikalavimai

- Funkcija **void**, gražinanti apskaičiuotus **vieno rinkinio** rezultatus: stačiakampio plotą, trumpesniosios kraštinės ilgį, ilgesniosios kraštinės ilgį, likusių nepanaudotų degtukų skaičių.

### 2. Apranga

Dominykas savo spintoje turi skirtingus aprangos komponentų kiekius: **a** batų porų, **b** kojinių porų, **c** kelnų, **d** sportinių marškinėlių ir **e** megztinių. Duomenys surašyti faile **Komponentai.txt**. Suskaičiuokite ir faile **Rezultatai.txt** išspausdinkite, kiek skirtingų apsirengimo variantų, sudarytų iš visų komponentų, Dominykas turi (laikykite, kad vienas aprangos komponentas gali būti panaudotas tik vienam apsirengimo variantui). Taip pat išspausdinkite, kiek spintoje liks nepanaudotų komponentų.

Komponentai.txt	Rezultatai.txt
3 5 7 8 6	3 0 2 4 5 3

#### Reikalavimai

- Parašykite **void** funkciją, skirtą duomenų skaitymui iš failo.
- Parašykite **vieną void** funkciją, skirtą visiems rezultatams surasti.
- Parašykite **void** funkciją, skirtą rezultatų išvedimui į failą.

### 3. Šachmatų figūrų komplektai

Šachmatų būrelį lankantis Andrius nekantriai laukia artėjančių šachmatų varžybų. Jis nusprendė padėti savo treneriui joms pasiruošti. Šachmatų treneris jam davė užduotį – surinkti trūkstantį skaičių baltų ir juodų figūrų komplektų. Kadangi Andrius žino tikslų skaičių turimų tam tikros

rūšies figūrų, jis nusprendė parašyti programą, kuri suskaičiuotų, kiek komplektų galima sudaryti. Padėkite jam!

**Parašykite programą, kuri:**

1. Rastų, kiek baltų ir juodų figūrų komplektų galima sudaryti iš turimų figūrų;
2. Apskaičiuotų likusių figūrų kiekį, kurios nebuvo panaudotos sudaryti komplektams;
3. Jei buvo surinktas pakankamas skaičius komplektų, į rezultatų failą išvestų „Pakanka“, jei nepakankamas – „Nepakanka“.

**Pradiniai duomenys:**

Duomenų failo „**figuros.txt**“ pirmoje eilutėje yra reikalingų baltų ir juodų figūrų komplektų kiekiai **b** ( $b \leq 50$ ) ir **j** ( $j \leq 50$ ). Kitose eilutėse yra turimi baltų ir juodų figūrų kiekiai. Vienai figūrai yra skiriama viena eilutė. Figūrų eilė duomenų faile yra tokia: bokštas, rikis, žirgas, karalius, valdovė, pėstininkas.

**Rezultatai:**

Rezultatų failo „**komplektai.txt**“ pirmoje eilutėje yra išvedama „Pakanka“, jei buvo sudaryta pakankamai komplektų ir „Nepakanka“, jei komplektų sudaryta per mažai. Antroje eilutėje yra išvedamas sudarytų baltų ir juodų komplektų kiekis. Kitose eilutėse yra paėliui išvedami likę nepanaudoti baltų ir juodų figūrų kiekiai. Vienai figūrai yra skiriama viena eilutė. Figūrų eilė rezultatų faile yra tokia: bokštas, rikis, žirgas, karalius, valdovė, pėstininkas.

figuros.txt	komplektai.txt
5 4	Pakanka
11 12	5 4
15 11	1 4
20 9	5 3
6 7	10 1
7 7	1 3
50 42	2 3
	10 10
6 4	Nepakanka
11 9	5 4
10 13	1 1
22 25	0 5
9 7	12 17
7 6	4 3
63 33	2 2
	23 1
1 1	Nepakanka
5 4	0 1
9 6	5 2
5 5	9 4
3 2	5 3
2 1	3 1
7 9	2 0
	7 1

**Reikalavimai**

- Visi kiekiai – teigiami **int** tipo kintamieji;
- Sukurkite **void** tipo funkciją, kuri, naudojant parametrus – nuorodas, grąžintų sudarytų komplektų kiekį ir likusius figūrų kiekius – bokšto, rikio, žirgo, karaliaus, valdovės ir pėstininko;
- Nenaudokite masyvų.



#### 4. Liūdna prognozė

2017 metais mokslininkų grupelė iš Lietuvos pasidomėjo savo šalies klimatu ir jo tendencijomis ateityje.

Mokslininkai nustatė šių metų vidutinę temperatūrą  $t$ , kritulių kiekį  $k$  ir vėjo greitį  $v$ .

Pasitelkę sudėtingus skaičiavimus, mokslininkai paskelbė liūdną prognozę visiems ateinantiems metams  $mt$  ( $mt > 2017$ ):

1. Kaskart, kai  $mt$  dalinsis iš 5, vidutinė metų temperatūra padidės 0.1 laipsnio.
2. Kaskart, kai  $mt$  dalinsis iš 3, vėjo greitis padidės 0.05 m/s.
3. Kaskart, kai  $mt$  bus pirminis skaičius, kritulių kiekis padidės 10 procentų.

Sukurkite programą, kuri parodytų:

1. Prognozės metus  $x$ ;
2. Kritulių kiekį  $x$  metais;
3. Vidutinę metinę temperatūrą  $x$  metais;
4. Kritulių kiekį  $x$  metais.

Pradiniai duomenys: **metų skaičius  $m$ , vidutinė metų temperatūra  $t$ , vėjo greitis  $v$  ir kritulių kiekis  $k$**  2017 metų duomenimis, įrašyti tekstiniame faile **duomenys.txt** į vieną eilutę ir atskiriami tarpu. Rezultatai yra saugomi tekstiniame faile **rezultatai.txt**. Į atskiras eilutes įrašomi **einamieji metai, vidutinė metų temperatūra  $t$ , vėjo greitis  $v$  ir kritulių kiekis  $k$**  po  $m$  metų.

Duomenų ir rezultatų failų pavyzdžiai:

1. duomenys.txt	2. rezultatai.txt
11 15 10 1000	2028 15.2 10.2 1100
3 20 25 500	2020 20.1 25.05 500
25 13 17 1000	2042 13.5 17.4 1331

#### Reikalavimai

- $m$  – teigiamas **int** tipo kintamasis.
- $t, k, v$  – teigiami **double** tipo kintamieji.
- Sukurti **void** tipo funkciją, kuri per parametrus grąžintų metų, temperatūros, kritulių ir vėjo pokytį.

#### 5. Juodasis penktadienis

Artėjant juodajam penktadieniui parduotuvės daro įvairias akcijas bei dideles nuolaidas.

Tad pirkėjai skuba ir seka visas naujienas, kad pavyktų nusipirkti įvairius daiktus kuo pigiau.

Pigu.lt, varle.lt ir 1a.lt paskelbė tokią akciją: specialiai el. parduotuvių parinktos prekės gauna net 45 proc. nuolaidą, jei jų likęs kiekis sandėlyje yra lyginis skaičius.

Failo **Duomenys.txt** pirmoje eilutėje įrašyta elektroninė parduotuvė (**string** tipas), antroje eilutėje – prekių skaičius (**int** tipas), trečioje eilutėje – prekių kiekiai, ketvirtoje eilutėje – kainos (**double** tipo).

Rezultatai:

a) ar pirkti prekę;

b) nupirktų prekių skaičius;

c) apskaičiuoti kiek pinigų buvo išleista paskutinei nupirktai prekei. Rezultatus išveskite dviejų skaitmenų po kablelio tikslumu.

Duomenys.txt	Rezultatai.txt
Pigu 5 2 5 6 12 29 10 15.5 12.5 50 49.99	Pigu.lt Pirktu prekiu kiekis: 3 Paskutines pirktos prekes kaina: 27.5
Varle 7 3 4 9 18 23 51 77 16 9.5 48.5 34 25.99 345.5 411.5	Varle.lt Pirktu prekiu kiekis: 2 Paskutines pirktos prekes kaina: 18.7
1a 4 2 6 46 129 4 120 12.5 15	1a.lt Pirktu prekiu kiekis: 3 Paskutines pirktos prekes kaina: 6.88

### Reikalavimai

Parašykite **void** tipo funkcijas:

- funkciją, kuri nuspręstų ar pirkti prekę, perduodant bool tipo kintamąjį per funkcijos parametrus;
- funkciją, kuri apskaičiuotų nupirktą prekių kiekį;
- funkciją, kuri apskaičiuotų kiek pinigų buvo išleista paskutinei nupirktai prekei.

### 6. Kilometrai

Pranas sugalvojo apskaičiuoti kiekvieną dieną tėčio nuvažiuojamą atstumą ir kuro sąnaudas. Tačiau kartais Pranas nevažiuoja su tėčiu, tad tėtis nurašo nuvažiuotą kilometražą ir kuro sąnaudas savo mylimam sūnui, kad šis galėtų atlikti savo skaičiavimus. Pranas aptingo ir sumanė, kad viską įmanoma automatizuoti ir paprašė jūsų pagalbos.

Failo "**Duomenys.txt**" pirmoje eilutėje įrašytas kelionių skaičius **n**. Kitose **n** eilučių įrašytas nuvažiuotas kelias ir kuro sąnaudos (double tipo).

Faile "**Rezultatai.txt**" reikia išvesti kiekvienos kelionės kuro sąnaudas šimtui kilometrų, visą nuvažiuotą kelią ir sunaudotą kurą, o paskutinėje eilutėje išvesti vidutines kuro sąnaudas šimtui kilometrų. Rezultatai double tipo, išvedami dviejų ženklų po kablelio tikslumu.

Duomenys.txt	Rezultatai.txt
3	26.09
11.5	3.0 32.89
15.2	5.0 6.67
120 8	146.70 16.00 10.91

## Reikalavimai

- Skaitymo funkcija void.
- Rašymo funkcija void.
- Kuro sąnaudų skaičiavimo funkcija void.
- Nenaudokite masyvų.

### 7. Kalėdinis karnavalas

Artėjant šventėms mokyklos bendruomenė organizuoja Kalėdinę šventę. Į ją taip pat buvo pakviestas Kalėdų Senelis. Padėkite renginio organizatoriams sudaryti renginio programą, įvertindami, jog Kalėdų senelis yra labai užimtas ir gali nespėti atvykti.

Pradiniai duomenys pateikti tekstiniame faile "**Duomenys.txt**". Pirmoje eilutėje nurodyti du sveikieji skaičiai – Kalėdų Senelio atvykimo laikas (valanda ir minutės). Antroje eilutėje pateikti du sveikieji skaičiai, nurodantys renginio pradžią (valanda ir minutės). Trečioje eilutėje nurodytas sveikasis skaičius  $n$ , kuris parodo, kiek dalių sudaro šio renginio programą. Kitose  $n$  eilutėse pateikti duomenys apie kiekvieną dalį atskirai: pavadinimas ir sveikasis skaičius – dalies trukmė minutėmis.

Jūsų užduotis – parašyti programą, kuri suskaičiuotų, kada prasideda ir kada baigiasi kiekviena renginio dalis, bei įtrauktų Kalėdų Senelio pasirodymą, jei jis atvyks. Žinoma, jog Kalėdų Senelis atvyksta ne anksčiau, nei prasideda renginys, bet gali atvykti pasibaigus renginiui. Dėl šios priežasties renginio organizatoriai neįtraukė jo dalies į pradinę programą. Yra sutarta, jog atvykęs Kalėdų Senelis negali pertraukti jau vykstančios dalies, todėl turi palaukti ir pasirodyti iš karto po jos. Kalėdų Senelio pasirodymo trukmė yra **10 minučių**.

Tekstiniame faile "**Programa.txt**" pateikite rezultatus (žr. pavyzdįje). Pirmoje eilutėje įrašykite tekstą "RENGINIO PROGRAMA:". Tolimesnėse eilutėse pateikite kiekvienos renginio dalies pavadinimą bei laikus, kada ji prasideda ir baigiasi (atskirtus tarpu). Išspausdinkite tekstą "PAPILDOMA INFORMACIJA:". Tada pateikite bendrą renginio trukmę ir žinutę "Kaledu senelis neatvyks", jei Kalėdų Senelis nespės į renginį.

#### 1 pavyzdys:

"Duomenys.txt"	"Programa.txt"
15 10 12 00 8 Sveikinimo_zodis 10 Protmusio_I_etapas 20 Ansamblis_"Vivos" 10 Protmusio_II_etapas 20 Sokeju_grupe_"Trepsis" 15 Pertrauka 10 Protmusio_finalas 30 Nugaletoju_paskelbimas 20	RENGINIO PROGRAMA: Sveikinimo_zodis 12:00 12:10 Protmusio_I_etapas 12:10 12:30 Ansamblis_"Vivos" 12:30 12:40 Protmusio_II_etapas 12:40 13:00 Sokeju_grupe_"Trepsis" 13:00 13:15 Pertrauka 13:15 13:25 Protmusio_finalas 13:25 13:55 Nugaletoju_paskelbimas 13:55 14:15 PAPILDOMA INFORMACIJA: Renginys truks 2val. 15min. Kaledu senelis neatvyks

## 2 pavyzdys:

"Duomenys.txt"	"Programa.txt"
12 15 12 00 8 Sveikinimo_zodis 10 Protmusio_I_etapas 20 Ansamblis_"Vivos" 10 Protmusio_II_etapas 20 Sokeju_grupe_"Trepis" 15 Pertrauka 10 Protmusio_finalas 30 Nugaletoju_paskelbimas 20	RENGINIO PROGRAMA: Sveikinimo_zodis 12:00 12:10 Protmusio_I_etapas 12:10 12:30 Kaledu_senelis 12:30 12:40 Ansamblis_"Vivos" 12:40 12:50 Protmusio_II_etapas 12:50 13:10 Sokeju_grupe_"Trepis" 13:10 13:25 Pertrauka 13:25 13:35 Protmusio_finalas 13:35 14:05 Nugaletoju_paskelbimas 14:05 14:25 PAPILDOMA INFORMACIJA: Renginys truks 2val. 25min.

### Reikalavimai

- Nenaudokite masyvų.
- Parašykite void funkciją su parametrais-nuorodomis, kuri apskaičiuotų viso renginio trukmę.
- Parašykite void funkciją su parametrais-nuorodomis, kuri apskaičiuotų kiekvienos renginio dalies pradžią ir pabaigą.
- Parašykite funkciją void su parametrais-nuorodomis, kuri patikrintų, kada įterpti Kalėdų Senelio dalį programoje.

## 8. Mokytojo Galvočiaus užduotis



**Parašykite programą**, kuri atliktų veiksmus su duotais 2 skaičiais: užrašytų šiuos skaičius iš kitos pusės, patikrintų ar naujieji skaičiai yra pirminiai. Jeigu gautieji skaičiai yra pirminiai, užrašytų didesniojo skaičiaus skaitmenų sumą, o jeigu ne – mažesniojo skaičiaus skaitmenų sumą.

**Pradiniai duomenys:** tekstiniame faile „Skaitmenys.txt“ pateikti pradiniai duomenys. Pateikiami du mokytojo Galvočiaus duoti skaičiai.

**Rezultatai:** tekstiniame faile „Rezultatai.txt“ pateikiami rezultatai. Pirmoje eilutėje užrašyti 2 skaičiai iš kitos pusės, kitoje eilutėje – skaitmenų suma pagal atitinkančią sąlygą (jeigu abu pirminiai – didesniojo skaičiaus skaitmenų sumą, kitu atveju – mažesniojo skaičiaus skaitmenų sumą).

### Pavyzdys:

Skaitmenys.txt	Rezultatai.txt
799 938	997 839 25

### Reikalavimai

- Skaičiai **int** tipo kintamieji;
- Sukurkite **void** tipo funkciją, kuri *skaitytų duomenis*;
- Sukurkite **void** tipo funkciją, kuri *grąžintų 2 iš kitos pusės skaičius*;
- Sukurkite **bool** tipo funkciją, kuri *patikrintų ar skaičius pirminis*;
- Sukurkite **int** tipo funkciją kuri *apskaičiuotų skaičiaus skaitmenų sumą*;
- **Nenaudokite** masyvų!

## 9. Nenaudėlis moksleivis

Vienas mokinys įsilaužė į klasės dienyno sistemą! Padėk jam "pakoreguoti" bendraklasių pažymius. Sistemos įsibrovėlis nori visiem savo draugam vienu balu pakelti pažymius, o savo priešams – vienu balu sumažinti.

Sukurk programą, kuri pakeistų bendraklasių pažymius, atsižvelgiant į jų statusus.

