KAUNO TECHNOLOGIJOS UNIVERSITETAS

Programavimo įvadas (P175B117)

Laboratorinio darbo ataskaita

Atliko:

IF 1/9gr. studentas

Tautvydas Petkus

2011 m. gruodžio 21 d.

Priėmė:

Doc. Vacius Jusas

KAUNAS 2011

TURINYS

1. Pirma klasė (L2) 3

1.1. Darbo užduotis 3

1.2. Pirma užduotis 3

1.2.1 Programos tekstas 3

1.2.2 Pradiniai duomenys ir rezultatai 4

1.3. Antra užduotis 5

1.3.1 Programos tekstas 5

1.3.2 Pradiniai duomenys ir rezultatai 7

1.4. Trečia užduotis 8

1.4.1 Programos tekstas 8

1.4.2 Pradiniai duomenys ir rezultatai 11

2. Objektų rinkinys (L3) 12

2.1. Darbo užduotis 12

2.2. Programos tekstas 12

2.3. Pradiniai duomenys ir rezultatai 14

3. Susiję objektų rinkiniai (L4) 15

3.1. Darbo užduotis 15

3.2. Programos tekstas 15

3.3. Pradiniai duomenys ir rezultatai 18

4. Paveldėjimas (L5) 19

4.1. Darbo užduotis 19

4.2. Programos tekstas 19

4.3. Pradiniai duomenys ir rezultatai 23

5. Operacijos su objektais (L6) 26

5.1. Darbo užduotis 26

5.2. Programos tekstas 26

5.3. Pradiniai duomenys ir rezultatai 32

6. Teksto analizė ir redagavimas (L7) 40

6.1. Darbo užduotis 40

6.2. Programos tekstas 40

6.3. Pradiniai duomenys ir rezultatai 42

# Pirma klasė (L2)

## Darbo užduotis

## Pirma užduotis

### Programos tekstas

#include<iostream>

usingnamespace std;

#include<string>

#include<fstream>

//--------------------------------------------------------------------

class Laikrastis

{

private:

int kaina;

string pav;

int kiekis;

public:

Laikrastis () {}

~Laikrastis () {}

Laikrastis(string pavReiksme, int kainaReiksme, int kiekisReiksme);

string ImtiPav();

int ImtiKaina();

int ImtiKieki();

};

//--------------------------------------------------------------------

class Pirkejas

{

private:

string pavarde, vardas, pavadinimas;

int pinigai;

public:

Pirkejas () {}

~Pirkejas () {}

Pirkejas(string pavardeReiksme, string vardasReiksme, string pavadinimasReiksme, int pinigaiReiksme);

string ImtiPavarde();

string ImtiVarda();

string ImtiPavadinima();

int ImtiPinigus();

};

void Nuskaitymas(int& pavad, int& LT, int& kiekis);

//====================================================================

int main()

{

int k1, k2;

string p1, p2, p3;

duom >> p1, k1, k2;

k = 0;

if (L1.ImtiKaina() != L2.ImtiKaina())

k++;

if (L1.ImtiKaina() != L3.ImtiKaina())

k++;

if (L2.ImtiKaina() != L3.ImtiKaina())

k++;

ofstream rez("Rezultatai.txt");

rez <<"Yra "<< k <<" skirtingos kainos"<< endl;

rez.close();

return 0;

}

//================================================================

Laikrastis::Laikrastis(string pavReiksme, int kainaReiksme, int kiekisReiksme) : pav(pavReiksme), kaina(kainaReiksme), kiekis(kiekisReiksme)

{}

//------------------------------------------------------------

string Laikrastis::ImtiPav()

{

return pav;

}

//------------------------------------------------------------------

int Laikrastis::ImtiKaina()

{

return kaina;

}

//------------------------------------------------------------------

int Laikrastis::ImtiKieki()

{

return kiekis;

}

//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Pirkejas::Pirkejas(string pavardeReiksme, string vardasReiksme, string PavadinimasReiksme, int pinigaiReiksme) :

pavarde(pavardeReiksme), vardas(vardasReiksme), pavadinimas(PavadinimasReiksme), pinigai(pinigaiReiksme)

{}

//---------------------------------------------------------------------------

string Pirkejas::ImtiPavarde()

{

return pavarde;

}

//---------------------------------------------------------------------------

string Pirkejas::ImtiVarda()

{

return vardas;

}

//---------------------------------------------------------------------------

string Pirkejas::ImtiPavadinima()

{

return pavadinimas;

}

//---------------------------------------------------------------------------

string Pirkejas::ImtiKieki()

{

return kiekis;

}

//---------------------------------------------------------------------------

void NuskaitymasL(string & pavad, int& LT, int& kiekybe)

{

ifstream duom("Duomenys.txt");

duom >> pavad >> LT >> kiekybe;

duom.close()

}

### Pradiniai duomenys ir rezultatai

Duomenys.txt:  
  
Laikrastis1 58

Laikrastis2 12

Laikrastis3 58

Rezultatai.txt:

Yra 2 skirtingos kainos

## Antra užduotis

### Programos tekstas

#include<iostream>

usingnamespace std;

#include<string>

#include<fstream>

//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Laikrascio klase~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

class Laikrastis

{

private:

double kaina;

string pav;

int kiekis;

public:

Laikrastis () {}

~Laikrastis () {}

Laikrastis(string pavReiksme, double kainaReiksme, int kiekisReiksme);

string ImtiPav();

double ImtiKaina();

int ImtiKieki();

void DetiPav(string a);

void DetiKaina(double a);

void DetiKieki(int a);

};

//------------------------------Pirkeju klase--------------------------------------

class Pirkejas

{

private:

string pavarde, vardas, pavadinimas;

double pinigai;

public:

Pirkejas () {}

~Pirkejas () {}

Pirkejas(string pavardeReiksme, string vardasReiksme, string pavadinimasReiksme, double pinigaiReiksme);

string ImtiPavarde();

string ImtiVarda();

string ImtiPavadinima();

double ImtiPinigus();

void DetiPavarde(string a);

void DetiVarda(string a);

void DetiPavadinima(string a);

void DetiPinigus(double a);

};

//-----------------------------------------------------------------------

void Nuskaitymas(Laikrastis L[], Pirkejas P[]);

void Skaiciavimai(Laikrastis L[], Pirkejas P[]);

//============================Pagrindine funkcija===========================================

int main()

{

Laikrastis L[3];

Pirkejas P[2];

Nuskaitymas(L, P);

Skaiciavimai(L, P);

return 0;

}

//=======================================================================

Laikrastis::Laikrastis(string pavReiksme, double kainaReiksme, int kiekisReiksme) : pav(pavReiksme), kaina(kainaReiksme), kiekis(kiekisReiksme)

{}

//------------------------------------------------------------

string Laikrastis::ImtiPav()

{

return pav;

}

//------------------------------------------------------------------

double Laikrastis::ImtiKaina()

{

return kaina;

}

//------------------------------------------------------------------

int Laikrastis::ImtiKieki()

{

return kiekis;

}

//---------------------------------------------------------------------

void Laikrastis::DetiKieki(int a)

{

kiekis = a;

}

//---------------------------------------------------------------------

void Laikrastis::DetiPav(string a)

{

pav = a;

}

//---------------------------------------------------------------------

void Laikrastis::DetiKaina(double a)

{

kaina = a;

}

//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Pirkejas::Pirkejas(string pavardeReiksme, string vardasReiksme, string pavadinimasReiksme, double pinigaiReiksme) :

pavarde(pavardeReiksme), vardas(vardasReiksme), pavadinimas(pavadinimasReiksme), pinigai(pinigaiReiksme)

{}

//---------------------------------------------------------------------------

string Pirkejas::ImtiPavarde()

{

return pavarde;

}

//---------------------------------------------------------------------------

string Pirkejas::ImtiVarda()

{

return vardas;

}

//---------------------------------------------------------------------------

string Pirkejas::ImtiPavadinima()

{

return pavadinimas;

}

//---------------------------------------------------------------------------

double Pirkejas::ImtiPinigus()

{

return pinigai;

}

//----------------------------------------------------------------------------

void Pirkejas::DetiPinigus(double a)

{

pinigai = a;

}

//----------------------------------------------------------------------------

void Pirkejas::DetiPavarde(string a)

{

pavarde = a;

}

//----------------------------------------------------------------------------

void Pirkejas::DetiVarda(string a)

{

vardas = a;

}

//----------------------------------------------------------------------------

void Pirkejas::DetiPavadinima(string a)

{

pavadinimas = a;

}

//---------------------------------------------------------------------------

void Nuskaitymas(Laikrastis L[], Pirkejas P[])

{

ifstream duom("Duomenys.txt");

string Lpav, Ppavard, Pvard, Plaikrast;

int Lkiekis= 0;

double Lkaina =0.0, Ppinigai= 0.0;

for (int i = 0; i <= 2; i++){

duom >> Lpav >> Lkaina >> Lkiekis;

L[i].DetiPav(Lpav);

L[i].DetiKaina(Lkaina);

L[i].DetiKieki(Lkiekis);

}

for (int i = 0; i <= 1; i++){

duom >> Ppavard >> Pvard >> Plaikrast >> Ppinigai;

P[i].DetiPavarde(Ppavard);

P[i].DetiVarda(Pvard);

P[i].DetiPavadinima(Plaikrast);

P[i].DetiPinigus(Ppinigai);

}

duom.close();

}

//=============================================================================

void Skaiciavimai(Laikrastis L[], Pirkejas P[])

{

ofstream rez("Rezultatai.txt");

for (int i = 0; i <= 1; i++)

for (int j = 0; j <= 2; j++)

if (P[i].ImtiPavadinima() == L[j].ImtiPav()) //Atrenkama, kuris laikrastis bus perkamas pirkejo, susiejamos dvi klases

if (P[i].ImtiPinigus() >= L[j].ImtiKaina())

if (L[j].ImtiKieki() >= 1){

L[j].DetiKieki(L[j].ImtiKieki() - 1);

P[i].DetiPinigus(P[i].ImtiPinigus() - L[j].ImtiKaina());

rez << P[i].ImtiVarda() <<' '<< P[i].ImtiPavarde() <<" nusipirks laikrasti"<< endl;

}

rez.close();

}

//----------------------------------------------------------------------------

### Pradiniai duomenys ir rezultatai

Pradiniai duomenys Duomenys.txt faile:  
  
Respublika 2.40 6

Varpas 8.00 1

Lrytas 1.00 16

Petraitis Petras Varpas 8.00

Jonas Jonaitis Varpas 2.00

Rezultatai Rezultatai.txt faile:

Petras Petraitis nusipirks laikrasti

## Trečia užduotis

### Programos tekstas

#include<iostream>

usingnamespace std;

#include<string>

#include<fstream>

//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

class Laikrastis

{

private:

double kaina;

string pav;

int kiekis;

public:

Laikrastis () {}

~Laikrastis () {}

Laikrastis(string pavReiksme, double kainaReiksme, int kiekisReiksme);

string ImtiPav();

double ImtiKaina();

int ImtiKieki();

void DetiPav(string a);

void DetiKaina(double a);

void DetiKieki(int a);

};

//--------------------------------------------------------------------

class Pirkejas

{

private:

string pavarde, vardas, pavadinimas;

double pinigai;

public:

Pirkejas () {}

~Pirkejas () {}

Pirkejas(string pavardeReiksme, string vardasReiksme, string pavadinimasReiksme, double pinigaiReiksme);

string ImtiPavarde();

string ImtiVarda();

string ImtiPavadinima();

double ImtiPinigus();

void DetiPavarde(string a);

void DetiVarda(string a);

void DetiPavadinima(string a);

void DetiPinigus(double a);

};

//-----------------------------------------------------------------------

void Nuskaitymas(Laikrastis L[], Pirkejas P[]);

void Skaiciavimai(Laikrastis L[], Pirkejas P[]);

bool Skolinimasis(Laikrastis L[], Pirkejas P[], int k, int k2, int bukle[]);

//=======================================================================

int main()

{

Laikrastis L[3];

Pirkejas P[2];

