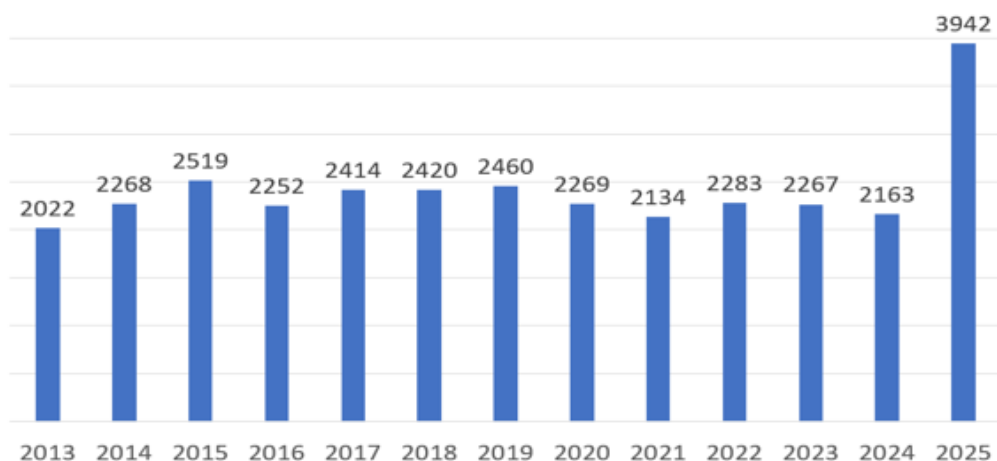


**INFORMATIKOS VALSTYBINIO BRANDOS
EGZAMINO II DALIES KANDIDATŲ DARBŲ
VERTINTOJŲ MOKYMŲ II DALIS**

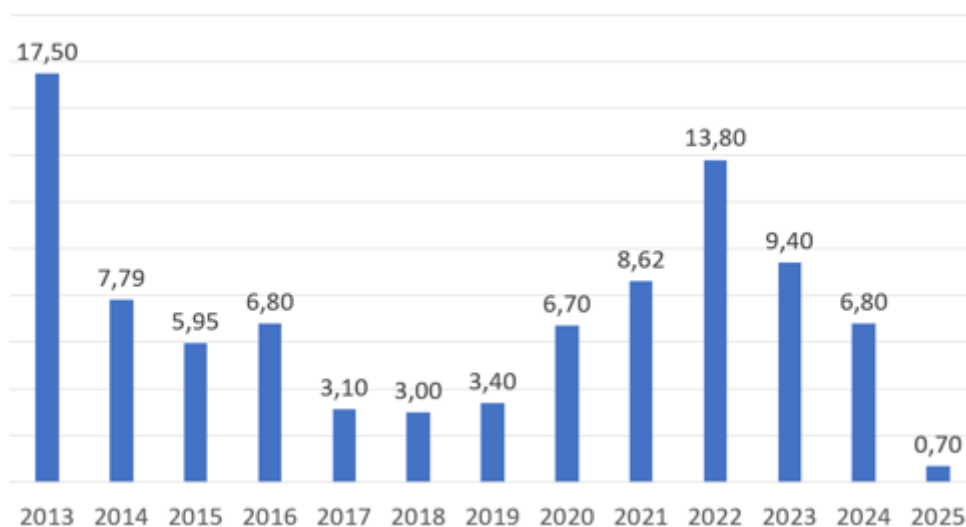
Įvadinė dalis

Dalykų bendrosiose programose siektini ugdymo rezultatai aprašyti kaip mokinių kompetencijų ugdymo pasiekimai. Informatikos (nuo 2025 m., informacinių technologijų – 2006–2024 m.) egzaminas vykdomas siekiant įvertinti kandidatų įgytas kompetencijas. 1 pav. pateikta informatikos VBE (iki 2025 m. informacinių technologijų) laikančių kandidatų skaičiaus dinamika.



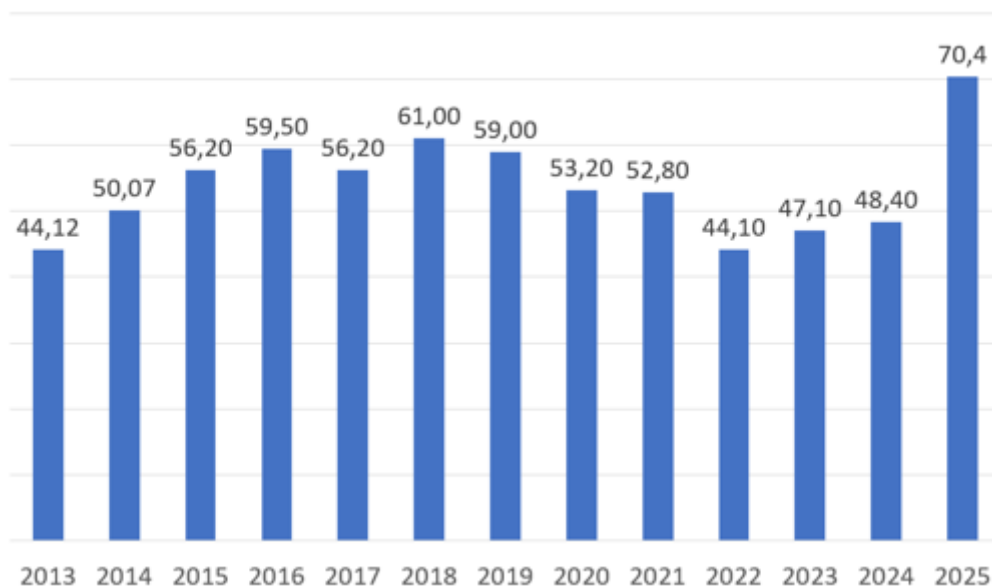
1 pav. Informatikos egzaminą (iki 2025 m. informacinių technologijų) laikančių kandidatų skaičiaus dinamika

2 pav. pateikti 2013–2025 m. egzamino neišlaikiusių kandidatų skaičiai procentais.



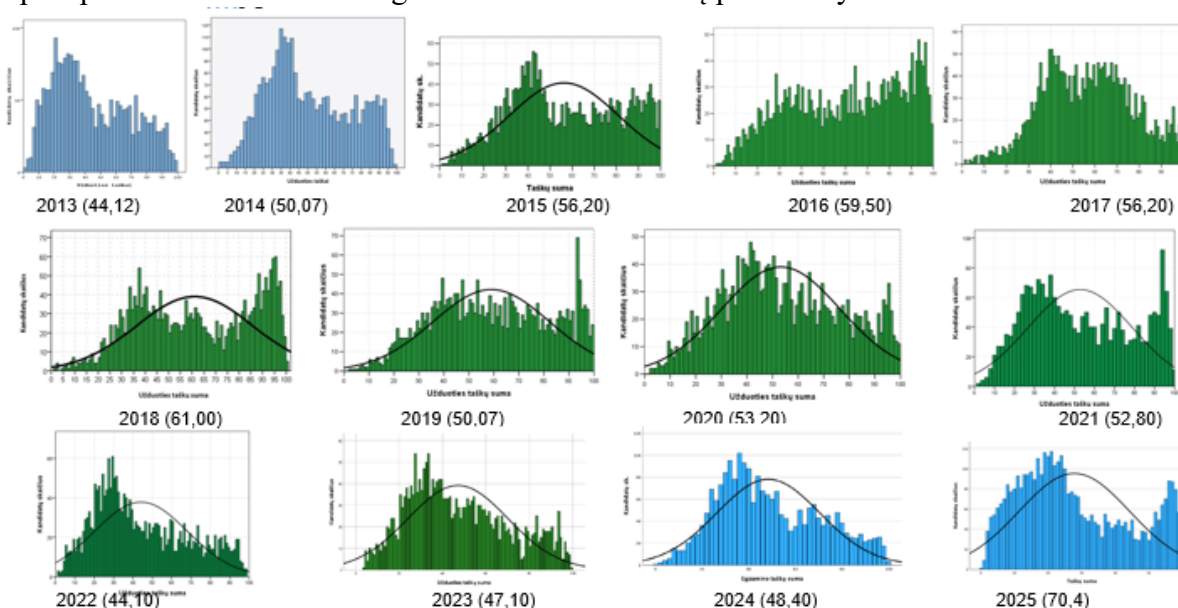
2 pav. 2013-2025 m. egzamino neišlaikiusių kandidatų skaičius procentais

3 pav. pateiktas 2013–2025 m. egzaminą laikusių kandidatų egzamino užduoties taškų vidurkis.



3 pav. 2013–2025 m. egzaminą laikusių kandidatų egzamino užduoties taškų vidurkis

4 pav. pateikti 2013–2025 m. egzamino užduoties taškų pasiskirstymai.



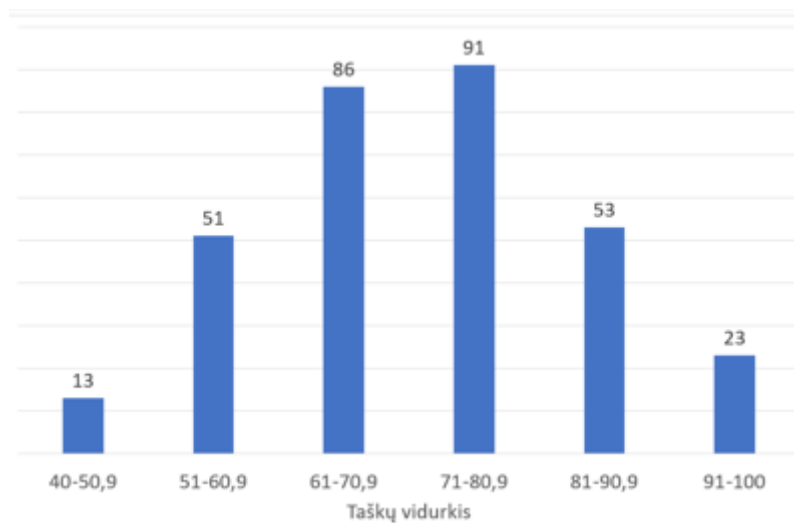
4 pav. 2013–2025 m. egzamino užduoties taškų pasiskirstymai

2025 m. informatikos egzamino II dalies užduočių sunkumas ir koreliacija su visa II egzamino dalies užduotimi pateikti 1 lentelėje.

1 lentelė. 2025 m. informatikos egzamino II dalies užduočių sunkumas ir koreliacija su visa II dalies užduotimi

Užduotis	Sunkumas	Koreliacija su visa užduotimi
Testas	50,2 (vidutinio sunkumo)	0,51
Duomenų tyrybos užduotis	66,8 (lengva)	0,80
Programavimo užduotis	30,3 (sunki)	0,95

2025 m. mokyklų skaičiaus pasiskirstymas pagal informatikos valstybinio brandos egzamino laikiusių kandidatų balų vidurkį (iš 402 mokyklų laikė 317 mokyklų kandidatai, tai sudaro 79 %) pateiktas 5 pav.



5 pav. 2025 m. mokyklų skaičiaus pasiskirstymas pagal informatikos VBE laikiusių kandidatų balų vidurkį

Pagrindinės išvalgos apie II egzamino dalies užduočių evoliuciją 2013-2025 m.

Testą 2013–2024 m. metais sudarė pasirenkamojo ir atvirojo tipo klausimai. Nuo 2025 m. įvyko esminis pokytis – pereita prie atvirų klausimų, reikalaujančių samprotavimo, paaiškinimo, pavyzdžių.

2013–2024 m. kandidatai atliko skaičiuoklės užduotį. Nuo 2025 m. užduotis vadinama Duomenų tyrybos užduotimi. Pagrindiniai pokyčiai – padidėjo naudojamų funkcijų skaičius, užduotis reikalauja ne tik atlikti skaičiavimus, bet ir interpretuoti pateiktus duomenis, argumentuotai paaiškinti gautus rezultatus.

2013–2024 m. metais kandidatai per egzaminą sprendė 2 programavimo užduotis. 2025 m. kandidatams buvo pateikta 1 programavimo užduotis. Programavimo užduotyse naudojamos sudėtingesnės duomenų struktūros, tačiau kandidatams suteikta daugiau laisvės jas pasirenkant ir taikant užduoties sprendimo algoritmus.

Informatikos VBE vertinimas, 2025 m. Programavimo dalis

Programavimo užduočių ir taškų skaičius 2006–2025 m. Iki 2024 m. egzaminas vadinosi informacinių technologijų valstybinis brandos egzaminas, nuo 2025 m. – informatikos valstybinis brandos egzaminas.

	Iki 2024 m.	Nuo 2025 m.
Užduočių skaičius	2	1-2
Taškų skaičius	50 I – 20; II – 30 I – 25; II – 25	30-35
Programavimo kalbos	<i>Pascal, C++, Python</i>	<i>C++, Python</i>

Nuo 2025 m. kandidatai, laikydami informatikos brandos valstybinį egzaminą, gali naudotis programavimo kalbų žinynais.

2025 m. informatikos egzamino II dalies programavimo užduotis

8.1.2. turi būti įdiegta ir pasiekama programinė įranga: „Python“ programavimo aplinka „LR NŠA Python“; ne ankstesnė kaip „CodeBlocks 17.12“ versija („Windows 10“, „Windows 11“); C++ programavimo kalbos žinynas; failų ir katalogų tvarkymo programa (įskaitant ir pakavimo programą): „Windows Explorer“ („My Computer“), „7-Zip“ arba alternatyvi; PDF failų peržiūros programa (ne ankstesnė kaip „Adobe Reader DC“ versija arba ne ankstesnė kaip „Foxit Reader 12.x.x“ versija arba lygiavertė); skaičiuoklė (ne ankstesnė kaip raštinės paketo „Microsoft Office 2019“ (išskyrus „Microsoft Office 365“, kuri yra personalizuota ir reikalauja prisijungimo prie interneto) ir ne ankstesnė kaip „LibreOffice 7.6.x Calc“ versija); naršyklė (ne ankstesnė kaip „Microsoft Edge 123“ versija, ne ankstesnė kaip „Mozilla Firefox 124.x“ versija, ne ankstesnė kaip „Google Chrome 124.x“ versija)

Bandė atlikti programavimo užduotis	2021 m.	2022 m.	2023 m.	2024 m.	2025 m.
1 programavimo užduotis	77 %	68 %	76 %	78 %	76 %
2 programavimo užduotis	67 %	59 %	59 %	59 %	

2025 m. C++ programavimo kalba užduotis atliko 90 %, Python – 10 % kandidatų.

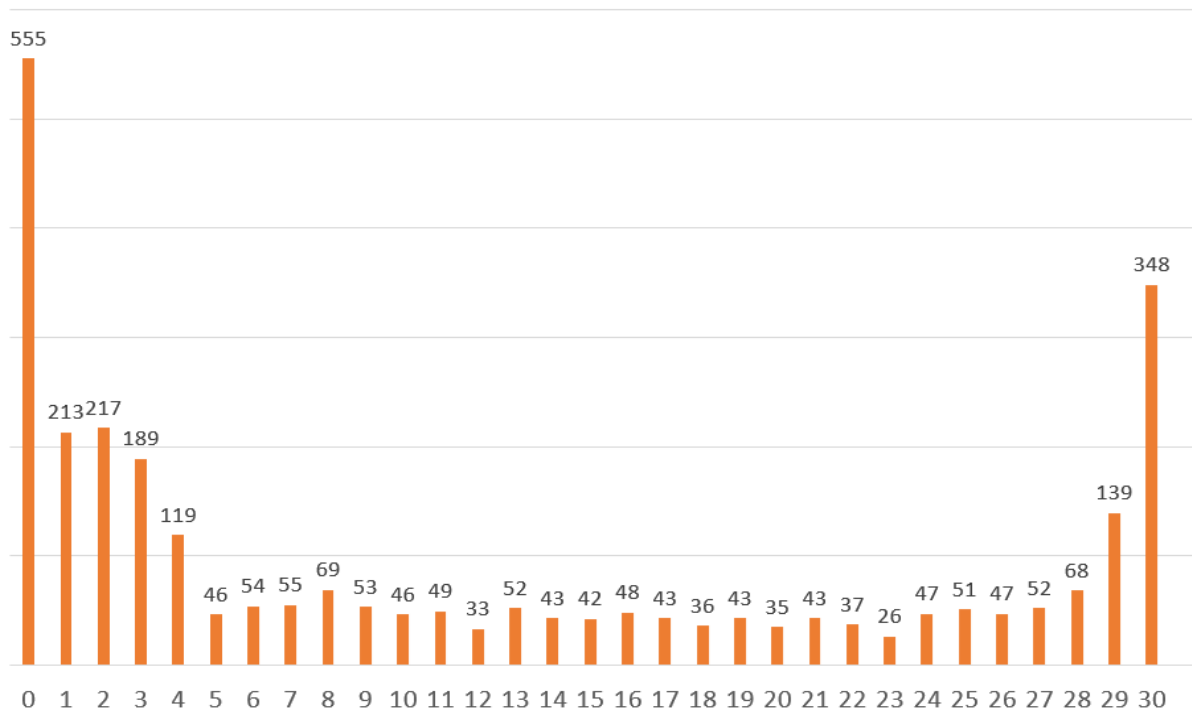
2022-2025 m. programavimo užduočių vidurkių palyginimas

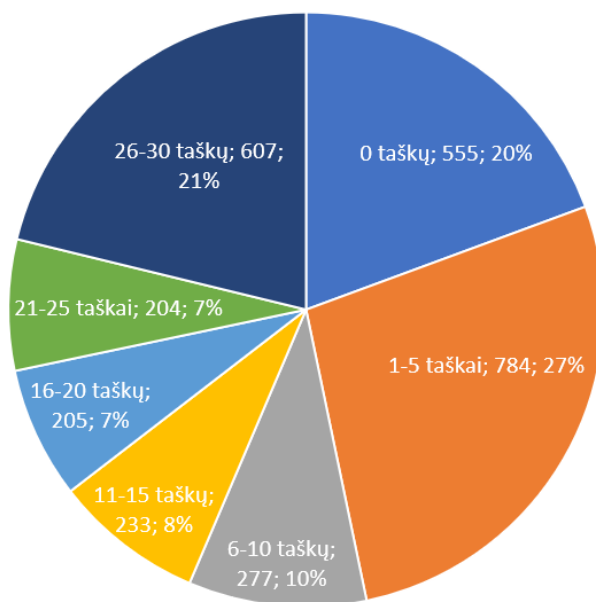
Metai	U1 vidurkis	U2 vidurkis
2022	12,4	9,5
2023	10,8	11,8
2024	12,2	8
2025	11,8	

2025 m. C++ ir Python programavimo kalba atliktų užduočių vidurkis nesiskiria.

