

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 tipu turi būti įrašytas žodis „NE“, pvz.:

LIETINGOS:

NE

Pradiniai duomenys	Rezultatai
10	
L 10.9	
S 11.2	LIETINGOS:
L 10.8	5 11.34 2
L 12.5	SAULETOS:
A 15.2	2 11.75 1
A 16.1	APSINIAUKUSIOS:
A 10.2	3 13.83 2
L 12.3	
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, grąžinanti apskaičiuotą reikšmę per funkcijos vardą. Funkciją panaudoti 3 kartus: lietingų, saulėtų ir apsiniaukusių dienų vidutinei temperatūrai skaičiuoti. Kiekio skaičiavimo funkcija, grąžinanti apskaičiuotą reikšmę per funkcijos vardą. Funkciją panaudoti 3 kartus: lietingų, saulėtų ir apsiniaukusių dienų kiekiui skaičiuoti.
- Funkcija, grąžinanti 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.

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 vienų 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.

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.

„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 Binks 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 grąžintų skaičių, kiek studentų gaus stipendijas (int).
- Realizuoti programos rezultatų spausdinimo į failą funkciją (void).

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 Paxeladinas 1 7 5 Akumuliumiotas 1 30 1.15	47.98 1 Hedeliksas

Reikalavimai

- Vaistą aprašanti struktūra.
- Vidutinę kainą skaičiuojanti funkcija.

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.

Sūrio pjaustymas

Parduotuvėse dažnai galima pamatyti pardavinėjamą įvairių 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 **mikrometrų** 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 kurias 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	14.611103142 16.269801734
10000 10000 20000 20000	24.092457788
40000 40000 50000 60000	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.

Automobiliai

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