

4. Teksto analizė ir redagavimas

Susipažinsite su:

- string klasės kintamaisiais.
- Klasės string metodais.
- Žodžių išskyrimu.
- Teksto analizės ir redagavimo elementais.

4.1. Eilutės simbolių analizė

- Raidžių pasikartojimų radimas.
- Raidžių konversijos metodų naudojimas.

Užduotis. Raidės.

Tekstiniame faile duotas tekstas. Parašykite programą, kuri surastų, kiek kartų tekste sutinkamos didžiosios ir mažosios lotyniškos raidės.

Pradiniai duomenys			
Petriukas, programuotojo sūnus, eina į pirmą klasę.			
Mokytoja vaikams aiškina:			
- Vaikai, pamokose reikia būti tyliems, jei norësite kažko paklausti, pirma pakelkite ranką.			
Petriukas iškart pakelia ranką. Mokytoja klausia:			
- Na, Petriuk, ką norėjai paklausti?			
- Nieko. Tiesiog testuoju sistemą.			
Rezultatai			
a	31		A 0
b	1		B 0
c	0		C 0
d	0		D 0
e	16		E 0
f	0		F 0
g	2		G 0
h	0		H 0
i	28		I 0
j	6		J 0
k	24		K 0
l	7		L 0
m	7		M 2
n	7		N 2
o	16		O 0
p	8		P 3
q	0		Q 0
r	13		R 0
s	16		S 0
t	16		T 1
u	10		U 0
v	1		V 1
w	0		W 0
x	0		X 0
y	3		Y 0
z	0		Z 0

Programos kūrimo eiga.

- Sukuriama klasė, kurios metoda suranda simbolių eilutėje raidžių pasikartojimų skaičius.
- Parašomas metoda, kuris pateikia duomenis raidžių dažnių suradimo klasei.
- Parašomas metoda, kuris spausdina gautus rezultatus.

¶ Pirmas žingsnis.

- Paskelbkite klasę ir parašykite metodą, kuris surastų duotoje simbolių eilutėje raidžių pasikartojimų skaičius:

```
//-----
/** Klasė, skirta skaičiuoti raidžių dažniams.
```

```

@class RaidziuDazniai */
class RaidziuDazniai
{
    private const int CMax = 256;
    private int[] Rn; // raidžių pasikartojimai
    public string eil { get; set; }
    public RaidziuDazniai()
    {
        eil = "";
        Rn = new int[CMax];
        for (int i = 0; i < CMax; i++)
            Rn[i] = 0;
    }

    public int Imti(char sim)
    {
        return Rn[sim];
    }

    /**
     * Skaičiuoja raidžių pasikartojimus
     */
    public void kiek()
    {
        for (int i = 0; i < eil.Length; i++)
        {
            if (('a' <= eil[i] && eil[i] <= 'z') ||
                ('A' <= eil[i] && eil[i] <= 'Z'))
                Rn[eil[i]]++;
        }
    }
}

```

⌚ Antras žingsnis.

- Patikrinkite, kaip dirba programa. Parašykite pgrindinį metodą `Main()`, kuris įvestų eilutę iš ekrano, ir parašykite metodą, kuris atspausdintų suskaičiuotus dažnius:

```

//-
class Program
{
    const string CFr = "...\\..\\Rezultatai.txt";

    static void Main(string[] args)
    {
        RaidziuDazniai eil = new RaidziuDazniai();
        Console.WriteLine("Įveskite eilutę iš mažujų ir didžiujų raidžių");
        string line = Console.ReadLine();
        eil.eil = line;
        eil.kiek();
        Spausdinti(CFr, eil);

        Console.WriteLine("Programa baigė darbą!");
    }

    /**
     * Spausdina į nurodytą failą raidžių dažnius dviem stulpeliais.
     * @param fv - failo vardas
     * @param eil - eilutės objektas
     */
    static void Spausdinti(string fv, RaidziuDazniai eil)
    {
        using (var fr = File.CreateText(fv))
        {
            for (char sim = 'a'; sim <= 'z'; sim++)
                fr.WriteLine("{0, 3:c} {1, 4:d} |{2, 3:c} {3, 4:d}",
                            sim, eil.Imti(sim),
                            Char.ToUpper(sim), eil.Imti(Char.ToUpper(sim)));
        }
    }
}

```

```

        }
    }
//-----

```

⌚ Trečias žingsnis.