2025 m. informatikos VBE programavimo užduotis: kandidatų skaičiaus pasiskirstymas pagal taškus.

Kandidatų pasiskirstymas pagal taškus





Programavimo praktinė užduotis publikuojama: https://www.nsa.smsm.lt/old/wp-content/uploads/2025/06/2025_I_VBEII_pagr.pdf

Sąlygos ir rezultatų pavyzdžio nesutapimas ir patikslinimas:

Rezultatai

Rezultatai pateikiami tekstiniame faile **Rezultatai.txt**.

- Pirmoje eilutėje turi būti įrašytas žodis *Statistika*.
- Tolimesnėse eilutėse turi būti pateikta socialinių-pilietinių veiklų trukmės statistika procentais:
 - informacija apie kiekvieną pilietinę-socialinę veiklą (atskirose eilutėse): socialinės-pilietinės veiklos trumpinys; procentų skaičius (šimtųjų tikslumu), nurodantis, kurią mokinių visų vykdytų socialinių-pilietinių veiklų trukmės dalį sudarė ši veikla; po skaičiaus turi būti procento ženklas „%“, nuo skaičiaus atskirtas vienu tarpo simboliu;
 - veiklos turi būti surikiuotos pagal veiklos trukmės procentų skaičių mažėjančiai;
 - turi būti pateikiamos tik tos veiklos, kurias vykdė mokiniai, surinkę ne mažiau kaip 20 valandų.*
- Naujoje eilutėje po socialinių-pilietinių veiklų trukmės statistika turi būti įrašytas žodis *Mokiniai*.
- Kitose (atskirose eilutėse) turi būti pateiktas mokinių, kurie surinko ne mažiau kaip 20 valandų, sąrašas: mokinio vardas; surinktų socialinės-pilietinės veiklos valandų skaičius. Sąrašas turi būti surikiuotas pagal mokinių vardus abėcėliškai.
- Duomenys visose eilutėse atskiriami vienu tarpo simboliu.

Nurodymai

- Sukurkite ir parašykite **vieną** funkciją, kuri surikiuoja socialinių-pilietinių veiklų statistiką pagal procento reikšmę (skaičių) mažėjančiai ir mokinių sąrašą abėcėliškai.

*Informacinėje sistemoje NECIS pateikta pastaba:

Programavimo užduotyje parašyta programa turi pateikti (1) visų mokinių vykdytų visų socialinių veiklų statistiką ir pateikti (2) mokinių, kurie surinko ne mažiau kaip 20 valandų, sąrašą.

Rezultatų failo pavyzdys	Paaškinimai
Statistika	Įrašytas žodis <i>Statistika</i>.
Dar. 29.21 %	Pateikta socialinių-pilietinių veiklų statistika: socialinės-pilietinės veiklos trumpinys; procentų skaičius (šimtųjų tikslumu), nurodantis, kurią mokinių visų vykdytų socialinių-pilietinių veiklų trukmės dalį sudarė ši veikla; po skaičiaus turi būti procento ženklas „%“, nuo skaičiaus atskirtas vienu tarpo simboliu.
Kul. 21.35 %	
Kit. 19.10 %	
Soc. 12.36 %	
Pil. 10.11 %	
Spo. 7.87 %	
Mokiniai	Sąrašas pateiktas tik tos veiklos, kurias mokiniai vykdė.
Matas 22	Įrašytas žodis <i>Mokiniai</i>.
Meda 35	- Pateiktas mokinių, kurie surinko ne mažiau kaip 20 valandų, sąrašas: mokinio vardas; surinktų valandų skaičius. - Socialinės-pilietinės veiklos surikiuotos pagal veiklos procentų skaičių mažėjančiai, o mokinių sąrašas surikiuotas pagal mokinio vardą abėcėliškai. - Duomenys visose eilutėse atskirti vienu tarpo simboliu.

Egzamino užduoties vertinimo instrukcija publikuojama: https://www.nsa.smsm.lt/old/wp-content/uploads/2025/07/2025_INF_VBE_PAGR-vertinimo-instrukcija.pdf

Testų, kuriuos naudojant buvo vertinami kandidatų darbai, rinkinys

Duomenys	Rezultatai	Ką tikrina
5 Jonas 3 Pil. 2 2 3 Soc. 1 3 Kul. 3 1 1 1 Kajus 2 Spo. 3 1 1 1 Kit. 1 2 Meda 4 Dar. 4 4 5 6 4 Soc. 2 4 4 Pil. 1 4 Spo. 1 4 Matas 2 Kit. 5 4 3 3 2 3 Dar. 2 3 4 Milda 1 Kul. 4 5 3 2 6	Statistika Dar. 29.21 % Kul. 21.35 % Kit. 19.10 % Soc. 12.36 % Pil. 10.11 % Spo. 7.87 % Mokiniai Matas 22 Meda 35	1. Sąlygos testas.
5 Jonas 4 Dar. 4 3 3 5 5 Eko. 2 3 3 Soc. 3 4 4 4 Kit. 3 1 1 1 Kajus 3 Kul. 3 3 4 4 Spo. 2 4 3 Soc. 1 3 Matas 2 Spo. 5 4 3 3 2 3 Pil. 2 3 4 Meda 4 Pil. 4 4 5 6 4 Eko. 3 5 5 5 Soc. 2 6 6 Kul. 3 4 4 4	Statistika Pil. 29.56 % Soc. 16.98 % Kul. 14.47 % Spo. 13.84 % Eko. 13.21 % Dar. 10.06 % Kit. 1.89 % Mokiniai Jonas 37 Kajus 21 Matas 22 Meda 58 Milda 21	2. Visos veiklos vykdomos. Visi mokiniai surinko virš 20 val. Netikrina rikiavimo.

Milda 1 Pil. 4 5 5 5 6		
5 Jonas 4 Kul. 4 3 3 5 5 Spo. 2 3 3 Soc. 3 4 4 4 Kit. 3 1 1 1 Kajus 3 Dar. 3 3 4 4 Eko. 2 4 3 Soc. 1 3 Matas 2 Eko. 5 4 3 3 2 3 Pil. 2 3 4 Meda 4 Pil. 4 4 5 6 4 Spo. 3 5 5 5 Soc. 2 6 6 Kul. 3 4 4 4 Milda 1 Kit. 4 5 5 5 6	Statistika Kul. 17.61 % Soc. 16.98 % Pil. 16.35 % Kit. 15.09 % Eko. 13.84 % Spo. 13.21 % Dar. 6.92 % Mokiniai Jonas 37 Kajus 21 Matas 22 Meda 58 Milda 21	3. Visos veiklos vykdomos. Visi mokiniai surinko virš 20 val. Tikrina veiklų rikiavimą.
5 Milda 1 Pil. 4 5 5 5 6 Matas 2 Spo. 5 4 3 3 2 3 Pil. 2 3 4 Kajus 3 Kul. 3 3 4 4 Spo. 2 4 3 Soc. 1 3 Meda 4 Pil. 4 4 5 6 4 Eko. 3 5 5 5 Soc. 2 6 6 Kul. 3 4 4 4 Jonas 4 Dar. 4 3 3 5 5	Statistika Pil. 29.56 % Soc. 16.98 % Kul. 14.47 % Spo. 13.84 % Eko. 13.21 % Dar. 10.06 % Kit. 1.89 % Mokiniai Jonas 37 Kajus 21 Matas 22 Meda 58 Milda 21	4. Visos veiklos vykdomos. Visi mokiniai surinko virš 20 val. Tikrina mokinių rikiavimą.

Eko. 2 3 3 Soc. 3 4 4 4 Kit. 3 1 1 1		
5 Matas 2 Spo. 5 4 3 3 3 3 Pil. 2 3 4 Jonas 2 Dar. 4 3 3 5 5 Eko. 2 3 2 Meda 3 Pil. 4 4 5 6 4 Eko. 3 6 6 5 Kul. 3 4 4 4 Milda 1 Pil. 4 5 5 5 6 Kajus 2 Kul. 3 4 4 5 Spo. 2 4 4	Statistika Pil. 35.07 % Kul. 18.66 % Spo. 17.91 % Eko. 16.42 % Dar. 11.94 % Mokiniai Jonas 21 Kajus 21 Matas 23 Meda 48 Milda 21	5. Yra nevykdomų veiklų. Visi mokiniai surinko virš 20 val. Tikrina rikiavimą.
1 Milda 1 Spo. 4 5 5 5 6	Statistika Spo. 100.00 % Mokiniai Milda 21	6. Viena veikla. Viena mokiny.
6 Jonas 4 Eko. 4 3 3 5 5 Kul. 2 3 3 Soc. 3 4 4 4 Kit. 3 1 1 1 Kajus 3 Kul. 3 1 1 4 Spo. 2 3 3 Soc. 1 4 Matas 2 Spo. 5 4 3 3 2 3 Pil. 2 3 2 Meda 4 Pil. 4 4 5 6 4 Dar. 3 5 5 5	Statistika Pil. 29.49 % Soc. 17.95 % Kul. 17.31 % Spo. 13.46 % Eko. 10.26 % Dar. 9.62 % Kit. 1.92 % Mokiniai Jonas 37 Matas 20 Meda 61 Milda 21	7. Visos veiklos vykdomos. Yra mokinių, nesurinkusių 20 val. Netikrina rikiavimo. * * Jei neina 7 testas, atidžiai žiūrėti, gal spręsta nepataisyta sąlyga.

Soc. 2 6 6 Kul. 3 5 5 5 Milda 1 Pil. 4 5 5 5 6 Simas 1 Pil. 1 1		
--	--	--

Mokinių darbų pavyzdžiai

Programos, įveikusios visus testus

1 pavyzdys

```
#include <bits/stdc++.h>

using namespace std;

const char CD[] = "Duomenys.txt";
const char CR[] = "Rezultatai.txt";

struct Mokiniai{
    string vard;
    int val;
};

struct Veiklos{
    double procentai;
    string dalykai;
    int val;
};

void Pildyti(vector <Veiklos> & V);
void Skaityti(vector <Veiklos> & V, vector <Mokiniai> & M);
void Skaiciuoti(vector <Veiklos> & V, vector <Mokiniai> & M);
bool Compare(Veiklos v1, Veiklos v2);
bool Compare1(Mokiniai m1, Mokiniai m2);
void Rezult(vector <Veiklos> & V, vector <Mokiniai> & M);

int main()
{
    vector <Veiklos> V;
    vector <Mokiniai> M;

    Pildyti(V);
    Skaityti(V, M);
    Skaiciuoti(V, M);
    sort(V.begin(), V.end(), Compare);
    sort(M.begin(), M.end(), Compare1);
    Rezult(V, M);

    return 0;
}

void Pildyti(vector <Veiklos> & V)
{
    Veiklos v;
    v.dalykai = "Pil.";
    v.val = 0;
```

```

        v.procentai = 0;
        V.push_back(v);
        v.dalykai = "Soc.";
        v.val = 0;
        v.procentai = 0;
        V.push_back(v);
        v.dalykai = "Kul.";
        v.val = 0;
        v.procentai = 0;
        V.push_back(v);
        v.dalykai = "Spo.";
        v.val = 0;
        v.procentai = 0;
        V.push_back(v);
        v.dalykai = "Eko.";
        v.val = 0;
        v.procentai = 0;
        V.push_back(v);
        v.dalykai = "Dar.";
        v.val = 0;
        v.procentai = 0;
        V.push_back(v);
        v.dalykai = "Kit.";
        v.val = 0;
        v.procentai = 0;
        V.push_back(v);
    }

void Skaityti(vector <Veiklos> & V, vector <Mokiniai> & M)
{
    ifstream fd(CD);

    string dalyk;
    int kd = 0, kdd = 0, k = 0, n;
    Mokiniai m;

    fd >> n;

    fd.ignore();

    for(int i = 0 ; i < n; i++)
    {
        string eil;
        getline(fd, eil);
        stringstream ss(eil);
        getline(ss, m.vard, ' ');
        ss >> kd;
        m.val = 0;
        for(int j = 0; j < kd; j++)
        {
            fd >> dalyk >> kdd;

```

```

        for(int r = 0; r < kdd; r++)
        {
            fd >> k;
            m.val += k;
            for(int d = 0; d < V.size(); d++)
            {
                if(V[d].dalykai == dalyk)
                {
                    V[d].val += k;
                }
            }
        }
        M.push_back(m);
        fd.ignore();
    }

    fd.close();
}

void Skaiciuoti(vector <Veiklos> & V, vector <Mokiniai> & M)
{
    int sumv = 0;

    for(int i = 0; i < M.size(); i++)
    {
        sumv += M[i].val;
    }

    for(int i = 0; i < V.size(); i++)
    {
        V[i].procentai = V[i].val*100.0/sumv;
    }
}

bool Compare(Veiklos v1, Veiklos v2)
{
    return v1.procentai > v2.procentai;
}

bool Compare1(Mokiniai m1, Mokiniai m2)
{
    return m1.vard < m2.vard;
}

void Rezult(vector <Veiklos> & V, vector <Mokiniai> & M)
{
    ofstream fr(CR);

```

```

fr << "Statistika" << endl;

for(int i = 0; i < V.size(); i++)
{
    if(V[i].procentai > 0)
    {
        fr << V[i].dalykai << " " << fixed << setprecision(2)
<< V[i].procentai << " %" << endl;
    }

}

fr << "Mokiniai" << endl;

for(int i = 0; i < M.size(); i++)
{
    if(M[i].val >= 20)
    {
        fr << M[i].vard << " " << M[i].val << endl;
    }

}

fr.close();
}

```

2 pavyzdys

```

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

struct Mokinyys
{
    string vardas;    // mokinio vardas
    int kiekis;       // kiek mokinyys turėjo veiki, kuriose rinko
valandas
    string Veiklos[7]; // veiklų pavadinimai
    int Valandos[7];   // kiek kiekviena mokinio veikla mokinio
turėjo iš viso surinkta valandų
    int valand[7];     // kiek kartu dalyvavo mokinyys veikloje
    double suma;       // kiek iš viso valandų surinko mokinyys
};

struct Veikla
{
    string pavadinimas; // veiklos pavadinimas
    double procentai;    // procentaliai išreikšta dalis kuria
visi mokiniai vykdė šią veiklą
    double suma;         // kiek bendrai valandų iš visų mokinių yra
veikloje
};
//-----

```

```

void Skaitymas( int & n, Mokinys M[]); // funkcija, skirta
nuskaityti duomenis
void Atrinkti( int n, Mokinys M[], int & m, Veikla V[]);
//funkcija, skirta atrinkti tuos pacius veiklos pavadinimus ir
sudeti valandas bendrai
int Yra( int m, Veikla V[], string pav);
void Procentai( int m, Veikla V[]); //funkcija, skaiciuojanti
procentus kiekvienai veiklai, kokia valandu dalis buvo skirta
veikla tarp visu veiklu
void Rikiavimas( int n, Mokinys M[], int m, Veikla V[]);
//rikiuojami vaiku vardai abeceliskai ir procentai mazejanciai
void Rasymas( int m, Veikla V[], int n, Mokinys M[]); //rasymas
//-----
int main()
{
    int n; Mokinys M[25];
    int m; Veikla V[25];

    Skaitymas( n, M);
    Atrinkti( n, M, m, V);
    Procentai( m, V);
    Rasymas( m, V, n, M);
    return 0;
}
//-----
void Skaitymas( int & n, Mokinys M[])
{
    ifstream fd("Duomenys.txt");
    int val;

    fd >> n >> ws;
    for( int i = 0; i < n; i++)
    {
        fd >> M[i].vardas >> M[i].kiekis;
        for( int j = 0; j < M[i].kiekis; j++)
        {
            double suma = 0.0;
            fd >> M[i].Veiklos[j] >> M[i].valand[j];
            for( int z = 0; z < M[i].valand[j]; z++)
            {
                fd >> val;
                suma += val;
            }
            M[i].Valandos[j] = suma;
            M[i].suma += M[i].Valandos[j];
            fd >> ws;
        }
    }
}
//-----

```

```

void Atrinkti( int n, Mokinyys M[], int & m, Veikla V[])
{
    m = 0;
    for( int i = 0; i < n; i++)
    {
        for( int j = 0; j < M[i].kiekis; j++)
        {
            int k = Yra( m, V, M[i].Veiklos[j]);
            if( k < 0)
            {
                V[m].pavadinimas = M[i].Veiklos[j];
                V[m].suma = M[i].Valandos[j];
                m++;
            }
            else if( k >= 0)
            {
                V[k].suma += M[i].Valandos[j];
            }
        }
    }
}

//-----
int Yra( int m, Veikla V[], string pav)
{
    int x = -1;
    for( int i = 0; i < m; i++)
    {
        if( pav == V[i].pavadinimas)
            x = i;
    }
    return x;
}

//-----
void Procentai( int m, Veikla V[])
{
    double visi = 0;
    for( int i = 0; i < m; i++)
    {
        visi += V[i].suma;
    }
    for( int i = 0; i < m; i++)
    {
        V[i].procentai = ((V[i].suma / visi)*100);
    }
}

//-----
void Rikiavimas( int n, Mokinyys M[], int m, Veikla V[])
{
    for( int i = 0; i < m-1 ; i++)
        for( int j = i+1 ; j < m; j++)

```

```

        if( V[i].procentai < V[j].procentai) swap( V[i], V[j]);

    for( int i = 0; i < n-1 ; i++)
        for( int j = i+1 ; j < n; j++)
            if( M[i].vardas > M[j].vardas) swap( M[i], M[j]);
}
//-----
void Rasymas( int m, Veikla V[], int n, Mokiny M[])
{
    ofstream fr("Rezultatai.txt");
    Rikiavimas( n, M, m, V);

    fr << "Statistika" << endl;
    for( int i = 0; i < m; i++)
    {
        fr << fixed << V[i].pavadinimas << " " <<
setprecision(2) << V[i].procentai << " %" << endl;
    }

    fr << "Mokiniai" << endl;
    for( int i = 0; i < n; i++)
    {
        if( M[i].suma >= 20)
            fr << fixed << M[i].vardas << " " << setprecision(0) <<
M[i].suma << endl;
    }
}

```

3 pavyzdys

```

#include<iostream>
#include<fstream>
#include<iomanip>
#include<vector>
using namespace std;
ifstream in("Duomenys.txt");
ofstream out("Rezultatai.txt");

struct tinkamimokiniai{
string vard;
int valsk=0;
};

struct veikla{
string pav;
int val=0;
float procentai;
};

void rikiavimas(vector<tinkamimokiniai> &m, vector<veikla> &v);