Nuskaitymas(L, P);

Skaiciavimai(L, P);

return 0;

}

//=======================================================================

Laikrastis::Laikrastis(string pavReiksme, double kainaReiksme, int kiekisReiksme) : pav(pavReiksme), kaina(kainaReiksme), kiekis(kiekisReiksme)

{}

//------------------------------------------------------------

string Laikrastis::ImtiPav()

{

return pav;

}

//------------------------------------------------------------------

double Laikrastis::ImtiKaina()

{

return kaina;

}

//------------------------------------------------------------------

int Laikrastis::ImtiKieki()

{

return kiekis;

}

//---------------------------------------------------------------------

void Laikrastis::DetiKieki(int a)

{

kiekis = a;

}

//---------------------------------------------------------------------

void Laikrastis::DetiPav(string a)

{

pav = a;

}

//---------------------------------------------------------------------

void Laikrastis::DetiKaina(double a)

{

kaina = a;

}

//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Pirkejas::Pirkejas(string pavardeReiksme, string vardasReiksme, string pavadinimasReiksme, double pinigaiReiksme) :

pavarde(pavardeReiksme), vardas(vardasReiksme), pavadinimas(pavadinimasReiksme), pinigai(pinigaiReiksme)

{}

//---------------------------------------------------------------------------

string Pirkejas::ImtiPavarde()

{

return pavarde;

}

//---------------------------------------------------------------------------

string Pirkejas::ImtiVarda()

{

return vardas;

}

//---------------------------------------------------------------------------

string Pirkejas::ImtiPavadinima()

{

return pavadinimas;

}

//---------------------------------------------------------------------------

double Pirkejas::ImtiPinigus()

{

return pinigai;

}

//----------------------------------------------------------------------------

void Pirkejas::DetiPinigus(double a)

{

pinigai = a;

}

//----------------------------------------------------------------------------

void Pirkejas::DetiPavarde(string a)

{

pavarde = a;

}

//----------------------------------------------------------------------------

void Pirkejas::DetiVarda(string a)

{

vardas = a;

}

//----------------------------------------------------------------------------

void Pirkejas::DetiPavadinima(string a)

{

pavadinimas = a;

}

//---------------------------------------------------------------------------

void Nuskaitymas(Laikrastis L[], Pirkejas P[])

{

ifstream duom("Duomenys.txt");

string Lpav, Ppavard, Pvard, Plaikrast;

int Lkiekis= 0;

double Lkaina =0.0, Ppinigai= 0.0;

for (int i = 0; i <= 2; i++){

duom >> Lpav >> Lkaina >> Lkiekis;

L[i].DetiPav(Lpav);

L[i].DetiKaina(Lkaina);

L[i].DetiKieki(Lkiekis);

}

for (int i = 0; i <= 1; i++){

duom >> Ppavard >> Pvard >> Plaikrast >> Ppinigai;

P[i].DetiPavarde(Ppavard);

P[i].DetiVarda(Pvard);

P[i].DetiPavadinima(Plaikrast);

P[i].DetiPinigus(Ppinigai);

}

duom.close();

}

//=============================================================================

void Skaiciavimai(Laikrastis L[], Pirkejas P[])

{

ofstream rez("Rezultatai.txt");

int bukle[2]; //Masyvas nurodantis, ar pirkejas JAU nusipirko laikrasti, ar dar ne

bukle[0] = 0;

bukle[1] = 0;

for (int i = 0; i <= 1; i++)

for (int j = 0; j <= 2; j++)

if (P[i].ImtiPavadinima() == L[j].ImtiPav())

if (P[i].ImtiPinigus() >= L[j].ImtiKaina())

if (L[j].ImtiKieki() >= 1){

L[j].DetiKieki(L[j].ImtiKieki() - 1);

P[i].DetiPinigus(P[i].ImtiPinigus() - L[j].ImtiKaina());

bukle[i] = 1;

rez << P[i].ImtiVarda() <<' '<< P[i].ImtiPavarde() <<" nusipirks laikrasti"<< endl;

}

elsecontinue; //Tesiamas toliau ciklas jeigu pirkimas praejo be skolu

else

if (Skolinimasis(L, P, i, j, bukle) == 1){ //Vyksta skolinimasis

bukle[i] = 1;

rez << P[i].ImtiVarda() <<' '<< P[i].ImtiPavarde() <<" nusipirks laikrasti pasiskolinus pinigu"<< endl;

}

rez.close();

}

//----------------------------------------------------------------------------

bool Skolinimasis(Laikrastis L[], Pirkejas P[], int k, int k2, int bukle[])

{

bool u;

for (int r = 0; r <= 2; r++) //Salyga reikalauja, jog pinigai butu skolinami tik is JAU nusipirkusio pirkejo, todel ciklas daromas daugiau kartu nei yra pirkeju

if (r % 2 != k)

if (bukle[r % 2] == 1)

if (P[r % 2].ImtiPinigus() >= (L[k2].ImtiKaina() - P[k].ImtiPinigus())){

P[k].DetiPinigus(0.0);

P[r % 2].DetiPinigus(P[r % 2].ImtiPinigus() - (L[k2].ImtiKaina() - P[k].ImtiPinigus()));

u = 1;

}

return u;

}

### Pradiniai duomenys ir rezultatai

Pradiniai duomenys Duomenys.txt faile:  
  
Respublika 2.40 6

Varpas 8.00 1

Lrytas 1.00 16

Petraitis Petras Lrytas 8.00

Jonas Jonaitis Varpas 2.00

Rezultatai Rezultatai.txt faile:

Petras Petraitis nusipirks laikrasti

Jonaitis Jonas nusipirks laikrasti pasiskolinus pinigu

# Objektų rinkinys (L3)

## Darbo užduotis

„4“ Skaičiuotė. Vaikai žaidžia slėpynių. Kad išsiskaičiuotų, kas eis ieškoti, vaikai sustoja ratu. Vienas iš jų skaičiuoja pradėdamas nuo savęs laikrodžio rodyklės kryptimi. Su kiekvienu skaičiuotės žodžiu pirštu parodo į eilinį vaiką. Tam, kuriam tenka paskutinis skaičiuotės žodis, išeina iš rato. Skaičiuotę tęsia rate toliau stovintis vaikas. Paskutinis likęs ir bus tas, kuris eis ieškoti. Parašykite programą, kuri apskaičiuotų, kokia tvarka vaikai iškrenta iš rato. Vaiko, nuo kurio reikia pradėti skaičiuoti, nustatymui panaudokite atsitiktinių skaičių generatorių.

Duomenys. Pirmojoje tekstinio failo U2.txt eilutėje parašytas vaikų skaičius n (1 <= d <= 20) ir skaičiuotės žodžių skaičius (1 <= m <= 50). Vaiką charakterizuoja pavardė, vardas. Informacija apie vieną vaiką užima vieną eilutę.

Rezultatai. Išspausdinkite vaikų sąrašą jų eilės iš rato iškritimo tvarka

Panagrinėkite variantą įvedimui, jei vardas iš dvejų žodžių.

## Programos tekstas

#include<iostream>

#include<fstream>

#include<string>

#include<stdio.h>

#include<time.h>

#include<stdlib.h>

usingnamespace std;

//--------------------------------------------------------------

const string Cduom = "U2.txt";

constint CmaxVaikas = 20;

constint CmaxSkaiciuote = 50;

//--------------------------------------------------------------

class Vaikas

{

private:

string pavarde;

string vardas;

bool iskritimas; //Nurodys, ar vaikas jau iskrites is skaiciuotes ar ne

public:

Vaikas();

Vaikas(string pavardeReiksme, string vardasReiksme, bool iskritimasReiksme);

~Vaikas();

string ImtiPavarde();

string ImtiVarda();

bool ImtiIskritima();

void DetiIskritima(bool a);

};

//--------------------------------------------------------------

void Nuskaitymas(string failoVardas, Vaikas V[], int& Svaikas, int& Sskaiciuote, int VaikasMax, int SkaiciuoteMax);

void Skaiciavimas(Vaikas V[], int Svaikas, int Sskaiciuote);

//=============================================================================

int main()

{

Vaikas V[CmaxVaikas];

int d = 0, m = 0;

Nuskaitymas(Cduom, V, d, m, CmaxVaikas, CmaxSkaiciuote);

Skaiciavimas(V, d, m);

return 0;

}

//=============================================================================

Vaikas::Vaikas(string pavardeReiksme, string vardasReiksme, bool iskritimasReiksme): pavarde(pavardeReiksme), vardas(vardasReiksme), iskritimas(iskritimasReiksme) { }

Vaikas::Vaikas(): pavarde(), vardas(), iskritimas(0) { }

Vaikas::~Vaikas() { }

//-----------------------------------------------------------------------------

string Vaikas::ImtiPavarde()

{

return pavarde;

}

//-----------------------------------------------------------------------------

string Vaikas::ImtiVarda()

{

return vardas;

}

//-----------------------------------------------------------------------------

bool Vaikas::ImtiIskritima()

{

return iskritimas;

}

//-----------------------------------------------------------------------------

void Vaikas::DetiIskritima(bool a)

{

iskritimas = a;

}

//-----------------------------------------------------------------------------

void Nuskaitymas(string failoVardas, Vaikas V[], int& Svaikas, int& Sskaiciuote, int VaikasMax, int SkaiciuoteMax)

{

string pavard, vard;

ifstream duom(failoVardas.c\_str());

duom >> Svaikas;

if (Svaikas > VaikasMax)

Svaikas = VaikasMax;

duom >> Sskaiciuote;

if (Sskaiciuote > SkaiciuoteMax)

Sskaiciuote = SkaiciuoteMax;

for (int i = 0; i<= Svaikas - 1; i++){

duom >> pavard;

getline (duom, vard);

//vardas nuskaitomas char masyvu ir konvertuojamas i string kintamaji

V[i] = Vaikas(pavard, vard, 0); //Galutinis duomenu priskyrimas

}

duom.close();

}

//----------------------------------------------------------------------------

void Skaiciavimas(Vaikas V[], int Svaikas, int Sskaiciuote)

{

srand(time(NULL));

int s, k;

bool check; //Sis kintamasis bus naudojamas nutraukiant while cikla, kurio metu bus praleidziami iskrite vaikai is rato

s = rand() % Svaikas; //Sugeneruojamas atsitiktinis vaiko, nuo kurio bus pradedama skaiciuoti numeris

int u = s;

for (int i = 1; i <= Svaikas; i++){

for (int k = 1; k <= Sskaiciuote; k++){

check = 0;

while (check != 1){

if (V[u%Svaikas].ImtiIskritima() == 0)

check = 1; //Jei Vaikas neiskrites, vaikas yra iskaitomas, einama prie sekancio skaiciuotes zodzio

else//Vaikas yra iskrites, kartojamas ciklas, einama prie sekancio vaiko, zodis tas pats

u++;

}

u++;

}

u--; //Kadangi paskutini zodi pavartojus, ciklas pereina prie sekancio vaiko, o mums reikia vaiko, kuris buvo ties paskutiniu zodziu

V[u%Svaikas].DetiIskritima(1); //Nurodoma, jog vaikas iskrites

cout << V[u%Svaikas].ImtiPavarde() << V[u%Svaikas].ImtiVarda() << endl;

}

## Pradiniai duomenys ir rezultatai

U2.txt

4 5

AaaaaAaaa

BbbbbBbbbbBbb

CccccCcccc C

DdddddDddd

Rezultatai ekrane: (parinktas atsitiktinis žmogus, nuo kurio pradės skaičiuoti- AaaaaAaaa)

AaaaaAaaa

CccccCcccc C

DdddddDddd

BbbbbBbbbbBbb

# Susiję objektų rinkiniai (L4)

## Darbo užduotis

12. *\*\*Spalvoti taškai.* Stačiakampėje koordinačių plokštumoje yra n (1 <= n <= 100) m (1 <= m <= 256) spalvų taškų. Jungiant taškus po tris, galima sudaryti daug trikampių. Spalvos, kurioms galima bandyti sudaryti trikampius, nurodytos kitame faile. Sudarykite programą, kuri tarp kiekvienos spalvos taškų, jeigu tai įmanoma, surastų tris taškus, kuriuos sujungus tarpusavyje būtų gauti didžiausi lygiašoniai trikampiai. Apskaičiuokite gautų trikampių perimetrus.

