

III gimnazijos klasė

Rikiavimo algoritmai (rikiavimas išrinkimu) (Aušra Markevičiūtė)

Pasiekimų sritis	(B) Algoritmai ir programavimas B2 Duomenų struktūrų naudojimas
Klasė	11 (III gimnazijos)
Tema	Rikiavimo algoritmai (rikiavimas išrinkimu).
Integruojami dalykai, pasiekimai	Anglų kalba, lietuvių kalba, matematika.
Kompetencijos	Pažinimo - mokiniai ugdomi informatinį mąstymą – gebėjimą spręsti įvairias aplinkos problemas (uždavinius) analizuodami ir apdorodami duomenis, taikyti rikiavimo išrinkimu algoritmą. Skaitmeninė - suvokia informacijos apdorojimo procesus, kuria skaitmeninį turinį programuodami. Komunikavimo - ugdomas gebėjimas pamokos metu teikti informatyvią grįžtamąją informaciją mokytojui, pagelbėti klasės draugams.
Tikslas	Gebėti taikyti klasikinį rikiavimo išrinkimu algoritmą.
Uždaviniai	1. Išnagrinėti pavyzdį, kuriame naudojamas duomenų rikiavimas. 2. Išsiaiškinti, kaip veikia rikiavimo išrinkimu algoritmas. 3. Įtvirtinti įgytas žinias sprendžiant uždavinius.
Planuojamas rezultatas	1. Suprasite ir gebėsite paaiškinti rikiavimo išrinkimu algoritmo veikimo principą. 2. Gebėsite pritaikyti rikiavimo išrinkimu algoritmą sprenddami uždavinius.
Specifinės priemonės / programinė įranga	Programinė įranga: 1. CodeBlocks, DevC++, onlinegdb.com ir kitos programavimo aplinkos. 2. Rekomenduojama naudoti programavimo uždavinių automatizuoto testavimo sistemą, pvz. APROMIS (programavimas.vgtu.lt), pikta.lt ir kt. 3. Interaktyvi lenta, multimedia projektorius. 4. <u>Literatūra ir kiti ištekliai:</u> <ul style="list-style-type: none"> • prof. Aleksandras TARGAMADZĖ, paskaitų užrašai. • Šiuolaikiškas Žvilgsnis į programavimą C++, Pasirenkamasis informacinių technologijų kursas XI-XII klasėms. R.Burbaitė, J.Blonskis, V.Bukšnaitis, Vilnius 2014. • open.ktu.edu programavimo pamokėlių ir pasirengimo informacinių technologijų valstybiniam brandos egzaminui kursai: <ul style="list-style-type: none"> ○ https://open.ktu.edu/course/view.php?id=88#section-5 ○ https://open.ktu.edu/course/view.php?id=75 • https://stackoverflow.com/questions/55325223/minimax-chess-algorithm-returns-bad-moves • https://www.youtube.com/watch?v=F67PIj0Dg_U • https://www.youtube.com/watch?v=2ItzEXHQx4 • https://visualgo.net/en/sorting?slide=1 • https://inf-knyga.nmakademija.lt/lt/latest/06_rikiavimas.html

	<ul style="list-style-type: none"> • https://atviri.emokymai.vu.lt/pluginfile.php/44637/mod_resource/content/0/17_Informatika_Rikiavimo_algoritmai.pdf • https://www.youtube.com/watch?v=dKExQ7_ylko
Mokymosi metodai	<p>Aiškinimas ir dialogas – analizuojant pavyzdį.</p> <p>Praktinis tyrimas – algoritmo užrašymas C++ kalba sprendžiant uždavinius.</p> <p>Darbas grupėse/porose – mokiniai bendradarbiauja tarpusavyje ir konsultuojasi su mokytoju.</p>
Mokinių atlikto darbo vertinimas ir įsivertinimas	<p>Slenkstinis – su mokytojo pagalba užrašo rikiavimo išrinkimu algoritmą, kai rikiuojama pagal vieną požymį.</p> <p>Patenkinamas – savarankiškai užrašo rikiavimo išrinkimu algoritmą, kai rikiuojama pagal vieną požymį.</p> <p>Pagrindinis – savarankiškai užrašo rikiavimo išrinkimu algoritmą, kai rikiuojama pagal kelis požymius.</p> <p>Aukštesnysis – sprendžia sunkesnius (olimpiadinius, valstybinių brandos egzaminų) uždavinius, kuriuose reikia pritaikyti rikiavimo algoritmą.</p> <p>Jei naudojama automatizuota programavimo uždavinių testavimo sistema, refleksijai galima panaudoti sistemos pateikiamus duomenis.</p> <p>Mokiniai įsivertina kas buvo sunkiausia, lengviausia, įdomiausia, ką sužinojo naujo – kaupiamasis vertinimas.</p>
Žinios prieš	<p>Gebėti skaityti pradinis duomenis iš failo, rezultatus išsaugoti faile.</p> <p>Mokėti sudėti pradinis duomenis ir rezultatus į masyvus, atlikti veiksmus su masyvų elementais.</p> <p>Naudotis elektroniniais žinynais. Taisyklingai vartoti kompiuterijos ir informacinių technologijų terminus, apibūdinti pagrindines sąvokas.</p>
Galimybės taikyti spec. poreikių mokiniams	<p>Spec. poreikių mokiniams galima pateikti sukurta projektą, kuriame jie galėtų atlikti paprastus papildymus pateiktuose programų koduose.</p>
Patarimai kolegoms, kurie naudos parengtą medžiagą	<ol style="list-style-type: none"> 1. Patys atlikite visas numatytas užduotis ir turėkite po ranka jų sprendimus. 2. Turėkite užduočių skirtingų gebėjimų mokiniams.

1 ETAPAS 🕒 15-20 minučių

1.1.Kartu su mokiniais nagrinėjama pavyzdinė programa „Populiariausios programavimo kalbos“.

TIOBE Software svetainėje (<http://www.tiobe.com>) skelbiami duomenys apie populiariausias programavimo kalbas: pavadinimas (skiriama 20 pozicijų), vieta 2022 metų spalio mėnesį, vieta 2023 metų spalio mėnesį, reitingas procentais, reitingo pokytis, procentais.

Parenkite programą, kuri surikiuotų kalbas reitingo didėjimo tvarka.