Faile **pazymiai.txt** pirmoje eilutėje pateiktas mokinių skaičius, kitose eilutėse pateikti mokinių pažymiai ir jų statusai. Rezultatų faile **rez.txt** kiekvienoje eilutėje turi būti pateikti pakoreguoti pažymiai. Atkreipk dėmesį, kad **pažymys gali būti tik intervale [1:10]**, todėl jei priešo pažymys = 1, tai jis daugiau nemažinamas.

pazymiai.txt	rez.txt
3	10
10 draugas	2
3 priešas	1
1 priešas	

### Reikalavimai

- Funkcija **void**, kuri pakeičia mokinio pažymį, atsižvelgiant į jo statusą.

## 10. Strateginio žaidimo kovos simuliacija

Strateginiame žaidime yra 5 karių rūšys. Kiekvienas karys turi sau priskirtą gynybos bei puolimo taškų skaičių. Programos užduotis – pagal suteiktus puolimo bei gynybos duomenis apskaičiuoti besiginančio žaidėjo kovoje prarastus karius bei nustatyti, ar gynyboje esantis žaidėjas laimėjo kovą.

Karių lentelė

Kario pavadinimas	Puolimo taškai	Gynybos taškai
Ietininkas	20	64
Lankininkas	54	54
Kryžiuotis	44	64
Barbaras	82	12
Raitas riteris	137	62

Įvesties paaiškinimas

Kiekvienas failas turi 10 eilučių duomenų. Eilutės skirstomos į dvi dalis po penkias eilutes. Eilutės nuo pirmos iki penktos imtinai priskiriamos puolėjo kariuomenei užrašyti. Toliau likusios eilutės (nuo šeštos iki dešimtos imtinai) yra priskiriamos besiginančio žaidėjo kariuomenei užrašyti.

Duomenų failas (data_input.txt)	Paaiškinimas
0	Puolėjo ietininkų skaičius
0	Puolėjo lankininkų skaičius
0	Puolėjo kryžiuočių skaičius
0	Puolėjo barbarų skaičius
0	Puolėjo r. riterių skaičius
0	Gynėjo ietininkų skaičius
0	Gynėjo lankininkų skaičius
0	Gynėjo kryžiuočių skaičius
0	Gynėjo barbarų skaičius
0	Gynėjo r. riterių skaičius

Skaičiavimų paaiškinimas

Kovos skaičiuojamos pagal puolėjo bendrą puolimo pajėgumą (**BPP**) bei gynėjo bendrą gynybos pajėgumą (**BGP**).

Skaičiuojant **BPP**, prieš tai nustatomas tam tikro **kario tipo** bendras visų karių pajėgumas.

**KARIO\_TIPO\_PAJĖGUMAS** = KARIŲ\_SKAIČIUS \* KARIO\_TIPO\_PUOLIMO\_TAŠKAI

Toliau, sumuojamas kiekvieno kario tipo pajėgumas su kitų karių pajėgumais. Ši suma yra **BPP** skaitinė išraiška.

Duomenų failas	Skaičiavimas
10	10 * 20 = 200
0	0 * 54 = 0
0	0 * 44 = 0
500	500 * 82 = 41000
100	100 * 137 = 13700
<b>BPP</b>	<b>54900</b>

Bendras gynėjo pajėgumas skaičiuojamas taip pat, tik **vietoj kario tipo puolimo taškų** naudoti to **kario tipo gynybos taškus**.

Žinant **BPP** ir **BGP**, pereinama prie laimėtojo nustatymo ir kovos aukų skaičiavimo.

Laimėtojas nustatomas lyginant **BPP** ir **BGP**. Jeigu **BPP** yra didesnis **arba lygus** bendro gynėjo pajėgumui, laikoma, kad puolėjas laimėjo kovą. Jeigu **BGP** yra didesnis už **BPP** – laimėjo gynėjas.

Siekiant nustatyti gynėjo prarastų karių skaičių, reikia apskaičiuoti pajėgumų santykį (**PS**).

Pajėgumų santykis skaičiuojamas taip:

jeigu **BPP** yra lygus **arba** didesnis už **BGP**:

**PS = BGP / BPP.**

Jeigu **BGP** didesnis už **BPP**:

**PS = BPP / BGP.**

Apskaičiavus **PS**, jis yra suapvalinamas iki 4 skaičių po kableliu ir toliau skaičiuojamos gynėjo aukos.

Jeigu gynėjas pralaimėjo, tai aukos (per kario tipą) skaičiuojamos taip:

$$\mathbf{KARIO\_TIPAS\_AUKOS = KARIŲ\_SKAIČIUS * (1 - PS)}$$

Jeigu gynėjas laimėjo, tai aukos (per kario tipą) skaičiuojamos taip:

$$\mathbf{KARIO\_TIPAS\_AUKOS = KARIŲ\_SKAIČIUS * PS}$$

Išvesties paaiškinimas

Pirmoje failo eilutėje išvesti **false**, jeigu gynėjas pralaimėjo, **true** – jeigu laimėjo. Toliau išrašyti prarastus **besiginančio žaidėjo karius**.

Išvesties failas (data_output.txt)	Paaiškinimas
true	Gynėjas laimėjo
41	Gynėjas prarado 41 ietininką
0	Gynėjas neprarado lankininkų
10	Gynėjas prarado 10 kryžiuočių
0	Gynėjas neprarado barbarų
0	Gynėjas neprarado r. riterių

Pavyzdys

Išvesties failas (data_input.txt)	Išvesties failas (data_output.txt)
0	false
0	437
0	874
0	0
5000	0
500	0
1000	
0	
0	
0	

**Reikalavimai**

- Nenaudoti masyvų.
- Skaičių tipų konversijai naudoti **static\_cast<type>(object);**  
[http://en.cppreference.com/w/cpp/language/static\\_cast#Example](http://en.cppreference.com/w/cpp/language/static_cast#Example)
- Naudoti **void** funkciją pajėgumų santykių (**PS**) skaičiavimui, kuri rezultatą grąžintų per parametrus.  
[https://www.tutorialspoint.com/cplusplus/cpp\\_function\\_call\\_by\\_reference.htm](https://www.tutorialspoint.com/cplusplus/cpp_function_call_by_reference.htm)
- Apvalinti pajėgumų santykį iki 4 skaičiaus po kableliu.  
Pavyzdys
- Skaičiuojant gynėjo aukų skaičių, reikšmę apvalinti naudojant **floor** funkciją.
- Naudokite įvesties failo vardą **data\_input.txt**, o išvesties **data\_output.txt**.

## 9. MASYVAI. SUMOS, KIEKIO, VIDURKIO SKAIČIAVIMAS

### 1. Krepšinio talentas

Aukštaūgis Tomas – savo miestelio krepšinio pažiba. Jis žaidžia krepšinį miestelio mėgėjų lygoje, tačiau nemato savo žaidimo statistikos. Sukurkite programą, kuri parodytų:

- taškų sumą **sumT**,
- atkovotų kamuolių sumą **sumK**,
- taškų vidurkį **vidT**,
- atkovotų kamuolių vidurkį **vidK**,
- atliktų dvigubų dublių skaičių sezone **dd**.

#### Pradiniai duomenys:

Rungtynių skaičius **n** ( $1 \leq n \leq 50$ ) įrašytas pirmoje eilutėje, pelnytų taškų skaičius **t** ir atkovotų kamuolių skaičius **k** įrašyti į kitas neilučių ir atskiriami tarpu **statistika.txt** duomenų faile.

#### Rezultatai:

Rezultatai yra saugomi tekstiniame faile **rezultatai.txt**. Pirmoje eilutėje įrašoma taškų suma **sumT** ir atkovotų kamuolių sumas **sumK**, atskiriant sumas tarpu. Antroje eilutėje įrašomas taškų vidurkis **vidT** ir atkovotų kamuolių vidurkis **vidK**, atskiriant vidurkius tarpu (vieno skaičiaus po kablelio tikslumu). Trečioje eilutėje įrašomas rungtynių skaičius **dd**, parodantis keliose rungtynėse Tomas pasiekė dvigubą dublį.

**Pastaba:** dvigubu dubliu vadinamas žaidėjo pasirodymas, kai du žaidėjo statistikos rodikliai yra dviženklis (šiuo atveju taškai ir atkovoti kamuoliai).

#### Duomenų ir rezultatų failų pavyzdžiai:

statistika.txt	rezultatai.txt
3 7 6 10 12 8 11	25 29 8.3 9.7 1
4 6 8 10 12 8 11 12 11	37 42 9.0 10.5 2
6 15 9 12 12 19 11 11 17 20 15 12 12	89 76 14.8 12.7 5



## Reikalavimai

- **t, k** – teigiami **int** tipo kintamieji;
- naudokite masyvus;
- void tipo funkcija **Skaityti()** duomenų skaitymui iš failo;
- void tipo funkcija **Rasyti()** rezultatų rašymui į failą;
- int tipo funkcija **Suma()** per funkcijos vardą gražinanti masyvo elementų sumą;
- int tipo funkcija **Dvigubi()** per funkcijos vardą gražinanti dvigubų dublių skaičių sezone.

### 2. Naujametinis pažadas

Naujų metų proga Justas pasižadėjo keisti savo gyvenimo būdą – jis nusprendė pradėti bėgioti. Užsibrėžęs pradinį tikslą – tam tikrą metrų skaičių **t**, jis pradėjo žymėti, kiek metrų kiekvieną dieną nubėgo. Padėkite Justui suskaičiuoti, kiek iš viso metrų jis nubėgo per **n** dienų, koks jo nubėgtų atstumų vidurkis ir kelias dienas jis pasiekė savo užsibrėžtą tikslą, jei kasdien po **p** metrų didino norimą nubėgti atstumą.

#### Pradiniai duomenys:

Duomenų failo „**atstumai.txt**“ pirmoje eilutėje yra pradinis atstumas metrais **t**. Antroje eilutėje yra metrų skaičius **p**. Trečioje eilutėje yra dienų skaičius **n** ( $n < 50$ ). Likusiose **n** eilučių yra Justo kiekvieną dieną nubėgti atstumai.

#### Rezultatai:

Rezultatų failo „**suvestine.txt**“ pirmoje eilutėje yra išvedama Justo iš viso nubėgtų metrų suma. Antroje eilutėje yra išvedamas Justo per dieną nubėgtų metrų vidurkis vienetų tikslumu. Trečioje eilutėje yra išvedamas dienų skaičius, per kurias Justas pasiekė arba viršijo užsibrėžtą tikslą.

atstumai.txt	suvestine.txt
3000	
300	
7	
3100	27600
3200	3942
4000	4
4200	
4350	
4150	
4600	
2500	
100	7625
3	2541
2425	1
2600	
2600	
4500	
500	
5	23270
4400	4654
4550	0
4600	
4720	
5000	

## Reikalavimai

- naudokite **masyvus**;
- **void** tipo funkcija **Skaityti()** duomenų skaitymui iš failo;
- **int** tipo funkcija **Sumuoti()** grąžinanti iš viso Justo nubėgtų metrų sumą per funkcijos vardą;
- **int** tipo funkcija **KiekIvykde()** grąžinanti dienų skaičių, kuriomis Justas įvykdė užsibrėžtą tikslą;
- **void** tipo funkcija **Rasyti()** rezultatų rašymui į failą.

### 3. Šaldytuvų parduotuvė

Pradinių duomenų faile nurodytas parduotuvėje esančių šaldytuvų kiekis **n** ir likusiose **n** eilučių – kiekvieno šaldytuvo gamintojas, modelis, talpa ir kaina. Tomas paprašė jūsų:

- Surasti skirtingų SIEMENS šaldytuvų modelių kiekį.
- Suskaičiuoti, kiek parduotuvėje yra šaldytuvų, kurių talpa yra didesnė už 100 litrų.
- Išspausdinti kiekvieną šaldytuvą (gamintoją bei modelį), kurio kaina yra didesnė už 1000 eurų. Jeigu nėra nei vieno tokio šaldytuvo, nespausdinkite nieko.

**PASTABA:** parduotuvėje gali būti keli to paties gamintojo ir modelio šaldytuvai.

Saldytuvai.txt	Rezultatai.txt
6	
SIEMENS KG54S8A 101 1001.99	2
SNAIGE AD55F88GH 101 1010.99	3
SAMSUNG GASDD879SS 99.99 999.99	SNAIGE AD55F88GH
SAMSUNG HSD898DD7 100 1000	SIEMENS GKKG787
SIEMENS GKKG787 87 1010	SIEMENS KG54S8A
SIEMENS KG54S8A 101 1001.99	

## Reikalavimai

- Naudoti masyvus.
- Tvarkingai struktūrizuoti kodą, naudoti funkcijas.

### 4. Maistingųjų medžiagų skaičiuoklė

Tavo užduotis – sukurti programą, kuri padėtų suskaičiuoti suvartojamą maistingųjų medžiagų kiekį ir įvertinti mitybos kokybę. Atlikdamas užduotį, remkis lentelių duomenimis apačioje.

Tekstiniame faile "Duomenys.txt" pateikti pradiniai duomenys: pirmoje eilutėje nurodytas sveikasis skaičius **n** – per parą suvartotų produktų skaičius. Kitose eilutėse yra **n double** tipo skaičių, atitinkamai nurodančių angliavandenių, baltymų bei riebalų kiekius **gramais**.

Tekstiniame faile "Rezultatai.txt" pateikiami skaičiavimų rezultatai. Pirmoje eilutėje nurodomas bendras angliavandenių, antroje – baltymų, trečioje – riebalų energijos kiekis **kalorijomis (kCal)**. Ketvirtoje eilutėje nurodytas bendras per parą gautas energijos kiekis kCal. Paskutinėje eilutėje atspausdintas tekstas "Taip" – jei mityba subalansuota, t.y. tenkina visus lentelėje "Subalansuotos mitybos rekomendacijos" punktus. Tekstas "Ne" reiškia, jog mityba nesubalansuota.

#### Kalorijų skaičius viename grame

Angliavandeniai	4
Baltymai	4
Riebalai	9

### Subalansuotos mitybos rekomendacijos

Angliavandeniai	55-65% energijos
Baltymai	15-20% energijos
Riebalai	maks. 30% energijos

(įskaitant nurodytus procentus)

#### Pavyzdys

Duomenys.txt
10 80 14.49 45 31.4 63.1 3.92 0 44.5 0 12 6.2 0.74 7.4 10.2 5.6 0.88 35.7 4.2 0 0.2 0.75 0.33 12 11 13.15 0.2 8.32 0.104 0 0.2

Rezultatai.txt
1177.64 284.48 414.486 1876.61 Taip

#### Reikalavimai

- Naudokite masyvus duomenims apie maistinių medžiagų kiekius saugoti.
- Parašykite void funkciją duomenų skaitymui iš failo.
- Parašykite void funkciją rezultatų įrašymui į failą.
- Parašykite funkciją double, kuri suskaičiuotų bendrą maistingųjų medžiagų kiekį gramais.
- Parašykite bool tipo funkciją, kuri patikrintų, ar mityba yra subalansuota.

## 5. Trejetukų žaidimas

Tomas ir Marius nusprendė pasivaržyti, kuris geriau sugeba spręsti matematinius uždavinius. Vienoje internetinėje užduočių sprendimo sistemoje jie pasirinko tris vienodus uždavinius ir juos išsprendė. Užduotys vertinamos šimtabalėje sistemoje. Parašykite programą, kuri įvertintų, kuris berniukas nugalėjo. Jei Tomas uždavinyje surinko daugiau balų, jam duodamas taškas, jei Marius – taškas atitenka jam, o jei berniukai surinko po vienodą balų skaičių – taško negauna nei vienas. Faile **balai.txt** pirmoje eilutėje pateikti Tomo rezultatai, o antroje – Mariaus, į failą **rezultatas.txt** įrašykite laimėjusio berniuko vardą, o jei abu surinko po tiek pat taškų, įrašykite žodį "lygiosios".

Pavyzdys:

balai.txt	rezultatas.txt
6 15 84 60 15 90	Marius

#### Reikalavimai

- **int** tipo masyvai, kuriuose saugomi kiekvieno berniuko balai;
- funkcija, kuri grąžina laimėjusio berniuko vardą arba "lygiosios" (**string rezultatas (int tomoBalai[], int mariausBalai[])**).

## 6. Statistinių duomenų analizė

Programai pateikiami skaičiai (imtis). Naudojantis žemiau pateikta informacija, suskaičiuoti:

- Imties vidurkį
- Imties plotį
- Imties dispersiją

Skaičiavimai

Skaičiuojant imties vidurkį, naudojama ši formulė:

$$\bar{x} = \frac{1}{n}(x_1 + x_2 + \dots + x_n)$$

*Paaiškinimas: sudedami visos imties skaičiai ir padauginama iš (1/n) (čia n – imties dydis). Skaičiuojant imties plotį, naudojama ši formulė:*

$$r = x_{max} - x_{min}$$

*Paaiškinimas: iš didžiausio skaičiaus imtyje atimamas tos pačios imties mažiausias skaičius.*

Skaičiuojant imties dispersiją, naudojama ši formulė:

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

*Paaiškinimas: iš kiekvieno imties elemento atimamas imties vidurkis. Šis skirtumas pakeliamas kvadratu. Taip padaroma su visais imties elementais ir gauti rezultatai susumuojami. Gauta suma padauginama iš (1/(n-1)) (čia n - imties dydis).*

Įvestis

Įvesties failas (**data\_input.txt**) turi tik vieną eilutę. Šioje eilutėje yra nežinomo dydžio teigiamų skaičių seka. Skaičiai atskirti tarpais.