Duomenys. Tekstiniame faile U2a.txt yra duomenys apie taškus. Pirmojoje tekstinio failo U2a.txt eilutėje parašytas skaičius n. Toliau eilutėmis surašytos n taškų spalvos ir koordinatės (x; y – sveikieji skaičiai) (-100 <= x, y <= 100). Spalvos pavadinimas gali būti iš dvejų žodžių.

Tekstiniame faile U2b.txt yra spalvų ir nurodymų sąrašas. Spalvos išvardintos stulpeliu, šalia spalvos pavadinimo žodelis „taip“ reiškia, kad galima, o žodelis „ne“ – negalima.

Rezultatai. Išspausdinkite spalvos pavadinimą, koordinates, perimetrą arba „nėra“, arba „negalima“

## Programos tekstas

Taskai.h:

#include<iostream>

usingnamespace std;

struct koordinates{

int x;

int y;

};

class Taskai

{

private:

string spalva;

int kiekis;

koordinates k[256];

public:

Taskai();

Taskai(string spalvaReiksme, int kiekisReiksme);

~Taskai();

string ImtiSpalva() {return spalva; }

void DetiSpalva(string a) {spalva = a; }

int ImtiKieki() {return kiekis; }

void DetiKieki(int a) {kiekis = a;}

int ImtiXk(int a) {return k[a].x; }

void DetiXk(int a, int u) {k[a].x = u; }

int ImtiYk(int a) {return k[a].y; }

void DetiYk( int a, int u) {k[a].y = u; }

};

TaskaiB.h:

#include<iostream>

usingnamespace std;

class TaskaiB

{

private:

string spalva;

bool dalyvis;

public:

TaskaiB();

TaskaiB(string spalvaReiksme, bool dalyvisReiksme);

~TaskaiB();

string ImtiSpalva() {return spalva; }

void DetiSpalva(string a) {spalva = a; }

bool ImtiDalyvi() {return dalyvis; }

void DetiDalyvi(bool a) {dalyvis = a; }

};

Main.cpp:

#include<iostream>

#include<fstream>

#include<string>

#include<iostream>

#include<cmath>

#include"Taskai.h"

#include"TaskaiB.h"

usingnamespace std;

//-----------------------------------------------------------------

const string Cduom1 = "U2a.txt";

const string Cduom2 = "U2b.txt";

constint Cspalva = 256;

constint Ctaskai = 100;

//---------------------------------------------------------------

void NuskaitymasB(string failoVardas, TaskaiB Tb[], int& Sspalvos, int SpalvaMax, Taskai T[]);

void NuskaitymasA(string failoVardas, Taskai T[], int Sspalvos, int TaskaiMax);

void Isvestis(Taskai T[], int Sspalvos, TaskaiB Tb[]);

void Trikampis(Taskai T[], int k);

//==============================================================

int main()

{

Taskai T[Cspalva];

TaskaiB Tb[Cspalva];

int n;

NuskaitymasB(Cduom2, Tb, n, Cspalva, T);

NuskaitymasA(Cduom1, T, n, Ctaskai);

Isvestis (T, n, Tb);

return 0;

}

//=============================================================

Taskai::Taskai(string spalvaReiksme, int kiekisReiksme): spalva(spalvaReiksme), kiekis(kiekisReiksme) { }

Taskai::Taskai(): spalva(""), kiekis(0) { }

Taskai::~Taskai() { }

//-------------------------------------------------------------

TaskaiB::TaskaiB(string spalvaReiksme, bool dalyvisReiksme): spalva(spalvaReiksme), dalyvis(dalyvisReiksme){ }

TaskaiB::TaskaiB(): spalva(""), dalyvis(0) { }

TaskaiB::~TaskaiB() { }

//-------------------------------------------------------------

void NuskaitymasB(string failoVardas, TaskaiB Tb[], int& Sspalvos, int SpalvaMax, Taskai T[])

{

ifstream duom(failoVardas.c\_str());

string spalv;

string dalyv;

int i = 0;

while (!duom.eof() && i <= SpalvaMax - 1) //U2b.txt nuskaitymas

{

getline (duom, spalv, ',');

T[i].DetiSpalva(spalv);

Tb[i].DetiSpalva(spalv);

duom >> dalyv;

if (dalyv == "taip")

Tb[i].DetiDalyvi(1);

else

if (dalyv == "ne")

Tb[i].DetiDalyvi(0);

T[i].DetiKieki(-1);

i++;

}

i--;

duom.close();

Sspalvos = i;

}

//-------------------------------------------------------------

void NuskaitymasA(string failoVardas, Taskai T[], int Sspalvos, int TaskaiMax)

{

ifstream duom(failoVardas.c\_str());

int x1, y1, m;

string spalv;

duom >> m;

if (m > TaskaiMax)

m = TaskaiMax;

for (int i = 1 ; i <= m; i++)

{

getline (duom, spalv, ',');

duom >> x1;

duom >> y1;

for (int j = 0 ; j <= Sspalvos; j++)

if (T[j].ImtiSpalva() == spalv){

T[j].DetiKieki(T[j].ImtiKieki()+1);

T[j].DetiXk(T[j].ImtiKieki(), x1);

T[j].DetiYk(T[j].ImtiKieki(), y1);

}

}

duom.close();

}

//-----------------------------------------------------------------

void Isvestis(Taskai T[], int Sspalvos, TaskaiB Tb[])

{

for (int i = 0; i <= Sspalvos; i++)

if (Tb[i].ImtiDalyvi() == 0)

cout << Tb[i].ImtiSpalva() <<" negalima"<< endl;

else

if (T[i].ImtiKieki() <= 1) //Jeigu yra tik 2 arba maziau tasku, trikampio neimanoma sudaryti (kiekis skaiciuojamas nuo 0)

cout << T[i].ImtiSpalva() <<" Nera"<< endl;

else

Trikampis(T, i);

}

//------------------------------------------------------------------

void Trikampis(Taskai T[], int k)

{

bool Egz = 0; //Sis kintamasis nurodys, ar egzistuoja bent vienas lygiasonis trikampis

double P, Pmax = 0;

double r1, r2, r3;

int taskas1, taskas2, taskas3;

for (int j1 = 0; j1 <= T[k].ImtiKieki() - 2; j1++)

for (int j2 = j1 + 1; j2 <= T[k].ImtiKieki() - 1; j2++)

for (int j3 = j2 + 1; j3 <= T[k].ImtiKieki(); j3++){

P = 0; //Apskaiciuojamos kiekvienos trikampio krastines

r1 = pow((T[k].ImtiXk(j1) - T[k].ImtiXk(j2))\*(T[k].ImtiXk(j1) - T[k].ImtiXk(j2)) + (T[k].ImtiYk(j1) - T[k].ImtiYk(j2)) \* (T[k].ImtiYk(j1) - T[k].ImtiYk(j2)), 0.5);

r2 = pow((T[k].ImtiXk(j2) - T[k].ImtiXk(j3))\*(T[k].ImtiXk(j2) - T[k].ImtiXk(j3)) + (T[k].ImtiYk(j2) - T[k].ImtiYk(j3)) \* (T[k].ImtiYk(j2) - T[k].ImtiYk(j3)), 0.5);

r3 = pow((T[k].ImtiXk(j3) - T[k].ImtiXk(j1))\*(T[k].ImtiXk(j3) - T[k].ImtiXk(j1)) + (T[k].ImtiYk(j3) - T[k].ImtiYk(j1)) \* (T[k].ImtiYk(j3) - T[k].ImtiYk(j1)), 0.5);

if (r1 == r2 || r2 == r3 || r3 == r1){

Egz = 1;

P = r1 + r2 + r3;

if (P > Pmax){

Pmax = P;

taskas1 = j1;

taskas2 = j2;

taskas3 = j3;

}

}

}

if (Egz == 1){

cout << T[k].ImtiSpalva() << endl;

cout << T[k].ImtiXk(taskas1) <<' '<< T[k].ImtiYk(taskas1) << endl;

cout << T[k].ImtiXk(taskas2) <<' '<< T[k].ImtiYk(taskas2) << endl;

cout << T[k].ImtiXk(taskas3) <<' '<< T[k].ImtiYk(taskas3) << endl;

cout << Pmax << endl;

}

else cout << T[k].ImtiSpalva() <<" Nera"<< endl;

}

## Pradiniai duomenys ir rezultatai

Pradiniai duomenys faile *U2a.txt* :

12

Raudona, -3 -2

Zalia, -6 2

Raudona, -5 10

Zalia, 0 10

Sviesi melyna, 0 12

Sviesi melyna, 1 8

Raudona, 2 3

Zalia, 4 6

Zalia, 4 11

Raudona, 2 -3

Raudona, 6 0

Zalia, 6 2

Pradiniai duomenys faile *U2b.txt* :

Raudona, taip

Sviesi melyna, taip

Zalia, ne

Ruda, taip

Rezultatai ekrane:

Raudona

2 3

2 -3

6 0

16

Sviesi melyna Nera

Zalia negalima

Ruda Nera

# Paveldėjimas (L5)

## Darbo užduotis

A2f34. \*\*Duota pirmame tekstiniame faile tokia informacija apie leidinius: leidinio kodas, leidinio pavadinimas, vieno mėnesio leidinio kaina. Kitame faile yra tokia informacija: prenumeratoriaus pavardė, adresas, laikotarpio pradžia(sveikasis skaičius 1..12), laikotarpio ilgis, leidinio kodas, leidinių kiekis. Užsakymas vyksta metų ribose. Nustatyti kiekvienam mėnesiui, kurio leidinio pajamos didžiausios. Sudaryti sąrašą leidinių, kurių pajamos mažesnės už vidutines. Rikiuoti(kaina, pavadinimas).

## Programos tekstas

LeidinysB.h:

#pragmaonce

#include<string>

usingnamespace std;

class Leidinys

{

private:

string kodas, pavadinimas;

double kaina;

public:

Leidinys() : kodas(""), pavadinimas(""), kaina(0.0) { }

Leidinys(string kodasR, string pavadinimasR, double kainaR): kodas(kodasR), pavadinimas(pavadinimasR), kaina(kainaR) {}

~Leidinys() {}

void DetiKoda(string a) { kodas = a; }

void DetiPavadinima(string a) { pavadinimas = a; }

void DetiKaina(double a) { kaina = a; }

string ImtiKoda() { return kodas; }

string ImtiPavadinima() { return pavadinimas; }

double ImtiKaina() { return kaina; }

};

Prenumeratoriai.h:

#pragmaonce

#include"leidinysB.h"

#include<string>

usingnamespace std;

class Prenumeratoriai :

public Leidinys

{

private:

string pavarde, adresas, Pkodas;

int pradzia, ilgis, kiekis;

public:

Prenumeratoriai() : Leidinys(), pavarde(""), adresas(""), pradzia(0), ilgis(0), kiekis(0){}

Prenumeratoriai(string pavardeR, string adresasR, int pradziaR, int ilgisR, int kiekisR)

: Leidinys(), pavarde(pavardeR), adresas(adresasR), pradzia(pradziaR), ilgis(ilgisR), kiekis(kiekisR) {}

~Prenumeratoriai() {}

void DetiPavarde(string b) {pavarde = b; }

void DetiAdresa(string b) {adresas = b; }

void DetiPradzia(int b) {pradzia = b; }

void DetiIlgi(int b) {ilgis = b; }

void DetiKieki(int b) {kiekis = b; }

string ImtiPavarde() {return pavarde; }

string ImtiAdresa() {return adresas; }

int ImtiPradzia() {return pradzia; }

int ImtiIlgi() {return ilgis; }

int ImtiKieki() {return kiekis; }

};

PajamosB.h:

#pragmaonce

#include<string>

#include"LeidinysB.h"

usingnamespace std;

class Pajamos: public Leidinys

{

private:

double uzdirbis;

double menesinis[12];

public:

Pajamos(double uzdirbisR = 0.0)

: Leidinys(), uzdirbis(uzdirbisR) {}

~Pajamos() {}

void DetiUzdirbi(double c) {uzdirbis = c; }

double ImtiUzdirbi() {return uzdirbis; }

void DetiMenesini(int i, double c){ menesinis[i] = c; }

double ImtiMenesini(int i) { return menesinis[i]; }

};

Main.cpp:

#include<iostream>

#include<fstream>

#include<string>

#include<iomanip>

#include"Prenumeratoriai.h"

#include"LeidinysB.h"

#include"PajamosB.h"

#include<iomanip>

usingnamespace std;

//-------------------------------------------------------------------

const string Cduom1 = "U1.txt";

const string Cduom2 = "U2.txt";

const string Crez = "Rezultatai.txt";

//-------------------------------------------------------------------

//-------------------------------------------------------------------

void Nuskaitymas(Pajamos J[], int& nj, string duomVardas);

void Nuskaitymas(Prenumeratoriai P[], int& np, string duomVardas, int nj, Pajamos J[]);

void PradDuomenys(Pajamos J[], Prenumeratoriai P[], int nj, int np, string rezVardas);

void PajamuApskaiciavimas(Pajamos J[], Prenumeratoriai P[], int nj, int np);

void PajamosMenesiui(Pajamos J[], int nj, string rezVardas);

void PajamosVidutine(Pajamos J[], int nj, string rezVardas);

//-------------------------------------------------------------------

int main()

{

Pajamos J[256];

Prenumeratoriai P[256];

int nj = 0, np = 0;

Nuskaitymas(J, nj, Cduom1);

Nuskaitymas(P, np, Cduom2, nj, J);

PradDuomenys(J, P, nj, np, Crez);

PajamuApskaiciavimas(J, P, nj, np);

PajamosMenesiui(J, nj, Crez);

PajamosVidutine(J, nj, Crez);

return 0;

}

//-----------------------------------------------------------------

void Nuskaitymas(Pajamos J[], int& nj, string duomVardas)

{

ifstream duom(duomVardas.c\_str());

string code, name;

nj = 0;

double price = 0.0;

while (!duom.eof()){

duom >> code;

getline(duom, name, ',');

duom >> price;

J[nj].DetiKoda(code);

J[nj].DetiPavadinima(name);

J[nj].DetiKaina(price);

J[nj].DetiUzdirbi(0.0);

nj++;

duom.ignore();

}

nj--;

}

//------------------------------------------------------------------

void Nuskaitymas(Prenumeratoriai P[], int& np, string duomVardas, int nj, Pajamos J[])

{

ifstream duom(duomVardas.c\_str());

string pav, adr, code;

int pradz, ilg, kiek;

np = 0;

while (!duom.eof()){

duom >> pav; P[np].DetiPavarde(pav);

getline(duom, adr, ','); P[np].DetiAdresa(adr);

duom >> pradz; P[np].DetiPradzia(pradz);

duom >> ilg; P[np].DetiIlgi(ilg);

duom >> code;

duom >> kiek; P[np].DetiKieki(kiek);

duom.ignore();

for(int i = 0; i <= nj; i++)

if (code == J[i].ImtiKoda()){

P[np].DetiPavadinima(J[i].ImtiPavadinima());

P[np].DetiKaina(J[i].ImtiKaina());

P[np].DetiKoda(J[i].ImtiKoda());

}

np++;

}

np--;

duom.close();

}

//------------------------------------------------------------------

void PradDuomenys(Pajamos J[], Prenumeratoriai P[], int nj, int np, string rezVardas)

{

ofstream rez(rezVardas.c\_str());

rez <<"--------------------------Laikrasciai-------------------------"<< endl;

rez << right << setw(21) <<"-----Kodas-----|"<< setw(31) <<"-----Pavadinimas-----|"<< setw(10) <<"-----kaina"<< endl;

for (int i = 0; i <= nj; i++)

rez << right << setw(20) << J[i].ImtiKoda()<<'|'<< setw(30) << J[i].ImtiPavadinima() <<'|'<< setw(10) << J[i].ImtiKaina() << endl;

rez <<"------------------------------------------------------------------------------------------------------------------------------"<< endl;

rez <<"---------------------------------------------Pirkejai-------------------------------------------------------------------------"<< endl;

rez << right << setw(21) <<"-----Pavarde-----|"<< setw(41) <<"-----Adresas-----|"<< setw(11) <<"-Pradzia-|"<<

setw(11) <<"--Ilgis-|"<< setw(21) <<"-----Kodas-----|"<< setw(10) <<"--Kiekis--"<< endl;

for (int i = 0; i <= np; i++)

rez << right << setw(20) << P[i].ImtiPavarde() <<'|'<< setw(40) << P[i].ImtiAdresa() <<'|'<< setw(10) << P[i].ImtiPradzia() <<'|'<<

setw(10) << P[i].ImtiIlgi() <<'|'<< setw(20) << P[i].ImtiKoda() <<'|'<< setw(10) << P[i].ImtiKieki() << endl;

rez <<"------------------------------------------------------------------------------------------------------------------------------"<< endl;

rez.close();

}

//--------------------------------------------------------------------

void PajamuApskaiciavimas(Pajamos J[], Prenumeratoriai P[], int nj, int np)

{

double u = 0.0;

double bendras = 0.0;

for (int i = 0; i <= nj; i++)

for (int j = 0; j <= 11; j++)

J[i].DetiMenesini(j, 0.0);

for (int i = 0; i <= np; i++)

for (int j = 0; j <= nj; j++)

if (P[i].ImtiKoda() == J[j].ImtiKoda()){

u = P[i].ImtiKieki() \* P[i].ImtiKaina();

for (int k = P[i].ImtiPradzia() - 1; k <= P[i].ImtiPradzia() + P[i].ImtiIlgi() - 1; k++){

J[j].DetiMenesini((k % 12), J[j].ImtiMenesini(k % 12) + u);

bendras += u;

}

J[j].DetiUzdirbi(J[j].ImtiUzdirbi() + bendras);

bendras = 0.0;

}

}

//---------------------------------------------------------------------

void PajamosMenesiui(Pajamos J[], int nj, string rezVardas)

{

string M[12]; M[0] = "Sausis"; M[1] = "Vasaris"; M[2] = "Kovas"; M[3] = "Balandis";

M[5] = "Birželis"; M[6] = "Liepa"; M[7] = "Rugpjūtis"; M[8] = "Rugsėjis";

M[9] = "Spalis"; M[10] = "Lapkritis"; M[11] = "Gruodis";

int Max = 0;

ofstream rez(rezVardas.c\_str(), ios::app);

rez << endl;

rez <<"------------------------------Menesio laikrasciai, gave daugiausiai pajamu ------------------------- "<<endl;

rez <<"---------------------------------------------------------------------------------------------------- "<<endl;

for (int i = 0; i <= 11; i++){

rez << right << setw(20) << M[i] <<"|";

for (int j = 0; j <= nj; j++)

if (J[j].ImtiMenesini(i) > J[Max].ImtiMenesini(i))

Max = j;

rez << right << setw(30) << J[Max].ImtiPavadinima() <<'|'<< setw(10) << J[Max].ImtiMenesini(i)<< endl;

}

rez.close();

}

//-----------------------------------------------------------------------

void PajamosVidutine(Pajamos J[], int nj, string rezVardas)

{

ofstream rez(rezVardas.c\_str(), ios::app);

double vid = 0.0;

int Nk = -1;

Pajamos U[256];

for (int i = 0; i <= nj; i++)

vid += J[i].ImtiUzdirbi();

vid = (vid / (nj + 1));

for (int i = 0; i <= nj; i++)

if (J[i].ImtiUzdirbi() < vid){

Nk++;

U[Nk] = J[i];

}

for (int i = 0; i <= Nk - 1; i++)

for (int j = i + 1; j <= Nk; j++)

if (U[i].ImtiUzdirbi() < U[j].ImtiUzdirbi()){

Pajamos C;

C = U[i];

U[i] = U[j];

U[j] = C;

}

else

if (U[i].ImtiUzdirbi() == U[j].ImtiUzdirbi())

if (U[i].ImtiPavadinima() < U[j].ImtiPavadinima()){

Pajamos C;

C = U[i];

U[i] = U[j];

U[j] = C;

}

rez << endl;

rez <<"---------------------------------------------------------------------------------------------------- "<<endl;

rez <<"------------------Laikraščiai, kurių pajamos mažesnės už vidutines---------------------------------- "<<endl;

rez <<"----------------------------vidurkis : "<< vid <<" Lt------------------------------------------------------------------------ "<<endl;

rez << right << setw(21) <<"-----Kodas-----|"<< setw(31) <<"-----Pavadinimas-----|"<< setw(11) <<"-----kaina|"<< setw(16) <<"---------pajamos"<< endl;

for (int i = 0; i <= Nk; i++)

rez << right << setw(20) << U[i].ImtiKoda()<<'|'<< setw(30) << U[i].ImtiPavadinima() <<'|'<< setw(10) << U[i].ImtiKaina() <<'|'<< setw(15) << U[i].ImtiUzdirbi() << endl;

rez.close();

}

## Pradiniai duomenys ir rezultatai

U1.txt:

123456789 Gyvenimo budas, 2.60

165413484 Lietuvos Rytas, 1.80

156841608 Respublika, 2.00

841561856 Vakaro Zinios, 0.50

984161986 Valstybe, 6.00

684168691 Iliustruotasis mokslas, 10.00

919651984 Ukininko patarejas, 1.40

988415961 Stilius, 6.00

951651681 PCgamer, 12.00

186168466 Akistata, 2.20

191655168 Rytai-Vakarai, 1.00

895589622 E!, 7.80

U2.txt:

Aleksandravicius Pusu g. 81, 6 11 186168466 2

Petraitis Rytu g. 2, 2 6 895589622 7

Irmantaitis Kalno g. 73-18, 1 11 165413484 4

Sakalauskas Taikos pr. 51, 11 7 123456789 6

Kaseta Vilties g. 132, 8 5 919651984 1

Norkeliunas Urmano g. 18, 12 10 684168691 1

Ivanauskas Pilies g. 3, 7 5 841561856 12

Poska Saules g. 18-13, 7 6 951651681 2

Kryzmantas Upes g. 2, 12 7 191655168 7

Zukas Vilnies g. 74, 9 2 156841608 20

Sevilis Ugnies g. 1, 7 12 123456789 1

Mikailionis Vytauto g. 6, 4 7 895589622 1

Zmitrulevicius Sauklio g. 12, 8 11 841561856 3

Smutginas Zalgirio g. 3, 1 12 165413484 2

Valiukevicius Jotvingiu g. 2, 10 6 919651984 5

Jonaitis Sauletekio g. 12, 8 9 123456789 1

Vanagas Balos g. 7, 3 5 919651984 3

Lukasevicius Ezero g. 15, 2 4 988415961 2

Obolovskis Krasto g. 81, 12 12 156841608 1

Saulusevicius Misko g. 14-18, 1 12 895589622 1

Navickas Studentu g. 2, 6 7 951651681 1

Lukosius Gatves g. 15, 2 9 919651984 6

Slivinskas Slyvu g. 6, 9 4 123456789 3

Rezultatai.txt(šiek tiek paredaguota, kad tilptų į vieną eilę):

--------------------------Laikrasciai-------------------------

-----Kodas-----| -----Pavadinimas-----|-----kaina

123456789| Gyvenimo budas| 2.6

165413484| Lietuvos Rytas| 1.8

156841608| Respublika| 2

841561856| Vakaro Zinios| 0.5

984161986| Valstybe| 6

684168691| Iliustruotasis mokslas| 10

919651984| Ukininko patarejas| 1.4

988415961| Stilius| 6

951651681| PCgamer| 12

186168466| Akistata| 2.2

191655168| Rytai-Vakarai| 1

895589622| E!| 7.8

----------------------------------------------------------------------------------