Pradinių duomenų faile *Duomenys.txt* įrašyta:

pirmoje eilutėje – programavimo kalbų skaičius n ($1 < n < 20$);

tolesnėse n eilučių – informacija apie kiekvieną programavimo kalbą: pavadinimas, vieta 2022 metų spalio mėnesį, vieta 2023 metų spalio mėnesį, reitingas ir reitingo pokytis.

Rezultatų faile *Rezultatai.txt* įrašyti kalbas, surikiuotas reitingo pokyčio didėjimo tvarka. Jei kalbų reitingo pokytis yra vienodas, jas palikti išdėstyti taip, kaip pradiniam sąrašė.

Reikalavimai programai:

1. naudokite struktūras ir masyvus su struktūros tipo elementais;
2. sukurkite pradinių duomenų skaitymo funkciją void C++;
3. sukurkite rezultatų rašymo į failą funkciją void C++;
4. sukurkite funkciją, rikiuojančią programavimo kalbas populiarumo pokyčio didėjimo tvarka.

Duomenys.txt	Rezultatai.txt
5	C# 5 5 7.71 3.29
C 2 2 12.08 -3.13	Java 4 3 8.92 -3.92
C# 5 5 7.71 3.29	C++ 3 4 10.67 0.74
C++ 3 4 10.67 0.74	C 2 2 12.08 -3.13
Java 4 3 8.92 -3.92	Python 1 1 14.82 -2.25
Python 1 1 14.82 -2.25	

Programos kūrimo algoritmas.

1 žingsnis. Aprašoma struktūra.

2 žingsnis. Sukuriama pradinių duomenų skaitymo iš failo funkcija.

3 žingsnis. Sukuriama rezultatų rašymo į failą funkcija.

4 žingsnis. Sukuriama rikiavimo funkcija.

```
//Programavimo kalbos
#include <bits/stdc++.h>
using namespace std;

const char CD[] = "Duomenys.txt";
const char CR[] = "Rezultatai.txt";
//1 žingsnis
struct Kalba {
    string pavadinimas;
    int vieta_2023;
    int vieta_2022;
```

```

    double reitingas;
    double pokytis;
};

void Skaityti (Kalba K[], int & n);
void Rasyti (Kalba K[], int n);
void Rikiuoti (Kalba K[], int n);

int main(){

    Kalba K[30]; int n;
    Skaityti (K, n);
    Rikiuoti (K, n);
    Rasyti (K, n);

    return 0;
}
//-----
//2 žingsnis
void Skaityti (Kalba K[], int & n){
    char eil[21];

    ifstream fd(CD);
    fd >> n;
    for (int i = 0; i < n; i++){
        fd.ignore();
        fd.get(eil, 21);
        K[i].pavadinimas = eil;
        fd >> K[i].vieta_2023 >> K[i].vieta_2022 >> K[i].reitingas >>
K[i].pokytis;
    }
    fd.close();
}
//-----
//3 žingsnis
void Rasyti (Kalba K[], int n){
    ofstream fr(CR);
    for (int i = 0; i < n; i++){
        fr << K[i].pavadinimas << K[i].vieta_2023 << " " <<
K[i].vieta_2022 << " " << K[i].reitingas << " " << K[i].pokytis << endl;
    }
    fr.close();
}
//-----
//4 žingsnis
void Rikiuoti (Kalba K[], int n){
    int m;
    for (int i = 0; i < n - 1; i++){
        m = i;
        for (int j = i + 1; j < n; j++){
            if (K[j].reitingas < K[m].reitingas) m = j;
        }
        swap(K[i], K[m]);
    }
}

```

}
}
1.2. Aiškinamas rikiavimo išrinkimu (minmax) algoritmas su sveikaisiais skaičiais, kad mokiniai suprastų, kaip jis veikia.

Aiškinimas:

Tarkime, masyvo A(n) elementus reikia išdėstyti didėjančia (mažėjančia) seka.

Skaičių seka:

110 34 9 17 -55 99

Didėjanti seka:

-55 9 17 34 99 110

Mažėjanti seka:

110 99 34 17 9 -55

Intervale nuo 0-nio iki (n-1)-ojo elemento randame min. narį. Jį keičiame vietomis su intervalo pradžios nariu.

Intervale nuo 1-ojo iki (n-1)-ojo elemento randame min. narį. Jį keičiame vietomis su intervalo pradžios nariu.

Intervale nuo 2-ojo iki (n-1)-ojo elemento randame min. narį. Jį keičiame vietomis su intervalo pradžios nariu.

Kartojame, kol intervale lieka tik viena reikšmė.

Atlikus viską, masyvo elementai išsidėstys didėjimo tvarka.

Mažiausią reikšmę surandame visame masyve, t. y. nuo nulinio iki n-1-ojo elemento.

110	34	9	17	-55	99
0	1	2	3	4	5

Sukeičiame vietomis surastąjį elemento ir nulinio elemento reikšmes.

-55	34	9	17	110	99
0	1	2	3	4	5

Mažiausią reikšmę surandame visame masyve, t. y. nuo pirmojo iki n-1-ojo elemento.

-55	34	9	17	110	99
0	1	2	3	4	5

Sukeičiame vietomis surastąjį elemento ir pirmojo elemento reikšmes.

-55	9	34	17	110	99
0	1	2	3	4	5

Mažiausią reikšmę surandame visame masyve, t. y. nuo antrojo iki n-1-ojo elemento.

-55	9	34	17	110	99
0	1	2	3	4	5

Sukeičiame vietomis surastojų elementų ir antrojo elementų reikšmes.

-55	9	17	34	110	99
0	1	2	3	4	5

Mažiausią reikšmę surandame visame masyve, t. y. nuo trečiojo iki n-1-ojo elemento.

-55	9	17	34	110	99
0	1	2	3	4	5

Sukeičiame vietomis surastojų elementų ir trečiojo elementų reikšmes.

-55	9	17	34	110	99
0	1	2	3	4	5

Mažiausią reikšmę surandame visame masyve, t. y. nuo ketvirtojo iki n-1-ojo elemento.

-55	9	17	34	110	99
0	1	2	3	4	5

Sukeičiame vietomis surastojų elementų ir ketvirtojo elementų reikšmes.

-55	9	17	34	99	111
0	1	2	3	4	5

Mažiausią reikšmę surandame visame masyve, t. y. nuo penktojo iki n-1-ojo elemento.

-55	9	17	34	99	111
0	1	2	3	4	5

Sukeičiame vietomis surastojų elementų ir ketvirtojo elementų reikšmes.