- Parašykite teksto lotyniškų raidžių dažnių skaičiavimo metodą:

```

//-----
/** Įveda iš nurodyto failo ir skaičiuoja raidžių dažnius.
@param fv - failo vardas
@param eil - eilutės objektas */
//-----
static void Dazniai(string fv, RaidziuDazniai eil)
{
    using (StreamReader reader = new StreamReader(fv))
    {
        string line;
        while ((line = reader.ReadLine()) != null)
        {
            eil.eil = line;
            eil.kiek();
        }
    }
}
//-----

```

- Papildykite pagrindinį metodą Main() krepiniu į teksto analizės metodą, panaikinkite kodą, kuris įvesdavo vieną eilutę iš ekrano:

```

//-----
const string CFd = "...\\..\\U1.txt";
const string CFr = "...\\..\\Rezultatai.txt";

static void Main(string[] args)
{
    RaidziuDazniai eil = new RaidziuDazniai();
    Dazniai(CFd, eil);
    Spausdinti(CFr, eil);

    Console.WriteLine("Programa baigė darbą!");
}
//-----

```

- Patirkinkite, kaip dirba programa.
- Patirkinkite programos darbą su skirtingais duomenimis. Pavyzdžiui, kaip programa dirba, jeigu tekste yra tik viena raidė? Arba kai tekste nėra raidžių.

Programos papildymas.

- Papildykite programą veiksmais, kuriais rastumėte dažniausiai vartojamą tekste raidę.
- Papildykite programą veiksmais, kuriais rezultatai būtų pateikiami raidžių pasikartojimų skaičių mažėjimo tvarka.

Savarankiško darbo užduotis.

Lietuviškuose tekstuose naudojamos lietuviškos abécélės raidės, kurių nėra lotyniškoje abécélėje. Rašydami programos papildymą įvertinkite, kad lietuviškos raidės simbolių lentelėje nėra surašytos eilės tvarka. Rezultatai turi būti surikiuoti alfabeto mažėjimo tvarka.

4.2. Teksto eilučių šalinimas

- Teksto eilučių analizė, naudojant klasės string metodą Length.
- Ilgiausios eilutės paieška.
- Nurodytos eilutės pašalinimas iš teksto.

Užduotis. Ilgiausia teksto eilutė.

Tekstiniame faile duotas tekstas. Tekstą sudaro viena ir daugiau eilučių. Parašykite programą, kuri pašalintų iš teksto ilgiausią eilutę.

Pradiniai duomenys ir rezultatai.

Pradiniai duomenys
<p>Petriukas, programuotojo sūnus, eina į pirmą klasę.</p> <p>Mokytoja vaikams aiškina:</p> <ul style="list-style-type: none"> - Vaikai, pamokose reikia būti tyliems, jei norėsite kažko paklausti, pirma pakelkite ranką. <p>Petriukas iškart pakelia ranką. Mokytoja klausia:</p> <ul style="list-style-type: none"> - Na, Petriuk, ką norėjai paklausti? - Niekö. Tiesiog testuoju sistemą.
Rezultatai
<p>Petriukas, programuotojo sūnus, eina į pirmą klasę.</p> <p>Mokytoja vaikams aiškina:</p> <p>pakelkite ranką.</p> <p>Petriukas iškart pakelia ranką. Mokytoja klausia:</p> <ul style="list-style-type: none"> - Na, Petriuk, ką norėjai paklausti? - Niekö. Tiesiog testuoju sistemą.

Programos kūrimo eiga.

- Parašomas metodas, kuris suranda tekste ilgiausią eilutę.
- Parašomas metodas, kuris iš teksto pašalina nurodytą eilutę. Šalinimas vykdomas perrašant duotą tekštą į naujają failą. Šalinamoji eilutė neperrašoma.

⌚ Pirmas žingsnis.

- Parašykite metodą, kuris duotame tekste surastą ilgiausią eilutę:

```
//-----
class Program
{
    const string CFd = "..\\..\\U1.txt";
    const string CFr = "..\\..\\Rezultatai.txt";

    static void Main(string[] args)
    {
        int nr;
        Skaityti(CFd, out nr);
        Console.WriteLine("Ilgiausios eilutės nr. {0, 4:d}", nr + 1);

        Console.WriteLine("Programa baigė darbą!");
    }

    //-
    /** Suranda ilgiausios eilutės numerį.
     * @param fv - duomenų failo varda
     * @param nr - ilgiausios eilutės numeris */
    //-----
    static void Skaityti(string fv, out int nr)
    {
        string[] lines = File.ReadAllLines(fv, Encoding.GetEncoding(1257));
        int ilgis = 0;
        nr = 0;
        int nreil = 0;
        foreach (string line in lines)
        {
            if (line.Length > ilgis)
            {
                ilgis = line.Length;
                nr = nreil;
            }
            nreil++;
        }
    }
}
```

- //
- Patikrinkite, kaip dirba programa. Ekrane turi būti matomas skaičius. Tai ilgiausios eilutės numeris. Duotam duomenų pavyzdžiui tai turi būti 3, kai eilutės numeruojamos pradedant 1.