---------------- --------Pirkejai-----------------------------

-----Pavarde-----| -----Adresas-----| -Pradzia-| --Ilgis-|---Kodas----|Kiekis--

Aleksandravicius| Pusu g. 81| 6| 11| 186168466| 2

Petraitis| Rytu g. 2| 2| 6|895589622| 7

Irmantaitis| Kalno g. 73-18| 1| 11|165413484| 4

Sakalauskas| Taikos pr. 51| 11| 7| 123456789| 6

Kaseta| Vilties g. 132| 8| 5|919651984| 1

Norkeliunas| Urmano g. 18| 12| 10| 684168691| 1

Ivanauskas| Pilies g. 3| 7| 5| 841561856| 12

Poska| Saules g. 18-13| 7| 6|951651681| 2

Kryzmantas| Upes g. 2| 12| 7| 191655168| 7

Zukas| Vilnies g. 74| 9| 2|156841608| 20

Sevilis| Ugnies g. 1| 7| 12| 123456789| 1

Mikailionis| Vytauto g. 6| 4| 7| 895589622| 1

Zmitrulevicius| Sauklio g. 12| 8| 11| 841561856| 3

Smutginas| Zalgirio g. 3| 1| 12| 165413484| 2

Valiukevicius| Jotvingiu g. 2| 10| 6| 919651984| 5

Jonaitis| Sauletekio g. 12| 8| 9| 123456789| 1

Vanagas| Balos g. 7| 3| 5| 919651984| 3

Lukasevicius| Ezero g. 15| 2| 4| 988415961| 2

Obolovskis| Krasto g. 81| 12| 12| 156841608| 1

Saulusevicius| Misko g. 14-18| 1| 12| 895589622| 1

Navickas| Studentu g. 2| 6| 7| 951651681| 1

Lukosius| Gatves g. 15| 2| 9| 919651984| 6

Slivinskas| Slyvu g. 6| 9| 4| 123456789| 3

---------------------------------------------------------------------------------

------------------------------Menesio laikrasciai, gave daugiausiai pajamu -------

----------------------------------------------------------------------------------

Sausis| PCgamer| 36

Vasaris| E!| 62.4

Kovas| E!| 62.4

Balandis| E!| 70.2

| E!| 70.2

Birželis| E!| 70.2

Liepa| E!| 70.2

Rugpjūtis| E!| 70.2

Rugsėjis| Respublika| 42

Spalis| Respublika| 42

Lapkritis| Respublika| 42

Gruodis| PCgamer| 36

----------------------------------------------------------------------------------

------------------Laikraščiai, kurių pajamos mažesnės už vidutines----------------

----------------------------vidurkis : 151.017 Lt---------------------------------

-----Kodas-----| -----Pavadinimas-----|-----kaina|---------pajamos

156841608| Respublika| 2| 146

165413484| Lietuvos Rytas| 1.8| 133.2

684168691| Iliustruotasis mokslas| 10| 110

988415961| Stilius| 6| 60

191655168| Rytai-Vakarai| 1| 56

841561856| Vakaro Zinios| 0.5| 54

186168466| Akistata| 2.2| 52.8

984161986| Valstybe| 6| 0

# Operacijos su objektais (L6)

## Darbo užduotis

A2f32. Duota tekstiniame faile tokia informacija apie leidinius: prenumeratoriaus pavardė, adresas, laikotarpio pradžia(sveikasis skaičius 1..12), laikotarpio ilgis, leidinio kodas, leidinių kiekis, agento kodas. Kitame tekstiniame faile yra informacija apie agentus: agento kodas, pavardė, vardas, adresas, telefonas. Suskaičiuoti kiekvienam agentui nurodyto mėnesio krūvį. Suformuoti sąrašą agentų, kurie dirba mažiau nei vidutinis krūvis nurodytam mėnesiui. Atspausdinti kiekvienam agentui nurodyto mėnesio krūvį ir nešiojamos prenumeratos sąrašus. Rikiuoti (adresas, pavardė).

## Programos tekstas

-----------------------------------------------------------------------------

Leidinys.h:

#pragmaonce

#include<string>

usingnamespace std;

class Leidinys

{

private:

string pavarde, adresas, kodas;

int pradzia, ilgis, kiekis;

public:

Leidinys() {}

~Leidinys() {}

string ImtiPavarde();

string ImtiAdresa();

string ImtiKoda();

int ImtiPradzia();

int ImtiIlgi();

int ImtiKieki();

void DetiPavarde(string a);

void DetiAdresa(string a);

void DetiKoda(string a);

void DetiPradzia(int a);

void DetiIlgi(int a);

void DetiKieki(int a);

booloperator< (const Leidinys & R);

};

Leidinys.cpp:

#include"Leidinys.h"

#include<string>

string Leidinys::ImtiPavarde() {return pavarde; }

string Leidinys::ImtiAdresa() {return adresas; }

string Leidinys::ImtiKoda() {return kodas; }

int Leidinys::ImtiPradzia() {return pradzia; }

int Leidinys::ImtiIlgi() {return ilgis; }

int Leidinys::ImtiKieki() {return kiekis; }

void Leidinys::DetiPavarde(string a) {pavarde = a; }

void Leidinys::DetiAdresa(string a) {adresas = a; }

void Leidinys::DetiKoda(string a) {kodas = a; }

void Leidinys::DetiPradzia(int a) {pradzia = a; }

void Leidinys::DetiIlgi(int a) {ilgis = a; }

void Leidinys::DetiKieki(int a) {kiekis = a; }

bool Leidinys::operator<(const Leidinys & R)

{

return (adresas < R.adresas || (adresas == R.adresas && pavarde < R.pavarde));

}

Agentas.h:

#pragmaonce

#include<string>

#include<iostream>

#include"Leidinys.h"

usingnamespace std;

class Agentas

{

private:

string Akodas, Apavarde, Avardas, Aadresas, telefonas;

int menesinis[36];

int menesiuKiekis;

int Akiekis;

Leidinys L[10];

public:

Agentas() {}

~Agentas() {}

//---------------------------

void DetiAkoda(string a);

void DetiApavarde(string b);

void DetiAvarda(string c);

void DetiAadresa(string d);

void DetiTelefona(string e);

void DetiAkieki(int a);

void DetiLeidini(string a, string b, string c, int d, int e, int f);

Leidinys ImtiLeidini(int a);

int ImtiAkieki();

string ImtiAadresa();

string ImtiApavarde();

string ImtiAkoda();

string ImtiAgentoDuom();

string ImtiLeidinioDuom(int a);

string ImtiVardoDuom();

string ImtiMenesioDuom(int a);

void Spausdinimas(string RezFailas);

void KruvioApskaiciavimasMenesiui();

int ImtiMenesioKruvi(int a);

void LeidiniuRikiavimas();

int ImtiMenesiuKieki();

booloperator< (const Agentas & R);

};

Agentas.cpp:

#include<string>

#include<iostream>

#include<sstream>

#include<fstream>

#include<iomanip>

#include"Leidinys.h"

#include"Agentas.h"

//---------------------------

void Agentas::DetiAkoda(string a) {Akodas = a; }

void Agentas::DetiApavarde(string b) {Apavarde = b; }

void Agentas::DetiAvarda(string c) {Avardas = c; }

void Agentas::DetiAadresa(string d) {Aadresas = d; }

void Agentas::DetiTelefona(string e) {telefonas = e; }

void Agentas::DetiAkieki(int a) {Akiekis = a; }

//---------------------------

void Agentas::DetiLeidini(string a, string b, string c, int d, int e, int f)

{

if (Akiekis < 100){

L[Akiekis].DetiPavarde(a);

L[Akiekis].DetiAdresa(b);

L[Akiekis].DetiKoda(c);

L[Akiekis].DetiPradzia(d);

L[Akiekis].DetiIlgi(e);

L[Akiekis].DetiKieki(f);

Akiekis++;

menesiuKiekis = 0;

}

}

//-----------------------------

Leidinys Agentas::ImtiLeidini(int a) {return L[a]; }

int Agentas::ImtiAkieki() {return Akiekis; }

string Agentas::ImtiAadresa() {return Aadresas; }

string Agentas::ImtiApavarde() {return Apavarde; }

string Agentas::ImtiAkoda() {return Akodas; }

int Agentas::ImtiMenesiuKieki() {return menesiuKiekis; }

//-------------------------------------

string Agentas::ImtiAgentoDuom()

{

stringstream ss;

ss << left << setw(10) << Akodas <<'|'<< setw(15) << Apavarde <<'|'<< setw(15) << Avardas <<'|'<< setw(20) << Aadresas <<'|'

<< right << setw(10) << telefonas;

return ss.str();

}

//-------------------------------------------

string Agentas::ImtiLeidinioDuom(int a)

{

stringstream ss;

ss <<" "<< left << setw(15) << L[a].ImtiPavarde() <<'|'<< setw(20) << L[a].ImtiAdresa() <<'|'<< right << setw(9)

<< L[a].ImtiKoda() <<'|'<< setw(3) << L[a].ImtiPradzia() <<" mėn."<<'|'<< setw(3) << L[a].ImtiIlgi() <<" mėn. trukmė"<<'|'<< setw(4) << L[a].ImtiKieki() <<" vnt.";

return ss.str();

}

//-------------------------------------------

string Agentas::ImtiVardoDuom()

{

stringstream ss;

ss << Apavarde <<" "<< Avardas;

return ss.str();

}

//-------------------------------------------

string Agentas::ImtiMenesioDuom(int a)

{

stringstream ss;

ss << a + 1 <<"-o mėnesio krūvis yra "<< menesinis[a] <<" leid.";

return ss.str();

}

//-------------------------------------------

void Agentas::Spausdinimas(string RezFailas)

{

ofstream rez(RezFailas.c\_str(), ios::app);

rez << ImtiAgentoDuom()<< endl;

for (int i = 0; i < Akiekis; i++)

rez << ImtiLeidinioDuom(i) << endl;

rez <<"-----------------------------------------------------------------------------"<< endl;

rez.close();

}

//-------------------------------------------

void Agentas::KruvioApskaiciavimasMenesiui()

{

for (int i = 0; i < 36; i++)

menesinis[i] = 0;

for (int i = 0; i < Akiekis; i++){

int n = L[i].ImtiPradzia() + L[i].ImtiIlgi();

if (n > 36)

n = 36;

for (int j = L[i].ImtiPradzia(); j < n; j++)

menesinis[j - 1] += L[i].ImtiKieki();

}

for (int i = 0; i < Akiekis; i++)

if ((L[i].ImtiPradzia() + L[i].ImtiIlgi() < 36) && (L[i].ImtiPradzia() + L[i].ImtiIlgi() > menesiuKiekis))

menesiuKiekis = L[i].ImtiPradzia() + L[i].ImtiIlgi();

else

if (L[i].ImtiPradzia() + L[i].ImtiIlgi() > 36)

menesiuKiekis = 36;

}

//-------------------------------------------

int Agentas::ImtiMenesioKruvi(int a) {return menesinis[a]; }

//-------------------------------------------

void Agentas::LeidiniuRikiavimas()

{

Leidinys C;

for (int i = 0; i < Akiekis - 1; i++)

for (int j = i + 1; j < Akiekis; j++)

if (L[i] < L[j]){

C = L[i];

L[i] = L[j];

L[j] = C;

}

}

bool Agentas::operator<(const Agentas & R)

{

return (Aadresas < R.Aadresas || (Aadresas == R.Aadresas && Apavarde < R.Apavarde));

}

Agentura.h:

#pragmaonce

#include"Agentas.h"

class Agentura

{

private:

int AgentuKiekis;

Agentas A[10];

public:

Agentura() {}

~Agentura() {}

void DetiAgenta(string a, string b, string c, string d, string e);

void NuskaitymasA(string DuomFailas);

void NuskaitymasB(string DuomFailas);

void SpausdinimasIfaila(string RezFailas);

void Rikiavimas();

void RikiavimoSpausdinimas(string RezFailas);

void MazaiDirbantysAgentai(string RezFailas);

void VidurkinisSpausdinimas(Agentas M[], int Mkiekis, double vidurkis, string RezFailas, int r);