Išvestis

Išvesties faile (**data\_output.txt**) spausdinti gautus skaičiavimus tokia tvarka:

1. Imties vidurkis
2. Imties plotis
3. Imties dispersija

Pavyzdys

Įvestis	Išvestis
38 33 25 3 39 28 33 35 41 19 10 15 18 47 20 40 47 28 13 19	27.55
	44.00
	159.42

### Reikalavimai

- Išvestį spausdinti paliekant du skaičius po kablelio (fixed, setprecision).
- Atskirti imties vidurkio, pločio bei dispersijos skaičiavimus į atskiras "double" arba "float" tipo funkcijas.
- Įvesties failą pavadinti **data\_input.txt**, o išvesties **data\_output.txt**.
- Jeigu naudosite masyvus, naudokite **vector** klasę.

## 10.MASYVAI. DIDŽIAUSIOS IR MAŽIAUSIOS REIKŠMĖS PAIEŠKA

### 1. Dėžės

Pranas neturėjo skaičiuotuvo ir pusę vasaros skaičiavo mintinai, tačiau darbe buvo nupirkta kompiuteris, kuris palengvintų jo darbą, tačiau jame nėra jokios programos, kuri padėtų, tad jis paprašė jūsų pagalbos. Jam reikia suskaičiuoti vidutinį dėžės svorį, surasti sunkiausią dėžę ir lengviausią dėžę, taip pat pasakyti, ar ją paveš keltuvas.

Faile "**Sandelys.txt**" rasite n, kiek yra dėžių, kitoje eilutėje pateikiami dėžių svoriai (**double**), trečioje eilutėje – keltuvo keliamoji galia (**double**).

Faile "**Rezultatai.txt**" reikia išvesti svorio vidurkį, sunkiausios dėžės numerį, lengviausios dėžės numerį ir "**Ne**", jei keltuvas nepaveš, arba "**Taip**", jei galės pavežti.

Sandelys.txt	Rezultatai.txt
3	14.12
10.2            11.5            20.65	2
20	0
	Ne

#### Reikalavimai

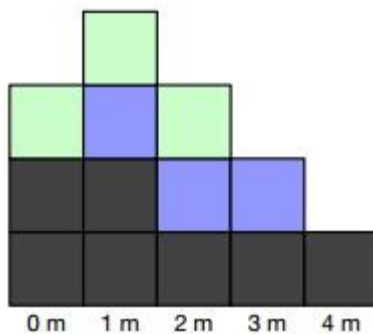
- Skaitymo funkcija void.
- Išvedimo funkcija void.
- Lengviausios dėžės numerio funkcija, grąžinanti dėžės numerį per funkcijos vardą.
- Sunkiausios dėžės numerio funkcija, grąžinanti dėžės numerį per funkcijos vardą.
- Vidurkio skaičiavimo funkcija, grąžinanti vidutinį dėžės svorį per funkcijos vardą.
- Loginė funkcija, grąžinanti reikšmę true, jei keltuvas paveža, arba false, jei nepaveža.

### 2. Karjeras

Ši sistema gali vienu metu iškasti viso karjero pločio duobę, kurios pradžia yra  $S_i$ , pabaiga –  $E_i$  metrų nuo karjero pradžios, o gylis –  $H_i$  metrų.

Sklypą (karjerą) galima pavaizduoti kaip dvimatį brėžinį, kuris vaizduoja karjero aukštį virš jūros lygio išilgai karjero, tiek pradiniu momentu, tiek vėliau, pradėjus naudingųjų iškasenų kasimą.

Automatinė kasimo sistema dirba vykdydama kasimo komandas. Kiekviena komanda sumažins karjero aukštį  $H_i$  metrų nurodytame intervale. 1 pav. parodyta, kaip keičiasi karjero aukštis įvykdžius dvi komandas, kurių pirma kasa intervale  $[0; 2]$ , antra –  $[1; 3]$ , abi iškasa 1 metro gylio duobes. Komandų iškasami plotai atitinkamai pažymėti žaliai ir mėlynai.



1 pav.

Karjero savininkas žino, kokias kasimo komandas duos šiai sistemai, bet prieš pradėdamas nori sužinoti, kaip atrodys karjeras įvykdžius visas komandas.

**! Parašykite programą, kuri surastų karjero aukštį įvykdžius visas kasimo komandas.**

**Pradiniai duomenys.** Pirmoje eilutėje pateikiami du sveikieji skaičiai: karjero ilgis  $N$  ir komandų skaičius  $K$ . Antroje eilutėje pateikiama  $N$  sveikųjų skaičių  $A_i$  – pradinis karjero aukštis virš jūros lygio  $i$  metrų nuo karjero pradžios. Pirmoji reikšmė nurodo aukštį  $0$  metrų nuo karjero pradžios. Toliau pateikiama  $K$  eilučių, kuriose yra po tris sveikuosius skaičius:  $i$ -tosios kasimo komandos pradžia  $S_i$ , pabaiga  $E_i$  ir gylis  $H_i$ .

**Rezultatai.** Išveskite  $N$  skaičių, kurių  $i$ -tasis nurodytų, koks būtų karjero aukštis  $i$  metrų nuo karjero pradžios įvykdžius visas komandas (galutinis aukštis gali būti neigiamas).

**Pavyzdys (atitinka 1 pav.):**

Duomenys.txt	Rezultatai.txt
5 2 3 4 3 2 1 0 2 1 1 3 1	2 2 1 1 1

### Reikalavimai

- **int** tipo kintamieji;
- Sukurkite **masyvą**, kuriame saugomi pradiniai karjero aukščiai virš jūros lygio  $i$  metrų nuo karjero pradžios;
- Sukurkite **void** tipo funkciją, kuri skaitytų duomenis.

### 3. Filmai

Duomenų failo pirmoje eilutėje parašytas filmų skaičius  $n$  ( $0 < n \leq 100$ ). Sekančiose  $n$  eilučių atitinkamai surašyti filmų biudžetai (naudokite long long tipą) bei IMDB reitingai (naudokite double tipą). Jums reikia:

- Rasti filmo numerį sąrašė, kurio IMDB reitingas yra didžiausias (jeigu yra keli tokie filmai, spausdinkite pirmą sąrašė).
- Rasti filmo numerį sąrašė, kurio biudžetas yra mažiausias (jeigu yra keli tokie filmai, spausdinkite paskutinį sąrašė).
- Išspausdinti visus filmus (jų sąrašė numerius), kurių biudžetas yra mažesnis už 100 000 000 JAV dolerių bei kurių IMDB reitingas yra didesnis už 8.5 (jeigu nėra nei vieno tokio filmo, nespausdinkite nieko).

Filmai.txt	Rezultatai.txt
6 1000000 8.6 120000 8.5 20000000 8.6 200000000 5.8 200000000 8.6 120000 7.7	1 6 1 3

### Reikalavimai

- Tvarkingai struktūrizuoti programos kodą, naudojant funkcijas.
- Naudoti masyvus.
- Nekeisti pradinių duomenų.
- Atlikdami paskutinį punktą, formuokite naują masyvą.

### 4. Grojaraštis

Artėjant kelionei automobiliu, nusprendėte įsigyti kompaktinį diską. Norite įsirašyti jums patinkančias dainas, kad nereikėtų klausytis nuobodžių radijo laidų ar nemėgstamos muzikos. Tekstiniame faile "Duomenys.txt" pateikti pradiniai duomenys. Pirmoje eilutėje nurodytos minutės, nusakančios disko **talpą**. Antroje eilutėje nurodytas sveikasis skaičius  $n$  - norimų įrašyti

dainų skaičius. Kitose n eilučių pateikiami du skaičiai, nurodantys dainos **trukmę** (minutės ir sekundės). Žinoma, jog maksimali dainos trukmė yra 15 minučių.

Suskaičiuokite, kokia yra grojaraščio su n dainų trukmė, ir įvertinkite, ar užteks disko talpos visoms dainoms įrašyti. Jeigu grojaraščio trukmė didesnė nei galima įrašyti, tuomet ištrinkite tas dainas, kurių trukmė grojaraštyje yra pati mažiausia, tol, kol dainos tilps diske.

Tekstiniame faile "Rezultatai.txt" pateikti skaičiavimų rezultatai. Pirmoje eilutėje nurodyta grojaraščio trukmė (minutėmis ir sekundėmis). Antroje eilutėje įrašytas tekstas „Taip“, jei visas grojaraštis telpa diske. Kitu atveju yra atspausdinamas tekstas „Ne“, taip pat nurodomas ištrintų dainų skaičius bei naujo grojaraščio trukmė (minutėmis ir sekundėmis).

### Pavyzdys

Duomenys.txt	Rezultatai.txt
80	85min. 39s.
16	Ne
5        25	3
3        15	78min. 14s.
6        29	
4        44	
7        45	
2        10	
3        55	
2        0	
4        58	
5        23	
6        52	
7        12	
4        38	
5        55	
8        49	
6 9	

### Reikalavimai

- Naudokite masyvus duomenims apie dainų trukmes saugoti.
- Parašykite void funkciją duomenų skaitymui iš failo.
- Parašykite void funkciją rezultatų įrašymui į failą.
- Parašykite funkciją, kuri suskaičiuotų grojaraščio trukmę.
- Parašykite funkciją, kuri grąžintų ištrintų dainų skaičių.

## 5. Kriptovaliutos

Du verslininkai, pamatę kylančias kriptovaliutų kainas, sumanė į jas investuoti. Vis dėlto, prieš investuojant reikėjo padaryti analizę, kuri atneš daugiausia pelno (Bitcoin, Litecoin, Ethereum, Zcash ir t.t.). Taigi, surinkę pastarųjų savaitių duomenis (kriptovaliutos kainas), padėkite jiems nuspręsti, kuri kriptovaliuta atneš daugiausia pelno.

Nuskaitykite duomenis iš failo **Duomenys.txt**. Pirmoje eilutėje pateikta, kelių savaitių duomenys surinkti. Antroje eilutėje pateiktas skirtingų kriptovaliutų kiekis. Trečioje ir tolesnėse eilutėse nurodytos kriptovaliutų kainos kiekvieną savaitę.

1. Duomenys.txt	1. Rezultatai.txt
2 10 1000.5 10 15 459.6 679 123 551 923 260 10 1150.7 50 10 400.3 892 124 580 1203 290 9.50	Valiuta ir jos pokytis 1.15.0125% 2.400% 3.-33.3333% 4.-12.9025% 5.31.3697% 6.0.813008% 7.5.26316% 8.30.3359% 9.11.5385% 10.-5% Daugiausia kaina pakilo 2 valiutos. Maziausia kaina pakilo/nusmuko 3 valiutos.
2 10 2331.5 1000.45 15 3452.6 987 1002 15 789 365 5 5120.7 957.4 10 3654.3 892 1004 98 987 290 9.98	Valiuta ir jos pokytis 1.119.631% 2.-4.30306% 3.-33.3333% 4.5.84197% 5.-9.62513% 6.0.199601% 7.553.333% 8.25.0951% 9.-20.5479% 10.99.6% Daugiausia kaina pakilo 7 valiutos. Maziausia kaina pakilo/nusmuko 3 valiutos.
2 10 31.5 564 0.7 824.6 1100.5 321 442 9233 491 10000.56 42.7 694 1.4 743.3 945 421 440 12310 491 11111.56	Valiuta ir jos pokytis 1.35.5556% 2.23.0496% 3.100% 4.-9.85933% 5.-14.1299% 6.31.1526% 7.-0.452489% 8.33.3261% 9.0% 10.11.1094% Daugiausia kaina pakilo 3 valiutos. Maziausia kaina pakilo/nusmuko 5 valiutos.

### Reikalavimai

- Kiekvienos savaitės kainas nuskaitykite į naujus **masyvus**: masyvo indeksas atitinka kriptovaliutą, reikšmė - jos kainą.
- Parašykite **funkciją**, kuri apskaičiuotų kriptovaliutų kainų kilimą/kritimą procentais nuo pirmos savaitės.
- Parašykite **funkciją**, kuri surastų, kurios kriptovaliutos vertė daugiausia pakilo.
- Parašykite **funkciją**, kuri surastų, kurios kriptovaliutos vertė mažiausia pakilo.

### 6. Stovykla

Krepšinio treneris nori pačius gambiausius sportininkus išleisti į krepšinio stovyklą. Dėl šios priežasties jis visą sezoną suvedinėjo statistiką ir nusprendė išrinkti tuos krepšinininkus, kurių naudingumą įvertino **m** balų ir daugiau. Parašykite programą, kuri rastų:

- tų krepšinininkų, kurie buvo išrinkti į stovyklą, numerių sąrašą;
- mažiausią naudingumo balą iš pradinio krepšinininkų sąrašo;
- didžiausią naudingumo balą iš pradinio krepšinininkų sąrašo.



### Pradiniai duomenys:

Duomenų failo **krepsininkai.txt** pirmoje eilutėje įrašytas krepšininkų skaičius  $n$  ( $1 < n < 50$ ) ir naudingumo balas  $m$ . Kitose eilučių įrašytas žaidėjo numeris ir naudingumo balas, kurie yra atskiriami tarpu. Vieno žaidėjo numeriui ir naudingumo balui yra skiriama viena eilutė.

### Rezultatai:

Rezultatai yra saugomi tekstiniame faile **atrinkti.txt**. Rezultatų failo pirmoje eilutėje yra išvedamas mažiausias ir didžiausias naudingumo balai, atskirti tarpu. Antroje rezultatų failo eilutėje yra išvedamas išrinktų krepšininkų numerių sąrašas kiekvieno krepšininko numerį atskiriant tarpu.

### Duomenų ir rezultatų failų pavyzdys:

krepsininkai.txt	atrinkti.txt
7 6.8	4.9 10.2
3 10.2	3 2 9 22
12 5.3	
2 6.8	
9 7.5	
14 5.1	
10 4.9	
22 7.1	

### Reikalavimai

- Išrinktų krepšininkų numeriai turi būti talpinami masyve.
- Funkcijos void duomenų skaitymui ir rašymui.
- Funkcija krepšininkų numerių išrinkimui.
- Funkcijos mažiausiam ir didžiausiam naudingumo balams rasti.

## 7. Slidinėjimas

Justas labai mėgsta slidinėti. Šį žiemos sezoną jis nusprendė atnaujinti savo slidinėjimo įrangą – nusipirkti naujas slides ir batus. Apsilankęs parduotuvėje jis pamatė, kad slidės ir batai pardavinėjami komplektais. Justas nori sumokėti ne mažiau nei **min** bet ne daugiau nei **max** eurų, kad nusipirktų kokybišką, bet nebrangų komplektą. Parduotuvė parduoda  $n$  ( $n \leq 50$ ) skirtingų komplektų.

Sukurkite programą, kuri parodytų:

1. Pigiausio ir brangiausio komplektų kainas, atitinkančias norimą sumą.
2. Komplektų, kainuojančių norimą sumą, kiekį.
3. Visų Justui tinkančių komplektų kainas.

Pradiniai duomenys: **mažiausia išleidžiama pinigų suma min, didžiausia išleidžiama pinigų max, komplektų kiekis parduotuvėje n**, įrašyti tekstiniame faile **kainos.txt** vienoje eilutėje ir atskiriami tarpais. Kitose  $n$  eilučių įrašomos ir tarpu atskiriamos vieno komplekto slidžių ir batų kainos. Rezultatai yra saugomi tekstiniame faile **rezultatai.txt**. Į pirmą eilutę įrašomos ir tarpu atskiriamos **mažiausia ir didžiausia tinkamų komplektų kainos**. Antroje eilutėje – **tinkamų komplektų kiekis k**. Trečioje eilutėje atskiriamos tarpu ir įrašomos **visos tinkamų komplektų kainos**.

Pradiniai duomenys	Rezultatai
100 200 4	
80 30	101 198
75 26	3
150 70	110 101 198
190 8	

## Reikalavimai

- **min,max,n** – teigiami **int** tipo kintamasis.
- **Naudoti masyvus.**
- Sukurti **int** tipo funkciją, kuri gražintų brangiausią komplektą,
- Sukurti **int** tipo funkciją, kuri gražintų pigiausią komplektą,
- Sukurti **int** tipo funkciją, kuri iš dviejų masyvų sudeda informaciją į naują masyvą ir gražina naujo masyvo ilgį.