-55	9	17	34	99	111
0	1	2	3	4	5

Surikiuoja skaičių masyvą A(n) didėjančia tvarka - *Modifikuotas algoritmas*. Taip C++ jis rašomas kodu.

```
void MinMax(int A[], int n)
{
    int c; // pagalbinis kintamasis reikšmių sukeitimui
    int m; // elemento su mažiausia reikšme indeksas
    for (int i = 0; i < n - 1; i++) {
        m = i;
        for (int j = i + 1; j < n; j++)
            if (A[j] < A[m])
                m = j;
        c = A[i];
        A[i] = A[m];
        A[m] = c;
    }
}
```

Mokytojui ruošiantis pamokai rekomenduojama peržiūrėti papildomai:



2 ETAPAS 🕒 15-20 minučių

Savarankiškas užduočių sprendimas

Slenkstiniam lygiui galima skirti pavyzdinėje programoje pakeisti rikiavimo požymį, ir/arba rikiavimo tvarką pasinaudojant mokytojo pagalba.

Kitų lygių mokiniai pavyzdinėje programoje turėtų atlikti pakeitimus savarankiškai:

Siūloma:

- Surikiuoti kalbas abėcėlės ir/arba nuo Z iki A tvarka.
- Surikiuoti kalbas pagal reitingo pokytį mažėjimo tvarka.
- Surikiuoti kalbas pagal 2022 ir/arba 2023 metų vietą didėjimo/mažėjimo tvarka.

Mokiniam, kurie atliko užduotis greičiau ir suprato rikiavimo išrinkimu algoritmo veikimą siūloma spręsti diferencijuotus uždavinius.

a. Rikiavimas pagal vieną požymį:

1. Gamykla gamina įvairaus tipo plytas (skiriama 20 pozicijų). Žinomas plytų aukštis, plotis, ilgis (matmenys pateikti mm) ir kaina.

Parenkite programą, kuri surikiuotų plytas pagal ilgį mažėjimo tvarka.

Pradinių duomenų faile *Duomenys.txt* įrašyta:

pirmoje eilutėje – plytos pavadinimas n ($1 < n < 20$);

tolesnėse n eilučių – informacija apie kiekvieną plytą: pavadinimas, aukštis, plotis, ilgis ir kaina.

Rezultatų faile *Rezultatai.txt* įrašyti plytas, surikiuotas ilgio mažėjimo tvarka.

Reikalavimai programai:

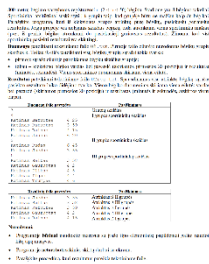
1. naudokite struktūras ir masyvus su struktūros tipo elementais;
2. sukurkite pradinių duomenų skaitymo funkciją void C++;
3. sukurkite rezultatų rašymo į failą funkciją void C++;
4. sukurkite funkciją, rikiuojančią plytas pagal ilgį mažėjimo tvarka.

Duomenys.txt	Rezultatai.txt
5	I tipo 88 120 255
II tipo 71 115 240 3.00	2.40
III tipo 61 110 250 2.60	III tipo 61 110 250
I tipo 88 120 255 2.40	2.60
IV tipo 75 112 245 2.55	IV tipo 75 112 245
V tipo 80 118 235 2.70	2.55
	II tipo 71 115 240
	3.00
	V tipo 80 118 235
	2.70

b. Rikiavimas pagal kelis požymius:

```
1. Programos pavadinimas: Rikiavimas pagal kelis požymius.
2. Programos tikslas: surikiuoti duomenis pagal kelis požymius.
3. Programos aprašymas:
   - Programoje yra duomenų masyvas, kuriame yra duomenys apie
     įvairius objektus.
   - Programoje yra funkcija, kuri surikiuoja duomenis pagal
     kelis požymius.
   - Programoje yra funkcija, kuri rašo rezultatus į failą.
4. Programos kodas:
   #include <iostream>
   #include <string>
   #include <vector>
   #include <algorithm>
   #include <fstream>
   using namespace std;
   struct Duomenys {
       string pavadinimas;
       int aukstis;
       int plotis;
       int ilgis;
       int kaina;
   };
   int main() {
       // ...
   }
5. Programos testavimas:
   - Programą testuoti galima su bet kokiais duomenimis.
   - Programą testuoti galima su duomenimis, kuriuose yra
     objektai, kurie turi vieną ar daugiau požymių, kuriems
     reikia surikiuoti.
6. Programos rezultatai:
   - Programos rezultatai yra surikiuoti duomenys, kurie
     išrašomi į failą.
7. Programos komentaras:
   - Programoje yra komentarai, kurie paaiškina kodą.
8. Programos autorius:
   - Programos autorius yra [Jūsų vardas].
9. Programos data:
   - Programos data yra [Jūsų data].
10. Programos versija:
   - Programos versija yra [Jūsų versija].
```

c. Aukštesniųjų gebėjimų mokiniams



3 ETAPAS ⌚ 5 minučių

Su mokiniais apibendrinamas atliktas darbas. Akcentuojama:

Tik rikiuotame rinkinyje galima iš esmės pagreitinti paiešką!

Rikiuojant reikia nurodyti, kokį objekto lauką (-us) naudosite rikiavimui.

Rikiavimą galima atlikti naudojant tiek skaitines reikšmes (skaičius), tiek ir simbolius ar tekstus ir sudėtingesnes duomenų struktūras.

1. Refleksija. Jei mokiniai dirbo pagal pavyzdinį kodą, galima jų paprašyti įvertinti sėkmes ir nesėkmes aptariant žodžiu ar užpildant mokytojo parengtą apklausą (galima naudoti Google forms, apklausa.lt ir pan.).
2. Jei mokiniai sprendė uždavinius savarankiškai, galima jų paprašyti įvardinti sunkiausias vietas.
3. Dalį refleksijos galima perkelti kitą pamoką, jei mokiniai nebaigė išspręsti uždavinių klasėje.