};

Agentura.cpp:

#include<string>

#include<fstream>

#include"Agentas.h"

#include<iomanip>

#include"Leidinys.h"

#include"Agentura.h"

void Agentura::DetiAgenta(string a, string b, string c, string d, string e) {

if (AgentuKiekis < 10){

A[AgentuKiekis].DetiAkoda(a);

A[AgentuKiekis].DetiApavarde(b);

A[AgentuKiekis].DetiAvarda(c);

A[AgentuKiekis].DetiAadresa(d);

A[AgentuKiekis].DetiTelefona(e);

A[AgentuKiekis].DetiAkieki(0);

AgentuKiekis++;

}

}

//----------------------------------------------

void Agentura::NuskaitymasA(string DuomFailas)

{

ifstream duom(DuomFailas.c\_str());

string kod = "", pavard = "", vard = "", adres = "", telef = "";

AgentuKiekis = 0;

while (!duom.eof()){

duom >> kod;

getline(duom, pavard, ',');

getline(duom, vard, ',');

getline(duom, adres, ',');

duom >> telef;

duom.ignore();

DetiAgenta(kod, pavard, vard, adres, telef);

}

duom.close();

}

//-----------------------------------------------

void Agentura::NuskaitymasB(string DuomFailas)

{

ifstream duom(DuomFailas.c\_str());

string pavard = "", adres = "", kod = "", Akod = "";

int pradz = 0, ilg = 0, kiek = 0;

while (!duom.eof()){

getline(duom, pavard, ',');

getline(duom, adres, ',');

duom >> pradz >> ilg >> kod >> kiek >> Akod;

duom.ignore();

for (int i = 0; i < AgentuKiekis; i++)

if (Akod == A[i].ImtiAkoda())

A[i].DetiLeidini(pavard, adres, kod, pradz, ilg, kiek);

}

duom.close();

}

//------------------------------------------------

void Agentura::SpausdinimasIfaila(string RezFailas) {

ofstream rez (RezFailas.c\_str());

for(int i = 0; i < AgentuKiekis; i++)

A[i].Spausdinimas(RezFailas);

}

//------------------------------------------------

void Agentura::Rikiavimas()

{

Agentas C;

for (int i = 0; i < AgentuKiekis; i++) A[i].LeidiniuRikiavimas();

for (int i = 0; i < AgentuKiekis - 1; i++)

for (int j = i + 1; j < AgentuKiekis; j++)

if (A[i] < A[j]){

C = A[i];

A[i] = A[j];

A[j] = C;

}

}

//------------------------------------------------

void Agentura::RikiavimoSpausdinimas(string RezFailas)

{

ofstream rez(RezFailas.c\_str(), ios::app);

for (int i = 0; i < AgentuKiekis; i++)

{

A[i].KruvioApskaiciavimasMenesiui();

rez << A[i].ImtiVardoDuom() << endl;

rez <<"-----------Leidiniai--------"<< endl;

for (int j = 0; j < A[i].ImtiAkieki(); j++)

rez <<" "<< A[i].ImtiLeidinioDuom(j) << endl;

rez <<"-----------Kruviai----------"<< endl;

int n = A[i].ImtiMenesiuKieki() / 12;

for (int j1 = 0; j1 <= n; j1++){

rez <<"========================"<< j1 + 1 <<"-i metai=========================="<< endl;

for (int j2 = 0; j2 < 12; j2++)

rez <<" "<< A[i].ImtiMenesioDuom(j2 + j1 \* 12) << endl;

rez <<"-------------------------------------------------------"<< endl;

}

}

rez.close();

}

//------------------------------------------------

void Agentura::MazaiDirbantysAgentai(string RezFailas)

{

Agentas M[10];

int Mkiekis = 0;

double vidurkiai[36];

for (int i = 0; i < 36; i++){

vidurkiai[i] = 0.0;

for (int j = 0; j < AgentuKiekis; j++)

vidurkiai[i] += A[j].ImtiMenesioKruvi(i);

vidurkiai[i] /= AgentuKiekis;

}

for (int i = 0; i < 36; i++){

Mkiekis = 0;

for (int j = 0; j < AgentuKiekis; j++)

if (A[j].ImtiMenesioKruvi(i) < vidurkiai[i]){

M[Mkiekis] = A[j];

Mkiekis++;

}

VidurkinisSpausdinimas(M, Mkiekis, vidurkiai[i], RezFailas, i);

}

}

//------------------------------------------------

void Agentura::VidurkinisSpausdinimas(Agentas M[], int Mkiekis, double vidurkis, string RezFailas, int r)

{

ofstream rez(RezFailas.c\_str(), ios::app);

if (vidurkis != 0.0){

rez << r + 1 <<"-as menesis. Menesio vidurkis: "<< vidurkis << endl;

for (int i = 0; i < Mkiekis; i++)

rez << M[i].ImtiAgentoDuom() <<'|'<< right << setw(4) << M[i].ImtiMenesioKruvi(r) <<" leid. "<< endl;

rez <<"-----------------------------------------------------"<< endl;

}

rez.close();

}

Main.cpp:

#include<iostream>

#include<string>

#include<iomanip>

#include<fstream>

#include"Agentura.h"

usingnamespace std;

//-----------------------------------

const string Cduom1 = "U1.txt";

const string Cduom2 = "U2.txt";

const string Crez = "Rezultatai.txt";

//-----------------------------------

int main()

{

Agentura I;

I.NuskaitymasA(Cduom1);

I.NuskaitymasB(Cduom2);

I.SpausdinimasIfaila(Crez);

I.Rikiavimas();

I.RikiavimoSpausdinimas(Crez);

I.MazaiDirbantysAgentai(Crez);

return 0;

}

## Pradiniai duomenys ir rezultatai

U1.txt:

321456461 Petraitis, Petras, lasknfogn 123, 836168436

864616841 Jonaitis, Jonas, dshubgkunvd 122, 841616357

321654648 Simaitis, Simas, dskgnsjklgh 2, 816543435

545646651 Aleksaitis, Aleksas, sdgjhsj 1, 865165134

641643615 Rimaitis, Rimantas, dsfjasnfjg 4, 876941364

168431565 Lukaitis, Lukas, sdadfjbfhaf 3, 831646874

U2.txt:

Urmaitis, sfstgdg 5, 2 14 65874631 4 641643615

Lauksaitis, kznvsvn 3, 4 12 65468476 3 864616841

Bbbbbbbb, awfafine 3, 7 13 97461967 4 168431565

Cccccccc, afafa6, 1 8 21646168 6 321456461

Dddddddd, agsghdhh5, 3 20 16954687 2 545646651

Eeeeeeee, asaf6, 5 13 46984164 7 864616841

Ffffffff, asg agg 1, 7 2 65498516 9 168431565

Gggggggg, safa fa 2, 8 13 46841684 4 641643615

Hhhhhhhh, aflmalfm 12, 8 7 65165468 1 321654648

Iiiiiiii, asfn algnal 3-3, 4 9 16546848 3 321654648

Jjjjjjjj, asfmaf 91, 9 9 12345678 2 321456461

Rezultatai.txt:

321456461 | Petraitis | Petras | lasknfogn 123 | 836168436

Cccccccc | afafa6 | 21646168| 1 mėn.| 8 mėn. trukmė| 6 vnt.

Jjjjjjjj | asfmaf 91 | 12345678| 9 mėn.| 9 mėn. trukmė| 2 vnt.

-----------------------------------------------------------------------------

864616841 | Jonaitis | Jonas | dshubgkunvd 122 | 841616357

Lauksaitis | kznvsvn 3 | 65468476| 4 mėn.| 12 mėn. trukmė| 3 vnt.

Eeeeeeee | asaf6 | 46984164| 5 mėn.| 13 mėn. trukmė| 7 vnt.

-----------------------------------------------------------------------------

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435

Hhhhhhhh | aflmalfm 12 | 65165468| 8 mėn.| 7 mėn. trukmė| 1 vnt.

Iiiiiiii | asfn algnal 3-3 | 16546848| 4 mėn.| 9 mėn. trukmė| 3 vnt.

-----------------------------------------------------------------------------

545646651 | Aleksaitis | Aleksas | sdgjhsj 1 | 865165134

Dddddddd | agsghdhh5 | 16954687| 3 mėn.| 20 mėn. trukmė| 2 vnt.

-----------------------------------------------------------------------------

641643615 | Rimaitis | Rimantas | dsfjasnfjg 4 | 876941364

Urmaitis | sfstgdg 5 | 65874631| 2 mėn.| 14 mėn. trukmė| 4 vnt.

Gggggggg | safa fa 2 | 46841684| 8 mėn.| 13 mėn. trukmė| 4 vnt.

-----------------------------------------------------------------------------

168431565 | Lukaitis | Lukas | sdadfjbfhaf 3 | 831646874

Bbbbbbbb | awfafine 3 | 97461967| 7 mėn.| 13 mėn. trukmė| 4 vnt.

Ffffffff | asg agg 1 | 65498516| 7 mėn.| 2 mėn. trukmė| 9 vnt.

-----------------------------------------------------------------------------

Aleksaitis Aleksas

-----------Leidiniai--------

Dddddddd | agsghdhh5 | 16954687| 3 mėn.| 20 mėn. trukmė| 2 vnt.

-----------Kruviai----------

========================1-i metai==========================

1-o mėnesio krūvis yra 0 leid.

2-o mėnesio krūvis yra 0 leid.

3-o mėnesio krūvis yra 2 leid.

4-o mėnesio krūvis yra 2 leid.

5-o mėnesio krūvis yra 2 leid.

6-o mėnesio krūvis yra 2 leid.

7-o mėnesio krūvis yra 2 leid.

8-o mėnesio krūvis yra 2 leid.

9-o mėnesio krūvis yra 2 leid.

10-o mėnesio krūvis yra 2 leid.

11-o mėnesio krūvis yra 2 leid.

12-o mėnesio krūvis yra 2 leid.

-------------------------------------------------------

========================2-i metai==========================

13-o mėnesio krūvis yra 2 leid.

14-o mėnesio krūvis yra 2 leid.

15-o mėnesio krūvis yra 2 leid.

16-o mėnesio krūvis yra 2 leid.

17-o mėnesio krūvis yra 2 leid.

18-o mėnesio krūvis yra 2 leid.

19-o mėnesio krūvis yra 2 leid.

20-o mėnesio krūvis yra 2 leid.

21-o mėnesio krūvis yra 2 leid.

22-o mėnesio krūvis yra 2 leid.

23-o mėnesio krūvis yra 0 leid.

24-o mėnesio krūvis yra 0 leid.

-------------------------------------------------------

Lukaitis Lukas

-----------Leidiniai--------

Bbbbbbbb | awfafine 3 | 97461967| 7 mėn.| 13 mėn. trukmė| 4 vnt.

Ffffffff | asg agg 1 | 65498516| 7 mėn.| 2 mėn. trukmė| 9 vnt.

-----------Kruviai----------

========================1-i metai==========================

1-o mėnesio krūvis yra 0 leid.

2-o mėnesio krūvis yra 0 leid.

3-o mėnesio krūvis yra 0 leid.

4-o mėnesio krūvis yra 0 leid.

5-o mėnesio krūvis yra 0 leid.

6-o mėnesio krūvis yra 0 leid.

7-o mėnesio krūvis yra 13 leid.

8-o mėnesio krūvis yra 13 leid.

9-o mėnesio krūvis yra 4 leid.

10-o mėnesio krūvis yra 4 leid.

11-o mėnesio krūvis yra 4 leid.

12-o mėnesio krūvis yra 4 leid.

-------------------------------------------------------

========================2-i metai==========================

13-o mėnesio krūvis yra 4 leid.

14-o mėnesio krūvis yra 4 leid.

15-o mėnesio krūvis yra 4 leid.

16-o mėnesio krūvis yra 4 leid.

17-o mėnesio krūvis yra 4 leid.

18-o mėnesio krūvis yra 4 leid.

19-o mėnesio krūvis yra 4 leid.