1) Antras žingsnis.

- Parašykite metodą, kuris duotame tekste pašalintų nurodytą eilutę:

```
//-----
    /** Spausdina tekštą į failą be ilgiausios eilutės.
    @param fv - duomenų failo vardas
    @param fvr - rezultatų failo vardas
    @param nr - ilgiausios eilutės numeris */
    ---
    static void Spausdinti(string fv, string fvr, int nr)
    {
        string[] lines = File.ReadAllLines(fv, Encoding.GetEncoding(1257));
        int nreil = 0;
        using (var fr = File.CreateText(fvr))
        {
            foreach (string line in lines)
            {
                if (nr != nreil)
                {
                    fr.WriteLine(line);
                }
                nreil++;
            }
        }
    }
//-----
```

- Papildykite pagrindinį metodą `Main()` kreipiniu į eilutės šalinimo metodą.

```
//-----
static void Main(string[] args)
{
    int nr;
    Skaityti(CFd, out nr);
    Spausdinti(CFd, CFr, nr);
    Console.WriteLine("Ilgiausios eilutės nr. {0, 4:d}", nr + 1);

    Console.WriteLine("Programa baigė darbą!");
}
//-----
```

- Patikrinkite, kaip dirba programa. Rezultatų faile turite matyti duoto teksto kopiją be ilgiausios eilutės.

Programos patikrinimas.

- Patikrinkite, kaip dirba programa, kai tekštą sudaro tik viena eilutė.
- Patikrinkite, kaip dirba programa, kai tekste yra kelios vienodos ilgiausios eilutės.

Programos papildymas.

Pakeiskite programą taip, kad būtų šalinamos visos ilgiausios eilutės, jeigu jų yra ne viena.

Savarankiško darbo užduotis.

Duotame tekste reikia pašalinti tuščias eilutes.

4.3. Teksto eilučių dalių šalinimas

- Teksto šalinimas, panaudojant `string` metodą `Remove()`.

Užduotis. C++ vienos eilutės komentavimas `//`.

Tekstiniame faile duotas C++ programos tekstas. Pašalinkite iš teksto komentarus, kurie žymimi `//`.

Pradiniai duomenys ir rezultatai.

Pradiniai duomenys	
<pre> void DuomenysInternet(Grybai & grybai) { ifstream fd(u2); // string pav, tip; // GrybasInfo s1; int ns = 0; bool yra = true; while(!fd.eof() && yra) { // kol yra duomenų ir jie telpa į masyva fd >> pav >> tip; s1.Deti (pav, tip); if(!fd.eof() && (ns - 1 < Grybai::CMax)) grybai[ns++] = s1; // išrašo nauja elementą else yra = false; } fd.close(); grybai.Deti(ns); } </pre>	

Rezultatai

Rezultatai	
<pre> void DuomenysInternet(Grybai & grybai) { ifstream fd(u2); int ns = 0; bool yra = true; while(!fd.eof() && yra) { fd >> pav >> tip; s1.Deti (pav, tip); if(!fd.eof() && (ns - 1 < Grybai::CMax)) grybai[ns++] = s1; else yra = false; } fd.close(); grybai.Deti(ns); } </pre>	

Analizės rezultatai

Analizės rezultatai	
<pre> // string pav, tip; // GrybasInfo s1; while(!fd.eof() && yra) { // kol yra duomenų ir jie telpa į masyva grybai[ns++] = s1; // išrašo nauja elementą </pre>	

Programos kūrimo eiga.

- Parašomas metodas, kuris įveda teksto eilutes, jas analizuojant ir formuoja rezultatų failą. Šis metodas kreipiasi į eilutės analizės metodą.
- Parašomas metodas, kuris analizuojant eilutę ir pašalina rastus komentarus.