Rikiavimo algoritmai (burbulo, įterpimu, sukeitimu) (Aušra Markevičiūtė)

Pasiekimų sritis	(B) Algoritmai ir programavimas B2 Duomenų struktūrų naudojimas
Klasė	11 (III gimnazijos)
Tema	Rikiavimo algoritmai (burbulo, įterpimu, sukeitimu).
Integruojami dalykai, pasiekimai	Anglų kalba, lietuvių kalba, matematika.
Kompetencijos	<p>Informatinio mąstymo subkompetencija – gebėjimą spręsti įvairias aplinkos problemas (uždavinius) analizuodami ir apdorodami duomenis, taikyti rikiavimo burbuliuku, įterpimu, sukeitimu algoritmus.</p> <p>Skaitmeninė - suvokia informacijos apdorojimo procesus, kuria skaitmeninį turinį programuodami.</p> <p>Komunikavimo - ugdomas gebėjimas pamokos metu teikti informatyvią grįžtamąją informaciją mokytojui, pagelbėti klasės draugams.</p>
Tikslas	Gebėti taikyti burbuliuko, įterpimo ir sukeitimo rikiavimo algoritmus.

Uždaviniai	<p>4. Išnagrinėsite pavyzdį, kuriame naudojamas duomenų rikiavimas.</p> <p>5. Išsiaiškinsite, kaip veikia burbuliuko, įterpimo ir sukeitimo rikiavimo algoritmai.</p> <p>6. Įtvirtinsite įgytas žinias sprendžiant uždavinius.</p>
Planuojamas rezultatas	<p>3. Suprasite ir gebėsite paaiškinti burbuliuko, įterpimo, sukeitimo rikiavimo algoritmų veikimo principus.</p> <p>4. Gebėsite pritaikyti burbuliuko, įterpimo, sukeitimo rikiavimo algoritmus sprenddami uždavinius.</p>
Specifinės priemonės / programinė įranga	<p>Programinė įranga:</p> <p>5. CodeBlocks, DevC++, onlinegdb.com ir kitos programavimo aplinkos.</p> <p>6. Rekomenduojama naudoti programavimo uždavinių automatizuoto testavimo sistemą, pvz. APROMIS (programavimas.vgtu.lt), pikta.lt ir kt.</p> <p>7. Interaktyvi lenta, multimedia projektorius.</p> <p>8. <u>Literatūra ir kiti ištekliai:</u></p> <ul style="list-style-type: none"> • prof. Aleksandras TARGAMADZĖ, paskaitų užrašai. • Šiuolaikiškas Žvilgsnis į programavimą C++, Pasirenkamasis informacinių technologijų kursas XI-XII klasėms. R.Burbaitė, J.Blonskis, V.Bukšnaitis, Vilnius 2014. • open.ktu.edu programavimo pamokėlių ir pasirengimo informacinių technologijų valstybiniam brandos egzaminui kursai: <ul style="list-style-type: none"> ○ https://open.ktu.edu/course/view.php?id=88#section-5 ○ https://open.ktu.edu/course/view.php?id=75 • https://stackoverflow.com/questions/55325223/minimax-chess-algorithm-returns-bad-moves • https://www.youtube.com/watch?v=F67PIj0Dg_U • https://www.youtube.com/watch?v=2ItzEXHQQx4 • https://visualgo.net/en/sorting?slide=1 • https://atviri.emokymai.vu.lt/pluginfile.php/44637/mod_resource/content/0/17_Informatika_Rikiavimo_algoritmai.pdf • https://www.youtube.com/watch?v=dKExQ7_ylko • https://informatikosvbe.lt/programavimas-cpp/masyvai-ir-algoritmai/vienmaciai/rikiavimas/#rikiavimas-burbuliuko-metodu-angl-bubble-sort • http://itpamoka.lt/ftp/4gimn/KTU%20pamokeles_2015_2016.pdf
Mokymosi metodai	<p>Aiškinimas ir dialogas – analizuojant pavyzdį.</p> <p>Praktinis tyrimas – algoritmo užrašymas C++ kalba sprendžiant uždavinius.</p> <p>Darbas grupėse/porose – mokiniai bendradarbiauja tarpusavyje ir konsultuojasi su mokytoju.</p>
Mokinių atlikto darbo vertinimas ir įsivertinimas	<p>Slenkstinis – su mokytojo pagalba užrašo burbuliuko, įterpimo, sukeitimo rikiavimo algoritmus, kai rikiuojama pagal vieną požymį.</p> <p>Patenkinamas – savarankiškai užrašo burbuliuko, įterpimo bei sukeitimo rikiavimo algoritmus, kai rikiuojama pagal vieną požymį.</p> <p>Pagrindinis – savarankiškai užrašo burbuliuko, įterpimo bei sukeitimo rikiavimo algoritmą, kai rikiuojama pagal kelis požymius.</p>

	<p>Aukštesnysis – sprendžia sunkesnius (olimpiadinius, valstybinių brandos egzaminų) uždavinius, kuriuose reikia pritaikyti rikiavimo algoritmus.</p> <p>Jei naudojama automatizuota programavimo uždavinių testavimo sistema, refleksijai galima panaudoti sistemos pateikiamus duomenis.</p> <p>Mokiniai įsivertina kas buvo sunkiausia, lengviausia, įdomiausia, ką sužinojo naujo – kaupiamasis vertinimas.</p>
Žinios prieš	<p>Gebėti skaityti pradinis duomenis iš failo, rezultatus išsaugoti faile.</p> <p>Mokėti sudėti pradinis duomenis ir rezultatus į masyvus, atlikti veiksmus su masyvų elementais.</p> <p>Naudotis elektroniniais žinynais. Taisyklingai vartoti kompiuterijos ir informacinių technologijų terminus, apibūdinti pagrindines sąvokas.</p>
Galimybės taikyti spec. poreikių mokiniams	<p>Spec. poreikių mokiniams galima pateikti sukurtą projektą, kuriame jie galėtų atlikti paprastus papildymus pateiktuose programų koduose.</p>
Patarimai kolegoms, kurie naudos parengtą medžiagą	<ol style="list-style-type: none"> 3. Patys atlikite visas numatytas užduotis ir turėkite po ranka jų sprendimus. 4. Turėkite užduočių skirtingų gebėjimų mokiniams.

1 ETAPAS 🕒 15-20 minučių. Rikiavimo burbulu, įterpimu ir sukeitimu algoritmai

1.3. Kartu su mokiniais nagrinėjama pavyzdinė programa „Populiariausios programavimo kalbos“.

TIOBE Software svetainėje (<http://www.tiobe.com>) skelbiami duomenys apie populiariausias programavimo kalbas: pavadinimas (skiriama 20 pozicijų), vieta 2022 metų spalio mėnesį, vieta 2023 metų spalio mėnesį, reitingas procentais, reitingo pokytis, procentais.

Parenkite programą, kuri surikiuotų kalbas reitingo didėjimo tvarka.