20-o mėnesio krūvis yra 0 leid.

21-o mėnesio krūvis yra 0 leid.

22-o mėnesio krūvis yra 0 leid.

23-o mėnesio krūvis yra 0 leid.

24-o mėnesio krūvis yra 0 leid.

-------------------------------------------------------

Petraitis Petras

-----------Leidiniai--------

Jjjjjjjj | asfmaf 91 | 12345678| 9 mėn.| 9 mėn. trukmė| 2 vnt.

Cccccccc | afafa6 | 21646168| 1 mėn.| 8 mėn. trukmė| 6 vnt.

-----------Kruviai----------

========================1-i metai==========================

1-o mėnesio krūvis yra 6 leid.

2-o mėnesio krūvis yra 6 leid.

3-o mėnesio krūvis yra 6 leid.

4-o mėnesio krūvis yra 6 leid.

5-o mėnesio krūvis yra 6 leid.

6-o mėnesio krūvis yra 6 leid.

7-o mėnesio krūvis yra 6 leid.

8-o mėnesio krūvis yra 6 leid.

9-o mėnesio krūvis yra 2 leid.

10-o mėnesio krūvis yra 2 leid.

11-o mėnesio krūvis yra 2 leid.

12-o mėnesio krūvis yra 2 leid.

-------------------------------------------------------

========================2-i metai==========================

13-o mėnesio krūvis yra 2 leid.

14-o mėnesio krūvis yra 2 leid.

15-o mėnesio krūvis yra 2 leid.

16-o mėnesio krūvis yra 2 leid.

17-o mėnesio krūvis yra 2 leid.

18-o mėnesio krūvis yra 0 leid.

19-o mėnesio krūvis yra 0 leid.

20-o mėnesio krūvis yra 0 leid.

21-o mėnesio krūvis yra 0 leid.

22-o mėnesio krūvis yra 0 leid.

23-o mėnesio krūvis yra 0 leid.

24-o mėnesio krūvis yra 0 leid.

-------------------------------------------------------

Simaitis Simas

-----------Leidiniai--------

Iiiiiiii | asfn algnal 3-3 | 16546848| 4 mėn.| 9 mėn. trukmė| 3 vnt.

Hhhhhhhh | aflmalfm 12 | 65165468| 8 mėn.| 7 mėn. trukmė| 1 vnt.

-----------Kruviai----------

========================1-i metai==========================

1-o mėnesio krūvis yra 0 leid.

2-o mėnesio krūvis yra 0 leid.

3-o mėnesio krūvis yra 0 leid.

4-o mėnesio krūvis yra 3 leid.

5-o mėnesio krūvis yra 3 leid.

6-o mėnesio krūvis yra 3 leid.

7-o mėnesio krūvis yra 3 leid.

8-o mėnesio krūvis yra 4 leid.

9-o mėnesio krūvis yra 4 leid.

10-o mėnesio krūvis yra 4 leid.

11-o mėnesio krūvis yra 4 leid.

12-o mėnesio krūvis yra 4 leid.

-------------------------------------------------------

========================2-i metai==========================

13-o mėnesio krūvis yra 1 leid.

14-o mėnesio krūvis yra 1 leid.

15-o mėnesio krūvis yra 0 leid.

16-o mėnesio krūvis yra 0 leid.

17-o mėnesio krūvis yra 0 leid.

18-o mėnesio krūvis yra 0 leid.

19-o mėnesio krūvis yra 0 leid.

20-o mėnesio krūvis yra 0 leid.

21-o mėnesio krūvis yra 0 leid.

22-o mėnesio krūvis yra 0 leid.

23-o mėnesio krūvis yra 0 leid.

24-o mėnesio krūvis yra 0 leid.

-------------------------------------------------------

Jonaitis Jonas

-----------Leidiniai--------

Lauksaitis | kznvsvn 3 | 65468476| 4 mėn.| 12 mėn. trukmė| 3 vnt.

Eeeeeeee | asaf6 | 46984164| 5 mėn.| 13 mėn. trukmė| 7 vnt.

-----------Kruviai----------

========================1-i metai==========================

1-o mėnesio krūvis yra 0 leid.

2-o mėnesio krūvis yra 0 leid.

3-o mėnesio krūvis yra 0 leid.

4-o mėnesio krūvis yra 3 leid.

5-o mėnesio krūvis yra 10 leid.

6-o mėnesio krūvis yra 10 leid.

7-o mėnesio krūvis yra 10 leid.

8-o mėnesio krūvis yra 10 leid.

9-o mėnesio krūvis yra 10 leid.

10-o mėnesio krūvis yra 10 leid.

11-o mėnesio krūvis yra 10 leid.

12-o mėnesio krūvis yra 10 leid.

-------------------------------------------------------

========================2-i metai==========================

13-o mėnesio krūvis yra 10 leid.

14-o mėnesio krūvis yra 10 leid.

15-o mėnesio krūvis yra 10 leid.

16-o mėnesio krūvis yra 7 leid.

17-o mėnesio krūvis yra 7 leid.

18-o mėnesio krūvis yra 0 leid.

19-o mėnesio krūvis yra 0 leid.

20-o mėnesio krūvis yra 0 leid.

21-o mėnesio krūvis yra 0 leid.

22-o mėnesio krūvis yra 0 leid.

23-o mėnesio krūvis yra 0 leid.

24-o mėnesio krūvis yra 0 leid.

-------------------------------------------------------

Rimaitis Rimantas

-----------Leidiniai--------

Urmaitis | sfstgdg 5 | 65874631| 2 mėn.| 14 mėn. trukmė| 4 vnt.

Gggggggg | safa fa 2 | 46841684| 8 mėn.| 13 mėn. trukmė| 4 vnt.

-----------Kruviai----------

========================1-i metai==========================

1-o mėnesio krūvis yra 0 leid.

2-o mėnesio krūvis yra 4 leid.

3-o mėnesio krūvis yra 4 leid.

4-o mėnesio krūvis yra 4 leid.

5-o mėnesio krūvis yra 4 leid.

6-o mėnesio krūvis yra 4 leid.

7-o mėnesio krūvis yra 4 leid.

8-o mėnesio krūvis yra 8 leid.

9-o mėnesio krūvis yra 8 leid.

10-o mėnesio krūvis yra 8 leid.

11-o mėnesio krūvis yra 8 leid.

12-o mėnesio krūvis yra 8 leid.

-------------------------------------------------------

========================2-i metai==========================

13-o mėnesio krūvis yra 8 leid.

14-o mėnesio krūvis yra 8 leid.

15-o mėnesio krūvis yra 8 leid.

16-o mėnesio krūvis yra 4 leid.

17-o mėnesio krūvis yra 4 leid.

18-o mėnesio krūvis yra 4 leid.

19-o mėnesio krūvis yra 4 leid.

20-o mėnesio krūvis yra 4 leid.

21-o mėnesio krūvis yra 0 leid.

22-o mėnesio krūvis yra 0 leid.

23-o mėnesio krūvis yra 0 leid.

24-o mėnesio krūvis yra 0 leid.

-------------------------------------------------------

1-as menesis. Menesio vidurkis: 1

545646651 | Aleksaitis | Aleksas | sdgjhsj 1 | 865165134| 0 leid.

168431565 | Lukaitis | Lukas | sdadfjbfhaf 3 | 831646874| 0 leid.

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435| 0 leid.

864616841 | Jonaitis | Jonas | dshubgkunvd 122 | 841616357| 0 leid.

641643615 | Rimaitis | Rimantas | dsfjasnfjg 4 | 876941364| 0 leid.

-----------------------------------------------------

2-as menesis. Menesio vidurkis: 1.66667

545646651 | Aleksaitis | Aleksas | sdgjhsj 1 | 865165134| 0 leid.

168431565 | Lukaitis | Lukas | sdadfjbfhaf 3 | 831646874| 0 leid.

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435| 0 leid.

864616841 | Jonaitis | Jonas | dshubgkunvd 122 | 841616357| 0 leid.

-----------------------------------------------------

3-as menesis. Menesio vidurkis: 2

168431565 | Lukaitis | Lukas | sdadfjbfhaf 3 | 831646874| 0 leid.

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435| 0 leid.

864616841 | Jonaitis | Jonas | dshubgkunvd 122 | 841616357| 0 leid.

-----------------------------------------------------

4-as menesis. Menesio vidurkis: 3

545646651 | Aleksaitis | Aleksas | sdgjhsj 1 | 865165134| 2 leid.

168431565 | Lukaitis | Lukas | sdadfjbfhaf 3 | 831646874| 0 leid.

-----------------------------------------------------

5-as menesis. Menesio vidurkis: 4.16667

545646651 | Aleksaitis | Aleksas | sdgjhsj 1 | 865165134| 2 leid.

168431565 | Lukaitis | Lukas | sdadfjbfhaf 3 | 831646874| 0 leid.

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435| 3 leid.

641643615 | Rimaitis | Rimantas | dsfjasnfjg 4 | 876941364| 4 leid.

-----------------------------------------------------

6-as menesis. Menesio vidurkis: 4.16667

545646651 | Aleksaitis | Aleksas | sdgjhsj 1 | 865165134| 2 leid.

168431565 | Lukaitis | Lukas | sdadfjbfhaf 3 | 831646874| 0 leid.

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435| 3 leid.

641643615 | Rimaitis | Rimantas | dsfjasnfjg 4 | 876941364| 4 leid.

-----------------------------------------------------

7-as menesis. Menesio vidurkis: 6.33333

545646651 | Aleksaitis | Aleksas | sdgjhsj 1 | 865165134| 2 leid.

321456461 | Petraitis | Petras | lasknfogn 123 | 836168436| 6 leid.

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435| 3 leid.

641643615 | Rimaitis | Rimantas | dsfjasnfjg 4 | 876941364| 4 leid.

-----------------------------------------------------

8-as menesis. Menesio vidurkis: 7.16667

545646651 | Aleksaitis | Aleksas | sdgjhsj 1 | 865165134| 2 leid.

321456461 | Petraitis | Petras | lasknfogn 123 | 836168436| 6 leid.

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435| 4 leid.

-----------------------------------------------------

9-as menesis. Menesio vidurkis: 5

545646651 | Aleksaitis | Aleksas | sdgjhsj 1 | 865165134| 2 leid.

168431565 | Lukaitis | Lukas | sdadfjbfhaf 3 | 831646874| 4 leid.

321456461 | Petraitis | Petras | lasknfogn 123 | 836168436| 2 leid.

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435| 4 leid.

-----------------------------------------------------

10-as menesis. Menesio vidurkis: 5

545646651 | Aleksaitis | Aleksas | sdgjhsj 1 | 865165134| 2 leid.

168431565 | Lukaitis | Lukas | sdadfjbfhaf 3 | 831646874| 4 leid.

321456461 | Petraitis | Petras | lasknfogn 123 | 836168436| 2 leid.

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435| 4 leid.

-----------------------------------------------------

11-as menesis. Menesio vidurkis: 5

545646651 | Aleksaitis | Aleksas | sdgjhsj 1 | 865165134| 2 leid.

168431565 | Lukaitis | Lukas | sdadfjbfhaf 3 | 831646874| 4 leid.

321456461 | Petraitis | Petras | lasknfogn 123 | 836168436| 2 leid.

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435| 4 leid.

-----------------------------------------------------

12-as menesis. Menesio vidurkis: 5

545646651 | Aleksaitis | Aleksas | sdgjhsj 1 | 865165134| 2 leid.

168431565 | Lukaitis | Lukas | sdadfjbfhaf 3 | 831646874| 4 leid.

321456461 | Petraitis | Petras | lasknfogn 123 | 836168436| 2 leid.

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435| 4 leid.

-----------------------------------------------------

13-as menesis. Menesio vidurkis: 4.5

545646651 | Aleksaitis | Aleksas | sdgjhsj 1 | 865165134| 2 leid.

168431565 | Lukaitis | Lukas | sdadfjbfhaf 3 | 831646874| 4 leid.

321456461 | Petraitis | Petras | lasknfogn 123 | 836168436| 2 leid.