```

```

int main(){
int n;
float suma=0;
in>>n;
bool nauja=true;
vector<veikla> v;
vector<tinkamimokiniai> m;
tinkamimokiniai tuscia;
veikla tuscias;//nariai, kurių reiks naudojant push_back
for(int i=0; i<n; i++){

    int ln;
    m.push_back(tuscia);
    in>>m[m.size()-1].vard>>ln;
    vector<veikla>laikinas;

    for(int u=0; u<ln; u++){
        int llm;
        laikinas.push_back(tuscias);
        in>>laikinas[u].pav>>llm;

        for(int q=0; q<llm; q++){
            int sk;
            in>>sk;
            laikinas[u].val+=sk;
            m[m.size()-1].valsk+=sk;
            suma+=sk;}
        }//nuskaitomi ir įrašomi duomenys į laikiną vektorinį
        struktūros masivą

        for(int ii=0; ii<laikinas.size(); ii++){
            for(int u=0; u<v.size(); u++){
                if(laikinas[ii].pav==v[u].pav){
                    v[u].val+=laikinas[ii].val;
                    nauja=false;
                    break;
                }
            }
            if(nauja)v.push_back(laikinas[ii]);
            nauja=true;//jei iki tol niekas nevykdė tokio dalyko
        }
        if(m[m.size()-1].valsk<20) m.pop_back();//panaikinam mokinį
        iš sąrašo, jei jis nesurinko 20 val

    }

    for(int i=0; i<v.size(); i++)v[i].procentai=v[i].val/suma;//surandama, kuria procentinę
    dalį sudaro

```

```

rikiavimas(m, v);

out<<"Statistika"<<endl;
for(int i=0; i<v.size(); i++) out<<v[i].pav<<"
"<<fixed<<setprecision(2)<<v[i].procentai*100<<" %"<<endl;

out<<"Mokiniai"<<endl;
for(int i=0; i<m.size(); i++) out<<m[i].vard<<"
"<<m[i].valsk<<endl;
//spausdinimas į failą
}

void rikiavimas(vector<tinkamimokiniai> &m, vector<veikla> &v){
for(int i=0; i<v.size(); i++){
    for(int u=i; u<v.size(); u++){
        if(v[i].procentai<v[u].procentai) swap(v[i], v[u]);
    }
}
for(int i=0; i<m.size(); i++){
    for(int u=i; u<m.size(); u++){
        if(m[i].vard>m[u].vard) swap(m[i], m[u]);
    }
}
}

```

4 pavyzdys

```

#include <iostream>
#include <fstream>
#include <iomanip>
using namespace std;

struct Mokinys {

string vardas;
string veikla;
int vsk; // veiklų skaičius
int vk; //kiek kartų vykdoma veikla
int viso=0; // visos veiklos trukmė

};

struct Veikla {

string pav; // veiklos pavadinimas
double proc; // procentų skaičius, nurodantis kuria trukmės dali
sudarė veikla

};

```

```

void Rikiavimas(Mokinys mok[], int &m, Veikla veik[]);

int main() {
int m;
int pil=0, soc=0, kul=0, spo=0, eko=0, dar=0, kit=0; // veiklų
valandų skaičius
double vval=0; // visų vykdytų veiklų trukmė

ifstream fd("Duomenys.txt");
ofstream fr("Rezultatai.txt");
fd >> m;

Mokinys mok[m];
Veikla veik[7];

// duomenų nuskaitymas ir veiklų valandų kiekio bei mokinio
valandų kiekio apskaičiavimas
for(int i=0; i<m; i++) {
    fd >> mok[i].vardas >> mok[i].vsk;
    for(int j=0; j<mok[i].vsk; j++) {
        fd >> mok[i].veikla >> mok[i].vk;
        for(int u=0; u<mok[i].vk; u++) {
            int val;
            fd >> val;
            mok[i].viso+=val;
            if(mok[i].veikla=="Pil.") {
                pil+=val;
            }
            if(mok[i].veikla=="Soc.") {
                soc+=val;
            }
            if(mok[i].veikla=="Kul.") {
                kul+=val;
            }
            if(mok[i].veikla=="Spo.") {
                spo+=val;
            }
            if(mok[i].veikla=="Eko.") {
                eko+=val;
            }
            if(mok[i].veikla=="Dar.") {
                dar+=val;
            }
            if(mok[i].veikla=="Kit.") {
                kit+=val;
            }
        }
    }
    vval+=mok[i].viso;
}

```

```

}

fd.close();

veik[0].pav="Pil.";
veik[1].pav="Soc.";
veik[2].pav="Kul.";
veik[3].pav="Spo.";
veik[4].pav="Eko.";
veik[5].pav="Dar.";
veik[6].pav="Kit.";

// veiklų procentų skaičiavimas
veik[0].proc=(100*pil)/vval;
veik[1].proc=(100*soc)/vval;
veik[2].proc=(100*kul)/vval;
veik[3].proc=(100*spo)/vval;
veik[4].proc=(100*eko)/vval;
veik[5].proc=(100*dar)/vval;
veik[6].proc=(100*kit)/vval;

Rikiavimas(mok, m, veik);

// rezultatų pateikimas
fr << "Statistika" << endl;

for(int i=0; i<7; i++) {
    if (veik[i].proc>0) {
        fr << fixed << setprecision(2) << veik[i].pav << " " <<
veik[i].proc << " %" << endl;
    }
}

fr << "Mokiniai" << endl;

for(int i=0; i<m; i++) {
    if(mok[i].viso>=20) {
        fr << mok[i].vardas << " " << mok[i].viso << endl;
    }
}

fr.close();
return 0;
}

void Rikiavimas(Mokinsys mok[], int &m, Veikla veik[]) {

// mokinių rikiavimas abecėliškai
for(int i=0; i<m; i++) {
    for(int j=0; j<m-1; j++) {
        if(mok[j].vardas > mok[j+1].vardas) {

```

```

        swap(mok[j], mok[j+1]);
    }
}

// veiklų rikiavimas pagal procentus mažėjančiai
for(int i=0; i<7; i++) {
    for(int j=0; j<6; j++) {
        if(veik[j].proc < veik[j+1].proc) {
            swap(veik[j], veik[j+1]);
        }
    }
}

```

5 pavyzdys

```

#include <iostream>
#include <fstream>
#include <iomanip>

using namespace std;

struct mokinys
{
    string vardas;
    int val;
};

struct veikla
{
    string pavadinimas;
    int val;
};

void sort_data(mokinys *mokiniai, int n, veikla *veiklos)
{
    //simple sorting

```

```

for (int i = 0; i < n; i++)
{
    for (int j = 0; j < i; j++)
    {
        if (mokiniai[i].vardas < mokiniai[j].vardas)
        {
            //variable swap using temporary variables
            string vardas = mokiniai[i].vardas;
            int val = mokiniai[i].val;
            mokiniai[i].vardas = mokiniai[j].vardas;
            mokiniai[i].val = mokiniai[j].val;
            mokiniai[j].vardas = vardas;
            mokiniai[j].val = val;
        }
    }
}

for (int i = 0; i < 7; i++)
{
    for (int j = 0; j < i; j++)
    {
        if (veiklos[i].val > veiklos[j].val)
        {
            //variable swap using temporary variables
            string pavadinimas = veiklos[i].pavadinimas;
            int val = veiklos[i].val;
            veiklos[i].pavadinimas =
veiklos[j].pavadinimas;
            veiklos[i].val = veiklos[j].val;
            veiklos[j].pavadinimas = pavadinimas;
            veiklos[j].val = val;
        }
    }
}

```

```

        }
    }

    return;
}

int main()
{
    ifstream duom("Duomenys.txt");
    ofstream rez("Rezultatai.txt");

    //setting up an array
    veikla *veiklos = new veikla[7];
    for (int i = 0; i < 7; i++)
    {
        veiklos[i].val = 0;
    }
    veiklos[0].pavadinimas = "Pil.";
    veiklos[1].pavadinimas = "Soc.";
    veiklos[2].pavadinimas = "Kul.";
    veiklos[3].pavadinimas = "Spo.";
    veiklos[4].pavadinimas = "Eko.";
    veiklos[5].pavadinimas = "Dar.";
    veiklos[6].pavadinimas = "Kit.";

    //reading inputs
    int n; duom >> n;
    mokiny *mokiniai = new mokiny[n];
    for (int i = 0; i < n; i++)
    {
        duom >> mokiniai[i].vardas;
    }
}

```

```

    mokiniai[i].val = 0;
    int m; duom >> m;
    for (int j = 0; j < m; j++)
    {
        int tmp1 = 0;
        string v; duom >> v;
        int c; duom >> c;
        for (int k = 0; k < c; k++)
        {
            int tmp2; duom >> tmp2;
            tmp1 += tmp2;
        }
        for (int k = 0; k < 7; k++)
        {
            if (v == veiklos[k].pavadinimas)
            {
                veiklos[k].val += tmp1;
            }
        }
        mokiniai[i].val += tmp1;
    }
}

```

```

//calling sort function

```

```

sort_data(mokiniai, n, veiklos);

```

```

// sum for calculating percentages

```

```

float veiklos_sum = 0;

```

```

for (int i = 0; i < 7; i++)

```

```

{
    veiklos_sum += veiklos[i].val;
}

```

```

    }

    //output
    rez << "Statistika" << endl;
    for (int i = 0; i < 7; i++)
    {
        if (veiklos[i].val == 0)
        {
            continue;
        }

        rez << veiklos[i].pavadinimas << " " << fixed <<
setprecision(2) << 100 * veiklos[i].val / veiklos_sum << " %"
<< endl;
    }
    rez << "Mokiniai" << endl;
    for (int i = 0; i < n; i++)
    {
        if (mokiniai[i].val < 20)
        {
            continue;
        }

        rez << mokiniai[i].vardas << " " << mokiniai[i].val <<
endl;
    }

    return 0;
}

```

6 pavyzdys

```

class Student():
    def __init__(self, Name):
        self.Name = Name
        self.TotalHours = 0
        self.Activities = []

```

```

class Activity():
    def __init__(self, Name, Hours):
        self.Name = Name
        self.Hours = sum([int(H) for H in Hours])
        self.SummedTime = 0
        self.Percentage = 0.00

def SortStudentsAndActivityData(Students, ActivityData):
    # Nurodymas parašytas neteisingai, jei norite rikiavimo
    funkcijos.
    Students = sorted(Students, key = lambda x: x.Name)
    ActivityData = sorted(ActivityData, key = lambda x:
x.Percentage, reverse = True)
    return Students, ActivityData

Students = []
ActivityData = []

TotalStudentHours = 0

# Duomenų nuskaitymas ir sutvarkymas.
with open("Duomenys.txt", "r") as f:
    n = int(f.readline())

    for i in range(n):
        # Mokinio vardas ir veiklų skaičius nuskaitymas į
        vienkartinį [Neprisimenu kaip "variable" yra lietuviškai],
        sukuriamas mokinio objektas.
        tmp = f.readline().split()
        Students.append(Student(tmp[0]))

        # Nuskaitymi ir sutvarkomi veiklų duomenys.
        for x in range(int(tmp[1])):
            tmp = f.readline().split()
            Students[i].Activities.append(Activity(tmp[0],
tmp[2:]))

            tmp = Students[i].Activities[x]
            Students[i].TotalHours += tmp.Hours

        # Šitą jovalą baisu net aiškinti.
        if len(ActivityData) > 0:
            HasActivity = False
            for y in ActivityData:
                if y.Name == tmp.Name:
                    y.SummedTime += tmp.Hours
                    HasActivity = True
                    break
            if HasActivity == False:
                tmp.SummedTime += tmp.Hours
                ActivityData.append(tmp)

```

```

        else:
            tmp.SummedTime += tmp.Hours
            ActivityData.append(tmp)

for i in Students:
    TotalStudentHours += i.TotalHours

for i in ActivityData:
    i.Percentage = ((i.SummedTime / TotalStudentHours) * 100)

Students, ActivityData = SortStudentsAndActivityData(Students,
ActivityData)

with open("Rezultatai.txt", "w") as f:
    f.write("Statistika\n")

    for i in ActivityData:
        tmp = "%.2f" % i.Percentage
        f.write(f"{i.Name} {tmp} %\n")

    f.write("Mokiniai\n")

    for i in Students:
        if i.TotalHours >= 20:
            f.write(f"{i.Name} {i.TotalHours}\n")

```

7 pavyzdys

```

def skaiciuoti(veikluStats, duomenys):
    veikluStats = sorted(veikluStats, key = lambda x: -
x["procDalis"])
    duomenys = sorted(duomenys, key=lambda x:(x["vardas"]))
    return veikluStats, duomenys
with open("Duomenys.txt", "r") as file:
    duomenys = []
    veiklos = ["Pil.", "Soc.", "Kul.", "Spo.", "Eko.", "Dar.",
"Kit."]
    veikluStats = []
    bendrasValanduSk = int(0)
    mokSK = int(file.readline())
    for i in range(mokSK):
        sarasas = {
            "vardas": "",
            "mokNR": i,
            "Pil.": int(0),
            "Soc.": int(0),
            "Kul.": int(0),
            "Spo.": int(0),
            "Eko.": int(0),
            "Dar.": int(0),
            "Kit.": int(0),

```

```

        "MokValSK": int(0)
    }
    duomenys.append(sarasas)

for i in range(mokSK):
    o = list(map(str, file.readline().split()))
    vardas = o[0]
    veikluNR = int(o[-1])
    for ii in range(veikluNR):
        y = str(file.readline())
        VALSUB = []
        pav = y[:4]
        VALSUB = list(map(int, y[7:].split()))
        bendrasValanduSk += sum(VALSUB)
        for x in duomenys:
            if(x["mokNR"] == i):
                x["vardas"] = vardas
                x["MokValSK"] += sum(VALSUB)
                if(pav == "Pil."):
                    x["Pil."] = sum(VALSUB)
                if(pav == "Soc."):
                    x["Soc."] = sum(VALSUB)
                if(pav == "Kul."):
                    x["Kul."] = sum(VALSUB)
                if(pav == "Spo."):
                    x["Spo."] = sum(VALSUB)
                if(pav == "Eko."):
                    x["Eko."] = sum(VALSUB)
                if(pav == "Dar."):
                    x["Dar."] = sum(VALSUB)
                if(pav == "Kit."):
                    x["Kit."] = sum(VALSUB)

for i in range(len(veiklos)):
    veik = {
        "pav": veiklos[i],
        "BendValanduSK": bendrasValanduSk,
        "DalykoSK": int(0),
        "procDalis": int(0)
    }
    veikluStats.append(veik)

for y in veikluStats:
    for x in duomenys:
        if(y["pav"] == "Pil."):
            y["DalykoSK"] += x["Pil."]
        if(y["pav"] == "Soc."):
            y["DalykoSK"] += x["Soc."]
        if(y["pav"] == "Kul."):
            y["DalykoSK"] += x["Kul."]

```

```

        if(y["pav"] == "Spo."):
            y["DalykoSK"] += x["Spo."]
        if(y["pav"] == "Eko."):
            y["DalykoSK"] += x["Eko."]
        if(y["pav"] == "Dar."):
            y["DalykoSK"] += x["Dar."]
        if(y["pav"] == "Kit."):
            y["DalykoSK"] += x["Kit."]

    for y in veikluStats:
        y["procDalis"] = y["DalykoSK"]/y["BendValanduSK"]*100

    veikluStats, duomenys = skaiciuoti(veikluStats, duomenys)
    file.close()
    with open("Rezultatai.txt", "w") as flie:
        flie.write(f'Statistika\n')
        for y in veikluStats:
            if(y["procDalis"]>0):
                flie.write(f'{y["pav"]} {y["procDalis"]:.2f} %\n')
        flie.write(f'Mokiniai\n')
        for x in duomenys:
            if(x["MokValSK"] > 19):
                flie.write(f'{x["vardas"]} {x["MokValSK"]}\n')
    flie.close()

```

8 pavyzdys

```

def rusiavimas(listas1,listas2,c): #c kintamasis kad atskirti
ko yra norima, didejimo ar mazejimo tvarka surusiuoti.
    if c==1: # mazejimo tvarka
        for i in range(len(listas1)-1):
            for j in range(i+1,len(listas2)):
                if listas1[i]<listas1[j]:

listas1[i],listas1[j]=listas1[j],listas1[i]

listas2[i],listas2[j]=listas2[j],listas2[i]
    if c==2: #Pagal abecele, t.y. didejimo tvarka
        for i in range(len(listas1)-1):
            for j in range(i+1,len(listas2)):
                if listas1[i]>listas1[j]:

listas1[i],listas1[j]=listas1[j],listas1[i]

listas2[i],listas2[j]=listas2[j],listas2[i]
    return listas1,listas2
with open ("duomenys.txt", "r") as f:
    m=f.readline() #mokiniu skaicius

```

```

m=int(m)
visi=[] #listas duomenim kaupiti
for x in range(m):
    vardas,n=f.readline().split()#nusiskaitau varda ir n -
vykdytu skirtingu soc.-pil. veiklu sk.
    n=int(n)
    d={
        "vardas": vardas,
        "bendrai":0,
        "Pil.":0,
        "Soc.":0,
        "Kul.":0,
        "Spo.":0,
        "Eko.":0,
        "Dar.":0,
        "Kit.":0
    }
    for y in range(n): #for tam, kad nusiskaityti veiklu
duomenis,t.y. valandas

        eil=f.readline() #eiluteje gali buti ivairus kiekis
duomenu todel skaitau ja visa
        pav=eil[:4]
        kiekis=eil[5]
        valandos=list(map(int,eil[7:].split()))
        sumaval=sum(valandos)
        if pav=="Pil.":
            d["Pil."]+=sumaval
        if pav=="Soc.":
            d["Soc."]+=sumaval
        if pav=="Kul.":
            d["Kul."]+=sumaval
        if pav=="Spo.":
            d["Spo."]+=sumaval
        if pav=="Eko.":
            d["Eko."]+=sumaval
        if pav=="Dar.":
            d["Dar."]+=sumaval
        if pav=="Kit.":
            d["Kit."]+=sumaval
        d["bendrai]+=sumaval
    visi.append(d)
o={ #bendras visu veiklu valandu skaicius
    "Pil.":0,
    "Soc.":0,
    "Kul.":0,
    "Spo.":0,
    "Eko.":0,
    "Dar.":0,
    "Kit.":0,
    "bendrai":0