1 Pirmas žingsnis.

- Parašykite metodą, kuris įvestų duomenis ir formuočia analizės bei rezultatų failus:

```

//-----
    /**
     * Skaito, analizuojant ir rašo į skirtinges failus.
     * @param fv - duomenų failo vardas
     * @param fvr - rezultatų failo vardas
     * @param fa - analizės failo vardas */
//-----
    static void Apdoroti(string fv, string fvr, string fa)
    {
        string[] lines = File.ReadAllLines(fv, Encoding.GetEncoding(1257));
        using (var fr = File.CreateText(fvr))
        {
            using (var far = File.CreateText(fa))
            {

```

```

        foreach (string line in lines)
        {
            if (line.Length > 0)
            {
                string nauja = line;
                if (BeKomentaru(line, out nauja))
                    far.WriteLine(line);
                if (nauja.Length > 0)
                    fr.WriteLine(nauja);
            }
            else
                fr.WriteLine(line);
        }
    }
//-----

```

1) Antras žingsnis.

- Parašykite metodą, kuris ieško ir šalina C++ komentarus, kurie žymimi //:

```

//-----
/** Pašalina iš eilutės komentarus ir grąžina požymį, ar šalino.
@param line - eilutė su komentarais
@param nauja - eilutė be komentaru */
//-----
static bool BeKomentaru(string line, out string nauja)
{
    nauja = line;
    for (int i = 0; i < line.Length - 1; i++)
        if (line[i] == '/' && line[i + 1] == '/')
    {
        nauja = line.Remove(i);
        return true;
    }
    return false;
}

```

- pagrindinis metodas Main() bus toks:

```

//-----
const string CFd = "...\\..\\Duomenys.txt";
const string CFr = "...\\..\\Rezultatai.txt";
const string CFa = "...\\..\\Analize.txt";

static void Main(string[] args)
{
    Apdoroti(CFd, CFr, CFa);

    Console.WriteLine("Programa baigė darbą!");
}
//-----

```

- Patikrinkite, kaip veikia programa.

Savarankiško darbo užduotis.

Duotame C++ programos tekste pašalinkite visus komentarus /* */ , //)

4.4. Žodžių išskyrimas eilutėje

- Žodžių išskyrimas eilutėje, panaudojant klasės string metodus.

Užduotis. Žodžių išskyrimas ir analizė.

Rasti, kiek tekste yra žodžių, kurių pirma ir paskutinė raidės vienodos. Žodžiai skiriami skyrikliais.

Pradiniai duomenys	Rezultatai
V.M. Putinas Margi sakalai Lydédami gëstančią žara vélai Pakilo į dangų margi sakalai. Paniekine žemés vylingus sapnus, Padangéje ištiesé savo sparnus. Ir taré margieji: negrišim į žemę, Kol josios kalnai ir pakalnés aptemę. ...	Žodžių skaičius: 6

Programos kūrimo eiga.

- Skaitomos teksto eilutės.
- Išskiriami žodžiai ir skaičiuojamas žodžių, kurių pirma ir paskutinė raidė vienodos, skaičius eilutėje.

1 Pirmas žingsnis.

Sukurkite metodą teksto skaitymui ir skaidymui į eilutes:

```
//-----
    /** Skaito failą ir analizuja eilutes.
     * @param fv      - duomenų failo vardas
     * @param skyrikliai - žodžių skyrikliai */
    //-----
    static int Apdoroti(string fv, char[] skyrikliai)
    {
        string[] lines = File.ReadAllLines(fv, Encoding.GetEncoding(1257));
        int sutampa = 0;
        foreach (string line in lines)
            if (line.Length > 0)
                sutampa += Zodziai(line, skyrikliai);
        return sutampa;
    }
//-----
```

2 Antras žingsnis.

- Parašykite metodą, kuris skaido eilutę į žodžius ir tikrina žodžių pirmą ir paskutinį simbolius:

```
//-----
    /** Skaido eilutę į žodžius ir analizuja žodžius.
     * @param eilute      - duomenų eilutė
     * @param skyrikliai - žodžių skyrikliai */
    //-----
    static int Zodziai(string eilute, char[] skyrikliai)
    {
        string[] parts = eilute.Split(skyrikliai,
                                       StringSplitOptions.RemoveEmptyEntries);
        int sutampa = 0;
        foreach (string zodis in parts)
            if (zodis[0] == zodis[zodis.Length - 1])
                sutampa++;
        return sutampa;
    }
//-----
```

- Patikrinkite kaip dirba programa su tekstu, esančiu faile Duomenys.txt:

```
//-----
const string CFd = "...\\..\\Duomenys.txt";

static void Main(string[] args)
{
    char[] skyrikliai = { ' ', '.', ',', '!', '?', ':', ';', '(', ')' , '\t' };
    Console.WriteLine("Sutampančių žodžių {0, 3:d}", Apdoroti(CFd, skyrikliai));

    Console.WriteLine("Programa baigė darbą!");
}
```

Programos papildymas.