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435| 1 leid.

-----------------------------------------------------

14-as menesis. Menesio vidurkis: 4.5

545646651 | Aleksaitis | Aleksas | sdgjhsj 1 | 865165134| 2 leid.

168431565 | Lukaitis | Lukas | sdadfjbfhaf 3 | 831646874| 4 leid.

321456461 | Petraitis | Petras | lasknfogn 123 | 836168436| 2 leid.

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435| 1 leid.

-----------------------------------------------------

15-as menesis. Menesio vidurkis: 4.33333

545646651 | Aleksaitis | Aleksas | sdgjhsj 1 | 865165134| 2 leid.

168431565 | Lukaitis | Lukas | sdadfjbfhaf 3 | 831646874| 4 leid.

321456461 | Petraitis | Petras | lasknfogn 123 | 836168436| 2 leid.

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435| 0 leid.

-----------------------------------------------------

16-as menesis. Menesio vidurkis: 3.16667

545646651 | Aleksaitis | Aleksas | sdgjhsj 1 | 865165134| 2 leid.

321456461 | Petraitis | Petras | lasknfogn 123 | 836168436| 2 leid.

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435| 0 leid.

-----------------------------------------------------

17-as menesis. Menesio vidurkis: 3.16667

545646651 | Aleksaitis | Aleksas | sdgjhsj 1 | 865165134| 2 leid.

321456461 | Petraitis | Petras | lasknfogn 123 | 836168436| 2 leid.

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435| 0 leid.

-----------------------------------------------------

18-as menesis. Menesio vidurkis: 1.66667

321456461 | Petraitis | Petras | lasknfogn 123 | 836168436| 0 leid.

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435| 0 leid.

864616841 | Jonaitis | Jonas | dshubgkunvd 122 | 841616357| 0 leid.

-----------------------------------------------------

19-as menesis. Menesio vidurkis: 1.66667

321456461 | Petraitis | Petras | lasknfogn 123 | 836168436| 0 leid.

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435| 0 leid.

864616841 | Jonaitis | Jonas | dshubgkunvd 122 | 841616357| 0 leid.

-----------------------------------------------------

20-as menesis. Menesio vidurkis: 1

168431565 | Lukaitis | Lukas | sdadfjbfhaf 3 | 831646874| 0 leid.

321456461 | Petraitis | Petras | lasknfogn 123 | 836168436| 0 leid.

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435| 0 leid.

864616841 | Jonaitis | Jonas | dshubgkunvd 122 | 841616357| 0 leid.

-----------------------------------------------------

21-as menesis. Menesio vidurkis: 0.333333

168431565 | Lukaitis | Lukas | sdadfjbfhaf 3 | 831646874| 0 leid.

321456461 | Petraitis | Petras | lasknfogn 123 | 836168436| 0 leid.

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435| 0 leid.

864616841 | Jonaitis | Jonas | dshubgkunvd 122 | 841616357| 0 leid.

641643615 | Rimaitis | Rimantas | dsfjasnfjg 4 | 876941364| 0 leid.

-----------------------------------------------------

22-as menesis. Menesio vidurkis: 0.333333

168431565 | Lukaitis | Lukas | sdadfjbfhaf 3 | 831646874| 0 leid.

321456461 | Petraitis | Petras | lasknfogn 123 | 836168436| 0 leid.

321654648 | Simaitis | Simas | dskgnsjklgh 2 | 816543435| 0 leid.

864616841 | Jonaitis | Jonas | dshubgkunvd 122 | 841616357| 0 leid.

641643615 | Rimaitis | Rimantas | dsfjasnfjg 4 | 876941364| 0 leid.

-----------------------------------------------------

# Teksto analizė ir redagavimas (L7)

## Darbo užduotis

16. \*\*Faile duotas tekstas. Skyrikliai žinomi. Išrinkite visus simetriškus teksto žodžius (žodžiai iš abiejų pusių skaitomi vienodai). Perkelkite šiuos žodžius su už jų esančiais skyrikliais į teksto pradžią. Formatuojamos eilutės ilgis neturi viršyti nurodyto dydžio.

## Programos tekstas

Main.cpp:

#include <iomanip>

using namespace std;

//----------------------------------

const int Ck = 64;

const string Cduom = "Duomenys.txt";

const string Crez = "Rezultatai.txt";

const string Canaliz = "Analize.txt";

const int CmaxEil = 35; //Nusako, kiek maximum gali buti simboliu eiluteje

//----------------------------------

void Nuskaitymas(string M[][Ck], string DuomFailas, int & n, int K[], string Z[][Ck]); //Nuskaitomi duomenys

void Paskirstymas(int n, string eilute, string M[][Ck], int K[], string Z[][Ck]); //Suskirsto eilute i zodzius ir skyriklius

void Rezultatai(string M[][Ck], string RezFailas, int n, int K[], string Z[][Ck]); //Isveda teksta i rezultatu faila

void Simetriskumas(int n, string M[][Ck], int K[], string Z[][Ck]); //bazine simetriskumo ir darbo su juo procedura

bool SimetriskumoPatikra(string zodis); //Patikrina, ar zodis simetriskas

void AnalizesIrasymas(string zodis, int i, int kuris, string RezFailas); //Isveda zodzio duomenis i analizes faila

void ZodisIpradzia(int i, int j, string M[][Ck], int K[], string Z[][Ck]); //Perkelia simetriska zodi su skyrikliu i teksto pradzia

void Tvarkymas(string M[][Ck], int & n, int K[], string Z[][Ck], int Max); //Tvarko eilutes, kad jos nevirsytu nustatyto limito

void Kelimas(string M[][Ck], int r1, int r2, string Z[][Ck], int K[], int & n); //Kelia likusius eilutes zodzius i sekancia eilute

//----------------------------------

int main()

{

setlocale(LC\_ALL, "Lithuanian");

string M[Ck][Ck];

string Z[Ck][Ck];

int n = 1;

int K[Ck]; for (int i = 0; i < Ck - 1; i++) K[i] = 0;

Nuskaitymas(M, Cduom, n, K, Z);

Simetriskumas(n, M, K, Z);

Tvarkymas(M, n, K, Z, CmaxEil);

Rezultatai(M, Crez, n, K, Z);

return 0;

}

//---------------------------------------------------

void Nuskaitymas(string M[][Ck], string DuomFailas, int & n, int K[], string Z[][Ck])

{

string eilute;

ifstream duom(DuomFailas.c\_str());

while (!duom.eof()){

getline (duom, eilute);

Paskirstymas (n, eilute, M, K, Z);

n++;

}

duom.close();

}

//---------------------------------------------------

void Paskirstymas(int n, string eilute, string M[][Ck], int K[], string Z[][Ck])

{

string Skirt = " .,!?:;()\t";

string zodis;

string zenklas = "";

int i = 0;

int j = 0;

while ((i = eilute.find\_first\_not\_of(Skirt, j)) != string::npos) {

j = eilute.find\_first\_of(Skirt, i);

int r = eilute.find\_first\_not\_of(Skirt, j);

if (r != -1 && j != -1)

zenklas = eilute.substr(j, r - j);

else

if (j!= -1)

zenklas = eilute.substr(j);

zodis = eilute.substr(i, j - i);

M[n][K[n]] = zodis;

Z[n][K[n]] = zenklas;

K[n]++;

}

}

//----------------------------------------------------

void Simetriskumas(int n, string M[][Ck], int K[], string Z[][Ck])

{

ofstream rez(Canaliz.c\_str());

rez << left << setw(15) << "-----Zodis-----" << '|' << right << setw(3) << "Eil" << '|' << setw(3) << "Poz" << endl;

rez << "------------------------------------" << endl;

rez.close();

for (int i = 1; i < n; i++)

for (int j = 0; j < K[i]; j++)

if (SimetriskumoPatikra(M[i][j])){

AnalizesIrasymas(M[i][j], i, j, Canaliz);

ZodisIpradzia(i, j, M, K, Z);

}

}

//----------------------------------------------------

bool SimetriskumoPatikra(string zodis)

{

bool T = 1;

int r = zodis.length();

int r2 = r / 2;

for (int i = 0; i <= r2; i++)

if (zodis[i] != zodis[r - 1 - i])

T = 0;

return T;

}

//----------------------------------------------------

void Rezultatai(string M[][Ck], string RezFailas, int n, int K[], string Z[][Ck])

{

ofstream rez(RezFailas.c\_str());

for (int i = 0; i < n; i++){

for (int j = 0; j < K[i]; j++)

rez << M[i][j] << Z[i][j];

rez << endl;

}

rez.close();

}

//----------------------------------------------------

void AnalizesIrasymas(string zodis, int i, int kuris, string RezFailas)

{

ofstream rez(RezFailas.c\_str(), ios::app);

rez << left << setw(15) << zodis << '|' << right << setw(3) << i << '|' << setw(3) << kuris + 1 << endl;

rez.close();

}

//----------------------------------------------------

void ZodisIpradzia(int i, int j, string M[][Ck], int K[], string Z[][Ck])

{

M[0][K[0]] = M[i][j];

Z[0][K[0]] = Z[i][j];

for (int m = j; m < K[i] - 1; m++){

M[i][m] = M[i][m + 1];

Z[i][m] = Z[i][m + 1];

}

M[i][K[i] - 1] = "";

Z[i][K[i] - 1] = "";

K[0]++;

K[i]--;

}

//-----------------------------------------------------

void Tvarkymas(string M[][Ck], int & n, int K[], string Z[][Ck], int Max)

{

for (int i = 0; i <n; i++){

string Sak = "";

for (int j = 0; j < K[i]; j++){

string r = Sak + M[i][j] + Z[i][j];

int r1 = r.length();

if (r1 > Max){

Kelimas(M, i, j, Z, K, n);

break;

}

else

Sak += M[i][j] + Z[i][j];

}

}

}

//------------------------------------------------------------------

void Kelimas(string M[][Ck], int r1, int r2, string Z[][Ck], int K[], int & n)

{

if (r1 == n - 1){

n++;

K[n - 1] = 0;

}

int m2 = 0; //Sis kintamasis nusako, kuris zodis yra keliamas i sekancia eilute

for (int i = r2; i < K[r1]; i++){

K[r1 + 1]++;

for (int j = K[r1 + 1] - 1; j > m2; j--){

M[r1 + 1][j] = M[r1 + 1][j - 1];

Z[r1 + 1][j] = Z[r1 + 1][j - 1];

}

M[r1 + 1][m2] = M[r1][i];

Z[r1 + 1][m2] = Z[r1][i];

m2++;

}

K[r1] = K[r1] - m2;

}

## Pradiniai duomenys ir rezultatai

Duomenys.txt:

Kalnuos dainuoja

Kalnuos dainuoja,,,rageliai uluulu,:, jo

Ir zydi geles tenai kalnuos.

Kai tykus vakaras ateina,

Saulute leidzias uz kalnu

Ir tolumoj nakties padangėj

Sumirga tukstanciai zvaigzdziu.

Kai saule rytmeti pakyla,

Zibutes rymo pakely

Ir piemenaites banda gena

Tenai toli olilo toli.

Tada as vėl taves ilgiuosi

Ir laukiu zydinciam sode,

O akys skesta tolumoje:

Gal pamatysiu as ava.

Analize.txt:

-----Zodis-----|Eil|Poz

------------------------------------

uluulu | 5| 4

olilo | 16| 3

O | 20| 1

ava | 21| 4

Rezultatai.txt: (Eilutės limitas – 35 simboliai)

uluulu,:, olilo O ava.

Kalnuos dainuoja

Kalnuos dainuoja,,,rageliai jo,:,

Ir zydi geles tenai kalnuos.

Kai tykus vakaras ateina,

Saulute leidzias uz kalnu

Ir tolumoj nakties padangėj

Sumirga tukstanciai zvaigzdziu.

Kai saule rytmeti pakyla,

Zibutes rymo pakely

Ir piemenaites banda gena

Tenai toli toli.

Tada as vėl taves ilgiuosi

Ir laukiu zydinciam sode,

akys skesta tolumoje:

Gal pamatysiu as