```

```

    }
    for x in visi:
        o["Pil."]+=x["Pil."]
        o["Soc."]+=x["Soc."]
        o["Kul."]+=x["Kul."]
        o["Spo."]+=x["Spo."]
        o["Eko."]+=x["Eko."]
        o["Dar."]+=x["Dar."]
        o["Kit."]+=x["Kit."]
        o["bendrai"]+=x["bendrai"]
    #susidarau listus, kad tiktu funkcijai
    o["Pil."]=o["Pil."]/o["bendrai"]
    o["Soc."]=o["Soc."]/o["bendrai"]
    o["Spo."]=o["Spo."]/o["bendrai"]
    o["Eko."]=o["Eko."]/o["bendrai"]
    o["Dar."]=o["Dar."]/o["bendrai"]
    o["Kit."]=o["Kit."]/o["bendrai"]
    o["Kul."]=o["Kul."]/o["bendrai"]

    procentai=[]
    procentai.append(o["Pil."])
    procentai.append(o["Soc."])
    procentai.append(o["Kul."])
    procentai.append(o["Spo."])
    procentai.append(o["Eko."])
    procentai.append(o["Dar."])
    procentai.append(o["Kit."])
    bendrai=o["bendrai"]
    kryptys=[] #pavadinimai siu kryptiu ir irgi susidedu i lista
kad galeciau rusiuoti veliau
    kryptys.append("Pil.")
    kryptys.append("Soc.")
    kryptys.append("Kul.")
    kryptys.append("Spo.")
    kryptys.append("Eko.")
    kryptys.append("Dar.")
    kryptys.append("Kit.")

    procentai,kryptys=rusiavimas(procentai,kryptys,1)
    #susidedu vardus ir jų bendras valandas į 2 listus kad galeciau
    rusiuoti
    vardai=[]
    bendrosval=[]
    for x in visi:
        vardai.append(x["vardas"])
        bendrosval.append(x["bendrai"])

    vardai,bendrosval=rusiavimas(vardai,bendrosval,2)
    with open ("rezultatai.txt", "w") as w:
        w.write(f'Statistika\n')

```

```

    for x in range(len(kryptys)):
        if procentai[x]!=0: #Isskiriui duomenis, kuriu nera
            w.write(f'{kryptys[x]}
{round(procentai[x]*100,2):.2f} %\n')
            w.write(f'Mokiniai\n')
            for x in range(len(vardai)):
                if bendrosval[x]>=20: #Iskiriui tik tuos kurie ne maziau
nei 20 val
                    w.write(f'{vardai[x]} {bendrosval[x]}\n')

```

9 pavyzdys

```

def rikiavimas(sarastas, rusiavimometodas): # Gaunu sarasa ir
metoda pagal kuri rusiuosiu savo saraša
    ilgis = len(sarastas)
    if rusiavimometodas == 1:
        for x in range(ilgis-1):
            for y in range(x+1, ilgis):
                if sarastas[x]["Procentas"] <
sarastas[y]["Procentas"]:
                    sarastas[x], sarastas[y] = sarastas[y],
sarastas[x]
            elif rusiavimometodas == 2:
                for x in range(ilgis-1):
                    for y in range(x+1, ilgis):
                        if sarastas[x]["Mokinys"] >
sarastas[y]["Mokinys"]:
                            sarastas[x], sarastas[y] = sarastas[y],
sarastas[x]
    return(sarastas)

with open("Duomenys.txt", "r", encoding="UTF-8") as file: #
Nusiskaitau failą
    mokiniai = []
    dalykai = []
    dalykulistas = ["Pil.", "Soc.", "Kul.", "Spo.", "Eko.",
"Dar.", "Kit."] # Pasirašau pradinį dalykų saraša
    for u in range(len(dalykulistas)):
        dalykas = {
            dalykulistas[u] : 0,
        }
        dalykai.append(dalykas)
    m = int(file.readline())
    for p in range(m): # Susirašau mokinio informaciją apie
dalykus
        vardas, veiklusk = file.readline().strip().split()
        mokinys = {
            "Mokinys" : vardas,
            "Veiklusk" : veiklusk,
            "Pil." : int(0),
            "Soc." : int(0),

```

```

        "Kul." : int(0),
        "Spo." : int(0),
        "Eko." : int(0),
        "Dar." : int(0),
        "Kit." : int(0),
        "Isviso" : int(0)
    }
    mokiniai.append(mokinys) # Mokinio informaciją įdedu į
sarašą
    for o in range(int(veiklusk)): # Suskaičiuoju kiek
mokinys iš turi pilietiško valandų.
        eilute = file.readline().strip().split()
        veikla, vykdytuveiklusk = eilute[0], eilute[1]
        eilute.pop(0)
        eilute.pop(0)
        hsk = 0
        for i in range(len(eilute)):
            hsk += int(eilute[i])
        mokiniai[p][veikla] = hsk
        mokiniai[p]["Isviso"] += hsk
        hsk = 0

isviso = 0
for t in range(len(dalykai)): # Skaičiuoju kiek iš viso dalykas
turi surinkęs valandų
    for r in range(m):
        dalykai[t][dalykulistas[t]] +=
mokiniai[r][dalykulistas[t]]
        isviso += dalykai[t][dalykulistas[t]]

for e in range(len(dalykai)): # Suskaičiuoju kokį procentą
dalyko valandosturi
    dalykai[e][dalykulistas[e]] =
dalykai[e][dalykulistas[e]]/isviso*100

naujassarasas = []
naujidalykai = []

for l in range(len(dalykulistas)): # Atsirenku tik tuos dalykus,
kurie turi daugiau valandų nei nulį
    if dalykai[l][dalykulistas[l]] > 0:
        naujassarasas.append(dalykai[l])
        naujidalykai.append(dalykulistas[l])

perrinktasmokiniai = []
for k in range(len(mokiniai)): # Perrenku mokinius, kurie turi
daugiau nei 20 valandų
    if mokiniai[k]["Isviso"] >= 20:
        perrinktasmokiniai.append(mokiniai[k])

statistikoslistas = []

```

```

for h in range(len(naujassarasas)): # Atskiriu dalykų statistiką
    ir susirašau į sąrašą

statistikoslistas.append(naujassarasas[h][naujidalykai[h]])

statistikos = []
for g in range(len(statistikoslistas)): # Pasidarau, jog dalykai
    būtų sujungti vienoje struktūroje su statistika
    statistika = {
        "Dalykas" : naujidalykai[g],
        "Procentas" : statistikoslistas[g]
    }
    statistikos.append(statistika)

surikiuotastatistika = rikiavimas(statistikos, 1)
surikiuotimok = rikiavimas(perrinktasmokiniai, 2)

with open("Rezultatai.txt", "w", encoding="UTF-8") as f:
    f.write(f'Statistika\n')
    for v in range(len(surikiuotastatistika)):
        f.write(f'{surikiuotastatistika[v]["Dalykas"]}\n')
    {surikiuotastatistika[v]["Procentas"]:.2f} %\n')
    f.write(f'Mokiniai\n')
    for b in range(len(surikiuotimok)):
        if b != len(surikiuotimok)-1:
            f.write(f'{surikiuotimok[b]["Mokinys"]}\n')
    {surikiuotimok[b]["Isviso"]}\n')
        else:
            f.write(f'{surikiuotimok[b]["Mokinys"]}\n')
    {surikiuotimok[b]["Isviso"]}\n')

```

10 pavyzdys

```

# Mokinio duomenų klasė
class mokinys:
    def __init__(self, vardas, veikla, trukme):
        self.vardas = vardas
        self.veikla = veikla
        self.trukme = trukme

# Veiklos duomenų klasė
class dalykai:
    def __init__(self, veikla, bendra_trukme):
        self.veikla = veikla
        self.bendra_trukme = bendra_trukme

# Funkcija skirta rikiuoti. sarasas - norimas surikiuoti
sąrašas; raktas - nurodo kokį sąrašą rikiuoja
def rikiavimas(sarasas, raktas):
    if raktas == "mokiniai":
        sarasas.sort(key=lambda x: x.vardas)

```

```

        elif raktas == "veiklos":
            sarasas.sort(key=lambda x: -x.bendra_trukme)

skaitymas = open("Duomenys.txt", "r")

kiek_mok = int(skaitymas.readline()) # Visų mokinių skaičius

mokiniai = [] # Klasės masyvas duomenims apie mokinius saugoti

# Skaitymas iš failo "Duomenys.txt" ir duomenų dėjimas į mokiniai
for i in range(kiek_mok):
    vardas, kiek_veikl = skaitymas.readline().split()

    veiklos = []
    trukmes = []

    for j in range(int(kiek_veikl)):
        eil = list(skaitymas.readline().split())
        veiklos.append(eil[0])
        trukme = 0

        for k in range(int(eil[1])):
            trukme += int(eil[k+2])
        trukmes.append(trukme)

    mokiniai.append(mokinys(vardas, veiklos, trukmes))

skaitymas.close()

mokiniai_visos_valandos = [] # Sąrašas saugoti duomenis apie
vaikus turinčius daugiau nei 20 valandų

# Randami vaikai su >20h
for i in range(kiek_mok):
    if sum(mokiniai[i].trukme) >= 20:
        mokiniai_visos_valandos.append(mokiniai[i])

isskirtines_veiklos:list = [] # Sąrašas su unikaliomis
veiklomis, kurias mokiniai atliko

# Iš mokiniai išrenkama unikalios veiklos ir surašoma į
isskirtines_veiklos
for i in range(len(mokiniai)):
    for n in range(len(mokiniai[i].veikla)):
        if len(isskirtines_veiklos) != 0:
            if mokiniai[i].veikla[n] not in
isskirtines_veiklos:

isskirtines_veiklos.append(mokiniai[i].veikla[n])

```

```

        else:
            isskirtines_veiklos.append(mokiniai[i].veikla[n])

veiklos_is_viso = [] # Klasės sąrašas skirtas saugoti duomenis
apie unikalės veiklas

# Perkeliame unikalio veiklų pavadinimai iš isskirtines_veiklos
į veiklos_is_viso
for i in range(len(isskirtines_veiklos)):
    veiklos_is_viso.append(dalykai(isskirtines_veiklos[i], 0))

# Paskaičiuojama kiekvienos unikalios veiklos visų mokinių
bendrai sudėtas atlikimo laikas
for i in range(len(isskirtines_veiklos)):
    for k in range(len(mokiniai)):
        for n in range(len(mokiniai[k].veikla)):
            if isskirtines_veiklos[i] == mokiniai[k].veikla[n]:
                veiklos_is_viso[i].bendra_trukme +=
mokiniai[k].trukme[n]

visu_dalyku_trukme = 0

# Randama bendra visų dalykų atliekamų visų vaikų bendra trukmė
for i in range(len(veiklos_is_viso)):
    visu_dalyku_trukme += veiklos_is_viso[i].bendra_trukme

# Randama kiekvienos veiklos procentinė atlikimo reikšmė
for i in range(len(veiklos_is_viso)):
    veiklos_is_viso[i].bendra_trukme =
round((float(veiklos_is_viso[i].bendra_trukme)/float(visu_daly
ku_trukme)*100), 2)

# Vykdoma rikiavimo funkcija
rikiavimas(veiklos_is_viso, "veiklos")
rikiavimas(mokiniai_visos_valandos, "mokiniai")

# Vyksta rašymas į "Rezultatai.txt" failą
rasyti = open("Rezultatai.txt", "w")
rasyti.write("Statistika\n")

# Surašomos veiklos su procentais
for i in range(len(veiklos_is_viso)):
    rasyti.write(f"{veiklos_is_viso[i].veikla}
{veiklos_is_viso[i].bendra_trukme:.2f} %\n")
rasyti.write("Mokiniai\n")

# Surašomi mokiniai surinkę >20h ir bendras valandų skaičius
for i in range(len(mokiniai_visos_valandos)):
    visa_trukme = 0
    # Suskaičiuojama bendras valandų skaičius
    for n in range(len(mokiniai_visos_valandos[i].trukme)):

```

```
        visa_trukme += mokiniai_visos_valandos[i].trukme[n]
    rasyti.write(f"{mokiniai_visos_valandos[i].vardas}
{visa_trukme}\n")

rasyti.close()
```

**Duomenų tyrybos užduoties
vertinimo aspektai, atsižvelgiant į
2025 m. informatikos VBE rezultatus**

Informatikos VBE II dalis

DT užduoties vertė
2025 m.

Mokymo(si) turinio sritys	Pasiekimų sritys						Užduoties taškai procentais
	Skaitmeninio turinio kūrimas (A)	Algoritmai ir programavimas (B)	Duomenų tyryba ir informacija (C)	Technologinių problemų sprendimas (D)	Virtualioji komunikacija ir bendradarbiavimas (E)	Saugus elgesys (F)	
Skaitmeninio turinio kūrimo mokymo(si) turinys							8
Algoritimų ir programavimo mokymo(si) turinys							54
Duomenų tyrybos ir informacijos mokymo(si) turinys							30
Technologinių problemų sprendimo							—

I. KLAUSIMAI IR STRUKTŪRINIAI KLAUSIMAI

Maksimalus vertinimas – 10 taškų

II. DUOMENŲ TYRYBOS PRAKTINĖ UŽDUOTIS

Maksimalus vertinimas – 20 taškų

III. PROGRAMAVIMO PRAKTINĖ UŽDUOTIS

Maksimalus vertinimas – 30 taškų

			8
	—	8	100

INFORMATIKA

Valstybinio brandos egzamino II dalies užduotis

Pagrindinė sesija

2025 m. birželio 12 d.

Trukmė – 3 val. (180 min.)

INFORMATIKA • Valstybinio brandos egzamino II dalies užduotis • Pagrindinė sesija

II. DUOMENŲ TYRYBOS PRAKTINĖ UŽDUOTIS

Maksimalus vertinimas – 20 taškų

Visa užduotis atliekama skaičiuokle.

INFORMATIKOS VALSTYBINIŲ BRANDOS EGZAMINŲ UŽDUOČIŲ APRAŠAS

Duomenų tyrybos praktinė užduotis.

Atliekant šią užduotį kompiuteriu, reikia (jei nenurodyta kitaip) pagal pateiktą sąlygą laikantis nurodymų atlikti pateiktų duomenų tyrybos veiksmus (pavyzdžiui, skaičiuoti, rikiuoti, atrinkti, grupuoti, pavaizduoti, analizuoti ir pan.) ir pateikti išvadas.

Kiekvienas teisingai atliktas veiksmas vertinamas 1–5 taškais.

Duomenų tyrybos užduočiai atlikti naudojama skaičiuoklė.

INFORMATIKOS BENDROJI PROGRAMA

Užduotis rengiama remiantis Programos *IV gimnazijos klasės mokymo(si) turiniu. Pagrindinio ugdymo* programos ir *III gimnazijos klasės* mokymo(si) turinys į užduotį įtraukiamas tik tiek, kiek būtina užduotims, parengtoms pagal IV gimnazijos klasės mokymo(si) turinį, atlikti.

31. Mokymo(si) turinys. IV gimnazijos klasė.

31.3. Duomenų tyrybos ir informacijos mokymo(si) turinys.

31.3.1. Informacijos (rezultatų) pateikimas. Mokomasi pateikti duomenis suvestinėmis lentelėmis, juos rikiuoti ir atrinkti pagal kelis kriterijus, grupuoti, skaičiuoti tarpines sumas, formuoti ataskaitas. Prisimenamos anksčiau išmoktos duomenų tyrybai naudotos funkcijos, diagramų braižymas (linijinės, stulpelinės, skritulinės, juostinės, histogramos, stačiakampės, sklaidos, kombinuotosios). Mokomasi apdoroti tekstinę informaciją naudojant darbui su tekstu skirtas funkcijas. Aptariami galimi sprendimai, formuluojamos išvados.

INFORMATIKOS BENDROJI PROGRAMA

Užduotis rengiama remiantis Programos *IV gimnazijos klasės mokymo(si) turiniu. Pagrindinio ugdymo* programos ir *III gimnazijos klasės* mokymo(si) turinys į užduotį įtraukiamas tik tiek, kiek būtina užduotims, parengtoms pagal IV gimnazijos klasės mokymo(si) turinį, atlikti.

30. Mokymo(si) turinys. III gimnazijos klasė.

30.3. Duomenų tyrybos ir informacijos mokymo(si) turinys.

.....

30.3.2. Duomenys apdorojami skaičiuokle naudojant kai kurias statistines, **matematinės**, logines, datos ir laiko, peržvalgų ir nuorodų funkcijas (pavyzdžiui, **apvalinimo**, apvalinimo iki mažesniojo, apvalinimo iki didesniojo, netuščių langelių skaičiavimo, **sąlyginio sumavimo**, sąlyginio langelių skaičiavimo, sąlyginio vidurkio, variantinės sąlygos, loginio IR, loginio ARBA, variantinės sąlygos sumavimo, variantinės sąlygos vidurkio, variantinės sąlygos langelių skaičiavimo, variantinės sąlygos minimumo (maksimumo), datos (metų, mėnesio, dienos, šiandienos), vertikaliosios paieškos), sujungimo operatorių &.

II. DUOMENŲ TYRYBOS PRAKTINĖ UŽDUOTIS

Maksimalus vertinimas – 20 taškų

Visa užduotis atliekama skaičiuokle.

Faile *Abiturientai.xlsx* (*Abiturientai.ods*), kurį sudaro 4 darbo lakštai, pateikti užduoties pradiniai duomenys.

Naudodamiesi pateiktais duomenimis ir laikydamiesi nurodymų, atlikite užduotis.