- Papildykite metodą `Zoddiai()` atvejui, kai žodis prasideda ir pasibaigia skirtingo dydžio (kodo) raidėmis (pvz.: 'A' ir 'a'). Atsižvelkite tik į lotyniškos abécélės raides.

Savarankiško darbo užduotis.

Parašykite programą, kuri surastų kiek tekste yra žodžių palindromų, vienodai skaitomų iš abiejų pusių, pvz., „sūnūs“, „émé“, „iki“ ir pan.

4.5. Eilučių redagavimas

- Žodžių išskyrimas eilutėje, panaudojant klasės `String` metodą `IndexOf()` ir `Substring()`.
- Eilutės konstravimas, naudojant tipą `StringBuilder`.

Užduotis. Žodžių išskyrimas ir eilutės redagavimas.

Tekste visus vardus (pvz., Arvydas) papildyti nurodyta pavarde (pvz., SABONIS). Žodžiai skiriami skyrikliais.

Pradiniai duomenys
Arvydas (g. 1964 m. gruodžio 19 d. Kaune) – Lietuvos krepšininkas, olimpinis ir pasaulio čempionas, nuo 2011 m. spalio 24 d. Arvydas Lietuvos krepšinio federacijos prezidentas. Profesionalaus žaidėjo karjerą Arvydas pradėjo 1981 m. Kauno krepšinio klube "Žalgiris". Tris sezonus iš eilės (1985–1987 m.) Arvydas padėjo komandai iškovoti SSRS krepšinio čempionato aukso medalius. 1982 m. SSRS rinktinės sudėtyje Arvydas dalyvavo pasaulio krepšinio čempionate ir laimėjo auksą.
Rezultatai
Arvydas Sabonis (g. 1964 m. gruodžio 19 d. Kaune) – Lietuvos krepšininkas, olimpinis ir pasaulio čempionas, nuo 2011 m. spalio 24 d. Arvydas Sabonis Lietuvos krepšinio federacijos prezidentas. Profesionalaus žaidėjo karjerą Arvydas Sabonis pradėjo 1981 m. Kauno krepšinio klube "Žalgiris". Tris sezonus iš eilės (1985–1987 m.) Arvydas Sabonis padėjo komandai iškovoti SSRS krepšinio čempionato aukso medalius. 1982 m. SSRS rinktinės sudėtyje Arvydas Sabonis dalyvavo pasaulio krepšinio čempionate ir laimėjo auksą.

Programos kūrimo eiga.

- Sukuriamas metodas, kuris organizuoja darbą su viena eilute.
- Sukuriamas metodas, kuris atlieka veiksmus eilutės viduje.
- Parašomas pagrindinis metodas.

1 Pirmas žingsnis.

- Sukurkite metodą darbui su viena eilute:

```
//-----
    /**
     * Analizuojama vieną eilutę.
     * @param fd      - analizuojama eilutė
     * @param skyrikliai - žodžių skyrikliai
     * @param vardas   - žodis, kurio ieškome
     * @param pavarde   - žodis, kuriuo papildome */
    //-
    static void Apdoroti(string fd, string skyrikliai, string vardas,
                          string pavarde)
    {
        string line = fd;
        StringBuilder nauja = new StringBuilder();
        Zodziai(line, skyrikliai, vardas, pavarde, nauja);
        Console.WriteLine(nauja);
    }
//-----
```

- Sukurkite metodą darbui su vienos eilutės žodžiais:

```
//-----
    /** Ieško eilutėje žodžių ir konstruoja naujā eilutę.
     * @param line      - duomenų eilutė
     * @param skyrikliai - žodžių skyrikliai
     * @param vardas    - žodis, kurio ieškome
     * @param pavarde   - žodis, kuriuo papildome
     * @param nauja     - rezultatų eilutė */
//-----
static void Zodziai(string line, string skyrikliai, string vardas,
                     string pavarde, StringBuilder nauja)
{
    string papild = " " + line + " ";
    int prad = 1;
    int ind = papild.IndexOf(vardas);
    while (ind != -1)
    {
        if (skyrikliai.IndexOf(papild[ind - 1]) != -1
            && skyrikliai.IndexOf(papild[ind + vardas.Length]) != -1)
        {
            nauja.Append(papild.Substring(prad, ind + vardas.Length - prad));
            nauja.Append(pavarde);
            prad = ind + vardas.Length;
        }
        ind = papild.IndexOf(vardas, ind + 1);
    }
    nauja.Append(line.Substring(prad - 1));
}
```