Pradinių duomenų faile *Duomenys.txt* įrašyta:

pirmoje eilutėje – programavimo kalbų skaičius n ($1 < n < 20$);

tolesnėse n eilučių – informacija apie kiekvieną programavimo kalbą: pavadinimas, vieta 2022 metų spalio mėnesį, vieta 2023 metų spalio mėnesį, reitingas ir reitingo pokytis.

Rezultatų faile *Rezultatai.txt* įrašyti kalbas, surikiuotas reitingo pokyčio didėjimo tvarka. Jei kalbų reitingo pokytis yra vienodas, jas palikti išdėstyti taip, kaip pradiniam sąrašė.

Reikalavimai programai:

1. naudokite struktūras ir masyvus su struktūros tipo elementais;
2. sukurkite pradinių duomenų skaitymo funkciją void C++;
3. sukurkite rezultatų rašymo į failą funkciją void C++;
4. sukurkite funkcijas, rikiuojančias programavimo kalbas populiarumo pokyčio didėjimo tvarka.

Duomenys.txt	Rezultatai.txt
5	C# 5 5 7.71 3.29
C 2 2 12.08 -3.13	Java 4 3 8.92 -3.92
C# 5 5 7.71 3.29	C++ 3 4 10.67 0.74
C++ 3 4 10.67 0.74	C 2 2 12.08 -3.13
Java 4 3 8.92 -3.92	Python 1 1 14.82 -2.25
Python 1 1 14.82 -2.25	

Programos kūrimo algoritmas.

1 žingsnis. Aprašoma struktūra.

2 žingsnis. Sukuriama pradinių duomenų skaitymo iš failo funkcija.

3 žingsnis. Sukuriama rezultatų rašymo į failą funkcija.

4 žingsnis. Sukuriamos rikiavimo (burbuliuku, įterpimu, sukeitimu) funkcijos.

```
//Programavimo kalbos
#include <bits/stdc++.h>
using namespace std;

const char CD[] = "Duomenys.txt";
const char CR[] = "Rezultatai.txt";
//1 žingsnis
struct Kalba {
    string pavadinimas;
    int vieta_2023;
    int vieta_2022;
    double reitingas;
```

```

    double pokytis;
};

void Skaityti (Kalba K[], int & n);
void Rasyti (Kalba K[], int n);
void Rikiuoti_isrinkimu (Kalba K[], int n);
void Rikiuoti_burbulu (Kalba K[], int n);
void Rikiuoti_iterpimu (Kalba K[], int n);
void Rikiuoti_sukeitimu (Kalba K[], int n);

int main(){

    Kalba K[30]; int n;
    Skaityti (K, n);
    //Rikiuoti_isrinkimu (K, n);
    //Rikiuoti_burbulu(K, n);
    //Rikiuoti_iterpimu(K, n);
    Rikiuoti_sukeitimu(K, n);
    Rasyti (K, n);

    return 0;
}
//-----
//2 žingsnis
void Skaityti (Kalba K[], int & n){
    char eil[21];

    ifstream fd(CD);
    fd >> n;
    for (int i = 0; i < n; i++){
        fd.ignore();
        fd.get(eil, 21);
        K[i].pavadinimas = eil;
        fd >> K[i].vieta_2023 >> K[i].vieta_2022 >> K[i].reitingas >>
K[i].pokytis;
    }
    fd.close();
}
//-----
//3 žingsnis
void Rasyti (Kalba K[], int n){
    ofstream fr(CR);
    for (int i = 0; i < n; i++){
        fr << K[i].pavadinimas << K[i].vieta_2023 << " " << K[i].vieta_2022
<< " " << K[i].reitingas << " " << K[i].pokytis << endl;
    }
    fr.close();
}
//-----
//4 žingsnis. Rikiavimas išrinkimu
void Rikiuoti_isrinkimu (Kalba K[], int n){
    int m;
    for (int i = 0; i < n - 1; i++){
        m = i;
        for (int j = i + 1; j < n; j++){
            if (K[j].reitingas < K[m].reitingas) m = j;
        }
        swap(K[i], K[m]);
    }
}
//-----

```

```

//Rikiavimas burbulu
void Rikiuoti_burbulu (Kalba K[], int n){
    int i = 0;
    while (i < n-1){
        if (K[i].reitingas < K[i+1].reitingas) i++;
        else {
            swap(K[i], K[i+1]);
            if (i > 1) i--;
            else i++;
        }
    }
}
//-----
//Rikiavimas ėterpimu
void Rikiuoti_iterpimu (Kalba K[], int n){
    for (int i = 1; i < n; i++){
        Kalba k = K[i]; int j = i;
        while (j > 0 && k.reitingas < K[j-1].reitingas){
            K[j] = K[j-1]; j--;
        }
        K[j] = k;
    }
}
//-----
//Rikiavimas sukeitimu
void Rikiuoti_sukeitimu (Kalba K[], int n){
    for (int i = 0; i < n - 2; i++)
        for (int j = 0; j < n - i - 1; j++)
            if (K[j].reitingas > K[j+1].reitingas)
                swap(K[j], K[j+1]);
}

```

Mokytojas algoritmus paaiškina žodžiu, mokiniai savarankiškai pritaiko (burbuliuko, ėterpimo, sukeitimo) algoritmus spręsdami praėjusios pamokos spręstus uždavinius.

Mokytojui ruošiantis pamokai rekomenduojama peržiūrėti papildomai:

<https://visualgo.net/en/sorting?slide=1>



2 ETAPAS 🕒 15-20 minučių

Savarankiškas užduočių sprendimas

Mokiniais galima pasiūlyti pakeisti praėjusios pamokos užduotyse rikiavimo algoritmus.

Slenkstiniam lygiui galima skirti pavyzdinėje programoje pakeisti pradinius duomenis, ir/arba rikiavimo tvarką pasinaudojant mokytojo pagalba.

Kitų lygių mokiniai pavyzdinėje programoje turėtų atlikti pakeitimus savarankiškai:

Siūloma:

- Surikiuoti savo pradinius duomenis-pagal kitą požymį.
- Surikiuoti savo pradinius duomenis-pagal kelis požymius.

Mokiniam, kurie atliko užduotis greičiau ir suprato rikiavimo išrinkimu algoritmo veikimą siūloma spręsti diferencijuotus uždavinius.

a. Rikiavimas pagal vieną požymį:

1. Klasėje mokosi šie mokiniai: Jonas, Petras, Gintas, Adomas, Povilas, Gabriele, Gintare, Julija. Padėkite juos surašyti abėcėlės tvarka.