2025 m. informatikos egzamino II dalies užduočių sunkumas ir koreliacija su visa II egzamino dalies užduotimi

Užduotis	Sunkumas	Koreliacija su visa užduotimi
Testas	50,2 (vidutinio sunkumo)	0,51
Duomenų tyrybos užduotis	66,8 (lengva)	0,80
Programavimo užduotis	30,3 (sunki)	0,95

Formulės ir funkcijos

1. Lakšte *Duomenys* esančioje lentelėje pateiktas abiturientų, stojusių į I pakopos studijas (penkis Lietuvos universitetus) 2023–2024 mokslo metais, skaičius pagal apskritis ir savivaldybes. Naudodamiesi tinkamomis formulėmis ir funkcijomis, užpildykite toliau nurodytus lakšto *Duomenys* langelius. Visos jūsų sukurtos formulės turi būti laisvai kopijuojamos iš vieno langelio į kitus.
 - 1.1. Langelyje **D63** įrašykite formulę, kuri apskaičiuotų santykį tarp abiturientų, stojusių į Vilniaus universitetą (VU) ir visų abiturientų skaičiaus Lietuvoje. Apskaičiuotas santykis turi būti suapvalintas iki šimtųjų, nekeičiant langelio formato.
 - 1.2. Langelyje **K3** įrašykite formulę, kuri apskaičiuotų langelyje **J3** nurodytos apskrities abiturientų skaičių.
 - 1.3. Langelio **D63** formulę nukopijuokite į langelius **E63:H63**; langelio **K3** formulę nukopijuokite į langelius **K4:K12**.
 - 1.4. Išnagrinėkite lentelės *Gyventojų ir abiturientų duomenys savivaldybėse 2023 metais* duomenis ir nustatykite, ar yra susiję abiturientų skaičius ir gyventojų skaičius. Išvadą ir jos pagrindimą įrašykite langelyje **K14**.
- Pastaba.** Langeliuose **D63** ir **K3** įrašytose formulėse turi būti naudojamos tokios langelių koordinatės, kad atliekant **1.3** punktą formulių prasmė nesikeistų.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	2023–2024 mokslo metais stojusių į I pakopos studijas Lietuvoje abiturientų skaičius pagal savivaldybes (universiteto duomenys)									Gyventojų ir abiturientų duomenys savivaldybėse 2023 metais						
2	Apskritis	Savivaldybė	Abiturientų skaičius savivaldybėje	VU	LSMU	KTU	VDU	VILNIUS TECH		Apskritis	Abiturientų skaičius	Gyventojų skaičius apskrityje				
3	Šiaulių	Akmenės r. sav.	60	8	9	2	1	6		Tauragės		89951				
4	Alytaus	Alytaus m. sav.	329	81	8	33	14	43		Utenos		122712				
5	Alytaus	Alytaus r. sav.	24	8	7	3	0	1		Telšių		129863				
6	Utenos	Anykščių r. sav.	85	23	0	4	5	6		Alytaus		132911				
7	Kauno	Birštono sav.	25	3	3	3	1	1		Marijampolės		133335				
8	Panevėžio	Biržų r. sav.	73	15	1	4	2	9		Panevėžio		206966				
9	Alytaus	Druskininkų r. sav.	72	16	4	2	7	7		Šiaulių		266804				
10	Vilniaus	Elektrėnų sav.	20	1	1	1	1	1		1.1. Langelyje D63 įrašykite formulę, kuri apskaičiuotų santykį tarp abiturientų, stojusių į Vilniaus universitetą (VU) ir visų abiturientų skaičiaus Lietuvoje. Apskaičiuotas santykis turi būti suapvalintas iki šimtųjų, nekeičiant langelio formato. Išvada:						
11	Utenos	Ignalinos r. sav.	10	1	1	1	1	1								
12	Kauno	Jonavos r. sav.	10	1	1	1	1	1								
13	Šiaulių	Joniškio r. sav.	10	1	1	1	1	1								
14	Tauragės	Jurbarko r. sav.	90	15	5	14	7	4								
15	Kauno	Kaišiadorių r. sav.	102	26	2	22	13	3								
16	Marijampolės	Kalvarijos sav.	25	3	3	2	2	6								
17	Kauno	Kauno m. sav.	1939	200	165	36	244	28								
18	Kauno	Kauno r. sav.	269	15	17	50	63	3								
19	Marijampolės	Kazlų Rūdos sav.	31	4	4	4	2	2								
20	Kauno	Kėdainių r. sav.	189	27	15	21	15	7								
21	Šiaulių	Kelmės r. sav.	83	10	7	13	10	1								
22	Klaipėdos	Klaipėdos m. sav.	886	201	38	62	28	64								
23	Klaipėdos	Klaipėdos r. sav.	132	16	12	12	10	5								
24	Klaipėdos	Kretingos r. sav.	193	47	6	22	6	14								
25	Panevėžio	Kupiškio r. sav.	64	15	3	6	4	6								
26	Alytaus	Lazdijų r. sav.	66	20	2	6	5	2								
27	Marijampolės	Marijampolės sav.	295	56	19	28	32	12								
28	Telšių	Mažeikių r. sav.	215	54	13	31	21	12								
29	Utenos	Molėtų r. sav.	64	23	2	3	1	5								
59	Vilniaus	Vilniaus m. sav.	2935	1178	74	30	60	391								
60	Vilniaus	Vilniaus r. sav.	333	73	9	0	7	35								
61	Utenos	Visagino sav.	57	9	2	1	4	9								
62	Utenos	Zarasų r. sav.	30	6	2	1	3	3								
63			Santykis:	0,24	0,05	0,06	0,07	0,08								
64																
65																

=ROUND(SUM(D3:D62)/SUM(\$C3:\$C62);2)

1.1. Langelyje **D63** įrašykite formulę, kuri apskaičiuotų santykį tarp abiturientų, stojusių į Vilniaus universitetą (VU) ir visų abiturientų skaičiaus Lietuvoje. Apskaičiuotas santykis turi būti suapvalintas iki šimtųjų, nekeičiant langelio formato.

Problematika

- *Nesupranta matematinės santykio raiškos (mokiniams, vertintojams);*
- *Abiturientų skaičius Lietuvoje = stojusių į 5 universitetus abiturientų suma (vertintojams);*
- *Koordinatinių naudojimas (mokiniams);*
- *Apvalinimo naudojimas (mokiniams);*
- *Neteisingi duomenys.*

=ROUND(SUM(D3:D62)/SUM(\$C3:\$C62);2)

59	Vilniaus	Vilniaus m. sav.	2935	1178	74	30	60	391
60	Vilniaus	Vilniaus r. sav.	333	73	9	0	7	35
61	Utenos	Visagino sav.	57	9	2	1	4	9
62	Utenos	Zarasų r. sav.	30	6	2	1	3	3
63		Santykis:	0,24	0,05	0,06	0,07	0,08	
64								

333	73	9	0	7	35
57	9	2	1	4	9
30	6	2	1	3	3
Santykis:	0,24 : 0,76	0,05 : 0,95	0,06 : 0,94	0,07 : 0,93	0,08 : 0,92

1.1. Langelyje **D63** įrašykite formulę, kuri apskaičiuotų santykį tarp abiturientų, stojusių į Vilniaus universitetą (VU) ir visų abiturientų skaičiaus Lietuvoje. Apskaičiuotas santykis turi būti suapvalintas iki šimtųjų, nekeičiant langelio formato.

1.1. Langelyje D63 įrašyta teisinga formulė, pavyzdžiui, =ROUND(SUM(D3:D62)/SUM(\$C3:\$C62);2)	(5)
1.1.1. Teisingai apskaičiuojamas abiturientų skaičius stojančių į VU, pavyzdžiui, =ROUND(SUM(D3:D62)/SUM(\$C3:\$C62);2)	1
1.1.2. Teisingai apskaičiuojamas visų abiturientų skaičių Lietuvoje, pavyzdžiui, =ROUND(SUM(D3:D62)/SUM(\$C3:\$C62);2)	1
1.1.3. Teisingai apskaičiuojamas santykis, pavyzdžiui, =ROUND(SUM(D3:D62)/SUM(\$C3:\$C62);2)	1
1.1.4. Apskaičiuotas santykis suapvalintas iki šimtųjų, pavyzdžiui, =ROUND(SUM(D3:D62)/SUM(\$C3:\$C62);2)	1
1.1.5. Formulėje teisingai naudojamos mišriosios koordinatės, pavyzdžiui, =ROUND(SUM(D3:D62)/SUM(\$C3:\$C62);2)	

59	Vilniaus	Vilniaus m. sav.	2935	1178	74	30	60	391
60	Vilniaus	Vilniaus r. sav.	333	73	9	0	7	35
61	Utenos	Visagino sav.	57	9	2	1	4	9
62	Utenos	Zarasų r. sav.	30	6	2	1	3	3
63		Santykis:	0,24	0,05	0,06	0,07	0,08	
64								

=ROUND(SUM(D3:D62)/SUM(\$C3:\$C62);2)

Klausimo Nr.	Taškų skaičius	Sunkumas		Skiriamoji geba	Koreliacija	Pasiekimų lygis
1.1.1	1	72,4	L	73,3	0,589	Slenkstinis
1.1.2	1	68,1	L	78,4	0,607	Slenkstinis
1.1.3	1	50,9	V	77,5	0,585	Slenkstinis
1.1.4	1	43,6	V	75,4	0,602	Slenkstinis
1.1.5	1	58,0	V	83,2	0,618	Pagrindinis

1.1.1. Teisingai apskaičiuojamas abiturientų skaičius stojančių į VU, pavyzdžiui, =ROUND(SUM(D3:D62)/SUM(\$C3:\$C62);2)	1
1.1.2. Teisingai apskaičiuojamas visų abiturientų skaičių Lietuvoje, pavyzdžiui, =ROUND(SUM(D3:D62)/SUM(\$C3:\$C62);2)	1
1.1.3. Teisingai apskaičiuojamas santykis, pavyzdžiui, =ROUND(SUM(D3:D62)/SUM(\$C3:\$C62);2)	1
1.1.4. Apskaičiuotas santykis suapvalintas iki šimtųjų, pavyzdžiui, =ROUND(SUM(D3:D62)/SUM(\$C3:\$C62);2)	1
1.1.5. Formulėje teisingai naudojamos mišriosios koordinatės, pavyzdžiui, =ROUND(SUM(D3:D62)/SUM(\$C3:\$C62);2)	

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	2023–2024 mokslo metais stojusių į I pakopos studijas Lietuvoje abiturientų skaičius pagal savivaldybes (universiteto duomenys)									Gyventojų ir abiturientų duomenys savivaldybėse 2023 metais						
2	Apskritis	Savivaldybė	Abiturientų skaičius savivaldybėje	VU	LSMU	KTU	VDU	VILNIUS TECH		Apskritis	Abiturientų skaičius	Gyventojų skaičius apskrityje				
3	Šiaulių	Akmenės r. sav.	60	8	9	2	1	6		Tauragės	361	89951				
4	Alytaus	Alytaus m. sav.	329	81	8	33	14	43		Utenos	441	122712				
5	Alytaus	Alytaus r. sav.	24	8	7	3	0	1		Telšių	558	129863				
6	Utenos	Anykščių r. sav.	85	23	0	4	5	6		Alytaus	560	132911				
7	Kauno	Birštono sav.	25	3	3	3	1	1		Marijampolės	561	133335				
8	Panevėžio	Biržų r. sav.	73	15	1	4	2	9		Panevėžio	932	206966				
9	Alytaus	Druskininkų sav.	73	16	4	3	7	7		Šiaulių	1159	266901				
10	Vilniaus	Elektrėnų sav.	102	24	6	7	8	9		Klaipėdos	1508	343760				
11	Utenos	Ignalinos r. sav.	44	18	0	0	0	0		Kauno	2907	587370				
12	Kauno	Jonavos r. sav.	177	25	15	27	25	5		Vilniaus	3901	876450				
13	Šiaulių	Joniškio r. sav.	84	15	4	5	8	6								
14	Tauragės	Jurbarko r. sav.	90	15	5					1.2. Langelyje K3 įrašykite formulę, kuri apskaičiuotų langelyje J3 nurodytos apskrities abiturientų skaičių.						
15	Kauno	Kaišiadorių r. sav.	102	26	7											
16	Marijampolės	Kalvarijos sav.	25	3	3											
17	Kauno	Kauno m. sav.	1939	200	163	30	244	28								
18	Kauno	Kauno r. sav.	269	15	17	50	63	3								
19	Marijampolės	Kazlų Rūdos sav.	31	4	4	4	2	2								
20	Kauno	Kėdainių r. sav.	189	27	15	21	15	7								
21	Šiaulių	Kelmės r. sav.	83	10	7	13	10	1								
22	Klaipėdos	Klaipėdos m. sav.	886	201	38	62	28	64								
23	Klaipėdos	Klaipėdos r. sav.	132	16	12	12	10	5								
24	Klaipėdos	Kretingos r. sav.	193	47	6	22	6	14								
25	Panevėžio	Kupiškio r. sav.	64	15	3	6	4	6								
26	Alytaus	Lazdijų r. sav.	66	20	2	6	5	2								
27	Marijampolės	Marijampolės sav.	295	56	19	28	32	12								
28	Telšių	Mažeikių r. sav.	215	54	13	31	21	12								
29	Utenos	Molėtų r. sav.	64	23	2	3	1	5								

=SUMIF(A\$3:A\$62;J3;C\$3:C\$62)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	2023–2024 mokslo metais stojusių į I pakopos studijas Lietuvoje abiturientų skaičius pagal savivaldybes (universiteto duomenys)									Gyventojų ir abiturientų duomenys savivaldybėse 2023 metais						
2	Apskritis	Savivaldybė	Abiturientų skaičius savivaldybėje	VU	LSMU	KTU	VDU	VILNIUS TECH		Apskritis	Abiturientų skaičius	Gyventojų skaičius apskrityje				
3	Šiaulių	Akmenės r. sav.	60	8	9	2	1	6		Tauragės	361	89951				
4	Alytaus	Alytaus m. sav.	329	81	8	33	14	43		Utenos	441	122712				
5	Alytaus	Alytaus r. sav.	24	8	7	3	0	1		Telšių	558	129863				
6	Utenos	Anykščių r. sav.	85	23	0	4	5	6		Alytaus	560	132911				
7	Kauno	Birštono sav.	25	3	3	3	1	1		Marijampolės	561	133335				
8	Panevėžio	Biržų r. sav.	73	15	1	4	2	9		Panevėžio	932	206966				
9	Alytaus	Druskininkų sav.	73	16	4	3	7	7		Šiaulių	1159	266901				
10	Vilniaus	Elektrėnų sav.	102	24	6	7	8	9		Klaipėdos	1508	343760				
11	Utenos	Ignalinos r. sav.	44	18	0	0	0	0		Kauno	2907	587370				
12	Kauno	Jonavos r. sav.	177	25	15	27	25	5		Vilniaus	3901	876450				
13	Šiaulių	Joniškio r. sav.	84	15	4	5	8	6								
14	Tauragės	Jurbarko r. sav.	90	15	5	3	1	2								
15	Kauno	Kaišiadorių r. sav.	102	26	7	1	1	1								
16	Marijampolės	Kalvarijos sav.	25	3	1	1	1	1								
17	Kauno	Kauno m. sav.	1939	200	16	33	47	20								
18	Kauno	Kauno r. sav.	269	15	17	50	63	3								
19	Marijampolės	Kazlų Rūdos sav.	31	4	4	4	2	2								
20	Kauno	Kėdainių r. sav.	100	27	15	21	15	7								

1.2. Langelyje K3 įrašykite formulę, kuri apskaičiuotų langelyje J3 nurodytos apskrities abiturientų skaičių.

1.2. Langelyje K3 įrašyta teisinga formulė, pavyzdžiui:

=SUMIF(A\$3:A\$62;J3;C\$3:C\$62)

1.2.1. Teisingai pasirinkta tikrinimo sąlygos sritis, pavyzdžiui,

=SUMIF(A\$3:A\$62;J3;C\$3:C\$62)

1.2.2. Teisingai pasirinkta tikrinimo sąlyga, apskrities pavadinimas, pavyzdžiui,

=SUMIF(A\$3:A\$62;J3;C\$3:C\$62)

1.2.3. Teisingai pasirinkta skaičiavimo sritis, pavyzdžiui,

=SUMIF(A\$3:A\$62;J3;C\$3:C\$62)

1.2.4. Formulėje teisingai naudojamos mišriosios koordinatės, pavyzdžiui,

=SUMIF(A\$3:A\$62;J3;C\$3:C\$62)

(4)

1

1

1

1

=SUMIF(A\$3:A\$62;J3;C\$3:C\$62)

Klausimo Nr.	Taškų skaičius	Sunkumas		Skiriamoji geba	Koreliacija	Pasiekimų lygis
1.2.1	1	73,7	L	68,1	0,551	Slenkstinis
1.2.2	1	75,9	L	66,9	0,555	Slenkstinis
1.2.3	1	72,2	L	62,4	0,501	Slenkstinis
1.2.4	1	53,0	V	79,6	0,594	Pagrindinis

1.2. Langelyje K3 įrašyta teisinga formulė, pavyzdžiui: =SUMIF(A\$3:A\$62;J3;C\$3:C\$62)	(4)
1.2.1. Teisingai pasirinkta tikrinimo sąlygos sritis, pavyzdžiui, =SUMIF(A\$3:A\$62;J3;C\$3:C\$62)	1
1.2.2. Teisingai pasirinkta tikrinimo sąlyga, apskrities pavadinimas, pavyzdžiui, =SUMIF(A\$3:A\$62;J3;C\$3:C\$62)	1
1.2.3. Teisingai pasirinkta skaičiavimo sritis, pavyzdžiui, =SUMIF(A\$3:A\$62;J3;C\$3:C\$62)	1
1.2.4. Formulėje teisingai naudojamos mišriosios koordinatės, pavyzdžiui, =SUMIF(A\$3:A\$62;J3;C\$3:C\$62)	1

1.3. Langelio D63 formulę nukopijuokite į langelius E63:H63; langelio K3 formulę nukopijuokite į langelius K4:K12.

59	Vilniaus	Vilniaus m. sav.	2935	1178	74	30	60	391
60	Vilniaus	Vilniaus r. sav.	333	73	9	0	7	35
61	Utenos	Visagino sav.	57	9	2	1	4	9
62	Utenos	Zarasų r. sav.	30	6	2	1	3	3
63		Santykis:	0,24	0,05	0,06	0,07	0,08	
64								

1.3. Langelio D63 formulę nukopijuota į langelius E63:H63; langelio K3 formulę nukopijuota į langelius K4:K12.

Pastaba.

Taškas skiriamas, jeigu nukopijuotos kandidato parašytos formulės.