Patikrinkite, kaip dirba programa su viena bandomaja eilute. Įveskite tekštą
Arvydas – Lietuvos krepšininkas Nr.1.

```
//-----
class Program
{
    static void Main(string[] args)
    {
        string skyr = " .,!?:;()\\t";
        string vardas = "Arvydas";
        string pavarde = " Sabonis";
        Console.WriteLine("Įveskite eilutę");
        string fv;
        fv = Console.ReadLine();
        Apdoroti(fv, skyr, vardas, pavarde);

        Console.WriteLine();
        Console.WriteLine("Programa baigė darbą!");
    }
}
//-----
```

- Ekrane turite matyti:

Arvydas Sabonis – Lietuvos krepšininkas Nr.1.

- Pakeiskite bandomąją eilutę, palikdami vieną žodį (vardą), daug vienodų žodžių (vardų) ir pan. Patikrinkite, ar gerai dirba sudarytas metodas.

Antras žingsnis.

- Parašykite metodą, kuris visame tekste atlieka aukščiau nurodytus veiksmus – žodžių (vardų) papildymą.

```
//-----
//-----
    /** Skaito failą ir analizuojasi eilutes.
     * @param fd - duomenų failo vidas
     * @param fr - rezultatų failo vidas
     * @param skyrikliai - žodžių skyrikliai
     * @param vardas - žodis, kurio ieškome
     * @param pavarde - žodis, kuriuo papildome */
//-----
```

```

    static void Apdoroti(string fd, string fr, string skyrikliai,
                          string vardas, string pavarde)
    {
        string[] lines = File.ReadAllLines(fd, Encoding.GetEncoding(1257));
        using (var far = File.CreateText(fr))
        {
            foreach (string line in lines)
            {
                StringBuilder nauja = new StringBuilder();
                Zodziai(line, skyrikliai, vardas, pavarde, nauja);
                far.WriteLine(nauja);
            }
        }
    }
//-----

```

- Patikrinkite, kaip dirba programa su tekstu, esančiu faile Duomenys.txt:

```

//-----
class Program
{
    const string CFd = "..\\..\\Duomenys.txt";
    const string CFr = "..\\..\\Rezultatai.txt";

    static void Main(string[] args)
    {
        string skyr = " .,!?:;()\\t'";
        string vardas = "Arvydas";
        string pavarde = " Sabonis";
        Apdoroti(CFd, CFr, skyr, vardas, pavarde);

        Console.WriteLine("Programa baigė darbą!");
    }
//-----

```

Programos papildymas.

- Papildykite metodą Zodziai() taip, kad papildomas žodis būtų įrašomas, paliekant tarp žodžių tarpelį (eilutės pavarde pradžioje esanti tarpelis prieš tai išmeskite).

Savarankiško darbo užduotis.

- Parašykite programą, kuri pašalintų iš teksto nurodytus žodžius kartu su už jų esančiais skyrikliais.

4.6. Analizės failo sukūrimas

- Analizės failo, kuris padėtų sekti teksto keitimo eiga, sukūrimas.
- Klasės string konstruktoriaus panaudojimas eilutės užpildymui simboliais – linijos brėžimui, metodo Replace panaudojimas.

Užduotis. Analizės failo, kuriame bus atspindėti tarpiniai veiksmai, sukūrimas.

Tekstiniame faile duotas tekstas. Žodžiai iš eilutės į eilutę nekeliami. Skyrikliai žinomi. Pašalinti iš kiekvienos eilutės ilgiausio žodžio (vieno) visas balses.