Reikalavimai programai:

1. naudokite struktūras ir masyvus su struktūros tipo elementais;
2. sukurkite pradinių duomenų skaitymo funkciją void C++;
3. sukurkite rezultatų rašymo į failą funkciją void C++;
4. pritaikykite duomenų rikiavimui burbuliuko rikiavimo algoritimą C++;
5. išveskite surikiuotus duomenis į ekraną C++.

Duomenys.txt	Rezultatai.txt
Jonas, Petras, Gintas, Adomas, Povilas, Gabriele, Gintare, Julija	Adomas Gabriele Gintare Gintas Jonas Julija Petras Povilas

Mokytojas sprendimą gali pasižiūrėti (pavyzdys Nr 4):
<https://informatikosvbe.lt/programavimas-cpp/masyvai-ir-algoritmai/vienmaciai/rikiavimas/#rikiavimas-burbuliuko-metodu-angl-bubble-sort>

b) Rikiavimas pagal du požymius:

Name gyvena keturi gyventojai, žinomas jų vardas ir amžius. Surikuosite gyventojus pagal alfabetą ir ir amžių didėjimo tvarka.

Reikalavimai programai:

1. naudokite struktūras ir masyvus su struktūros tipo elementais;
2. sukurkite pradinių duomenų skaitymo funkciją void C++;
3. sukurkite rezultatų rašymo į failą funkciją void C++;
4. pritaikykite duomenų rikiavimui burbuliuko rikiavimo algoritimą C++;
5. išveskite surikiuotus duomenis į ekraną C++.

Duomenys.txt	Rezultatai.txt
Jonas 58 Petras 30 Inga 35 Aina 30	Aina 30 Petras 30 Inga 35 Jonas 58

Mokytojas sprendimą gali pasižiūrėti (pavyzdys Nr 6):
<https://informatikosvbe.lt/programavimas-cpp/masyvai-ir-algoritmai/vienmaciai/rikiavimas/#rikiavimas-burbuliuko-metodu-angl-bubble-sort>

Siuvykla siuva įvairaus tipo drabužius (skiriama 20 pozicijų). Žinomas drabužių asortimentas, jų kiekis ir kaina.

Parentkite programą, kuri surikiuotų: 1) siuvykloje siuvenus drabužius pagal alfabetą ir jų kiekį; 2) surikiuokite siuvenus gaminius nuo brangiausio iki pigiausio.

Pradinių duomenų faile *Duomenys.txt* įrašyta:

pirmoje eilutėje – gaminio pavadinimas n ($1 < n < 20$);

tolesnėse n eilučių – informacija apie kiekvieną gaminį: pavadinimas, jų kiekis, kaina.

Rezultatų faile *Rezultatai.txt* įrašyti gaminius, surikiuotus pagal alfabetą bei jų kiekį; surikiuotus gaminius nuo brangiausio iki pigiausio.

Reikalavimai programai:

1. naudokite struktūras ir masyvus su struktūros tipo elementais;
2. sukurkite pradinių duomenų skaitymo funkciją void C++;
3. sukurkite rezultatų rašymo į failą funkciją void C++;
4. sukurkite funkcijas, rikiuojančias gaminius: 1) pagal alfabetą bei jų kiekį; 2) surikiuotus gaminius nuo brangiausio iki pigiausio.

Duomenys.txt	Rezultatai.txt
5	pagal alfabetą bei jų kiekį
Sijonas 71 30.00	Džemperis 80 35.70
Suknelė 61 36.60	Kelnės 88 31.00
Kelnės 88 31.00	Palaidinė 75 25.55
Palaidinė 75 25.55	Sijonas 71 30.00
Džemperis 80 35.70	Suknelė 61 36.60
	Pagal kainą
	Suknelė 61 36.60
	Džemperis 80 35.70
	Kelnės 88 31.00
	Sijonas 71 30.00
	Palaidinė 75 25.55

c. Aukštesniųjų gebėjimų mokiniams

Prekių apskaita

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

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

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

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

```

prekes.txt
-----
Prekes: 10
-----
Prekės pavadinimas | Kaina
-----
Arbata | 0.99
Kava | 1.20
Mėlynė | 1.50
Pienas | 1.80
Sūris | 2.00
-----
pakeitimai.txt
-----
Prekes: 10
-----
Prekės pavadinimas | Kaina | Keitimai
-----
Arbata | 0.99 | +
Kava | 1.20 | +
Mėlynė | 1.50 | +
Pienas | 1.80 | +
Sūris | 2.00 | +
-----

```

Reikalavimai programai:

1. naudokite struktūras ir masyvus su struktūros tipo elementais;
2. sukurkite pradinių duomenų skaitymo funkciją void C++;
3. sukurkite rezultatų rašymo į failą funkciją void C++;
4. sukurkite funkcijas, rikiuojančias duomenis.

3 ETAPAS 🕒 5 minučių

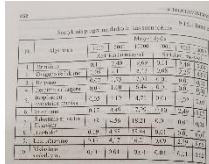
Su mokiniais apibendrinamas atliktas darbas. Akcentuojama:

Rikiavimas išrinkimu naudingas kai pradinis masyvas surikiuotas arba beveik surikiuotas. Dideliame nerikiuotame masyve greičiau veiks sukeitimo algoritmas nei burbuliuko. Burbuliuko algoritme atliekama daugiau veiksmų.

Burbulo algoritmo veikimo principas primena virimą, kai oro burbulai kyla į paviršių, dėl to jis ir vadinamas burbulo rikiavimo metodu. Burbulo rikiavimo metodas yra vienas iš paprastų, bet nelabai efektyvių rikiavimo algoritmų.

Rikiuojant reikia nurodyti, kokį objekto lauką (-us) naudosite rikiavimui.

Rikiavimą galima atlikti naudojant tiek skaitines reikšmes (skaičius), tiek ir simbolius ar tekstus ir sudėtingesnes duomenų struktūras.



4. Refleksija. Jei mokiniai dirbo pagal pavyzdinį kodą, galima jų paprašyti įvertinti sėkmes ir nesėkmes aptariant žodžiu ar užpildant mokytojo parengtą apklausą (galima naudoti Google forms, Mentimeter, apklausa.lt ir pan.).
5. Jei mokiniai sprendė uždavinius savarankiškai, galima jų paprašyti įvardinti sunkiausias vietas.
6. Dalį refleksijos galima perkelti į kitą pamoką, jei mokiniai nebaigė išspręsti uždavinių klasėje.