Gyventojų ir abiturientų duomenys savivaldybėse 2023 metais		
Apskritis	Abiturientų skaičius	Gyventojų skaičius apskrityje
Tauragės	361	89951
Utenos	441	122712
Telšių	558	129863
Alytaus	560	132911
Marijampolės	561	133335
Panevėžio	932	206966
Šiaulių	1159	266901
Klaipėdos	1508	343760
Kauno	2907	587370
Vilniaus	3901	876450

Klausimo Nr.	Taškų skaičius	Sunkumas		Skiriamoji geba	Koreliacija	Pasiekimų lygis
1.3	1	67,7	L	70,5	0,549	Slenkstinis

Problematika

- Tik dalis nukopijuota;

2025 m. Formulės ir funkcijos

Klausimo Nr.	Taškų skaičius	Sunkumas		Skiriamoji geba	Koreliacija	Pasiekimų lygis
1.1.1	1	72,4	L	73,3	0,589	Slenkstinis
1.1.2	1	68,1	L	78,4	0,607	Slenkstinis
1.1.3	1	50,9	V	77,5	0,585	Slenkstinis
1.1.4	1	43,6	V	75,4	0,602	Slenkstinis
1.1.5	1	58,0	V	83,2	0,618	Pagrindinis
1.2.1	1	73,7	L	68,1	0,551	Slenkstinis
1.2.2	1	75,9	L	66,9	0,555	Slenkstinis
1.2.3	1	72,2	L	62,4	0,501	Slenkstinis
1.2.4	1	53,0	V	79,6	0,594	Pagrindinis
1.3	1	67,7	L	70,5	0,549	Slenkstinis

2023 m.

63,6

2024 m.

	Sunkumas	Skiriamoji geba	Koreliacija
SK 1 Formulės ir funkcijos	45,1	55,5	0,743

SK1. Formulių kūrimas ir standartinių funkcijų naudojimas	12	53,2	V	65,9	0,763
---	----	------	---	------	-------

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	2023–2024 mokslo metais stojusių į I pakopos studijas Lietuvoje abiturientų skaičius pagal savivaldybes (universiteto duomenys)									Gyventojų ir abiturientų duomenys savivaldybėse 2023 metais						
2	Apskritis	Savivaldybė	Abiturientų skaičius savivaldybėje	VU	LSMU	KTU	VDU	VILNIUS TECH		Apskritis	Abiturientų skaičius	Gyventojų skaičius apskrityje				
3	Šiaulių	Akmenės r. sav.	60	8	9	2	1	6		Tauragės	361	89951				
4	Alytaus	Alytaus m. sav.	329	81	8	33	14	43		Utenos	441	122712				
5	Alytaus	Alytaus r. sav.	24	8	7	3	0	1		Telšių	558	129863				
6	Utenos	Anykščių r. sav.	85	23	0	4	5	6		Alytaus	560	132911				
7	Kauno	Birštono sav.	25	3	3	3	1	1		Marijampolės	561	133335				
8	Panevėžio	Biržų r. sav.	73	15	1	4	2	9		Panevėžio	932	206966				
9	Alytaus	Druskininkų sav.	73	16						1.4. Išnagrinėkite lentelės Gyventojų ir abiturientų duomenys savivaldybėse 2023 metais duomenis ir nustatykite, ar yra susiję abiturientų skaičius ir gyventojų skaičius. Išvadą ir jos pagrindimą įrašykite langelyje K14.						
10	Vilniaus	Elektrėnų sav.	102	24												
11	Utenos	Ignalinos r. sav.	44	18												
12	Kauno	Jonavos r. sav.	177	25												
13	Šiaulių	Joniškio r. sav.	84	15												
14	Tauragės	Jurbarko r. sav.	90	15	5	14	7	4		Išvada:	Kuo didesnis gyventojų skaičius apskrityse, tuo abiturientų skaičius yra didesnis.					
15	Kauno	Kaišiadorių r. sav.	102	26	2	22	13	3								
16	Marijampolės	Kalvarijos sav.	25	3	3	2	2	6								
17	Kauno	Kauno m. sav.	1939	200	165	36`	244	28								
18	Kauno	Kauno r. sav.	269	15	17	50	63	3								
19	Marijampolės	Kazlų Rūdos sav.	31	4	4	4	2	2								
20	Kauno	Kėdainių r. sav.	189	27	15	21	15	7								
21	Šiaulių	Kelmės r. sav.	83	10	7	13	10	1								
22	Klaipėdos	Klaipėdos m. sav.	886	201	38	62	28	64								
23	Klaipėdos	Klaipėdos r. sav.	132	16	12	12	10	5								
24	Klaipėdos	Kretingos r. sav.	193	47	6	22	6	14								
25	Panevėžio	Kupiškio r. sav.	64	15	3	6	4	6								
26	Alytaus	Lazdijų r. sav.	66	20	2	6	5	2								
27	Marijampolės	Marijampolės sav.	295	56	19	28	32	12								
28	Telšių	Mažeikių r. sav.	215	54	13	31	21	12								
29	Utenos	Molėtų r. sav.	64	23	2	3	1	5								

Problematika

- Išvada ir pagrindimas (vertintojams):

Problematika

- Išvada ir pagrindimas (vertintojams);
-

1.4. Išnagrinėkite lentelės *Gyventojų ir abiturientų duomenys savivaldybėse 2023 metais* duomenis ir nustatykite, ar yra susiję abiturientų skaičius ir gyventojų skaičius. Išvadą ir jos pagrindimą įrašykite langelyje **K14**.

1.4. Langelyje **K14** pateikta išvada ir jos pagrindimas gali būti suformuluoti įvairiai, tačiau esmė yra tokia, pavyzdžiui,

Abiturientų skaičius yra tiesiogiai susijęs su gyventojų skaičiumi apskrityse. Visais atvejais duomenyse matoma: kuo didesnis gyventojų skaičius apskrityse, tuo abiturientų skaičius yra didesnis.

Pastaba.

Taškas skiriamas, jeigu išvada ir pagrindimas yra teisingi.

Gyventojų ir abiturientų duomenys savivaldybėse 2023 metais		
Apskritis	Abiturientų skaičius	Gyventojų skaičius apskrityje
Tauragės	361	89951
Utenos	441	122712
Telšių	558	129863
Alytaus	560	132911
Marijampolės	561	133335
Panevėžio	932	206966
Šiaulių	1159	266901
Klaipėdos	1508	343760
Kauno	2907	587370
Vilniaus	3901	876450

Išvada: Kuo didesnis gyventojų skaičius apskrityse, tuo abiturientų skaičius yra didesnis.

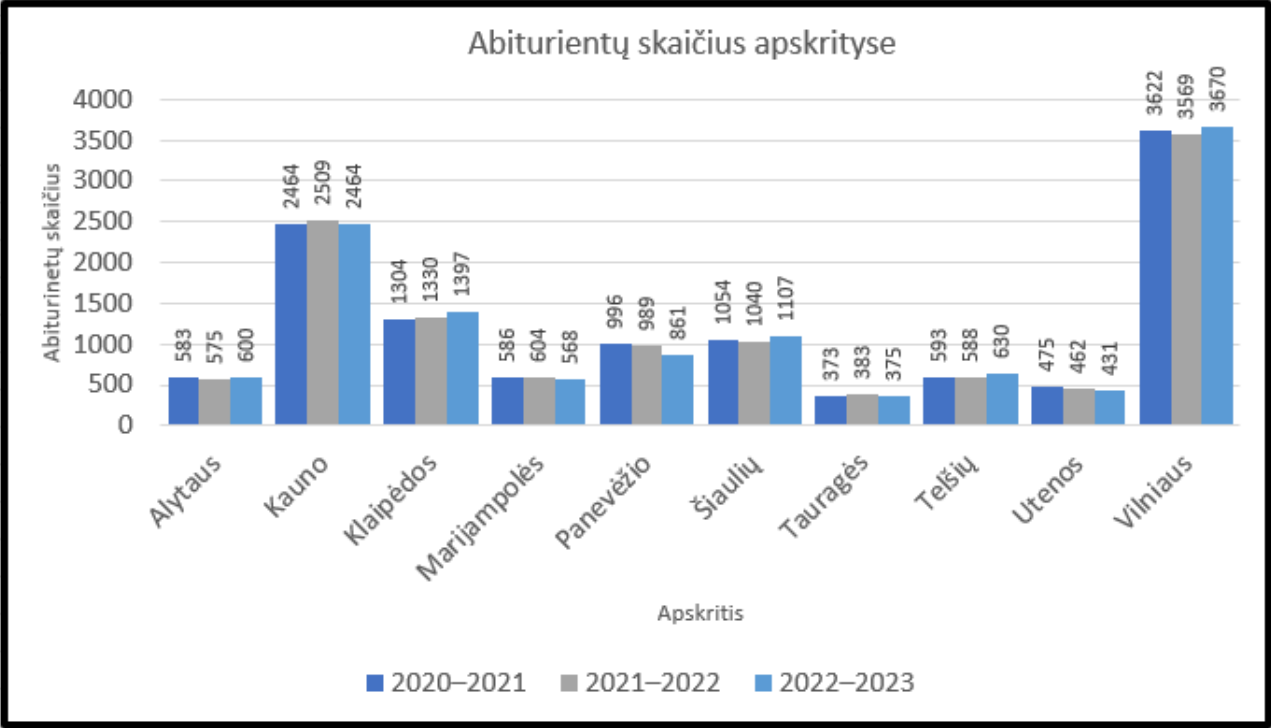
Klausimo Nr.	Taškų skaičius	Sunkumas		Skiriamoji geba	Koreliacija	Pasiekimų lygis
1.4	1	56,9	v	70,5	0,534	Aukštesnysis

Diagramos

2. Lakšte *Diagrama* esančios lentelės *Abiturientų, stojusių į I pakopos studijas Lietuvoje, skaičius pagal apskritis* duomenys pavaizduoti stulpeline diagrama.
- 2.1. Diagramą pavadinkite *Abiturientų skaičius apskrityse*, diagramos legendą pateikite diagramos apačioje.
- 2.2. Atsižvelgdami į diagramoje pateiktus duomenis, pridėkite diagramos reikšmių (y) ir kategorijų (x) ašis ir jas tinkamai pavadinkite.
- 2.3. Diagramos reikšmių (y) ašies didžiausią reikšmę nustatykite 4000. Skaitinės vertės turi būti rodomos kas 500 vienetų.
- 2.4. Išnagrinėkite diagramos duomenis ir padarykite išvadą apie abiturientų skaičiaus kitimo tendencijas Klaipėdos ir Panevėžio apskrityse. Išvadą įrašykite langelyje **B14**.

(5 taškai)

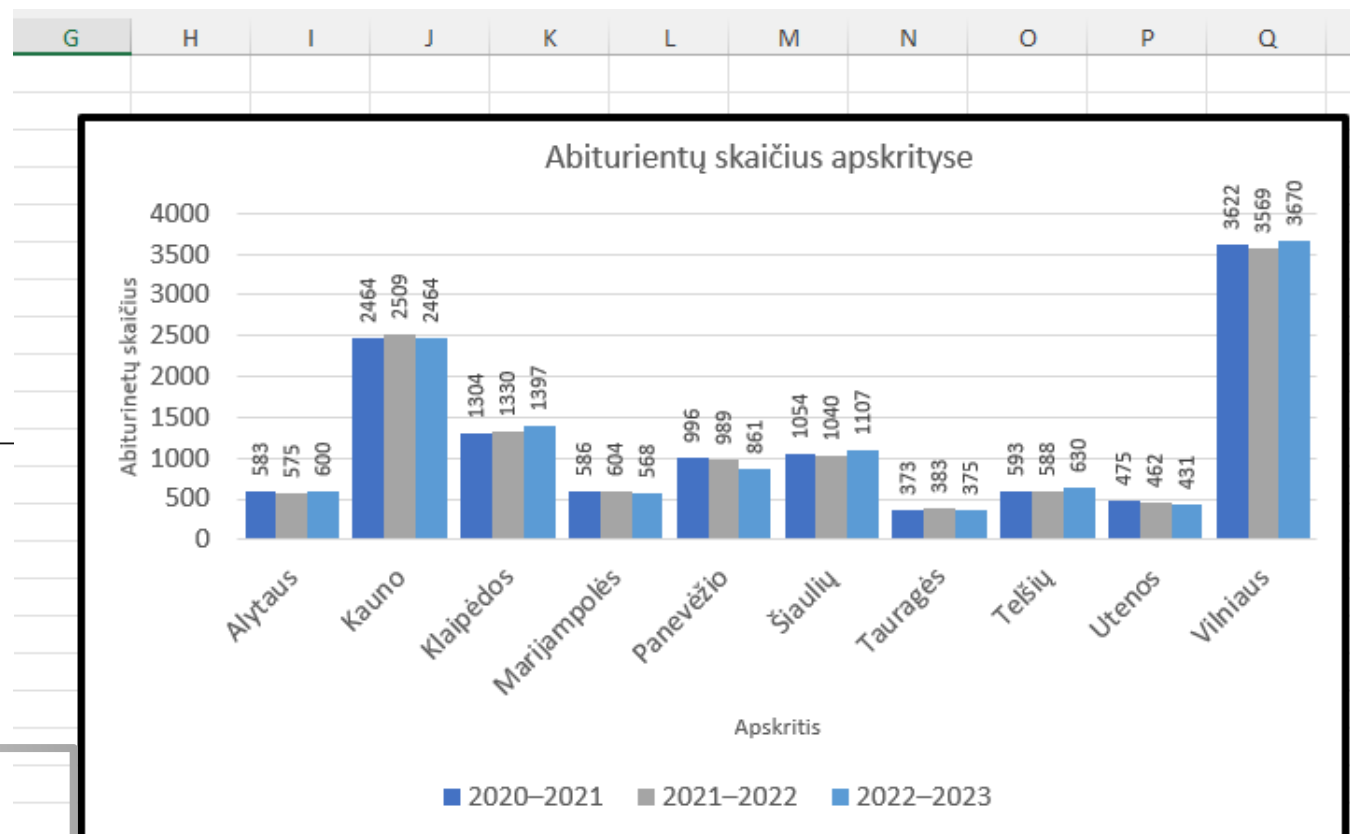
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Abiturientų, stojusių į I pakopos studijas Lietuvoje, skaičius pagal apskritis																
2	Apskritis	2020–2021	2021–2022	2022–2023													
3	Alytaus	583	575	600													
4	Kauno	2464	2509	2464													
5	Klaipėdos	1304	1330	1397													
6	Marijampolės	586	604	568													
7	Panevėžio	996	989	861													
8	Šiaulių	1054	1040	1107													
9	Tauragės	373	383	375													
10	Telšių	593	588	630													
11	Utenos	475	462	431													
12	Vilniaus	3622	3569	3670													
13																	
14	Išvada:																
15																	
16																	
17																	
18																	
19																	
20																	
21																	
22																	



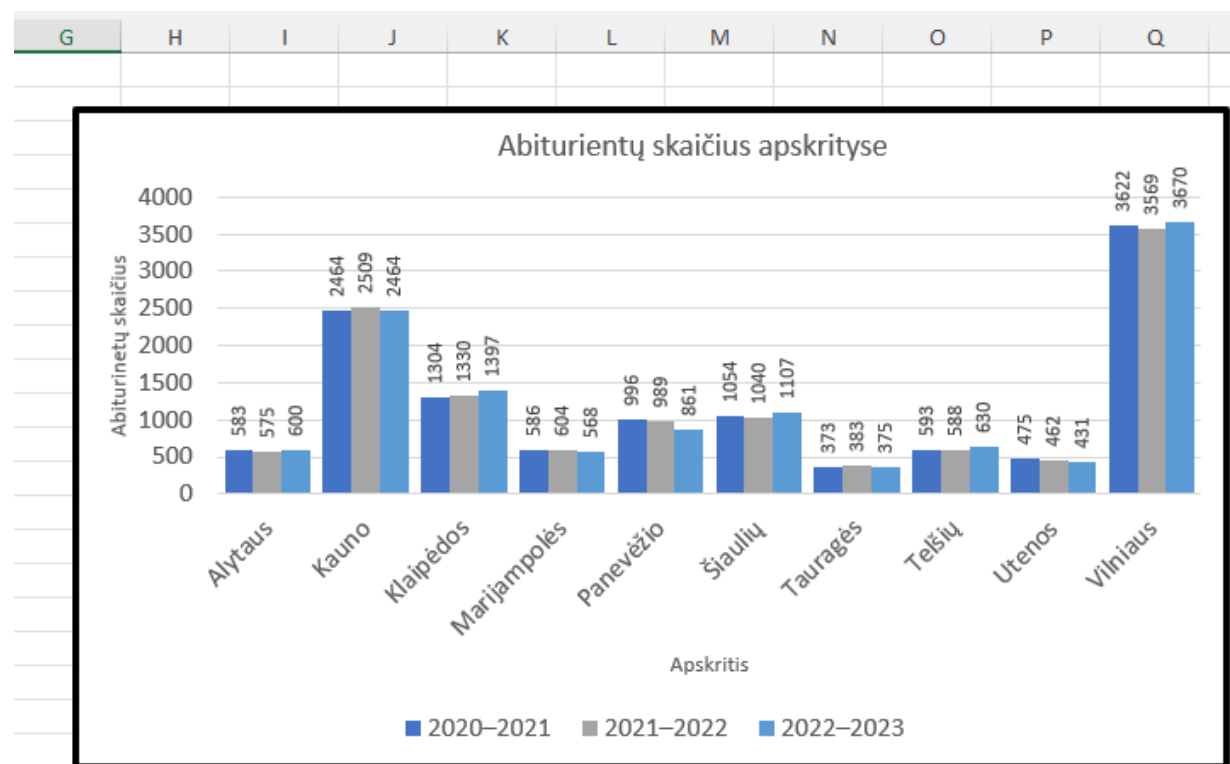
2	2.1. Diagramos pavadinimas – <i>Abiturientų skaičius apskrityse</i> ; diagramos legendą pateikta diagramos apačioje. Pastaba. <i>Taškas skiriamas, jeigu teisingai įvykdyti abu nurodymai.</i>	(1)	5
	2.2. Atsižvelgiant į diagramoje pateiktus duomenis, ašių pavadinimai tinkamai pavadinti, pavyzdžiui, 2.2.1. Diagramos reikšmių (y) ašies pavadinimas – <i>Abiturientų skaičius</i> . 2.2.2. Diagramos kategorijų (x) ašies pavadinimas – <i>Apskritis</i> .	(2)	
	Pastaba. <i>Taškai skiriami, jeigu kandidatas prasmingai pavadino ašis.</i>	1 1	
	2.3. Nustatyta didžiausia reikšmių (y) ašies reikšmė – 4000; skaitinės vertės nustatytos rodyti kas 500 vienetų. Pastaba. <i>Taškas skiriamas, jeigu teisingai įvykdyti abu nurodymai.</i>	(1)	

Problematika

- *Atidumas (įvykdyti abu reikalavimai);*
- *Legendos perkėlimas;*



2	2.1. Diagramos pavadinimas – <i>Abiturientų skaičius apskrityse</i> ; diagramos legendą pateikta diagramos apačioje. Pastaba. <i>Taškas skiriamas, jeigu teisingai įvykdyti abu nurodymai.</i>	(1)	5
	2.2. Atsižvelgiant į diagramoje pateiktus duomenis, ašių pavadinimai tinkamai pavadinti, pavyzdžiui, 2.2.1. Diagramos reikšmių (y) ašies pavadinimas – <i>Abiturientų skaičius</i> . 2.2.2. Diagramos kategorijų (x) ašies pavadinimas – <i>Apskritis</i> .	(2)	
		1	
		1	
	Pastaba. <i>Taškai skiriami, jeigu kandidatas prasmingai pavadino ašis.</i>		
	2.3. Nustatyta didžiausia reikšmių (y) ašies reikšmė – 4000; skaitinės vertės nustatytos rodyti kas 500 vienetų. Pastaba. <i>Taškas skiriamas, jeigu teisingai įvykdyti abu nurodymai.</i>	(1)	



Klausimo Nr.	Taškų skaičius	Sunkumas		Skiriamoji geba	Koreliacija	Pasiekimų lygis
2.1	1	75,2	L	36,4	0,326	Slenkstinis
2.2.1	1	78,3	L	51,1	0,464	Pagrindinis
2.2.2	1	78,3	L	50,9	0,463	Pagrindinis
2.3	1	85,5	LL	39,5	0,412	Slenkstinis

2025 m. Diagramos

Klausimo Nr.	Taškų skaičius	Sunkumas		Skiriamoji geba	Koreliacija	Pasiekimų lygis
2.1	1	75,2	L	36,4	0,326	Slenkstinis
2.2.1	1	78,3	L	51,1	0,464	Pagrindinis
2.2.2	1	78,3	L	50,9	0,463	Pagrindinis
2.3	1	85,5	LL	39,5	0,412	Slenkstinis

79,3

2024 m.