### 8. Stalo žaidimų parduotuvė

Stalo žaidimų parduotuvės konsultantas Simonas nusprendė suskaičiuoti priedą prie savo mėnesinio atlyginimo. Nuo kiekvieno parduoto stalo žaidimo jis gauna 10% žaidimo kainos prie atlyginimo priedo.

**! Parašykite programą,** kuri suskaičiuotų Simono mėnesinio atlyginimo priedą ir rastų tą stalo žaidimą, iš kurio Simonas gavo didžiausią dalį priedo.

**Pradiniai duomenys.** Tekstiniame faile „**Zaidimai.txt**“ pateikti pradiniai duomenys. **Pirmoje failo eilutėje** nurodyta, kiek skirtingų stalo žaidimų tipų **n** yra parduotuvėje. Likusiose failų eilutėse nurodyta vieno stalo žaidimo kaina **k** (realus skaičius) ir parduotų žaidimų kiekis **y** (sveikasis skaičius).

**Rezultatai.** Tekstinio failo „**Priedas.txt**“ pirmoje eilutėje pateikiamas Simono priedas prie atlyginimo (realusis skaičius, du skaitmenys po kablelio), antroje eilutėje pateikiama kaina to stalo žaidimo, iš kurio Simonas gavo didžiausią dalį priedo.

#### Pavyzdys:

Zaidimai.txt	Priedas.txt
5	
8.13	3
50.99	1 14.84
20.50	2 50.99
4.65	4
13.45	1

## Reikalavimai

- Sukurkite **void** tipo funkciją, kuri *skaitytų duomenis*;
- Sukurkite **void** tipo funkciją, kuri *spausdintų duomenis*;
- Sukurkite **double** tipo funkciją, kuri *grąžintų nuo vieno stalo žaidimo gaunamą dalį prie priedo*;
- Sukurkite **double** tipo funkciją, kuri *grąžintų priedą prie atlyginimo*;
- Sukurkite **double** tipo funkciją, kuri *grąžintų kainą to stalo žaidimo, iš kurio Simonas gavo didžiausią dalį priedo*;
- Skaičių apvalinimas vyksta pateikiant tik **galutinį rezultatą**;
- Naudokite **masyvus**!

# 11.MASYVO ELEMENTŲ ŠALINIMAS. NAUJŲ ELEMENTŲ ĮTERPIMAS

## 1. Staigmena

Mokslo metų pabaigos šventei Antanas sugalvojo padaryti staigmeną klasės draugams. Jis kompiuteriu atsitiktinai sugeneravo  $n$  ( $n \leq 100$ ) natūraliųjų skaičių ir padarė korteles su tais skaičiais. Kiekvieno kortelę ištraukusio draugo laukia staigmena: atvirukas, kalendoriukas, saldainis ir pan.

Antanas draugams korteles davė ištraukti prieš mokslo metų pabaigą, ant skirtingų popieriaus skiaučių užsirašė, kuris draugas kurią kortelę ištraukė, tačiau vieną popieriaus skiautę su ant kortelių surašytais skaičiais ir draugų vardais pametė. Popieriaus skiautę surado tik paskutinį vakarą prieš mokslo metų pabaigą. Tai buvo stagimena pačiam Antanui. Jam reikia įterpti rastoje skiautėje įrašytus kortelių numerius į jau esamą sąrašą, kad žinotų, kokią staigmeną kuriam draugui padaryti. Pirminiame Antano sąrašė kortelės surašytos taip, kad ant jų užrašyti skaičiai yra išdėstyti didėjančiai.

Parašykite programą, kuri į esamą sąrašą reikiamose vietose įterptų ant rastos popieriaus skiautės surašytus duomenis.

Pradinių duomenų failo staigmena.txt pirmoje eilutėje įrašyta, kiek kortelių yra pirminiame Antano sąrašė. Antroje eilutėje išvardinti ant kortelių surašyti skaičiai didėjimo tvarka. Trečioje failo eilutėje įrašyta, kiek kortelių yra surastoje popieriaus skiautėje. Ketvirtoje eilutėje – kokie skaičiai užrašyti ant kortelių. Skaičiai išdėstyti atsitiktine tvarka.

Rezultatų faile staigmenarez.txt turi būti išvestas galutinis sąrašas, gautas įterpus į pirminį sąrašą duomenis iš popieriaus skiautės. Vienai kortelei skiriama viena eilutė.

Pradiniai duomenys	Rezultatai
	5
	7
5	8
5 8 9 12 15	9
3	12
7 14 12	12
	14
	15

### Reikalavimai

- Pradinių duomenų skaitymo funkcija void.
- Rezultatų rašymo funkcija void.
- Funkcija, surandanti vietą, kur reikia įterpti naują kortelę.
- Vienos kortelės įterpimo funkcija.
- Programoje gali būti daugiau funkcijų.
- NEGALIMA formuoti naujo sąrašo. Ant popieriaus skiautės surašyti duomenys turi būti įterpti į pradinį sąrašą.

## 2. Juodųjų skylių spiečiaus virtimas viena skyle

Adomas sugalvojo sumodeliuoti juodosios skylės susiformavimą iš spiečiuje esančių juodųjų skylių. Jis surašė  $n$  natūraliųjų skaičių. Kiekvienas iš tų skaičių yra laikomas spiečiaus juodąja skyle. Pirmasis skaičius lyginamas su antruoju. Jei pirmasis skaičius didesnis už antrąjį, tuomet jis laikomas juodąja skyle ir įtraukia antrąjį, jei antrasis didesnis, tuomet jis bus laikomas juodąja skyle ir įtrauks pirmąjį skaičių. Kai juodoji skylė įtraukia kaimyną, ji padidėja vienetu. Jei kaimyninės skylės yra vienodo dydžio, tuomet jos susilieja ir tampa dvigubai didesne juodąja

skyle. Veiksmai kartojami tol, kol lieka vienas skaičius. Jis ir bus laikomas galutine juodaja skyle.

Pradinių duomenų faile spiecious.txt nurodytas juodųjų skylių skaičius spiečiuje ir išvardinti juodųjų skylių dydžiai.

Rezultatų faile spieciousrez.txt turi būti išspausdintas galutinis juodosios skylės, susiformavusios iš spiečiaus skylių, dydis.

Pradiniai duomenys	Paaiškinimas
5 12 15 18 12 29	12 15 18 12 29 Pirmąją juodąją skylę įtraukia antroji: 16 18 12 29 Pirmąją juodąją skylę įtraukia antroji: 19 12 29 Antrąją juodąją skylę įtraukia pirmoji: 20 29
Rezultatai	Pirmąją juodąją skylę įtraukia antroji: 30

### Reikalavimai

- Pradinių duomenų skaitymo funkcija void;
- Rezultatų rašymo funkcija void;
- Vieno elemento šalinimo funkcija;
- Funkcija, grąžinanti galutinį juodosios skylės dydį.

### 3. Iššūkis vienuoliktokams

Vienos mokyklos vienuoliktokams, besimokantiems informatiką A kursu, labai sunkiai sekasi programavimas. Vienuoliktokai pradėjo derybas su mokytoja dėl pusmečio galutinio įvertinimo skaičiavimo. Vienuoliktokai mokytojai pasiūlė išvesti pusmečio įvertinimą atmetus už programavimą gautus pažymius.

Mokytoja sutiko su vienuoliktokų siūlomomis taisyklėmis, bet tik su sąlyga, jei jie parašys programą, skaičiuojančią, koks bus įvertinimas išbraukus iš pažymių sąrašo už programavimą gautus pažymius (naujas pažymių sąrašas negali būti formuojamas, už programavimą gauti pažymiai yra neigiami skaičiai) ir apskaičiuotų likusių pažymių vidurkį.

Vienuoliktokai nepriėmė mokytojos iššūkio. Pabandykite šį iššūkį įveikti jūs.

Pradinių duomenų failo pazymiai.txt pirmoje eilutėje įrašytas mokinių skaičius n. Tolesnėse n eilučių įrašyta informacija apie kiekvieną mokinį: mokinio kodas (triženklis natūralusis skaičius), kelis pažymius mokinyso gavo per pusmetį ir toliau mokinio gauti pažymiai.

Rezultatų faile pazymiairez.txt įrašykite kiekvieno mokinio kodą ir įvertinimą, apskaičiuotą pagal naujas taisykles.

Pradiniai duomenys	Rezultatai
3	
100 5 7 -4 8 -3 8	100 8
110 4 8 -2 7 -2	110 8
111 6 -4 10 9 -3 8 9	111 9

### Reikalavimai

- Pradinių duomenų skaitymo funkcija (joje gali būti kreipiniai į kitas funkcijas).
- Vieno elemento šalinimo funkcija.
- Funkcija, grąžinanti apskaičiuotą vidurkį.
- Rezultatų rašymo į failą funkcija.

#### 4. Monetų kolekcionieriai

Domantas ir Simonas yra aistringi monetų kolekcionieriai. Jie ne tik kolekcionuoja monetas, bet ir jomis keičiasi.

Pradinių duomenų failo monetos.txt eilutėje įrašyti du sveikieji skaičiai  $d$  ir  $s$  ( $d \leq 100$ ,  $s \leq 100$ ) – Domanto ir Simono kolekcijose esančių monetų kiekiai. Tolesnėje eilutėje įrašyta  $d + s$  sveikųjų skaičių – kolekcijose esančių monetų nominalai (monetų nominalai gali būti natūralieji skaičiai nuo 1 iki 1000). Pirmieji  $s$  skaičių – Domanto kolekcija, likusieji – Simono. Monetų nominalai išdėstyti atsitiktine tvarka ir gali kartotis, pvz.: 1 5 3 1 3 3 5.

Draugai keičiasi monetomis tuomet, kai vienas iš jų neturi tam tikro nominalo monetos, o kitas turi daugiau negu vieną tokią monetą. Keičiantis monetomis iš draugo, kuris atiduoda monetą, sąrašo pašalinama sąrašo pirma esanti tokio nominalo moneta, kito draugo sąrašo ji įterpiama prieš pirmą rastą didesnio nominalo monetą.

Parašykite programą, kuri rezultatų faile monetosrez.txt išvestų draugų kolekcijas jiems pasikeitus monetomis. Pirmoje eilutėje išvedama Domanto, antroje – Simono kolekcija.

Pradiniai duomenys	Rezultatai
5 7 2 5 7 9 2 7 1 9 4 3 4 1	1 4 5 7 9 2 2 7 9 3 4 1

#### Reikalavimai

- Pradinių duomenų skaitymo funkcija.
- Loginė funkcija, tikrinanti, ar kolekcionierius turi tam tikro nominalo monetą.
- Loginė funkcija, tikrinanti, ar kolekcionierius turi kelias tam tikro nominalo monetas.
- Funkcija, gražinanti vietą, iš kurios reikia šalinti elementą.
- Funkcija, gražinanti vietą, kur reikia įterpti elementą.
- Elemento įterpimo funkcija.
- Elemento šalinimo funkcija.
- Funkcija, kurioje yra keitimosi mechanizmas (kreiptis 2 kartus).
- Rezultatų rašymo funkcija (kreiptis 2 kartus).

## 12.SIMBOLIAI. SIMBOLIŲ EILUTĖS

### 1. Anagrama

Žodis vadinamas kito žodžio anagrama tada, kai sutampa **ir abiejų žodžių raidės, ir jų kiekis**. Perskaitykite duomenis iš failo ir atlikite veiksmus su simboliais taip, kad patikrintumėte, ar iš pirmojo žodžio galima sudaryti antrąjį, kad neliktų nė vienos nepanaudotos raidės. Duomenų faile **anagrama.txt** pirmasis žodis pateiktas pirmoje eilutėje, antrasis – antroje. Rezultatų faile **anagramarez.txt** turi būti įrašytas žodis **taip**, jei iš pirmojo žodžio galima sudaryti antrąjį, arba žodis **ne**, jei iš pirmojo žodžio negalima sudaryti antrojo. Pradinių duomenų faile žodžiai bus pateikti tik mažosiomis raidėmis.

anagrama.txt	anagramarez.txt
semti mesti	taip
likti kelti	ne

#### Reikalavimai

- Pradinių duomenų skaitymo funkcija **void()**.
- Tikrinimui, ar žodis yra anagrama, sukurti loginę funkciją.

## 2. Stalo žaidimas „Informacijos matavimo vienetai“

Informatikos mokytoja, norėdama, kad devintokai greičiau išmokytų įvertinti informacijos kiekį, pasiūlė mokiniams pažaisti stalo žaidimą „Informacijos matavimo vienetai“. Ji padarė kortelių, ant kurių surašyti informacijos kiekio matavimo vienetai. Kortelės išdalinamos žaidime dalyvaujantiems mokiniams po lygiai. Žaidėjai iš eilės deda po vieną kortelę ant stalo ir visas korteles nuo stalo pasiima žaidėjas, ant kurio kortelės buvo užrašytas didžiausias informacijos matavimo kiekio vienetas. Jei keli mokiniai padėjo po kortelę, ant kurios užrašytas didžiausias matavimo vienetas, tai jie pasiima kiekvienas savo kortelę, o likusios kortelės išimamos iš žaidimo. Nuo stalo paimtos kortelės eina į žaidėjo banką, o tolesniam žaidimui imamos naujos kortelės iš dar turimų išdalintų.

Žaidimą laimi tas mokinytis, kuris surenka daugiausiai taškų. Laimėtojų gali būti ir ne vienas. Ant kortelių surašyti simboliai ir simbolių eilutės: b – bitas (kortelės vertė 1 taškas), B – baitas (kortelės vertė 2 taškai), kB – kilobaitas (kortelės vertė 3 taškai), MB – megabaitas (kortelės vertė 4 taškai), GB – gigabaitas (kortelės vertė 5 taškai), TB – terabaitas (kortelės vertė 6 taškai), PB – petabaitas (kortelės vertė 7 taškai), EB – eksabaitas (kortelės vertė 8 taškai), ZB – zetabaitas (kortelės vertė 9 taškai), YB – jotabaitas (kortelės vertė 10 taškų).

Pradinių duomenų failo **zaidimas.txt** pirmoje eilutėje įrašytas kortelių skaičius **n** ir žaidime dalyvaujančių mokinių skaičius **m** (**n visada be liekanos dalijasi iš m, mažiausia n reikšmė lygi m reikšmei,  $2 < m < 6, n < 31$** ). Tolesnėse n failo eilučių surašyta po vieną informacijos kiekio vieneto pavadinimą. Kortelės imamos iš eilės ir dalinamos kiekvienam mokiniui. Kai išdalinamos pirmosios kortelės, toliau vėl dalinama ir taip tol, kol bus išdalintos visos kortelės. Gavę korteles, mokiniai ima jas iš eilės tokia tvarka, kokia kortelės buvo išdalintos ir deda ant stalo. Laikantis žaidimo taisyklių, žaidėjai ima korteles nuo stalo ir deda į banką. Žaidimo pabaigoje apskaičiuojamas kiekvieno žaidėjo surinktas taškų skaičius ir skelbiami laimėtojai.

Rezultatų faile **zaidimasrez.txt** turi būti **m** eilučių su kiekvieno žaidėjo banke esančiomis kortelėmis, kurios viena nuo kitos atskiriamos tarpais ir surinktų taškų skaičius. Vienam žaidėjui skiriama viena eilutė. Žaidėjo banke korteles dėliokite taip, kaip jos buvo išdalintos. Jei kuris nors žaidėjas banke neturi nė vienos kortelės, tuomet žaidėjo eilutėje turi būti įrašytas nulis.

Paskutinėje rezultatų failo eilutėje turi būti nurodytas žaidimo laimėtojo numeris. Jei laimėtojai yra keli, jų numeriai turi būti atskirti tarpais ir išdėstyti didėjimo tvarka. Žaidėjai numeruojami iš eilės nuo 1 iki m.

zaidimas.txt	zaidimasrez.txt
6 2 ZB TB GB B b MB	ZB TB GB B 22 b MB 5 1  MB b 5 TB ZB B GB 22
6 2 TB ZB B GB MB b	2

## Reikalavimai

- Naudokite simbolių eilučių masyvus.
- Funkcija, gražinanti didžiausią taškų skaičių per funkcijos vardą. Funkciją galite panaudoti keletą kartų: ieškodami ant stalo sudėtų kortelių didžiausio taškų skaičiaus ir ieškodami geriausių žaidėjų.
- Funkcija, susiejanti informacijos matavimo vienetų simbolius su jų verte taškais. Funkcija gražina suformuotus taškų ir matavimo vienetų simbolių eilučių masyvus, kuriuos susieja masyvų indeksai.
- Rašymo funkcija void () .
- Programoje patartina naudoti ir daugiau funkcijų. Jas sukurkite savo nuožiūra.