Rikiavimo algoritmų taikymas (Aušra Markevičiūtė)

Pasiekimų sritis	(B) Algoritmai ir programavimas B2 Duomenų struktūrų naudojimas
Klasė	11 (III gimnazijos)
Tema	Rikiavimo algoritmų taikymas.
Integruojami dalykai, pasiekimai	Anglų kalba, lietuvių kalba, matematika.
Kompetencijos	Informatinio mąstymo subkompetencija – gebėjimą spręsti įvairias aplinkos problemas (uždavinius) analizuodami ir apdorodami duomenis, taikyti rikiavimo algoritmus. Skaitmeninė - suvokia informacijos apdorojimo procesus, kuria skaitmeninį turinį programuodami.

	Komunikavimo - ugdomas gebėjimas pamokos metu teikti informatyvią grįžtamąją informaciją mokytojui, pagelbėti klasės draugams.
Tikslas	Gebėti taikyti klasikinį rikiavimo išrinkimu, burbuliuko, įterpimo bei sukeitimo algoritmus.
Uždaviniai	7. Įtvirtinsite įgytas žinias sprendžiant uždavinius.
Planuojamas rezultatas	5. Gebėsite pritaikyti rikiavimo išrinkimu, burbuliuko, įterpimo bei sukeitimo algoritmus sprenddami uždavinius.
Specifinės priemonės / programinė įranga	<p>Programinė įranga:</p> <p>9. CodeBlocks, DevC++, onlinegdb.com ir kitos programavimo aplinkos.</p> <p>10. Rekomenduojama naudoti programavimo uždavinių automatizuoto testavimo sistemą, pvz. APROMIS (programavimas.vgtu.lt), pikta.lt ir kt.</p> <p>11. Interaktyvi lenta, multimedia projektorius.</p> <p>12. Literatūra ir kiti ištekliai:</p> <ul style="list-style-type: none"> • prof. Aleksandras TARGAMADZĖ, paskaitų užrašai. • Šiuolaikiškas Žvilgsnis į programavimą C++, Pasirenkamasis informacinių technologijų kursas XI-XII klasėms. R.Burbaitė, J.Blonskis, V.Bukšnaitis, Vilnius 2014. • open.ktu.edu programavimo pamokėlių ir pasirengimo informacinių technologijų valstybiniam brandos egzaminui kursai: <ul style="list-style-type: none"> ○ https://open.ktu.edu/course/view.php?id=88#section-5 ○ https://open.ktu.edu/course/view.php?id=75 • https://stackoverflow.com/questions/55325223/minimax-chess-algorithm-returns-bad-moves • https://www.youtube.com/watch?v=F67PIj0Dg_U • https://www.youtube.com/watch?v=2ItzEXHQx4 • https://visualgo.net/en/sorting?slide=1 • https://inf-knyga.nmakademija.lt/lt/latest/06_rikiavimas.html • https://atviri.emokymai.vu.lt/pluginfile.php/44637/mod_resource/content/0/17_Informatika_Rikiavimo_algoritmai.pdf • https://www.youtube.com/watch?v=dKExQ7_ylko • http://itpamoka.lt/ftp/4gimn/KTU_pamokeles_2013_2014.pdf
Mokymosi metodai	Praktinis tyrimas – algoritmo užrašymas C++ kalba sprendžiant uždavinius. Darbas grupėse/porose – mokiniai bendradarbiauja tarpusavyje ir konsultuojasi su mokytoju.
Mokinių atlikto darbo vertinimas ir įsivertinimas	<p>Slenkstinis – su mokytojo pagalba sprendžia uždavinį taikydami rikiavimo išrinkimu, burbuliuko, įterpimo bei sukeitimo algoritmus, kai rikiuojama pagal vieną požymį.</p> <p>Patenkinamas – savarankiškai užrašo rikiavimo išrinkimu, burbuliuko, išrinkimo, sukeitimo algoritmus, kai rikiuojama pagal vieną požymį.</p> <p>Pagrindinis – savarankiškai užrašo rikiavimo išrinkimu, burbuliuko, išrinkimo, sukeitimo algoritmus, kai rikiuojama pagal kelis požymius.</p>

	<p>Aukštesnysis – sprendžia sunkesnius (olimpiadinius, valstybinių brandos egzaminų) uždavinius, kuriuose reikia pritaikyti rikiavimo išrinkimu, burbuliuko, įterpimo bei sukeitimo algoritmus.</p> <p>Jei naudojama automatizuota programavimo uždavinių testavimo sistema, refleksijai galima panaudoti sistemos pateikiamus duomenis.</p> <p>Mokiniai įsivertina kas buvo sunkiausia, lengviausia, įdomiausia, ką sužinojo naujo – kaupiamasis vertinimas.</p>
Žinios prieš	<p>Gebėti skaityti pradinis duomenis iš failo, rezultatus išsaugoti faile.</p> <p>Mokėti sudėti pradinis duomenis ir rezultatus į masyvus, atlikti veiksmus su masyvų elementais.</p> <p>Naudotis elektroniniais žinynais. Taisyklingai vartoti kompiuterijos ir informacinių technologijų terminus, apibūdinti pagrindines sąvokas.</p>
Galimybės taikyti spec. poreikių mokiniams	<p>Spec. poreikių mokiniams galima pateikti sukurta projektą, kuriame jie galėtų atlikti paprastus papildymus pateiktuose programų koduose.</p>
Patarimai kolegoms, kurie naudos parengtą medžiagą	<ol style="list-style-type: none"> 5. Patys atlikite visas numatytas užduotis ir turėkite po ranka jų sprendimus. 6. Turėkite užduočių skirtingų gebėjimų mokiniams.

1 ETAPAS ⌚ 15-20 minučių

Mokytojas pateikia užduotis, mokiniai pagal gebėjimus pasirenka užduotis ir sprendžia. Savarankiškas užduočių sprendimas.

a. Rikiavimas pagal vieną požymį:

1. Dėžės sandėlyje

Sandėlyje yra įvairaus dydžio dėžių (skiriama 20 pozicijų). Žinoma: iš ko pagaminta dėžė, dėžių ilgis, plotis, aukštis, (matmenys pateikti cm) ir svoris.

Parenkite programą, kuri surikiuotų dėžes pagal aukštį didėjimo tvarka.