SK2. Diagramos tvarkymas	4	80,2	LL	31,9	0,445
--------------------------	---	------	----	------	-------

2023 m.

SK 2 Diagrama	81,1	15,9	0,331
---------------	------	------	-------

2.4. Išnagrinėkite diagramos duomenis ir padarykite išvadą apie abiturientų skaičiaus kitimo tendencijas Klaipėdos ir Panevėžio apskrityse. Išvadą įrašykite langelyje B14.

(5 taškai)

	A	B	C	D	E	F
1	Abiturientų, stojusių į I pakopos studijas Lietuvoje, skaičius pagal apskritis					
2	Apskritis	2020–2021	2021–2022	2022–2023		
3	Alytaus	583	575	600		
4	Kauno	2464	2509	2464		
5	Klaipėdos	1304	1330	1397		
6	Marijampolės	586	604	568		
7	Panevėžio	996	989	861		
8	Šiaulių	1054	1040	1107		
9	Tauragės	373	383	375		
10	Telšių	593	588	630		
11	Utenos	475	462	431		
12	Vilniaus	3622	3569	3670		
13						
14	Išvada:	Klaipėdoje abiturientų skaičius didėjo, o Panevėžio apskrityje – mažėjo.				
15						
16						
17						
18						
19						
20						
21						

2.4. Langelyje B14 pateikta išvada gali būti suformuluota įvairiai, tačiau išvados esmė yra tokia, pavyzdžiui,

Iš reikšmių, pateiktų stulpelinėse diagramose, matoma, jog nuo 2020 iki 2023 metų Klaipėdoje abiturientų skaičius didėjo, o Panevėžio apskrityje – mažėjo.

2.4	1	80,5	LL	35,0	0,344	Aukštesnysis
-----	---	------	----	------	-------	--------------

Duomenų atranka

3. Lakšte *Atranka* pateikta lentelė 2023–2024 mokslo metais stojusių į I pakopos studijas Lietuvoje abiturientų skaičius pagal savivaldybes (kolegijų duomenys). Šios lentelės duomenims parinkite tinkamus automatinės atrankos kriterijus, kad būtų pateikti tik Vilniaus, Kauno bei Klaipėdos apskričių savivaldybėse esančių abiturientų skaičius. Be to, į pateikiamus atrankos rezultatus neįtraukite savivaldybių, kuriose yra mažiau kaip 100 abiturientų, duomenų.

2023–2024 mokslo metais stojusių I pakopos studijas Lietuvoje abiturientų skaičius pagal savivaldybes (kolegijų duomenys)

Apskritis	Savivaldybė	Abiturientų skaičius savivaldybėje	Alytaus kolegija	Kauno kolegija	Klaipėdos kolegija	Marijampolės kolegija	Panevėžio kolegija	Šiaulių kolegija	Vilniaus kolegija
Kauno	Kauno m. sav.	1939							
Kauno	Kaišiadorių r. sav.	102							
Vilniaus	Vilniaus r. sav.	333							
Klaipėdos	Klaipėdos r. sav.	132							
Klaipėdos	Šilutės r. sav.	145							
Klaipėdos	Klaipėdos m. sav.	886							
Klaipėdos	Kretingos r. sav.	193							
Vilniaus	Vilniaus m. sav.	2935							
Vilniaus	Elektrėnų sav.	102							
Vilniaus	Ukmergės r. sav.	119							
Kauno	Kauno r. sav.	269							
Vilniaus	Trakų r. sav.	126							
Kauno	Kėdainių r. sav.	189	0	39	3	0	1	2	5
Kauno	Jonavos r. sav.	177	0	29	0	1	0	3	5
Vilniaus	Šalčininkų r. sav.	132	0	1	0	0	0	0	22
Kauno	Raseinių r. sav.	128	0	32	1	1	0	2	2

Filtravimo panelė:

- Rūšiuoti nuo A iki Z
- Rūšiuoti nuo Z iki A
- Rūšiuoti pagal spalvą
- Lapo rodinys
- Valyti filtrą nuo Apskrities
- Filtruoti pagal spalvą
- Teksto filtrai
- Ieškoti
- (Žymėti viską)
 - ☐ Alytaus
 - ☒ Kauno
 - ☒ Klaipėdos
 - ☐ Marijampolės
 - ☐ Panevėžio
 - ☐ Šiaulių
 - ☐ Tauragės
 - ☐ Telšių
 - ☐ Utenos
 - ☒ Vilniaus

Pasirinktasis automatinis filtras:

Rodyti eilutes, kuriose:
Abiturientų skaičius savivaldybėje
 daugiau arba lygu
☐ Ir ☒ Arba

 Naudokite ? bet kokiam vienam simboliui žymėti
 Naudokite * bet kokiai simbolių sekai žymėti

Gera! Atšaukti

3. Lakšte *Atranka* pateikta lentelė 2023–2024 mokslo metais stojusių į I pakopos studijas Lietuvoje abiturientų skaičius pagal savivaldybes (kolegijų duomenys). Šios lentelės duomenims parinkite tinkamus automatinės atrankos kriterijus, kad būtų pateikti tik Vilniaus, Kauno bei Klaipėdos apskričių savivaldybėse esančių abiturientų skaičius. Be to, į pateikiamus atrankos rezultatus neįtraukite savivaldybių, kuriose yra mažiau kaip 100 abiturientų, duomenų.

(2 taškai)

Pasirinktinis automatinis filtras

Rodyti eilutes, kuriose:

Abiturientų skaičius savivaldybėje

daugiau arba lygu 100

☐ Į ☒ Arba

Naudokite ? bet kokiam vienam simboliui žymėti

Naudokite * bet kokiai simbolių sekai žymėti

Gerai Atšaukti

leškoti

☒ (Žymėti viską)

☐ Alytaus

☒ Kauno

☒ Klaipėdos

☐ Marijampolės

☐ Panevėžio

☐ Šiaulių

☐ Tauragės

☐ Telšių

☐ Utenos

☒ Vilniaus

Gerai Atšaukti

Problematika

- „Rankinis“ duomenų parinkimas;
- Mažiau kaip 100 = „daugiau arba lygu“ 100;
- Pivot table;
-

Duomenų atranka

3. Lakšte *Atranka* pateikta lentelė 2023–2024 mokslo metais stojusių į I pakopos studijas Lietuvoje abiturientų skaičius pagal savivaldybes (kolegijų duomenys). Šios lentelės duomenims parinkite tinkamus automatinės atrankos kriterijus, kad būtų pateikti tik Vilniaus, Kauno bei Klaipėdos apskričių savivaldybėse esančių abiturientų skaičius. Be to, į pateikiamus atrankos rezultatus neįtraukite savivaldybių, kuriose yra mažiau kaip 100 abiturientų, duomenų.

(2 taškai)

3.1. Atrenkami duomenys, kurių apskritis yra: Vilniaus, Kauno, Klaipėdos.	1	2
3.2. Atrenkami duomenys, kurių savivaldybėse abiturientų skaičius yra ne mažesnis nei 100.	1	
Pastaba. <i>Taškas skiriamas, jeigu naudojamas skaičių filtras.</i>		

Klausimo Nr.	Taškų skaičius	Sunkumas		Skiriamoji geba	Koreliacija	Pasiekimų lygis
3.1	1	78,2	L	51,5	0,454	Slenkstinis
3.2	1	40,4	V	60,9	0,483	Pagrindinis

2025 m. Duomenų atranka

Klausimo Nr.	Taškų skaičius	Sunkumas		Skiriamoji geba	Koreliacija	Pasiekimų lygis
3.1	1	78,2	L	51,5	0,454	Slenkstinis
3.2	1	40,4	V	60,9	0,483	Pagrindinis

59,3

2024 m.

SK3. Duomenų atranka	2	64,1	L	59,5	0,532
----------------------	---	------	---	------	-------

2023 m.

SK 3 Duomenų atranka	72,3	40,3	0,451
----------------------	------	------	-------

Duomenų rikiavimas

4. Lakšte Rikiavimas pateikta lentelė 2023–2024 mokslo metais stojusių į I pakopos studijas Lietuvoje abiturientų skaičius pagal savivaldybes (kolegijų duomenys). Šios lentelės duomenis surikiuokite pagal du kriterijus: pagal abiturientų skaičių savivaldybėje mažėjančiai ir Kauno kolegijos abiturientų skaičių didėjančiai.

(2 taškai)

	A	B	C	D	E	F	G	H	I	J
1	2023–2024 mokslo metais stojusių į I pakopos studijas Lietuvoje abiturientų skaičius pagal savivaldybes (kolegijų duomenys)									
2	Apskritis	Savivaldybė	Abiturientų skaičius savivaldybėje	Alytaus kolegija	Kauno kolegija	Klaipėdos kolegija	Marijampolės kolegija	Panevėžio kolegija	Šiaulių kolegija	Vilniaus kolegija
3	Vilniaus	Švenčionių r. sav.	87	0	0	0	0	0	0	16
4	Kauno	Kauno m. sav.	1939	2	216	2	1	0	0	12
5	Kauno	Kaišiadorių r. sav.	102	0	7	0	2	0	0	4
6	Telšių	Rietavo r. sav.	19	0	1	2	0	0	0	1
7	Vilniaus	Vilniaus r. sav.	333	0	2	0	0	0	0	32
8	Klaipėdos	Palangos m. sav.	86	0	1	7	0	0	1	2
9	Utenos	Anykščių r. sav.	85	0	3	0	0	0	0	10
10	Tauragės	Tauragės r. sav.	134	0	16	5	0	0	0	2
11	Panevėžio	Rokiškio r. sav.	103	0	10	0	0	2	1	21
12	Telšių	Mažeikių r. sav.	215	0	16	20	0	0	5	14
13	Klaipėdos	Klaipėdos r. sav.	132	0	6	22	0	0	0	4
14	Klaipėdos	Šilutės r. sav.	145	0	16	24	0	0	0	0
15	Marijampolės	Vilkaviškio r. sav.	113	1	15	0	10	0	0	3
16	Utenos	Ignalinos r. sav.	44	0	2	0	0	0	1	11
17	Šiaulių	Radvilišio r. sav.	147	0	30	2	0	2	13	11
18	Klaipėdos	Klaipėdos m. sav.	886	0						
19	Klaipėdos	Kretingos r. sav.	193	0						
20	Utenos	Utenos r. sav.	161	0						

	A	B	C	D	E	F	G	H	I	J
1	2023–2024 mokslo metais stojusių į I pakopos studijas Lietuvoje abiturientų skaičius pagal savivaldybes (kolegijų duomenys)									
2	Apskritis	Savivaldybė	Abiturientų skaičius savivaldybėje	Alytaus kolegija	Kauno kolegija	Klaipėdos kolegija	Marijampolės kolegija	Panevėžio kolegija	Šiaulių kolegija	Vilniaus kolegija
3	Vilniaus	Vilniaus m. sav.	2935	0	19	2	0	0	1	292
4	Kauno	Kauno m. sav.	1939	2	216	2	1	0	0	12
5	Klaipėdos	Klaipėdos m. sav.	886	0	15	131	0	0	0	23
6	Šiaulių	Šiaulių m. sav.	654	0	29	8	0	3	89	30
7	Panevėžio	Panevėžio m. sav.	549	0	35	9	1	57	5	35
8	Vilniaus	Vilniaus r. sav.	333	0	2	0	0	0	0	32
9	Alytaus	Alytaus m. sav.	329	35	30	0	1	0	0	17
10	Marijampolės	Marijampolės sav.	295	0	28	1	37	0	0	8
11	Kauno	Kauno r. sav.	269	0	57	2	0	1	0	4
12	Telšių	Mažeikių r. sav.	215	0	16	20	0	0	5	14
13	Klaipėdos	Kretingos r. sav.	193	0	9	26	0	0	0	7
14	Kauno	Kėdainių r. sav.	189	0	39	3	0	1	2	5
15	Kauno	Jonavos r. sav.	177	0	29	0	1	0	3	5
16	Telšių	Telšių r. sav.	169	0	17	18	0	0	3	9
17	Utenos	Utenos r. sav.	161	0	3	0	0	0	0	11
					20	33	0	0	1	4
					30	2	0	2	13	11
					16	24	0	0	0	0

Rūšiavimas

+

Pridėti lygį

×

Naikinti lygį

Kopijuoti lygį

^

↓

Parinkty...

☒ Mano duomenyse yra antraščių

Stulpelis

Rūšiuoti pagalAbiturientų skaičiu

Po to pagalKauno kolegija

Rūšiuoti

Langelio reikšmės

Langelio reikšmės

Tvarka

Nuo didžiausios iki mažiausios

Nuo mažiausios iki didžiausios

Duomenų rikiavimas

4.1. Surikiuota pagal abiturientų skaičių savivaldybėje mažėjančia tvarka (C stulpelis).	1	2
4.2. Surikiuota pagal Kauno kolegijos abiturientų skaičių didėjančia tvarka (E stulpelis).	1	
Pastabos. <i>Taškai skiriami, jeigu surikiuota tik duomenų lentelė, be antraščių. Jei rikiavimai atlikti teisingai, tik sukeistas kriterijų eiliškumas, skiriamas vienas taškas.</i>		

Rūšiavimas

☒ Mano duomenyse yra antraščių

Stulpelis	Rūšiuoti	Tvarka
Rūšiuoti pagal Abiturientų skaičiu	Langelio reikšmės	Nuo didžiausios iki mažiausios
Po to pagal Kauno kolegija	Langelio reikšmės	Nuo mažiausios iki didžiausios

Problematika

- „Sugadinti“ duomenys;
- Vienas rikiavimas;
- Nesimato rikiavimo parinkčių (Rūšiuoti ir filtruoti->Pasirinktinis rūšiavimas);
-

Duomenų rikiavimas

4.1. Surikiuota pagal abiturientų skaičių savivaldybėje mažėjančia tvarka (C stulpelis).	1	2
4.2. Surikiuota pagal Kauno kolegijos abiturientų skaičių didėjančia tvarka (E stulpelis).	1	
Pastabos. <i>Taškai skiriami, jeigu surikiuota tik duomenų lentelė, be antraščių.</i> <i>Jei rikiavimai atlikti teisingai, tik sukeistas kriterijų eiliškumas, skiriamas vienas taškas.</i>		

Klausimo Nr.	Taškų skaičius	Sunkumas		Skiriamoji geba	Koreliacija	Pasiekimų lygis
4.1	1	71,2	L	57,8	0,480	Slenkstinis
4.2	1	72,7	L	59,2	0,491	Slenkstinis

2025 m. Duomenų rikiavimas

Klausimo Nr.	Taškų skaičius	Sunkumas		Skiriamoji geba	Koreliacija	Pasiekimų lygis
4.1	1	71,2	L	57,8	0,480	Slenkstinis
4.2	1	72,7	L	59,2	0,491	Slenkstinis

72,0

2024 m.

SK4. Duomenų rikiavimas	2	80,6	LL	48,7	0,511
-------------------------	---	------	----	------	-------

2023 m.

SK 4 Duomenų rikiavimas	84,7	38,3	0,476
-------------------------	------	------	-------

Skaičiuoklės funkcijos vartotinos VBE užduotyse

1. Matematinės funkcijos

- $+$, $-$, $*$, $/$ – aritmetinės operacijos.
- **SUM** – suma.
- **MULT** – daugyba.
- **ABS** – absoliučioji reikšmė (modulis).
- **SQRT** – kvadratinė šaknis.
- **POWER**, $^$ – kėlimas laipsniu.

2. Apvalinimo funkcijos

- **ROUND** – apvalinimas iki artimiausio sveikojo skaičiaus.
- **ROUNDUP** – apvalinimas iki didesnio skaičiaus.
- **ROUNDDOWN** – apvalinimas imi mažesnio skaičiaus.
- **TRUNC** – trupmeninės dalies atmetimas.

3. Statistinės funkcijos

- **AVERAGE** – vidurkis.
- **MIN** – mažiausia reikšmė.
- **MAX** – didžiausia reikšmė.
- **MEDIAN** – mediana.
- **MODE** – dažniausiai pasikartojanti reikšmė.
- **COUNT** – reikšmių skaičius.

3. Sąlyginės funkcijos

- **IF** – vienos sąlygos tikrinimo.
- **IFS** – kelių sąlygų tikrinimo.
- **SUMIF** – suma, atitinkanti sąlygą.
- **COUNTIF** – skaičiavimas pagal sąlygą.
- **AVERAGEIF** – vidurkis pagal sąlygą.
- **SUMIFS** – suma pagal kelias sąlygas.
- **COUNTIFS** – skaičiavimas pagal kelias sąlygas.
- **AVERAGEIFS** – vidurkis pagal kelias sąlygas.
- **MINIFS** – mažiausia reikšmė pagal sąlygą.
- **MAXIFS** – didžiausia reikšmė pagal sąlygą.

Skaičiuoklės funkcijos vartotinos VBE užduotyse

4. Loginės funkcijos

- **AND** – loginis „ir“.
- **OR** – loginis „arba“.
- **NOT** – loginis neiginys.