### 3. ATBASH slaptaraštis

Atsiradus raštui, kurį supranta visi raštingi žmonės, natūralu, kad atsirado poreikis ieškoti būdų, skirtų informacijai paslėpti (užkoduoti). Pirmieji skaitymą „apsunkino“ egiptiečiai, kurie vietoj raidžių naudojo savus hieroglifus, tačiau dar iki šiol nėra aišku, ar taip buvo daroma dėl to, kad informaciją galėtų perskaityti tik tie, kam tas raštas skirtas, ar dėl kitų priežasčių. Tačiau yra aišku, kad slaptaraštį tikrai naudojo žydų raštininkai. Vienas iš jų naudojamų slaptaraščių – ATBASH, kurio esmė, kad vietoj vienerių raidžių yra naudojamos kitos. Sukurkite programą, kuri iššifruotų eilutes, užkoduotas ATBASH slaptaraščiu.

ATBASH slaptaraščio esmė tokia, kad pirmoji pateiktos abėcėlės raidė atitinka paskutiniąją, antroji – priešpaskutinę ir t.t. Pavyzdžiui, lietuviškai abėcėlei būtų pritaikomi tokie raidžių pakeitimai:

A	Ą	B	C	Č	D	E	Ę	È	F	G	H	I	Į	Y	J
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Ž	Z	V	U	Ū	Ų	T	Š	S	R	P	O	N	M	L	K

Tačiau Jūsų užduotis: sukurti programą, kuri naudoja **lotynišką 26 raidžių abėcėlę** ir iššifruoja tekstiniame faile pateiktus sakinius (iššifruojama kiekviena faile pateikta eilutė).

Programos duomenys: duomenų failo **duomenys.txt pirmoje eilutėje** – tekstiniame faile esančių užšifruotų eilučių kiekis (n). Kitose n eilučių pateiktos ATBASH šifru užkoduotos eilutės.

Programos rezultatai: rezultatų failo **rezultatai.txt** atskirose eilutėse surašytos iššifruotos duomenų faile buvusios eilutės (kiekvieną duomenų failo eilutę atitinka viena rezultatų failo eilutė).

Pradiniai duomenys	Rezultatai
1 Ozyzh	Labas
2 Ozyzh Svool_Dliow!	Labas Hello_World!
3 Ozyzh Svool_Dliow! OzYzH_oZyZh	Labas Hello_World! LaBaS_lAbAs

#### Pastabos:

- Šifruojami tik lotyniškoje abėcėlėje esantys simboliai, jei eilutėje yra kitų simbolių – jie paliekami tokie, kokie yra.
- Atkreipkite dėmesį į tai, kad didžiosios raidės iššifravus lieka didžiosiomis. Atitinkamai ta pati taisyklė galioja ir mažosioms raidėms.
- Naudokite lotynišką abėcėlę (26 raidžių).
- Šiame uždavinyje žodžiai atskiriami apatiniu brūkšneliu.

## Reikalavimai

- Parašyti duomenų skaitymo funkciją **void()**.
- Parašyti iššifruotų eilučių spausdinimo funkciją **void()**.
- Sukurti funkciją **Atbash()**, kuri per savo vardą grąžintų iššifruotą eilutę. Funkcijos tipas – **string()**.
- Realizuoti funkciją, kuri kiekvienai nuskaitytai duomenų failo eilutei pritaikytų funkciją **Atbash()**. Funkcijos tipas – **void()**.

## 4. Atblokavimas

Kadangi technologijos sparčiai tobulėja, telefonų gamintojai stengiasi padaryti kuo patogesnius telefonus. Kad atrakintų telefoną, Alex'ui reikia ištarti dvi raides. Alex'o priešas, Arkadijus, nori atrakinti Alex'o telefoną ir pavogti svarbią informaciją. Deja, Arkadijus gali ištarti tik **n** dviejų raidžių kombinacijų. Ištartimo eilė nesvarbi, taipogi jis gali ištarti iškart keletą raidžių kombinacijų (nebūtinai skirtingų) kad gautų raidžių kombinaciją, kurioje būtų slaptažodis. Parašykite programą, kuri nustatytų, ar Arkadijui pavyks.

Pradinių duomenų faile **duomenys.txt** yra tokie duomenys:

Pirmoje duomenų failo eilutėje dvi raidės – slaptažodis.

Antroje eilutėje **n** – kiek raidžių kombinacijų gali ištarti Arkadijus.

Tolimesnėse **n** eilučių seka raidės, kurias Arkadijus gali ištarti.

Į rezultatų failą **atsakymas.txt**

Išveskite vieną žodį – **Taip** arba **Ne**

duomenys.txt	atsakymas.txt
ya 4 ah oy to ha	Taip

Pavyzdžio paaiškinimas: Arkadijus gali ištarti "oyah" kuris turi dvi slaptažodžio raides – "ya", atrakinimas pavyks.

## Reikalavimai

Kad gautumėte maksimalų įvertinimą, neviršykite 100 eilučių.

## 5. Teksto komentarų analizė

Duotas teksto fragmentas, kuriame tarp eilučių yra įterpta komentarų. Kiekvienas komentaras atskirtas simboliais "\*\*\*" iš abiejų pusių. Rastus komentarus reikia iškirpti iš teksto ir kartu įsiminti, iš kurios eilutės jis paimtas. Rezultatų faile išspausdinti tekstą be komentarų su sunumeruotomis eilutėmis, už teksto išspausdinti komentarų sąrašą: iš kurios eilutes jis iškirptas ir pats komentaras.

duomenys.txt Lorem ipsum dolor sit amet, consectetur adipiscing elit.**Pirmas komentaras** Nam auctor neque tortor, sit amet lacinia erat eleifend sit amet. Cras nibh arcu, scelerisque non tempor sagittis, suscipit ac orci. Sed sollicitudin ante diam, et porttitor metus vulputate ut. Donec sit amet dui nunc.**Komentaras trecioje eiluteje** Pellentesque diam risus,**Komentaras toje pacioje eiluteje** condimentum elementum diam sit amet, accumsan **Komentaras prasidedantis pradzioje**ullamcorper lectus. Curabitur luctus egestas risus, et vestibulum arcu sollicitudin eget. Proin elementum consequat urna ac luctus. Sed eu pellentesque justo, tempor porttitor dui. Sed sollicitudin sem ut magna aliquet aliquet.**Pabaiga****Komentaras einantis iskarto uz kito komentaro**
---



rezultatai.txt

1: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam auctor neque tortor, sit amet lacinia erat eleifend sit  
2: amet. Cras nibh arcu, scelerisque non tempor sagittis, suscipit ac orci. Sed sollicitudin ante diam, et porttitor  
metus  
3: vulputate ut. Donec sit amet dui nunc. Pellentesque diam risus, condimentum elementum diam sit amet, accumsan  
4: ullamcorper lectus. Curabitur luctus egetas risus, et vestibulum arcu sollicitudin eget. Proin elementum  
consequat urna  
5: ac luctus. Sed eu pellentesque justo, tempor porttitor dui. Sed sollicitudin sem ut magna aliquet aliquet.

Komentariai

1: Pirmas komentaras  
3: Komentaras trecioje eiluteje  
3: Komentaras toje pacioje eiluteje  
4: Komentaras prasidedantis pradžioje  
5: Pabaiga  
5: Komentaras einantis iskarto uz kito komentaro

### Reikalavimai

- Sukurti teksto nuskaitymo į **string** masyvą funkciją.
- Sukurti teksto apdorojimo funkciją, kuri atskiria komentarus nuo teksto.
- Sunumeruoti rezultatuose teksto eilutes.
- Išspausdinti, iš kurios eilutės paimtas komentaras.

### 6. IP adresai

Sukurkite programą, kuri nustatytų IP adreso validumą (tinkamumą) pagal pateiktas **taisykles**:

- IP adresas turi būti sudarytas iš 4 dalių.
- IP adresas turi būti vien iš skaičių ir taško simbolių.
- IP adreso dalys gali būti nuo 0 iki 255

Duomenų faile (Duomenys.txt) pateikiamas nežinomo dydžio (bet ne daugiau nei 10) IP adresų sąrašas, po vieną eilutėje.

Rezultatų faile turite išspausdinti IP adresą ir atsakymą, teisingas jis ar neteisingas.

Duomenys.txt	Rezultatai.txt
199.145.148.97	199.145.148.97 Teisingas
83.33.28.184	83.33.28.184 Teisingas
144.174.24	144.174.24 Neteisingas
219.135.ABC.202	219.135.ABC.202 Neteisingas
256.0.15.1	256.0.15.1 Neteisingas

### Reikalavimai

- Skaitymo funkcija.
- Spausdinimo funkcija.
- Vieno IP adreso tikrinimo funkcija, grąžinanti bool tipo reikšmę.

### 7. Straipsnių pavadinimai

Įsivaizduojamoje planetoje Gyvūnų žemė yra leidžiamas laikraštis. Laikraščio kiekvieno straipsnio pavadinimas  $s_1$  sudarytas iš mažųjų 26 lotyniškos abėcėlės raidžių. Planetos gyventojas, ruoniukas Jonas, nori nusipirkti keletą tokių laikraščių, iškirpti antraštes bei suklijuoti jas į vieną didelį žodį. Suklijavęs antraštes, Jonas iškerpa tam tikras raides iš gauto žodžio, kad būtų gautas naujas žodis  $s_2$ . Laikoma, kad iškirpus raidę iš žodžio, nepaliekamas tarpas, t.y., žodis išlieka vientisas tik be iškirptų raidžių. Be to, iškirptų raidžių keletą kartu naudoti negalima.

Pavyzdžiui, jeigu straipsnio pavadinimas yra "ada", suklijavus du tokius pačius pavadinimus, gauname "adaada". Jeigu pašalinamos 1 ir 6 raidės, gaunamas naujas žodis "daad".

Kiek mažiausiai reikia laikraščio antraščių  $s_1$ , kad Jonas jas suklijavęs bei pašalinęs tam tikras raides gautų žodį  $s_2$ ?

Duomenų faile **in.txt** yra dvi teksto eilutės. Pirmojoje eilutėje yra straipsnio pavadinimas  $s_1$ , antrojoje žodis  $s_2$ . Žodžiai gali būti sudaryti iš mažųjų 26 lotynų abėcėlės raidžių ( $1 \leq |s_1| \leq 10^4, 1 \leq |s_2| \leq 10^6$ ).

Rezultatų faile **out.txt**, jeigu neįmanoma suklijuoti žodžio  $s_2$  pagal nurodytas sąlygas, išvedamas "-1" (be kabučių). Kitu atveju turi išvestas skaičius, kuris apibrėžia **mažiausią reikalingą laikraščio antraščių  $s_1$  kiekį**, kuris reikalingas Jonui suklijuoti žodį  $s_2$ .

#### Pastabos

- Atkreipkite dėmesį į tai, kad žinomas maksimalus  $s_1$ .
- Simbolius galima užrašyti sveikaisiais skaičiais. ASCII lentelėje galima rasti, jog a-z mažosios lotyniškos raidės atitinka 97-122 sveikuosius skaičius.

in.txt	out.txt	Detalus pavyzdys
jonas qwerty	-1	3 pavyzdžio paaiškinimas, kaip gautas rezultatas. 1            2            3            4 
jonas jonas	1	
qwerty qwqyyq	4	
ab baaabba	6	

#### Reikalavimai

- Parašyti duomenų skaitymo funkciją **void()**.
- Uždavinį išspręsti optimizuotai. Kad tai būtų pasiekta, per  $s_1$  ir  $s_2$  galima iteruoti maksimaliai vieną kartą. Be to, turi būti sukurta funkcija, kurios vienas iš parametrų dvimatis masyvas **int[][26]**, kurį funkcija turi užpildyti. Šiame dvimačiame masyve saugoma kiekvienai  $s_1$  pozicijai arčiausia kitų abėcėlės raidžių pozicija žodyje.
- Sukurti funkciją, kuri per savo vardą grąžintų reikalingų  $s_1$  žodžių kiekį, o vienas iš funkcijos parametrų būtinai būtų apskaičiuotas dvimatis masyvas **int[][26]**. Funkcijos tipas **int()**.

### 8. Pasaka

Sukurkite programą, kuri padėtų autoriui pakeisti tekste esančius vardus. Turite vieną duomenų failą – **pasaka.txt**, kuriame surašytas tekstas, tada yra **viena eilutė kurioje padėtas ; (kablitaškis)** ir po jos **kiekvienoje naujoje eilutėje** surašyti vardų rinkiniai: **senas\_vardas naujas\_vardas**. Nuskaitytą tekstą su juo atlikite veiksmus taip, kad tekste senus vardus pakeistų nauji. Tekstą su pakeistais vardais spausdinkite į failą **pasakarez.txt**. Turi būti pakeisti tik vardai, kiti simboliai ir žodžiai turi išlikti tokie kokie buvo. **Į linksnius nekreipkite dėmesio**. Keiskite tik tuos vardus, kurių linksniai sutampa.

pasaka.txt	pasakarez.txt
Tomas_nusipirko_dviratį. Jam_nepatiko_įo_naujas_dviratis Todėl_Tomas_įį_pardavė_įr_nusipirko paspirtuką. ; Tomas Marius	Marius_nusipirko_dviratį. Jam_nepatiko_įo_naujas_dviratis Todėl_Marius_įį_pardavė_įr_nusipirko paspirtuką.
Lukas_yra_ledų_gurmanas. Lukas_labai_mėgsta_vanilinius. Todėl_įam_Karolis_davė_du_eurus. Lukas_padėkojo_draugui_už_paskolintus_pinigus ledams. ; Lukas Darius Karolis Tomas	Darius_yra_ledų_gurmanas. Darius_labai_mėgsta_vanilinius. Todėl_įam_Tomas_davė_du_eurus. Darius_padėkojo_draugui_už_paskolintus_pinigus ledams.
Paulius_mėgsta_sportą. Įis_į_komandą_kviečia_draugus. Darius_Andrius_įr_Lukas sutiko_būti_su_įuo_komandoje. ; Paulius Petras Darius Domas Andrius Algis Lukas Laurynas	Petras_mėgsta_sportą. Įis_į_komandą_kviečia_draugus. Domas_Algis_įr_Laurynas sutiko_būti_su_įuo_komandoje

### Reikalavimai

- Pradinių duomenų skaitymo funkcija **void()**.
- Funkcija **void()**, pertvarkanti tekstą .
- Duomenų rašymo į failą funkcija **void()**.

### 9. Kiti formatai

Programai pateikiamas tekstas be tarpų vienoje eilutėje faile **nesuprantamas.txt**, tame tekste visos š raidės pakeistos dviem simboliais: &;. Programa turi pakeisti visus užrašus &; į š raides ir išsaugoti rezultatą faile **suprantamas.txt**.

nesuprantamas.txt	suprantamas.txt
abcabc&;	abcabcš
abc&;abc	abcšabc
abc&&;;abc	abc&š;abc

### Reikalavimai

- Funkcija, kuri pakeistų &; į š raides nepašalinant tvarkos ir kitų raidžių.
- Funkcija, kuri išsaugotų rezultatus faile.

## 13. STRUKTŪROS, STRUKTŪRŲ MASYVAI

### 1. Rudens dargana

Jaunasis meteorologas Saulius keliasdešimt dienų fiksavo rudens orus. Kiekvieną dieną jis registravo, kokia diena (lietinga – L, saulėta – S, apsiniaukusi – A) ir tos dienos oro temperatūrą (realusis skaičius). Saulius stebėjimų duomenis surašė pradinių duomenų faile **oras.txt**. Pirmoje failo eilutėje įrašytas dienų skaičius **n**, likusiose **n** eilučių – kiekvienos dienos duomenys – simbolis, apibūdinantis dieną ir tos dienos oro temperatūra.

Parašykite programą, kuri į rezultatų failą **dargana.txt** išvestų rezultatus, kaip pateikta pavyzdyje: nurodomas dienos tipas, kelios buvo to tipo dienos, kokia buvo vidutinė to tipo dienų oro temperatūra, kiek buvo to tipo dienų, kurių oro temperatūra buvo didesnė už to tipo dienų vidutinę oro temperatūrą.

Jei nėra kurio nors tipo dienų, tuomet faile po dienos tipo turi būti įrašytas žodis „NE“, pvz.:

LIETINGOS:

NE

Pradiniai duomenys	Rezultatai
10	
L 10.9	
S 11.2	
L 10.8	LIETINGOS:
L 12.5	5 11.34 2
A 15.2	SAULETOS:
A 16.1	2 11.75 1
A 10.2	APSINIAUKUSIOS:
L 12.3	3 13.83 2
L 10.2	
S 12.3	

### Reikalavimai

- Sukurta ir naudojama struktūra, aprašanti vieną stebėjimą.
- Sukurtas ir naudojamas struktūrų masyvas.
- Pradinių duomenų skaitymo funkcija void.
- Vidutinės oro temperatūros skaičiavimo funkcija, gražinanti apskaičiuotą reikšmę per funkcijos vardą. Funkciją panaudoti 3 kartus: lietingų, saulėtų ir apsiniaukusių dienų vidutinei temperatūrai skaičiuoti.
- Kiekio skaičiavimo funkcija, gražinanti apskaičiuotą reikšmę per funkcijos vardą. Funkciją panaudoti 3 kartus: lietingų, saulėtų ir apsiniaukusių dienų kiekiui skaičiuoti.
- Funkcija, gražinanti apskaičiuotą vieno tipo dienų skaičių, kurių temperatūra buvo didesnė už to tipo dienų vidutinę temperatūrą. Funkciją panaudoti 3 kartus: lietingų, saulėtų ir apsiniaukusių dienų kiekiui skaičiuoti.
- Rezultatų rašymo funkcija void. Funkciją panaudoti 3 kartus: lietingų, saulėtų ir apsiniaukusių dienų rezultatams išvesti.