Pradiniai duomenys
Kūčių ryta
<p>Anksti Kūčių ryta šeimininkė budina savo vyra:</p> <p>- Eik greičiau, saulei netekėjus, kur dalgės kabos, ištverk dalges. Dalges padék po stogu, o dalgiaukočius sudék svirnan.</p> <p>Šeimininkė ieško kubilo lanko, kad būt visai apskritas, nepertrūkės niekur. Ta lanka neša vištų tvartan, vidury tvarto paguldo. O tada šeimininkė skuba tvartant prie kodžio, kur būna žirniai supilti. Šeimininkė tuos žirnus semia negailėdama didžiuli gorčią, kad visos vištос prilestu lig soties. Šeimininkė pila tuos žirnus tan kubilo lankan, kad nei vienas žirnis nebūt už kubilo lanko - kad vištос visos détu kiaušinius vienan daiktan, nemėtytū kiaušinių. Berdama žirnus tan lankan, šeimininkė garsiai sako vištom:</p> <p>- Žiūrėkite, kad nei vieno kiaušinio, nei vieno niekur nepamestut, visus vienon vieton dékite!</p> <p>Na ir visos vištос šeimininkės įsakymą vykdo.</p>

Analizė			
<hr/>			
Ilgiausias žodis Pradžia Ilgis			
- - - - -			
Kūčių	0	5	
šeimininkė	20	10	
netekėjus	25	9	
dalgiakočius	18	12	
nepertrükės	58	11	
šeimininkė	59	10	
Šeimininkė	48	10	
negailėdama	0	11	
žirnius	10	7	
kiaušinius	30	10	
šeimininkė	28	10	
nepamestut	56	10	
vienon	0	6	
šeimininkės	19	11	
- - - - -			
Rezultatai			
Kč ryta			
Anksti Kūčių ryta šmnnk budina savo vyra:			
- Eik greičiau, saulei ntkjs, kur dalgés kabø, ištverk dalges. Dalges padék po stogu, o dlkčs sudék svirnan.			
Šeimininké ieško kubilo lanko, kad būt visai apskritas, nprtrks niekur. Ta lanka neša vištų tvartan, vidury tvarto paguldo. O tada šmnnk skuba tvartant prie kodžio, kur būna žirniai supilti. Šmnnk tuos žirnius semia ngldm didžiuli gorčiu, kad visos vištос prilestu lig soties. Šeimininké pila tuos žrns tan kubilo lankan, kad nei vienas žirnis nebūt už kubilo lanko - kad vištос visos détu kšns vienan daiktan, nemetytū kiaušinių. Berdama žirnius tan lankan, šmnnk garsiai sako vištom:			
- Žiürékite, kad nei vieno kiaušinio, nei vieno niekur npmstt, visus vnn vieton dékite!			
Na ir visos vištос šmnnks ısakymą vykdo.			

Programos kūrimo eiga.

- Metodas `Ilgiausias()` ieško eilutėje ilgiausio žodžio ir grąžina jį.
- Metodas `BeBalsiu()` iš ilgiausio žodžio šalina balses.
- Metodas `Apdoroti()` skaito tekštą po eilutę, kviečia metodus `Ilgiausias()` ir `BeBalsiu()` bei formuoja rezultatų ir analizės failus.

1 Pirmas žingsnis.

- Sukurkite metodą ilgiausio žodžio paieškai eilutėje:

```
//-
    /** Ieško ilgiausio žodžio eilutėje ir grąžina rezultatą per vardą.
     * @param eilute - duomenų eilutė
     * @param skyrikliai - žodžių skyrikliai */
    //-
    static string Ilgiausias(string eilute, char[] skyrikliai)
    {
        string[] parts = eilute.Split(skyrikliai,
            StringSplitOptions.RemoveEmptyEntries);
        string ilgiaus = "";
        foreach (string zodis in parts)
            if (zodis.Length > ilgiaus.Length)
                ilgiaus = zodis;
        return ilgiaus;
    }
//-
```

2 Antras žingsnis.

- Sukurkite metodą, kuris iš duoto žodžio pašalintų nurodytus simbolius:

```
//-
    /** IŠ žodžio pašalima balses ir grąžina rezultatą pervardą.
     * @param eilute - žodis su balsėmis
     * @param balses - abécélės balsės */
    //-
    static StringBuilder BeBalsiu(string eilute, string balses)
    {
        StringBuilder nauja = new StringBuilder();
        for (int i = 0; i < eilute.Length; i++)
            if (balses.IndexOf(eilute[i]) == -1)
                nauja.Append(eilute[i]);
        return nauja;
    }
//-
```

⌚ Trečias žingsnis.