6. Datos ir laiko funkcijos

- **TODAY** – dabartinė data.
- **DATE** – sukurama data iš duotų skaičių – metų, mėnesio ir dienos.
- **DAY** – gražina dieną.
- **DAYS** – dienų skirtumas tarp datų.
- **MONTH** – gražina mėnesį.
- **YEAR** – gražina metus.

7. Paieškos ir nuorodų funkcijos

- **INDEX** – gražina reikšmę pagal eilutę ir stulpelį.
- **MATCH** – ieško reikšmės ir gražina jos vietą.
- **VLOOKUP** – vertikalioji paieška.

8. Teksto funkcijos

- **LEFT** - paima simbolius iš eilutės kairės.
- **RIGHT** - paima simbolius iš eilutės dešinės.
- **LEN** - eilutės ilgis.
- **MID** - paima iš eilutės simbolius nuo nurodytos pozicijos nurodytą kiekį.
- **TRIM** - pašalina tarpus nuo eilutės pradžios ir galo.
- **FIND** - ieško įnuodytos eilutės kitoje eilutėje.
- **SUBSTITUTE** - nurodytoje eilutėje ieško pateiktos eilutės ir keičia kita įnuodyta.
- **LOWER** - keičia raides į mažąsias.
- **UPPER** - keičia raides į didžiąsias.
- **PROPER** - keičia raides į mažąsias, pirmąją raidę paverčia didžiąja.
- **&** - sujungimo operatorius

2025 m. informatikos VBE II dalies Testo vertinimo ypatybės

2025 m. informatikos VBE II dalies klausimų parametrų suvestinė

Klausimo Nr.	Taškų skaičius	Sunkumas		Skiriamoji geba	Koreliacija	Pasiekimų lygis
1	1	40,6	V	27,9	0,241	Slenkstinis
2	1	64,6	L	36,8	0,311	Slenkstinis
3.1	2	55,9	V	23,6	0,295	Slenkstinis
3.2	1	75,7	L	3,5	0,033	Slenkstinis
4.1	2	35,6	S	50,8	0,504	Slenkstinis
4.2	1	99,9	LL	0,0	-0,011	Pagrindinis
5.1	1	51,7	V	27,5	0,233	Aukštesnysis
5.2	1	35,0	S	52,5	0,456	Slenkstinis

1 klausimas

1. Internetinėse svetainėse yra naudojami hipertekstai. Paaiškinkite, kas yra hipertekstas. (1 taškas)

Taškai	Mokymosi turinio / Pasiekimų sritis	Kognityvinių gebėjimų grupė	Pasiekimų lygis
1	Skaitmeninio turinio kūrimas	Žinios	Slenkstinis

Taškų pasiskirstymas (%)		Sunkumas	Skiriamoji geba	Koreliacija
0	1	64,6	36,8	0,311
35,4	64,6			

1 klausimo atsakymų vertinimas

1. Internetinėse svetainėse yra naudojami hipertekstai. Paaiškinkite, kas yra hipertekstas. (1 taškas)

Vertinimo instrukcija		
Hipertekstas yra tekstas, su kitais teksta susietas nuorodomis (saitais). Į hipertekstą gali būti įterpta įvairių tipų objektų: piešinių, garso failų, programų ir kt.		
Kandidatų atsakymai (kalba originali)		Vertinimas
1	Hipertekstas tai elektroninė nuoroda.	0
2	Hiperteksas, tai internetinėse svetainėse esantys tekstai, kurie galėtų patraukti žmogaus akį.	0
3	Hipertekstas – tokių svetainės elementų kaip teksto elementai, mygtukai, nuorodos, visuma.	0
4	Hipertekstas yra internetinės svetainės dalis.	0
5	Hipertekstas – tai spalvotai arba pabrauktai pažymėtas žodis ar frazė, kurią paspaudus vartotojas nukreipiamas į kitą interneto svetainę ar dokumentą.	1
6	Hipertekstas – tekstas, į kurį įdėta nuoroda, ją paspaudus tekstas veikia kaip nuoroda.	1
7	Tekstas, už kurio užšifruotas hipersaitas (nuoroda).	1

2 klausimas

2. Pateikite du pavyzdžius, kur gali būti taikoma vektorinė grafika. (1 taškas)

Taškai	Mokymosi turinio / Pasiekimų sritis	Kognityvinių gebėjimų grupė	Pasiekimų lygis
1	Skaitmeninio turinio kūrimas	Žinios	Slenkstinis

Taškų pasiskirstymas (%)		Sunkumas	Skiriamoji geba	Koreliacija
0	1	40,6	27,9	0,241
59,4	40,6			

2 klausimo atsakymų vertinimas

2. Pateikite du pavyzdžius, kur gali būti taikoma vektorinė grafika. (1 taškas)

Vertinimo instrukcija

Reklaminiai plakatai, skrajutės, techninės schemos, knygų iliustracijos, grafiniai elementai, logotipai, prekių ženklai, animacija ir kita.

Kandidatų atsakymai (kalba originali)		Vertinimas
1	Matematinėms funkcijoms pavaizduoti	0
2	vaizdo įrašų kurimas, nuotraukų redagavimas	0
3	Animacijoje, vaizdo įrašuose	0
4	Paint, Photoshop.	0
5	Maketavime, animacijos kūryboje	1
6	Dizaine ir architektūroje	1
7	Įmonės logotipai, animacija.	1

3.1 klausimas

3. Įsivaizduokite, kad kompiuteryje sukūrėte paveikslą, naudodami RŽM (angl. RGB) spalvų modelį.

3.1. Kas gali nutikti spalvoms, kai sukurtą paveikslą konvertuosite į ŽPGJ (angl. CMYK) spalvų modelį? Atsakymą pagrįskite. (2 taškai)

Taškai	Mokymosi turinio / Pasiekimų sritis	Kognityvinių gebėjimų grupė	Pasiekimų lygis
2	Skaitmeninio turinio kūrimas	Žinios	Slenkstinis

Taškų pasiskirstymas (%)			Sunkumas	Skiriamoji geba	Koreliacija
0	1	2	55,9	23,6	0,295
16,3	55,6	28,1			

3.1 klausimo atsakymų vertinimas (1)

3. Įsivaizduokite, kad kompiuteryje sukūrėte paveikslą, naudodami RŽM (angl. RGB) spalvų modelį.
- 3.1. Kas gali nutikti spalvoms, kai sukurtą paveikslą konvertuosite į ŽPGJ (angl. CMYK) spalvų modelį? Atsakymą pagrįskite. (1 taškas)

Vertinimo instrukcija

Kai kurios paveiksle naudojamos spalvos gali pasikeisti. Praktiškai RŽM spalvų modelis turi platesnį spalvų spektrą nei ŽPGJ, todėl, jeigu spalva iš RŽM spalvų modelio nepatenka į ŽPGJ spalvų diapazoną, programa automatiškai parenka artimiausią įmanomą atspalvį.

Pastaba.

Taškas skiriamas už įvardytą pasekmę ir antras taškas už pagrindimą.

3.1 klausimo atsakymų vertinimas (2)

Kandidatų atsakymai (kalba originali)		Vertinimas
1	Invertuojasi spalvos	0
2	Spalvos liks tokios pačios, kadangi kompiuterio monitoriaus galia yra per maža tam, kad perduoti ŽPGJ spalvų modelio ryškumą bei parodyti skirtumą tarp RŽM ir ŽPGJ spalvų modelių.	0
3	Spalvos bus atvirkščios, nes taip veikia spalvų kodai.	0
4	Gali pasikeisti spalvų atspalviai	1
5	RŽM - elektroninių prietaisų ekranuose naudojamos spalvos pateikti visą spalvų spektrą. ŽPGJ- spausdinimo prietaisų naudojamos spalvos.	1
6	spalvos gali pasikeisti, nes yra keičiamas formatas	1
7	Spalvos gali pasikeisti ir neatitikti RGB modelio. RGB turi tris spalvas raudoną, žalią ir mėlyną. O CMYK -neturi raudonos, žalios ir mėlynos. Tai reiškia, kad verčiant iš RGB į CMYK spalvos yra maišomos, kad gauti atitikmenis.	2
8	Spalvos gali išsikraipyti, nesutapti, nes spalvų modeliai sudaryti iš skirtingų pagrindinių spalvų.	2
9	Spalvos pasikeičia, nes RGB ir CMYK spalvų modeliai naudoja skirtingus spalvų maišymo principus – RGB naudoja šviesos priedą, o CMYK pigmentų atėmimą. Dėl to jų kodai ir reikšmės nėra tiesiogiai suderinami.	2

3.2 klausimas

3.2. Kuris iš šių dviejų spalvų modelių dažniausiai naudojamas elektroninėje leidyboje – RŽM ar ŽPGJ? (1 taškas)

Taškai	Mokymosi turinio / Pasiekimų sritis	Kognityvinių gebėjimų grupė	Pasiekimų lygis
1	Skaitmeninio turinio kūrimas	Žinios	Slenkstinis

Taškų pasiskirstymas (%)		Sunkumas	Skiriamoji geba	Koreliacija
0	1	75,7	3,5	0,033
24,3	75,7			

3.2 klausimo atsakymų vertinimas

3.2. Kuris iš šių dviejų spalvų modelių dažniausiai naudojamas elektroninėje leidyboje – RŽM ar ŽPGJ? (1 taškas)

Vertinimo instrukcija

RŽM (angl. RGB).

Kandidatų atsakymai (kalba originali)		Vertinimas
1	CMYK / ŽPGJ	0
2	RGB / RŽM	1

4.1 klausimas

4. Duomenys šifruojami, naudojant įvairius metodus. Vienas iš jų yra asimetrinis šifravimas, kai taikomi privatieji ir viešieji šifravimo raktai.

4.1. Įvardykite šių abiejų raktų paskirtį. (2 taškai)

Taškai	Mokymosi turinio / Pasiekimų sritis	Kognityvinių gebėjimų grupė	Pasiekimų lygis
2	Saugus elgesys	Taikymas	Slenkstinis

Taškų pasiskirstymas (%)			Sunkumas	Skiriamoji geba	Koreliacija
0	1	2	35,6	50,8	0,504
54,5	19,7	25,8			

4.1 klausimo atsakymų vertinimas (1)

4. Duomenys šifruojami, naudojant įvairius metodus. Vienas iš jų yra asimetrinis šifravimas, kai taikomi privatieji ir viešieji šifravimo raktai.

4.1. Įvardykite šių abiejų raktų paskirtį. (2 taškai)

Vertinimo instrukcija

- Privatusis raktas yra laikomas paslapyje ir naudojamas duomenims iššifruoti / užšifruoti. Tik pats asmuo turi prieigą prie privačiojo rakto. (1 taškas)
- Viešasis raktas yra prieinamas visiems ir naudojamas duomenims užšifruoti / iššifruoti. (1 taškas)

4.1 klausimo atsakymų vertinimas (2)

Kandidatų atsakymai (kalba originali)		Vertinimas
1	Viešasis raktas yra prieinamas visiems Privatusis raktas yra skirtas pačiam asmeniui	0
2	Privatusis raktas: Privatus raktas yra taikomas, kai šifruojami duomenys yra privatūs ir norima, kad juos pasiektų tik tie asmenys, kuriems yra duotas šifravimo raktas. Viešasis raktas: Viešasis raktas yra taikomas, kai šifruojami duomenys nėra privatūs ir žinoma, kad šiuos duomenis gali prieiti kiekvienas.	0
3	P. šifravimo raktas, kuris yra slapas nuo kitų, pašalinių žmonių. Naudojamas privačiai informacijai atšifruoti. V. Tai raktas, kuris yra plačiai prieinamas visuomenei. Jį galima rasti internete, jis lengvai prieinamas.	1
4	Privatus raktas žinutės gavėjui <u>atkoduoti</u> (suteikia galimybę skleidusių žinutę perskaityti). Viešasis raktas: Skleidusių žinutę <u>užkoduoti</u> .	1
5	Privatusis raktas: Naudojamas autentifikuoti vartotoją, kad būtų aišku, kad tikrai tas vartotojas išsiuntė duomenis, arba autentifikuoti vartotoją, kad būtų aišku, kad tikrai tik tas vartotojas gali pasiekti duomenis. Viešasis raktas: Naudojamas visų ir duomenis pasiekti gali tik turintis atitinkamą privatų raktą.	2
6	Viešasis raktas – šifruoja informaciją (prieinamas visiems). Privatusis raktas – iššifruoja informaciją (žinomas tik savininkui).	2

4.2 klausimas

4.2. Nurodykite vieną asimetrinio šifravimo algoritmą. (1 taškas)

Taškai	Mokymosi turinio / Pasiekimų sritis	Kognityvinių gebėjimų grupė	Pasiekimų lygis
1	Saugus elgesys	Žinios	Pagrindinis

Taškų pasiskirstymas (%)		Sunkumas	Skiriamoji geba	Koreliacija
0	1	99,9	0	-0,011
0,1	99,9			

4.2 klausimo atsakymų vertinimas

4.2. Nurodykite vieną asimetrinio šifravimo algoritmą. (1 taškas)

Vertinimo instrukcija

RSA, ECC ir kita.

Pastaba. Klausimas neatitinka informatikos programos. Taškas skiriamas visiems kandidatams.

Kandidatų atsakymai (kalba originali)		Vertinimas
1	paimama žinutė, užšifruojama naudojant viešąjį raktą, siunčiama gavėjui, iššifruojama naudojant privatųjį raktą, gaunama iššifruota žinutė	1
2	Kleopatra	0
3	AES, DES, Blowfisz	0
4	Hash, SHA-256, MD5	0
5	VPN, SSL, HTTPS	0
6	Captcha	0
7	Cezario šifras	0

5.1 klausimas

5. Į savo asmeninį elektroninį paštą gavote vienos įmonės rinkodaros skyriaus kvietimą dalyvauti suasmenintoje apklausoje. Joje prašoma nurodyti jūsų amžių, lytį ir nuomonę apie prekes, kurias įsigijote.

5.1. Ar įmonė turi teisę rinkti ir saugoti tokius duomenis be jūsų sutikimo? Pagrįskite savo atsakymą. (1 taškas)

Taškai	Mokymosi turinio / Pasiekimų sritis	Kognityvinių gebėjimų grupė	Pasiekimų lygis
1	Saugus elgesys	Taikymas	Aukštesnysis

Taškų pasiskirstymas (%)		Sunkumas	Skiriamoji geba	Koreliacija
0	1	51,7	27,5	0,233
48,3	51,7			

5.1 klausimo atsakymų vertinimas (1)

5. Į savo asmeninį elektroninį paštą gavote vienos įmonės rinkodaros skyriaus kvietimą dalyvauti suasmenintoje apklausoje. Joje prašoma nurodyti jūsų amžių, lytį ir nuomonę apie prekes, kurias įsigijote.

5.1. Ar įmonė turi teisę rinkti ir saugoti tokius duomenis be jūsų sutikimo? Pagrįskite savo atsakymą. (1 taškas)

Vertinimo instrukcija

Nors šiuo atveju tiesiogiai nėra nurodomi asmens duomenys, tačiau siunčiama suasmeninta apklausa apie prekes, kurias pirkė konkretus asmuo. Turint informaciją apie jų amžių ir lytį, galima netiesiogiai identifikuoti asmenį. Todėl įmonė neturi teisės rinkti ir saugoti jūsų asmens duomenų be jūsų sutikimo.

Pastaba.

Kadangi užduoties formuluotė gali būti suprasta nevienareikšmiškai, taškas skiriamas ir tuomet, jei kandidatas argumentuoja, kodėl įmonė turi teisę rinkti duomenis.

5.1 klausimo atsakymų vertinimas (2)

Kandidatų atsakymai (kalba originali)		Vertinimas
1	Taip, nes šie duomenys <u>nėra asmens duomenys</u> ir pagal juos negalima identifikuoti asmens ir nėra pažeidžiamos jo asmens duomenų teisės.	0
2	Neturi jokios teisės, be jūsų leidimo.	0
3	Taip, nes jie nėra itin svarbūs.	0
4	Įmonė neturi teisės rinkti tokių duomenų be žmogaus sutikimo, nes tie duomenys priklauso žmogui ir papuolus į nepatikimų žmonių rankas gali blogai baigtis.	0
5	Neturi, nes tai yra mano asmeniniai duomenys, kurie gali mane identifikuoti. Gali šiuos duomenis rinkti jei tik aš sutinku.	1
6	Tai yra jautrūs asmeniniai duomenys, tad be sutikimo jais naudotis yra griežtai draudžiama.	1
7	Ne negalima, nes šie duomenys saugomi BDAR. Jie yra asmeniniai ir be leidimo negalima jų saugoti.	1
8	Taip, nes naudojant šią informaciją (amžius, lytis, nuomonė) nėra galimybės nustatyti mano tikslios tapatybės.	1
9	Turi, nes vien pagal lytį ir žmogaus amžių negalima nustatyti žmogaus tapatybės.	1
10	Jeigu tie duomenys yra naudojami statistinių duomenų tikslams ir vartotojas pagal juos negali būti indentifikuotas. pvz panevėžio mieste gyvena 23 metų vyras.	1

5.2 klausimas

5.2. Įvardykite teisės aktą, kuriame aprašytos asmens duomenų tvarkymo ir saugojimo taisyklės. (1 taškas)

Taškai	Mokymosi turinio / Pasiekimų sritis	Kognityvinių gebėjimų grupė	Pasiekimų lygis
1	Saugus elgesys	Žinios	Slenkstinis

Taškų pasiskirstymas (%)		Sunkumas	Skiriamoji geba	Koreliacija
0	1	35,0	52,5	0,456
65,0	35,0			

5.2 klausimo atsakymų vertinimas

5.2. Įvardykite teisės aktą, kuriame aprašytos asmens duomenų tvarkymo ir saugojimo taisyklės. (1 taškas)

Vertinimo instrukcija

Bendrasis duomenų apsaugos reglamentas (BDAR arba GDPR).
Lietuvos Respublikos asmens duomenų teisinės apsaugos įstatymas.

Kandidatų atsakymai (kalba originali)		Vertinimas
1	Lietuvos Respublikos Teisės Aktai.	0
2	Lietuvos konstitucija.	0
3	Asmens duomenų tvarkymo politika	0
4	BDAR / GDPR / DBAR / BADR / BDASR	1
5	Europos Parlamento ir Tarybos reglamentas (ES) 2016/679	1
6	Bendrasis duomenų saugojimo reglamentas	1