### 2. Krepšinio rungtynės

Stipriausioje pasaulio krepšinio lygoje "NBA" per vieną dieną įvyko daug rungtynių tarp komandų. Krepšinio analitikai daug diskutuoja, kas tą rungtynių dieną pasirodė geriausiai, tačiau

niekaip negali apsispręsti. Padėkite jiems palengvinti šį darbą ir sukurkite programą, kuri atliktų veiksmus su žaidėjų statistikos rodikliais ir nuspręstų, kuris žaidėjas pasirodė geriausiai!

Pradinių duomenų faile (duomenys.txt) duomenys pateikti tokia tvarka: pirmoje eilutėje n – žaidusių žaidėjų kiekis, kitose n eilučių pateikti žaidėjų vienu rungtynių duomenys: vardas, pavardė, pelnytų taškų skaičius, rezultatyvių perdavimų skaičius, atkovotų kamuolių skaičius. Rezultatų failas rezultatai.txt.

#### Užduotis:

- Rasti ir į rezultatų failą išvesti vardus, pavardes ir taškus tų žaidėjų, kurie pelnė ne mažiau nei 30 taškų (rezultatai rikiuojami mažėjimo tvarka pagal taškus).
- Rasti ir į rezultatų failą išvesti naudingiausio žaidėjo vardą, pavardę bei naudingumo balų skaičių.

Naudingumo balai skaičiuojami taip:

- Už vieną pelnytą tašką skiriamas vienas balas.
- Už vieną rezultatyvų perdavimą skiriami du balai.
- Už vieną atkovotą kamuolį skiriamas vienas balas.

Pradiniai duomenys ir rezultatai:

duomenys.txt	rezultatai.txt
7 Bradley Beal 27 1 4 DeMar DeRozan 33 6 8 Lauri Markkanen 26 2 13 Lonzo Ball 11 11 16 Stephen Curry 39 7 11 Kyle Kuzma 17 3 3 Jimmy Butler 26 4 10	Stephen Curry 39 DeMar DeRozan 33 Naudingiausias: Stephen Curry 64

#### Reikalavimai

- Sukurta ir panaudota struktūra, aprašanti žaidėjo vienu rungtynių statistiką.
- Sukurtas ir panaudotas struktūrų masyvas.
- Pradinių duomenų skaitymo funkcija void.
- Funkcija int, grąžinanti kiekį žaidėjų, įmetusių ne mažiau ( $\geq$ ) nei 30 taškų.
- Funkcija int, apskaičiuojanti žaidėjo naudingumo balus.
- Rezultatų rašymo funkcija void.

### 3. Sėkmės žaidimas

Aidenas ir Mia yra geri draugai. Jie turi ir savo mėgstamą žaidimą – sėkmės žaidimą. Žaidimas žaidžiamas metant žaidimų kauliuku, ant kurio gali būti užrašyti bet kokie sveikieji skaičiai. Vienas po kito žaidėjai meta žaidimų kauliuką ir išridena skaičius. Aidenas mandagus vaikiną ir visada užleidžia draugei pradėti pirmai (o gal čia kažkoks planas?). Žaidimo pabaigoje visi skaičiai yra palyginami ir taškai apskaičiuojami tokiu būdu:

- Jeigu vieno žaidėjo skaičius buvo didesnis (3 taškais) tame pačiame ėjime, tas žaidėjas gauna +3 taškus.
- Jeigu skirtumas buvo mažesnis negu 3 taškai, +2 taškus gauna žaidėjas, kurio skaičius buvo mažesnis.
- Jeigu skaičiai yra lygūs, abudu žaidėjai gauna po +1 tašką.
- Kitais atvejais – niekas negauna taškų.

Duomenų faile **duomenys.txt** įrašyti tokie duomenys:

Aideno ir draugės pavardės, ūgis ir amžius.

Tolesnėje eilutėje įrašytas metimų skaičius **n**.

Ketvirtoje eilutėje yra **n** skaičių, kurie buvo išmesti žaidžiant žaidimą. Skaičiai eina vienas po kito.

Rezultatų faile **rezultatai.txt** išveskite laimėtojo vardą, pavardę, ūgį padaugintą iš amžiaus, ir kiek taškų surinko laimėtojas.

duomenys.txt	rezultatai.txt
Aidenavičius 1.78 17 Miaukauskaitė 1.68 17 6 1 7 4 3 5 3	Aidenas Aidenavičius 30.26 7

## Reikalavimai

- Žmogaus struktūra: vardas, pavardė, ūgis, amžius ir visų taškų suma.
- Veiksmai atliekami su struktūra. Rezultatai išvedami kaip struktūros elementai.

### 4. „Talento stipendijų“ konkursas

Kauno technologijos universitete geriausiai besimokantiems studentams yra skiriamos „Talento“ stipendijos. Parašykite programą, kuri apskaičiuotų, kiek studentų gaus šias stipendijas.

„Talento“ stipendija yra skiriama už gerus akademinis rezultatus bei aktyvią papildomą veiklą (tam, kad būtų paprasčiau, darome prielaidą, kad stipendija yra gaunama tik už studento akademinis rezultatus). Skaičiavimas, ar studentas gaus stipendiją, yra labai paprastas: yra apskaičiuojamas kiekvieno studento praeito semestro visų galutinių modulių įvertinimų vidurkis ir jei tas vidurkis yra **didesnis arba lygus** „Talento“ stipendijų skyrimo komisijos nustatytam slenkstiniam vidurkiui, tai studentas stipendiją gauna.

Programos duomenys: pradinių duomenų failo **duomenys.txt** pirmoje eilutėje nurodytas skaičius **n** – studentų, pateikusių paraiškas stipendijai gauti, skaičius. Kitose **n** eilučių yra pateikta informacija apie studentus: vardas, pavardė, tada pateiktas skaičius **m**, kuris nurodo, kiek modulių praėjusio semestro metu mokėsi studentas, o toliau esantys **m** skaičių atitinka galutinius įverčius (pažymius), kuriuos studentas gavo už kiekvieną atsiskaitytą modulį. Paskutinė eilutėje pateiktas skaičius **vid**, nurodantis, kokio vidurkio užtenka studentui, kad jis gautų "Talento" stipendiją (slenkstinis vidurkis).

Programos rezultatai: rezultatų failo **rezultatai.txt** pirmoje eilutėje pateikiamas skaičius **k** – kiek studentų gavo stipendijas. Tolimesnėse **k** eilučių pateikiama informacija apie studentus: Vardas, **tarpo simbolis**, pavardė, **tarpo simbolis**, studento praeito semestro modulių galutinių įverčių vidurkis (**2 skaičių po kablelio tikslumu**).

Pradiniai duomenys	Rezultatai
1 Alfredas Binkis 6 10 9 10 10 8 9 9.2	1 Alfredas Binkis 9.33
1 Alfredas Binkis 6 10 9 10 9 8 9 9.2	0
3 Alfredas Binkis 6 10 9 10 10 8 9 Balys Sruoga 5 9 9 9 9 9 Klaudijus Makgaiveris 6 10 9 10 9 10 10 9.3	2 Alfredas Binkis 9.33 Klaudijus Makgaiveris 9.67

## Reikalavimai

- Panaudoti struktūros duomenų tipą studento informacijai saugoti.
- Realizuoti programos duomenų nuskaitymo funkciją (void).
- Programos duomenis nuskaityti ir išsaugoti studentų struktūrų masyve.
- Realizuoti funkciją, kuri apskaičiuoja kiekvieno studento modulių vidurkį (void).
- Realizuoti funkciją, kuri per savo vardą gražintų skaičių, kiek studentų gaus stipendijas (int).
- Realizuoti programos rezultatų spausdinimo į failą funkciją (void).

## 5. Vaistinė

Atėjus šaltajam metų laikui, padidėja tikimybė peršalti ar susirgti gripu. Vaistų yra nemažai, tačiau skiriasi jų kaina ir laikas per kurį pasveikstama.

Sukurkite programą, suskaičiuojančią vidutinę gydymosi kainą iš pateiktų vaistų sąrašo ir atrenkančią tinkamus vaistus, kuriais gydytis kaina neviršys pateiktos pinigų sumos.

Duomenų failo struktūra:

- Pirma eilutė: vaistų skaičius ir pinigų suma.
- Likusios eilutės: vaisto pavadinimas, kiek tablečių per dieną gerti, kiek dienų reikia vartoti, tabletės kaina.

Rezultatų failo struktūra:

- Vidutinė gydymosi kaina.
- Kelių vaistų kaina neviršija pinigų sumos.
- Išvardinami atrinkti vaistai po vieną pavadinimą eilutėje.

Pvz.:

Duomenys.txt	Rezultatai.txt
5 50 Prospan 2 21 2 Paxeladine 1 7 5 Acc 1 30 1.15 Flavamed 2 15 3.3 Brontex 3 14 2	67.3 2 Paxeladine Acc
Duomenys.txt	Rezultatai.txt
5 90 Bronchipret 2 20 2.3 Stodal 3 5 6.4 Pertusinas 1 30 2.5 Gelomyrtol 3 14 1.1 Ambroxol 1 60 0.59	68.92 3 Pertusinas Gelomyrtol Ambroxol
Duomenys.txt	Rezultatai.txt
6 30 Hedeliksas 1 7 1.5 Gertolis 1 15 5.9 Ambroksolis 1 60 0.59 Prospan 2 21 2 Pakseladinas 1 7 5 Akumuliumiotas 1 30 1.15	47.98 1 Hedeliksas

## Reikalavimai

- Vaistą aprašanti struktūra.
- Vidutinę kainą skaičiuojanti funkcija.

### 6. Studentų maratonas

Maratono bėgimas yra viena iš populiariausių bėgimo rungčių. Tavo užduotis bus išanalizuoti bėgimo rezultatus ir pateikti tam tikrą statistiką.

Reikia parašyti programą ,apskaičiuojančią bendrą bėgimo laiko trukmę ir vidutinį bėgikų greitį. Rezultatuose pateikti pirmus 5 bėgikus kurių greitis yra didesnis už vidurkį (jeigu tik 3 bėgikai greičiau bėgo nei vidurkis, tai išspausdinti tik 3), bendrą bėgikų laiką sekundėmis ir vidutinį bėgikų greitį.

Duomenų faile pateikta informacija: bėgiko vardas ir pavardės pirma raidė atskirta tašku, sugaištas laikas bėgant sekundėmis ir bėgimo atstumas metrais. Nubėgtas atstumas gali skirtis (vieni bėgo ilgesnį atstumą, kiti trumpesnį, pagal savo pajėgumus).

duomenys.txt	rezultatai.txt
20 Liam.R 705 5000 Sarah.M 709 5000 Stewart.M 730 5000 Rebecca.L 753 5000 Jessica.S 763 5000 Lauren.S 765 5000 Piers.G 770 5000 Rebecca.B 774 5000 Austin.P 777 5000 Jasmine.M 787 5000 Rachel.M 798 5000 Lisa.A 802 5000 Penelope.F 805 5000 Stephen.M 809 5000 Pippa.G 814 5000 Matt.D 819 5000 Sean.K 857 5000 Jonathan.J 865 5000 Vanessa.P 883 5000 Angela.M 887 5000	Liam.R 7.09m/s Sarah.M 7.05m/s Stewart.M 6.85m/s Rebecca.L 6.64m/s Jessica.S 6.55m/s Bendras laikas: 15872s Greicio vidurkis: 6.33m/s
0	Bendras laikas: 0s Greicio vidurkis: 0.00m/s

## Reikalavimai

- Bėgiką aprašanti struktūra.
- Funkcija skirta nuskaityti duomenis.
- Funkcija, skaičiuojanti bendrą bėgimo laiką.
- Funkcija, skaičiuojanti visų bėgikų greičių vidurkį.



## 7. Sūrio pjaustymas

Parduotuvėse dažnai galima pamatyti pardavinėjamą įvairų gabaliukais supjaustytą sūrį. Sūrio gaminimo pramonė konstruoja įrenginius, kurie galėtų supjaustyti sūrį vienodos masės gabaliukais. Tačiau įrenginių kūrėjai susiduria su problemomis, kai norima pjaustyti šveicarišką sūrį. Pavyzdžiui, Emmentaler sūryje yra skylių ir skylės yra skirtingų dydžių. Gabaliukas, kuriame yra skylių, sveria mažiau, negu sūrio gabaliukas be skylių. Atsiranda iššūkis supjaustyti sūrį, kuris turi skylių, vienodos masės gabaliukais.

Naudojant naujausias technologijas galima nustatyti skylių vietą ir dydį **mikrometru** tikslumu. Sprendžiant šią problemą galima laikyti, kad skylės yra tobulos sferos formos.

Kiekvieno nesupjaustyto sūrio gabalo matmenys yra 100 x 100 x 100 **milimetrų**. Tikslas – supjaustyti šį gabalą į  $s$  vienodos masės gabaliukų. Kiekvienas sūrio gabaliukas bus 100 mm pločio, 100 mm aukščio. Jūsų pagalbos reikės nustant kiekvieno gabaliuko storį.

### Įvestis

Pirmoje failo **in.txt** eilutėje yra du sveikieji skaičiai  $n$  ir  $s$ , kur  $0 \leq n \leq 125$  skylių skaičius sūryje ir  $1 \leq s \leq 100$  yra skaičius gabaliukų, į kurios reikia supjaustyti sūrį. Likusiose  $n$  eilučių – 4 sveikieji skaičiai  $r$ ,  $x$ ,  $y$  ir  $z$  apibūdinantys sferą, kur  $r$  – sferos spindulys,  $x$ ,  $y$  ir  $z$  sferos centro koordinatės. Šių skaičių matavimo vienetai – **mikrometrai**. Pjūviai daromi lygiagrečiai  $z$  ašiai.

Nesupjaustytas sūris užima taškus  $(x, y, z)$ , kur  $0 \leq x, y, z \leq 100\,000$ , išskyrus tuos, kurie yra tam tikros skylės dalis.

Laikoma, kad skylės nepersidengia, bet gali liestis. Be to, skylės neišeina už sūrio ribų, bet gali liesti sienas.

### Išvestis

I failą **out.txt** išvesti  $s$  gabaliukų storius, pradedant nuo sūrio pabaigos, kur  $z = 0$ . Didžiausia galima absoliuti paklaida yra  $10^{-9}$  (išvesti 9 skaičius po kablelio).

in.txt	out.txt
0 4	25.000000000 25.000000000 25.000000000 25.000000000
in.txt	out.txt
2 5 10000 10000 20000 20000 40000 40000 50000 60000	14.611103142 16.269801734 24.092457788 27.002992272 18.023645064

### Pastabos

- Rekomenduojama naudoti **binary search** gabaliuko storio skaičiavimui.
- Pjaunant gabaliuką, įmanoma, jog tam tikros skylės yra kertamos, todėl gali reikėti apskaičiuoti dalinį sferos tūrį, priklausomai nuo to, kurioje vietoje buvo kiršta skylė.

### Reikalavimai

- Panaudoti struktūros duomenų tipą sferos informacijai saugoti. Be to, šioje struktūroje sukurti funkciją sferos tūrio apskaičiavimui.
- Programos duomenis nuskaityti ir išsaugoti sferų struktūrų masyve.

## 8. Automobiliai

Rokas yra jaunas studentas, todėl jam svarbu sekti savo išlaidas. Rokas turi automobilį, tačiau jo automobilyje technika, apskaičiuojanti vidutinės sąnaudas, yra sugedusi. Rokas kiekvieną kartą

nuvažiuojes į kuro kolonėlę registruoja duomenis faile **duomenys.txt**. Rokas visada pila pilną baką. Pirmoje failo eilutėje įrašytas Roko užregistruotų duomenų kiekis **n**, likusiose **n** eilučių – kiekvienos kelionės duomenys – nuvažiuotų kilometrų skaičius ir sunaudoto kuro kiekis.

Parašykite programą, kuri padėtų Rokui suskaičiuoti savo automobilio vidutines kuro sąnaudas 100 km dešimtųjų tikslumu ir išspausdintų jas faile **sanaudos.txt**.