Pradinių duomenų faile *Duomenys.txt* įrašyta:

pirmoje eilutėje – iš ko padaryta dėžė n ($1 < n < 20$);

tolesnėse n eilučių – informacija apie kiekvieną dėžę: iš ko padaryta dėžė, dėžių ilgis, plotis, aukštis ir svoris (g).

Rezultatų faile *Rezultatai.txt* įrašyti dėžės, surikiuotas pagal aukštį didėjimo tvarka.

Reikalavimai programai:

1. naudokite struktūras ir masyvus su struktūros tipo elementais;
2. sukurkite pradinių duomenų skaitymo funkciją void C++;
3. sukurkite rezultatų rašymo į failą funkciją void C++;
4. sukurkite funkcijas, rikiuojančias dėžes pagal aukštį didėjimo tvarka.

Duomenys.txt	Rezultatai.txt
5	
Kartonas 61 25 24 300	Kartonas 61 25 24 300
Medis 51 40 42 560	Popierius 25 35 37 285
Plastikas 45 38 55 240	Medis 51 40 42 560
Popierius 25 35 37 285	Fanera 70 49 53 456
Fanera 70 49 53 456	Plastikas 45 38 55 240

2. Užduotis. Surikiuokite dėžes pagal plotį mažėjimo tvarka.

3. Užduotis. Surikiuokite dėžes pagal tai iš ko pagaminta nuo A iki Z.

b. Rikiavimas pagal kelis požymius:

7. Automobilių ieškojimo varžybos

Tomas ir Benas varžosi, kuris automobilių turguje ras brangiausią ir pigiausią automobilis. 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 vaikino vardas ("Brangiausias: vardas"), antroje eilutėje – pigiausią automobilį radusio vaikino vardas ("Pigiausias: vardas"), trečioje eilutėje – kuris vaikas, 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.

Reikalavimai programai:

1. naudokite struktūras ir masyvus su struktūros tipo elementais;
2. sukurkite pradinių duomenų skaitymo funkciją void C++;
3. sukurkite rezultatų rašymo į failą funkciją void C++;

4. sukurkite funkcijas, rikiuojančias duomenis.

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

4. 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 dešimtainių 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.).

Reikalavimai programai:

1. naudokite struktūras ir masyvus su struktūros tipo elementais;
2. sukurkite pradinių duomenų skaitymo funkciją void C++;

3. sukurkite rezultatų rašymo į failą funkciją void C++;
4. sukurkite funkcijas, rikiuojančias duomenis.
5. vieno žmogaus gimimo metų, mėnesio ir dienos išskyrimo iš asmens kodo funkcija void().
6. mėnesių, išreikštų skaičiais nuo 01 iki 12, konvertavimo į žodžius funkcija, gražinanti reikšmę per funkcijos vardą.
7. rikiavimo pagal amžių funkcija, į kurią reikia kreiptis 2 kartus: su vyrų ir moterų struktūrų masyvais.
8. Rezultatų išsaugojimo faile funkcija, į kurią reikia kreiptis 2 kartus: su vyrų ir moterų

struktūrų masyvais. 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

5. Skaniausi saldainiai.

Saulius yra saldainių ekspertas. Jis ragauja įvairių gamintojų produkciją ir įvertina saldainių skonį 5 balų skalėje (1 – neskanus; 2 – iš bėdos galima valgyti; 3 – galima valgyti, bet yra ir skanesnių; 4 – skanus; 5 – labai skanus).

Parenkite programą, kuri Sauliui pateiktų kiekvieno gamintojo saldainių sąrašą skanumo didėjimo tvarka.

- Pradinių duomenų faile Duomenys.txt įrašyta:
- pirmoje eilutėje – saldainių gamintojų skaičius n;
- toliau surašyti duomenys apie kiekvieną gamintoją:
- vienoje eilutėje įrašytas gamintojo pavadinimas (skiriama 25 pozicijos) ir saldainių rūšių skaičius r;
- tolesnėse r eilučių įrašytas saldainių rūšies pavadinimas (skiriama po 20 pozicijų) ir sveikasis skaičius nuo 1 iki 5, nurodantis saldainio skanumą.

Rezultatų faile Rezultatai.txt turi būti įrašyta:

- kiekvieno gamintojo pavadinimas;
- to gamintojo saldainiai, išrikiuoti skanumo didėjimo tvarka. Jei yra kelios to paties skanumo saldainių rūšys, tai jos gali būti išdėstytos bet kuria tvarka.

Reikalavimai programai:

1. naudokite įrašus (struktūras) ir masyvus su įrašo (struktūros) tipo elementais;
2. sukurkite pradinių duomenų skaitymo procedūrą (funkciją void C++);

3. sukurkite rezultatų rašymo į failą procedūrą (funkciją void C++);
4. sukurkite funkcijas, rikiuojančias saldinius skanumo didėjimo tvarka.

c) Aukštesniųjų gebėjimų mokiniams

6. 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 šeimnininkų 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: šeimnininkų pavardė, katino vardas ir rezultatas (minutės ir sekundės).

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

Reikalavimai programai:

1. naudokite struktūras ir masyvus su struktūros tipo elementais;
2. sukurkite pradinių duomenų skaitymo funkciją void C++;
3. sukurkite rezultatų rašymo į failą funkciją void C++;
4. sukurkite funkcijas, rikiuojančias duomenis.
5. gali būti papildomų funkcijų.

7. 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 veiksmai (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.

Reikalavimai programai:

1. naudokite struktūras ir masyvus su struktūros tipo elementais;
2. sukurkite pradinių duomenų skaitymo funkciją void C++;
3. sukurkite rezultatų rašymo į failą funkciją void C++;
4. sukurkite funkcijas, rikiuojančias duomenis.

<code>piestukai.txt</code>	<code>piestukairez.txt</code>
3	
Melynas PBLDBPLPB	Zalias 140
Raudonas	Melynas 81
DBDBDBDLDBDBDBL	Raudonas 62
Zalias P	

Su mokiniais apibendrinamas atliktas darbas.

7. Refleksija. Mokiniai įvertina sėkmes ir nesėkmes aptariant žodžiu ar užpildant mokytojo parengtą apklausą (galima naudoti Google forms, Mentimeter, apklausa.lt ir pan.).
8. Jei mokiniai sprendė uždavinius savarankiškai, galima jų paprašyti įvardinti sunkiausias vietas.
9. Dalį refleksijos galima perkelti į kitą pamoką, jei mokiniai nebaigė išspręsti uždavinių klasėje.
10. Mokiniai gali įvardinti kuo galėjo padėti klasės draugams.