Pradiniai duomenys	Rezultatai
5	
100	5
200	10
300	15
200	10
300 15	5

### Reikalavimai

- Sukurta ir naudojama struktūra, aprašanti kelionės duomenis.
- Sukurtas ir naudojamas struktūrų masyvas.
- Pradinių duomenų skaitymo funkcija void.
- Nuvažiuotų kilometrų sumos skaičiavimo funkcija, grąžinanti apskaičiuotą reikšmę per funkcijos vardą.
- Viso sudeginto kuro kiekio skaičiavimo, grąžinanti apskaičiuotą reikšmę per funkcijos vardą.
- Vidutinių kelionės sąnaudų skaičiavimo funkcija, grąžinanti apskaičiuotą reikšmę per funkcijos vardą.
- Rezultatų rašymo funkcija void.

## 14. STRUKTŪRŲ MASYVAI. DIDŽIAUSIOS IR MAŽIAUSIOS REIKŠMĖS PAIEŠKA

### 1. Automobilių ieškojimo varžybos

Tomas ir Benas varžosi, kuris automobilių turguje ras brangiausią ir pigiausią automobilius. Draugai susitarė, kad Tomas ieškos tik BMW markės automobilių, o Benas – Audi. Laimės tas, kuris ras ir brangiausią, ir pigiausią automobilį.

Pradinių duomenų faile duomenys.txt duomenys pateikti tokia tvarka: pirmoje eilutėje **n** – automobilių kiekis, kitose **n** eilučių – duomenys apie kiekvieną automobilių turguje esantį automobilį: markė, modelis, kaina. Duomenys skiriami tarpais.

Rezultatų faile rezultatai.txt išvedami tokie rezultatai: pirmoje eilutėje – brangiausią automobilį radusio vaikinų vardas ("Brangiausias: vardas"), antroje eilutėje – pigiausią automobilį radusio vaikinų vardas ("Pigiausias: vardas"), trečioje eilutėje – kuris vaikinai, pagal Tomo ir Beno priimtas taisykles, laimėjo varžybas ("Laimėjo: vardas"). Jei niekas nelaimėjo varžybų, išvedama "Lygiosios".

Pastaba: automobilių turguje bus tik po vieną pigiausią ir brangiausią automobilį.

Pavyzdys:

duomenys.txt	rezultatai.txt
8 Audi 80 600 BMW 330 3600 Audi A4 2700 Volkswagen Golf 1300 Audi A6 5100 BMW 320 6000 BMW 530 2000 Volvo V50 3000	Brangiausias: Tomas Pigiausias: Benas Lygiosios

### Reikalavimai

- Sukurta ir panaudota struktūra, aprašanti vieno automobilio duomenis.
- Pradinių duomenų skaitymo funkcija void.
- Sukurtas pradinių duomenų struktūrų masyvas.
- Sukurtas naujas Tomo rastų automobilių struktūrų masyvas ("BMW").
- Sukurtas naujas Beno rastų automobilių struktūrų masyvas ("Audi").
- Funkcija, randanti brangiausią automobilį.
- Funkcija, randanti pigiausią automobilį.
- Rezultatų rašymo funkcija void.

## 2. Idėja Lietuvai

**IDĖJA LIETUVAI** – nacionalinė iniciatyva (<https://www.idejalietuvai.lt>), siekianti suburti nepriklausomos šalies šimtmečio slenkstį liudijančius protus, kuriems rūpi mūsų valstybės išlikimas, stiprybė ir gerovė. Kaip gyvensime po 5, 10 ar... 100 metų? Kokią Lietuvą paliksime ateinančio šimtmečio kartoms? Kokių valstybės masto idėjų įgyvendinimas užtikrins Lietuvos gerovę ateityje?

Trys reikšmingiausios idėjos, dėl kurių sutars žiniasklaida, ekspertai ir visuomenė, bus įvardytos Valstybės stiprybės akte, kurį pasirašys visa Lietuva.

Visos idėjos suskirstytos į  $n$  ( $2 < n < 20$ ) grupių. Kiekvienoje grupėje yra  $k$  idėjų (skirtingų grupių  $k$  reikšmės skiriasi,  $2 < k < 30$ ). Kiekvieną idėją apibūdina trumpas aprašas ir žmonių, balsavusių už tai, kad idėją verta įgyvendinti, skaičius.

Pradiniai duomenys surašyti faile `ideja.txt`. Pirmoje failo eilutėje nurodytas idėjų grupių skaičius  $n$ . Antroje eilutėje nurodytas grupės pavadinimas ir kiek idėjų  $k$  yra toje grupėje. Tolesnėse  $k$  eilučių įrašyta informacija apie vieną grupės idėją: idėjos aprašas ir žmonių, balsavusių už tai, kad idėją verta įgyvendinti, skaičius. Tolesnėse pradinių duomenų failo eilutėse laikantis to paties principo aprašomos kitų grupių idėjos.

Parašykite programą, kuri į rezultatų failą `idejalietuvai.txt` įrašytų išrinktas 3 reikšmingiausias idėjas. Faile turi būti įrašyta, kuriai grupei idėja priklauso, idėjos aprašas ir žmonių, balsavusių už tai, kad idėją verta įgyvendinti, skaičius. Rezultatai vienas nuo kito atskiriami tarpais.

Pastaba: pradinių duomenų ir rezultatų failuose bus naudojamos tik lotyniškos didžiosios raidės. Jei aprašai sudaryti iš kelių žodžių, tuomet žodžiai vienas nuo kito skiriami apatiniu brūkšneliu. Žodžiai gali būti trumpinami.

Pradinių duomenų failo pavyzdys:

2

SVIETIMAS 3

LIET.\_RISTYNES\_MOKSL.\_UGDYMUI 1

PO\_"IPAD"\_KIEKV.\_PENSININKUI 3

10000\_JAUNUJU\_INOVATORIU 3

ENERGETIKA 2

KOMPENS.\_FOTOEL.\_ELEKTROMOB. 3

SAULES\_ELEM.\_IR\_PUSL.\_KURIMAS 22

Rezultatų failo pavyzdys:

ENERGETIKA SAULES\_ELEM.\_IR\_PUSL.\_KURIMAS 22

SVIETIMAS PO\_"IPAD"\_KIEKV.\_PENSININKUI 3

SVIETIMAS 10000\_JAUNUJU\_INOVATORIU 3

ENERGETIKA KOMPENS.\_FOTOEL.\_ELEKTROMOB. 3

SVIETIMAS LIET.\_RISTYNES\_MOKSL.\_UGDYMUI 1

### Reikalavimai

- Sukurkite struktūrą, aprašančią vieną idėją.
- Sukurkite pradinių duomenų struktūrų masyvą.
- Sukurkite skaitymo funkciją void.
- Sukurkite didžiausios reikšmės paieškos funkciją, gražinančią reikšmę per funkcijos vardą.
- Į didžiausios reikšmės paieškos funkciją kreipkitės 3 kartus, ieškodami kiekvienos iš 3 reikšmingiausių idėjų. Jei kelios idėjos yra vienodo reikšmingumo, tuomet jos visos turi įrašytos į rezultatų failą.
- Sukurkite įrašymo į rezultatų funkciją void.

### 3. Moksliniai tyrimai ir eksperimentinė plėtra (MTEP)

Lietuvos Respublikos statistikos departamentas (<http://osp.stat.gov.lt/>) pateikia informaciją apie asmenis, dalyvaujančius MTEP. Pirmoje pradinių duomenų failo **tyrimai.txt** eilutėje įrašytas sektorių, vykdančių MTEP, skaičius **s**, pradiniai analizės metai **prmetai** ir galutiniai analizės metai **gmetai**. Tolesnėse **s** eilučių surašyti sektorių pavadinimai (vienas ar keli žodžiai, vienas nuo kito skiriami apatiniu brūkšneliu, naudojamos vien lotyniškos raidės) ir tiek sveikųjų skaičių dvejetų, kiek yra analizės metų. Pirmasis dvejeto skaičius rodo, kiek vyrų dalyvavo MTEP, antrasis – moterų skaičių.

**s** reikšmė ir metų skaičius  $\geq 1$  ir  $\leq$  už 50.

Parašykite programą, kuri analizuoja pateiktus statistinius duomenis:

- kuriame sektoriuje analizuojamame metų intervale dirbo daugiausia moterų. Jei yra keli tokie sektoriai, turi būti išvedamas tas sektorius, kuris pradiniam sąraše yra aukščiausiai;
- kuriame sektoriuje analizuojamame metų intervale dirbo mažiausia vyrų. Jei yra keli tokie sektoriai, turi būti išvedamas tas sektorius, kuris pradiniam sąraše yra žemiausiai;
- koks yra kiekvienais tyrimo metais MTEP dalyvaujančių asmenų skaičius bendrai paėmus visus sektorius (skaičiuoti vienais metais MTEP dalyvaujančių asmenų sumą).

Rezultatai rezultatų faile **tyrimairez.txt** turi būti išsaugoti taip, kaip pateikta pavyzdyje.

Pradinių duomenų failo pavyzdys:

3 2012 2014

Mokslas 6921 9663 7036 9491 6757 9218

Valdzia 1154 1502 1217 1552 1238 1615

Verslas 1928 935 2276 1179 2917 1488

Rezultatų failo pavyzdys:

Mokslas 9663

Valdzia 1154

2012 22103

2013 22751

2014 23233

## Reikalavimai

- Uždaviniui spręsti reikalingas duomenų struktūras pasirinkite savo nuožiūra.
- Pradinių duomenų skaitymo funkcija void.
- Didžiausios ir mažiausios reikšmės paieškos funkcijos, grąžinančios apskaičiuotas reikšmes per funkcijos vardą.
- Sumos skaičiavimo funkcija, grąžinanti apskaičiuotą reikšmę per funkcijos vardą.

### 4. Kontrolinio darbo rezultatai

Mokiniai per kontrolinį darbą turėjo atlikti kelias užduotis, kurių kiekvienos įvertinimas sudarė tam tikrą bendro įvertinimo dalį.

Pradinių duomenų failo **kontrolinis.txt** pirmoje eilutėje įrašytas kontrolinį darbą atlikusių mokinių skaičius  $n$  ( $2 < n < 30$ ) ir kelias užduotis  $k$  ( $2 < k < 7$ ) mokiniai turėjo atlikti. Antroje pradinių duomenų failo eilutėje įrašyta  $k$  realiųjų skaičių – kiekvienos užduoties įvertinimo dalis formuojant bendrą kontrolinio darbo įvertinimą. Tolesnėse  $n$  eilučių surašyta informacija apie kiekvieną mokinį: vardas, pavardė ir  $k$  sveikųjų skaičių, kiekvienos užduoties įvertinimas dešimtbalėje sistemoje.

Parašykite programą, kuri į rezultatų failą **kontrolinisrez.txt** išvestų informaciją:

- kokį galutinį įvertinimą gavo kiekvienas mokinys. Galutinis įvertinimas yra dešimtbalės sistemos sveikasis skaičius, gautas pritaikius matematinio apvalinimo taisykles;
- mokinių, gavusių geriausius kontrolinio darbo įvertinimus, sąrašą (mokiniai sąrašė eina tokia tvarka, kaip ir pradiniam sąrašė);
- mokinių, gavusių blogiausius kontrolinio darbo įvertinimus, sąrašą (mokiniai sąrašė eina tokia atbuli tvarka, negu išvardinti pradiniam sąrašė);
- rezultatų faile išveskite informaciją tokia tvarka, kaip pateikta rezultatų failo pavyzdyje.

Pradiniai duomenys	Rezultatai
3 5 0.2 0.2 0.2 0.2 0.2 Jonas Jonaitis 10 10 10 10 10 Petras Petraitis 8 8 8 8 8 Antanas Antanaitis 5 5 5 5 5	VISI Jonas Jonaitis 10 Petras Petraitis 8 Antanas Antanaitis 5 GERIAUSI Jonas Jonaitis 10 BLOGIAUSI Antanas Antanaitis 5
3 5 0.2 0.2 0.2 0.2 0.2 Jonas Jonaitis 10 10 10 10 10 Petras Petraitis 5 5 5 5 5 Antanas Antanaitis 5 5 5 5 5	VISI Jonas Jonaitis 10 Petras Petraitis 5 Antanas Antanaitis 5 GERIAUSI Jonas Jonaitis 10 BLOGIAUSI Antanas Antanaitis 5 Petras Petraitis 5
3 5 0.2 0.2 0.2 0.2 0.2 Jonas Jonaitis 10 10 10 10 10 Petras Petraitis 10 10 10 10 10 Antanas Antanaitis 10 10 10 10 10	VISI Jonas Jonaitis 10 Petras Petraitis 10 Antanas Antanaitis 10 GERIAUSI Jonas Jonaitis 10 Petras Petraitis 10 Antanas Antanaitis 10 BLOGIAUSI Antanas Antanaitis 10

	Petras Petraitis 10 Jonas Jonaitis 10
--	--

### Reikalavimai

- Sukurkite struktūrą, saugančią mokinio vardą ir galutinį įvertinimą.
- Sukurkite 3 struktūrų masyvus – visų, geriausių ir silpniausių mokinių.
- Sukurkite pradinių duomenų skaitymo funkciją void.
- Sukurkite vieno mokinio galutinio įvertinimo skaičiavimo funkciją, gražinančią apskaičiuotą reikšmę per funkcijos vardą. Į šią funkciją galite kreiptis skaitymo funkcijoje, arba sukurti papildomą void funkciją, kurioje kreipsitės skaičiuodami galutinį įvertinimą.
- Sukurkite geriausio įvertinimo vietos paieškos funkciją, gražinančią reikšmę per funkcijos vardą.
- Sukurkite blogiausio įvertinimo vietos paieškos funkciją, gražinančią reikšmę per funkcijos vardą.
- Sukurkite void funkciją, formuojančią geriausiai parašiusių mokinių struktūrų masyvą iš pradinio masyvo.
- Sukurkite void funkciją, formuojančią blogiausiai parašiusių mokinių struktūrų masyvą iš pradinio masyvo.
- Sukurkite rezultatų rašymo funkciją, į kurią kreipkitės 3 kartus: įrašydami visų, geriausiai ir blogiausiai parašiusių kontrolinį darbą mokinių rezultatus.

## 15. STRUKTŪRŲ MASYVAI. DUOMENŲ ATRANKA, RIKIAVIMAS

### 1. Krepšinio rungtynės

Krepšinio rungtynių sekretorius registruoja kiekvieno žaidėjo pelnytus taškus vos tik jam juos pelnius. Jis juos rašo į failą **duomenys.txt**. Pirmoje failo eilutėje užrašytas užregistruotų duomenų kiekis **n**. Tolesnėse **n** eilučių - įvesti duomenys - žaidėjo numeris, komanda, pelnyti taškai.

Parašykite programą kuri suskaičiuotų kiek kuris žaidėjas buvo rezultatyviausias, kiek jis įmetė taškų ir kuri komanda laimėjo ir išspausdintų rezultatus faile **rezultatai.txt**.

duomenys.txt	rezultatai.txt
5 7 Zalgiris 2 9 Zalgiris 3 3 Rytas 1 9 Zalgiris 2 3 Rytas 2	9 5 Žalgiris

### Reikalavimai

- Sukurta ir naudojama struktūra žaidėjo taškams registruoti.
- Sukurti ir naudojami struktūrų masyvai - pirmos komandos protokolui, antros komandos protokolui, bendrai statistikai.
- Pradinių duomenų skaitymo funkcija void.
- Bendro struktūrų masyvo formavimo funkcija void.
- Rezultatyviausio žaidėjo nustatymo funkcija gražinanti reikšmę per funkcijos vardą.
- Komandos laimėtojos nustatymo funkcija gražinanti reikšmę per funkcijos vardą.
- Rezultatų rašymo funkcija void.

### 2. Geriausias futbolininkas

Futbolininkai A ir B varžosi tarpusavyje, kuris įmuš daugiau įvarčių per nustatytą varžybų skaičių. Sukurk programą, kuri apskaičiuotų ir įvardintų, kuriam futbolininkui sekėsi geriau!

Duomenų faile ("duomenys.txt") duomenys pateikti tokia tvarka: pirmoje eilutėje - A futbolininko vardas ir pavardė (du atskiri žodžiai), antroje eilutėje - B futbolininko vardas ir pavardė (taip pat du atskiri žodžiai), trečioje eilutėje - užfiksuotų varžybų skaičius n, o likusiose

n eilučių pateikta po du skaičius - pirmas skaičius yra A futbolininko įvarčių skaičius, o antras - B futbolininko įvarčių skaičius.

Užduotis:

- Nuskaityti duomenis į du atskirus masyvus (atskirai futbolininkams A ir B).
- Rasti, kuris futbolininkas įmušė daugiau įvarčių per visas varžybas.
- Rezultatų faile ("rezultatai.txt") išvesti futbolininko vardą, pavardę, o kitoje eilutėje - mažėjimo tvarka surikiuotus atskirų varžybų įvarčių kiekius.

Pastaba: Futbolininkai niekada nebus įmušę po vienodai įvarčių.

Duomenų failo pavyzdys:

duomenys.txt	rezultatai.txt
Cristiano Ronaldo Lionel Messi 5 2 1 2 2 1 3 2 1 2 3	Lionel Messi 3 3 2 1 1

### Reikalavimai

- Duomenų skaitymo funkcija void.
- Rezultatų rašymo funkcija void.
- Du struktūrų masyvai - futbolininkui A ir B.
- Vieno futbolininko visų įvarčių skaičiavimo funkcija.
- Rikiavimo funkcija void.

### 3. Giminės istorija

Antanas tvarkydamas senelių sodybos palėpę surado dokumentą su žmonių vardais ir asmens kodais. Senelio paklausė, kokių žmonių duomenys surašyti popieriaus lape. Senelis Antanui atsakė, kad tame lape surašyta giminės istorija. Antanas internete susirado informaciją, kaip formuojamas asmens kodas ir nusprendė iššifruoti dokumentą.

Lietuvos gyventojų registre kiekvienas gyventojas turi unikalų asmens kodą, kuris suformuotas pagal standartą Lietuvos RST 1185-91 „Asmens kodas. Sudėtis ir struktūra“, įsigaliojusį nuo 1991 m. gruodžio 1 d. Tai yra unikali vienuolikos dešimtinių skaitmenų seka, žyminti asmenį ir naudojama duomenims apie jį kaupti valstybės tvarkomose duomenų bazėse.

Asmens kodas susideda iš 11 skaitmenų, pvz.: 33309240064:

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

Pradinių duomenų failo **gimine.txt** pirmoje eilutėje įrašyta, kiek įrašų n ( $n < 100$ ) yra Antano surastame dokumente. Tolesnėse n eilučių įrašytas vardas (vienas žodis) ir asmens kodas. Vienam asmeniui skirta viena eilutė.

Rezultatų faile **giminerez.txt** turi būti pirmiausia surašytos giminės moterys, pradedant nuo vyriausios, baigiant jauniausia, nurodant gimimo metus, mėnesį ir dieną. Po to nuo vyriausio iki jauniausio išvardinti giminės vyrai, nurodant gimimo mėnesį ir dieną. Jei vyrų, arba moterų lape nebuvo, tuomet išvedamas žodis NERASTA (žr. pradinių duomenų ir rezultatų pavyzdžius).

Vardą, metus, mėnesį ir dieną vienas nuo kito atskirkite tarpais. Mėnesių pavadinimus rašykite be diakritinių ženklų (nenaudokite ą, č, ę, è ir t.t.).

Pradiniai duomenys	Rezultatai
6 Romas 36508230023 Antanina 43210060068 Asta 60111300321 Veronika 40203190010 Pranas 30501150007 Martynas 50310241123	MOTERYS Veronika 1902 kovo 19 Antanina 1932 spalio 6 Asta 2001 lapkricio 30 VYRAI Pranas 1905 sausio 15 Romas 1965 rugpjucio 23 Martynas 2003 spalio 24
3 Emilija 43210060068 Karolina 60111300321 Zuzana 40203190010	MOTERYS Zuzana 1902 kovo 19 Emilija 1932 spalio 6 Karolina 2001 lapkricio 30 VYRAI NERASTA

### Reikalavimai

- Naudokite struktūras ir struktūrų masyvus.
- Pradinių duomenų skaitymo funkcija void().
- Vieno žmogaus gimimo metų, mėnesio ir dienos išskyrimo iš asmens kodo funkcija void().
- Mėnesių, išreikštų skaičiais nuo 01 iki 12, konvertavimo į žodžius funkcija, gražinanti reikšmę per funkcijos vardą.
- Rikiavimo pagal amžių funkcija, į kurią reikia kreiptis 2 kartus: su vyrų ir moterų struktūrų masyvais.
- Rezultatų išsaugojimo faile funkcija, į kurią reikia kreiptis 2 kartus: su vyrų ir moterų struktūrų masyvais.

### 4. Katinų slidinėjimo varžybos

Katinų bėgimo varžyboms registravosi  $n$  ( $2 \leq n \leq 50$ ) katinų. Stadione yra 8 slidinėjimo takeliai. Katinai atsitiktinai suskirstyti į  $k$  grupių taip, kad grupėje būtų ne mažiau kaip du katinai.

Parašykite programą, kuri iš kiekvienos grupės atrinktų pusę katinų slidininkų, pasiekusių geriausius rezultatus. Jeigu grupėje yra nelyginis skaičius katinų, tada atrenkama vienu katinu mažiau (pvz., iš penkių slidininkų atrenkami du pasiekusieji geriausius rezultatus). Žinoma, kad visi katinų pasiekti rezultatai yra skirtingi.

Duomenys pateikiami tekstiniame faile **pradinis.txt**. Pirmoje failo eilutėje nurodomas grupių skaičius  $k$ . Toliau iš eilės pateikiami visų katinų grupių sąrašai tokia tvarka:

- pirmoje sąrašo eilutėje pateikiamas katinų skaičius grupėje;
- toliau – kiekvieno katino šeiminių pavardė ir katino vardas (sudaro po vieną žodį, be diakritinių ženklų) ir rezultatas (minutės, sekundės). Vieno katino duomenims skiriama viena eilutė.

Rezultatai pateikiami tekstiniame faile **galutinis.txt**. Spausdinamas visų atrinktų katinų sąrašas pasiekto rezultato laiko didėjimo tvarka. Vieno katino duomenims skiriama viena eilutė: šeiminių pavardė, katino vardas ir rezultatas (minutės ir sekundės).



Pradiniai duomenys	Rezultatai
3	
4	
Petraiciu Rainis 4 25	
Jonaiciu Juodis 3 59	
Stankunu Rudis 4 15	
Masiliunu Mazius 6 20	Staskunu Baltis 3 55
2	Kalniu Micius 3 58
Petkunu Rainis 6 45	Jonaiciu Juodis 3 59
Staskunu Baltis 3 55	Sereiku Kicius 4 2
5	Stankunu Rudis 4 15
Kalniu Micius 3 58	
Sereiku Kicius 4 2	
Dambrausku Rainis 4 5	
Dubausku Micius 4 4	
Kikuciu Rainis 4 6	

### Reikalavimai

- Naudokite masyvus su struktūros tipo elementais.
- Pradinių duomenų skaitymo funkcija void().
- Rikiavimo funkcija void().
- Rašymo į failą funkcija void().
- Gali būti papildomų funkcijų.

### 5. Pieštukai

Turime  $n$  ( $1 \leq n \leq 100$ ) spalvotų pieštukų naudojimo istoriją. Pieštukai naudojimo metu buvo drožiami peiliu arba drožtuku. Nudrožtais pieštukais buvo braižomi brėžiniai. Taip pat jie galėjo būti nulūžę. Naudojimo istorija koduojama raidėmis: D – drožimas drožtuku, P – drožimas peiliu, B – braižymas, L – lūžimas.

Parašykite programą, kuri nustatytų likusį pieštukų ilgį ir surikiuotų pieštukus nuo ilgiausio iki trumpiausio. Jei pieštukų ilgiai sutampa, tuomet pieštukai turi būti rikiuojami pagal spalvą abėcėlės tvarka.

1. Naudojimo pradžioje visi pieštukai yra nauji, nedrožti ir yra 15 cm ilgio.
2. Drožiant pieštuką peiliu, jis sutrumpėja 10 mm.
3. Drožiant pieštuką drožtuku, jis sutrumpėja 7 mm.
4. Jeigu braižant brėžinius pieštukas nenulūžta, jis sutrumpėja 7 mm, jeigu prieš tai buvo drožtas peiliu.
5. Jeigu braižant brėžinius pieštukas nenulūžta, jis sutrumpėja 5 mm, jeigu prieš tai buvo drožtas drožtuku.
6. Pieštukas gali nulūžti drožiant arba braižant.
7. Jeigu pieštukas lūžta, tai jis sutrumpėja 10 mm (buvo drožtas peiliu) arba 7 mm (buvo drožtas drožtuku).

Pradiniai duomenys yra faile **piestukai.txt**. Pirmoje failo eilutėje yra pieštukų skaičius  $n$ , kitose  $n$  eilučių yra pieštukų naudojimo istorijos. Eilutės pradžioje yra pieštuko spalva (vienas žodis). Toliau eilutėje yra pieštuko naudojimo veiksmas (mažiau negu 100).

Rezultatų faile **piestukairez.txt** turi būti spausdinami pieštukai, surikiuoti ilgio mažėjimo tvarka. Spausdinama pieštuko spalva ir jo ilgis, naudojimosi istorijos pabaigoje, milimetrais.

Pradiniai duomenys	Rezultatai
3 Melynas PBLDBPLPB Raudonas DBDBDBDLDBDBDBL Zalias P	Zalias 140 Melynas 81 Raudonas 62

### Reikalavimai

- Naudokite masyvą su struktūros tipo elementais. Gali būti naudojami ir kitų tipų masyvai.
- Rikiavimo funkcija void().
- Rašymo į failą funkcija void().

## 16.ELEMENTŲ ŠALINIMAS IR ĮTERPIMAS IŠ/Į STRUKTŪRŲ MASYVĄ

### 1. Permoka už elektros energiją

Energijos skirstymo operatorius gavo duomenis apie n klientų atliktas įmokas tam tikru laikotarpiu. Kai kurie klientai permokėjo už sunaudotą elektros energiją. Energijos skirstymo operatorius, siekdamas išvengti nuostolių, jei elektros energija pabrangtų, nori išbraukti permokėjusius klientus iš sąrašo. Parašykite programą, kuri iš pradinio sąrašo išbrauktų už elektros energiją permokėjusius klientus. Laikoma, kad klientas permokėjo už elektros sunaudojimą, jei tą dieną, kai darbuotojas tikrino skaitiklio rodmenis, skirtumas tarp skaitiklio parodymų, kuriuos deklaravo gyventojas ir darbuotojo užfiksuotų skiriasi daugiau kaip 50 kilovatvalandžių, t.y. gyventojas nurodyti skaitiklio parodymai yra didesni už darbuotojo užfiksuotus. Tie gyventojai, kurie paskutinį kartą deklaravo duomenis daugiau kaip prieš 14 dienų, permokėjusiais nelaikomi, nors jų įrašyti skaitiklio rodmenys yra didesni už darbuotojo užfiksuotus daugiau kaip 50 kilovatvalandžių.

Pradinių duomenų failo permoka.txt pirmoje eilutėje įrašytas klientų skaičius n ( $n \leq 100$ ), elektros skaitiklio tikrinimo data (metai, mėnuo, diena). Tolesnėse n eilučių įrašyta informacija apie kiekvieną klientą: pavardė, paskutinė mokėjimo data, kliento deklaruoti skaitiklio rodmenys, energijos skirstymo operatoriaus užfiksuoti duomenys.

Rezultatų faile permokarez.txt turi būti pateiktas gyventojų sąrašas, gautas išbraukus permokėjusius gyventojus. Naujo sąrašo įrašai turi būti išdėstyti taip: pavardė, paskutinė mokėjimo data, kliento deklaruoti skaitiklio rodmenys, energijos operatoriaus užfiksuoti duomenys. Jei visi gyventojai permokėjo už elektros energiją, tuomet rezultatų faile turi būti įrašyta: SARASAS TUSCIAS.

Pradiniai duomenys	Rezultatai
3 2018 01 09 Jonaitis 2017 12 30 175 200 Antanaitis 2017 11 25 400 300 Petraitis 2017 12 30 600 500	Jonaitis 2017 12 30 175 200 Antanaitis 2017 11 25 400 300

### Reikalavimai

- Naudokite struktūras ir masyvus su struktūros tipo elementais.
- Pradinių duomenų skaitymo funkcija.
- Datų skirtumo skaičiavimo funkcija (-os).
- Vieno elemento iš sąrašo šalinimo funkcija.
- Perbėgimo per pradinį sąrašą, šalinant visus permokėjusius klientus, funkcija.
- Rezultatų rašymo į failą funkcija.

## 2. FLL regioninių varžybų apibendrintas protokolas

Pasaulinės First LEGO league varžybos vienu metu vyko keliuose regionuose. Kiekviename regione dalyvavo tam tikras skaičius komandų. Į pagrindinę varžybų koordinavimo būstinę kiekvienas regionas atsiuntė protokolus, kuriuose nurodyti komandų pavadinimai ir galutiniai įvertinimai. Varžybų būstinėje bendras protokolas formuojamas įterpiant į pirmojo regiono protokolą kitų regionų duomenis. Visų regionų protokoluose duomenys išdėstyti galutinių įvertinimų mažėjimo tvarka (vienodų įvertinimų nebuvo).

Sudarykite programą, kuri suformuotų bendrą protokolą.

Pradinių duomenų failo protokolas.txt pirmoje eilutėje nurodytas regionų skaičius  $n$  ( $n \leq 20$ ).

Toliau surašyti kiekvieno regiono duomenys: regione dalyvavusių komandų skaičius  $k$  ( $k \leq 20$ ), komandų pavadinimai (vienas žodis) ir vertinimai taškais (sveikasis skaičius).

Rezultatų faile protokolasrez.txt turi būti įrašytos komandos nuo stipriausios iki silpniausios, nurodant komandos pavadinimą, surinktų taškų skaičių ir iš kurio regiono komanda yra (1, 2 ir t.t.).

Pradiniai duomenys	Rezultatai
2	
3	
Kvatokliai 45	Informikai 75 2
Svajokliai 15	Programmeriai 65 2
Zirzliai 10	Ereliai 55 2
5	Kvatokliai 45 1
Informikai 75	Kolibriai 40 2
Programmeriai 65	Svajokliai 15 1
Ereliai 55	Rimtuoliai 12 2
Kolibriai 40	Zirzliai 10 1
Rimtuoliai 12	

### Reikalavimai

- Naudokite struktūras ir masyvus su struktūrų tipo elementais.
- Pradinių duomenų skaitymo funkcija, kuri suformuoja pirmojo regiono komandų sąrašą, į kurį bus įterpiamos likusių regionų komandos. Skaitymo funkcijoje gali būti kreipinių į kitas funkcijas.
- Vietos, kur įterpti naują elementą, nustatymo funkcija.
- Vieno elemento įterpimo funkcija.
- Rezultatų rašymo į failą funkcija.

## 3. Prekių galiojimo laikas

Parduotuvės darbuotojai tikrina prekių galiojimo laiką. Prekės, kurių galiojimo laikas baigiasi greičiau kaip po 3 dienų, perkliamos į kitą lentyną.

Parašykite programą, kuri iš pradinio prekių sąrašo išbrauktų ir į naują sąrašą perkeltų prekes, kurių galiojimo laikas baigsis greičiau kaip po 3 dienų.

Pradinių duomenų failo prekes.txt pirmoje eilutėje įrašytas prekių skaičius  $n$  ( $n \leq 500$ ) ir data (metai, mėnuo, diena), kada prekės tikrinamos. Tolesnėse  $n$  eilučių pateikiama informacija apie kiekvieną prekę: pavadinimas (vienas žodis), galiojimo pabaigos data (metai, mėnuo, diena).

Rezultatų faile prekesrez.txt turi būti pateikti 2 sąrašai: pradinis sąrašas, kuriame liko prekės, kurioms iki galiojimo pabaigos liko 3 ir daugiau dienų ir prekių, kurių galiojimo laikas baigiasi greičiau negu po 3 dienų, sąrašas. Jei kuriame nors sąrašė prekių nėra, tuomet išvedamas žodis NERA. Rezultatų išvedimo pavyzdys pateiktas žemiau.

3 2018 02 14 Duona 2018 02 15 Sviestas 2018 03 15 Makaronai 2018 12 15	ILGAS GALIOJIMO LAIKAS Sviestas 2018 03 15 Makaronai 2018 12 15 TRUMPAS GALIOJIMO LAIKAS Duona 2018 02 15
3 2018 02 14 Duona 2018 02 19 Sviestas 2018 03 15 Makaronai 2018 12 15	ILGAS GALIOJIMO LAIKAS Duona 2018 02 19 Sviestas 2018 03 15 Makaronai 2018 12 15 TRUMPAS GALIOJIMO LAIKAS NERA

### Reikalavimai

- Naudokite struktūras ir masyvus su struktūrų tipo elementais.
- Pradinių duomenų skaitymo funkcija.
- Datų skirtumo skaičiavimo funkcija (-os).
- Vieno elemento šalinimo funkcija.
- Naujo sąrašo formavimo funkcija.
- Rezultatų rašymo į failą funkcija (kreiptis 2 kartus: su ilgo galiojimo laiko prekių sąrašu ir trumpo galiojimo laiko sąrašu).