- Parašykite metodą `Apdoroti()` viso darbo organizavimui:

```
//-
    /** Skaito failą, analizuojant eilutes, kuria rezultatų ir analizés failus.
     * @param fd          - duomenų failo vardas
     * @param fr          - rezultatų failo vardas
     * @param fa          - analizés failo vardas
     * @param skyrikliai - žodžių skyrikliai
     * @param balses      - abécélės balsės */
    //-
    static void Apdoroti(string fd, string fr, string fa, char[] skyrikliai,
                          string balses)
    {
        string[] lines = File.ReadAllLines(fd, Encoding.GetEncoding(1257));
        string eilute = new String('-', 38);
        using (var far = File.CreateText(fr))
        {
            using (var faa = File.CreateText(fa))
            {
                faa.WriteLine(eilute);
                faa.WriteLine("| Ilgiausias žodis | Pradžia | Ilgis |");
                faa.WriteLine(eilute);
                foreach (string line in lines)
                    if (line.Length > 0)
                    {
                        string ilgiaus = Ilgiausias(line, skyrikliai);
                        string ilgiausBe = BeBalsiu(ilgiaus, balses).ToString();
                        faa.WriteLine("| {0,-16} | {1, 7:d} | {2, 5:d} |",
                                     ilgiaus, line.IndexOf(ilgiaus), ilgiaus.Length);
                        string nauja = line.Replace(ilgiaus, ilgiausBe);
                        // trumpiausio taip pakeisti negalima,
                        // gali būti kito žodžio dalimi, sprendimas 5.5 poskyryje
                        far.WriteLine(nauja);
                    }
                else
                    far.WriteLine(line);
                faa.WriteLine(eilute);
            }
        }
//-
```

- Analizés faile rezultatus kaupsime lentelėje. Lentelės linijų brėžimui skelbiame `string` kintamąjį su konstruktoriu:

```
string eilute = new string('-', 38);
```

- Toks paskelbimas užpildo eilutės kintamąjį 38 brūkšneliais. Privalumai yra tokie:

- nereikia brėžti ilgos brūkšninės linijos;

- kintamąjį paskelbėme vieną kartą, o galėsime naudoti tiek kartų, kiek reikia.

⌚ Ketvirtas žingsnis.

- Parašykite pagrindinį metodą `Main()`:

```
//-----
class Program
{
    const string CFd = "..\\..\\Duomenys.txt";
    const string CFr = "..\\..\\Rezultatai.txt";
    const string CFa = "..\\..\\Analize.txt";

    static void Main(string[] args)
    {
        const string balses = "AEIYOUaeiyouAqEeEiUuUu";
        char[] skyrikliai = { ' ', '.', ',', '!', '?', ':', ';', '(', ')', '\t' };
        Apdoroti(CFd, CFr, CFa, skyrikliai, balses);

        Console.WriteLine("Programa baigė darbą!");
    }
}
//-----
```

4.7. Kontroliniai klausimai

- Kokie metodai naudojami žodžio išskyrimui eilutėje?
- Kaip reikia kreiptis į eilutę, norint išrinkti vieną eilutės simbolį?
- Koks tipas yra naudojamas kintamojo, kuris galėtų saugoti tik vieną simbolį, paskelbimui?
- Kokiu būdu galėtume patalpinti simbolį į nurodytą eilutės vietą?
- Užrašykite ne mažiau trijų skirtingų būdų, kaip suteikti pradines reikšmes eilutei.
- Ką vadiname žodžiu?
- Ar galima lyginti dvi eilutes tarpusavyje? Jei taip, pateikite pavyzdį su dviem eilutėmis ir pasakykite, kuri eilutė yra didesnė.
- Kokiam tikslui naudojamas metodas `IndexOf`?
- Kaip užrašomas `string` konstruktorius, kuris leidžia užpildyti eilutę nurodytu simboliu?
- Kokiam tikslui naudojamas `string` eilutės `Split` metodas?
- Kokiam tikslui `string` eilutės `Split` metode naudojamas parametras `StringSplitOptions.RemoveEmptyEntries`?
- Koks turi būti skyriklių tipas `string` klasės `Split` metodui?
- Ar gali metodai `Remove`, `Insert`, `Replace` pakeistą eilutę priskirti buvusiai eilutei? Atsakymą pakomentuokite.
- Kada reikia naudoti `StringBuilder` klasę? O kodėl tuo tikslu negalima panaudoti `string` klasės?

Užduotys

U4-1. Ilgiausias žodis

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Pašalinkite kiekvienos eilutės ilgiausią žodį. Jei yra keli ilgiausi žodžiai, tuomet reikia pašalinti juos visus. Skyrikliai nešalinami.

U4-2. Trumpiausias žodis, prasidedantis balse

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Pašalinkite kiekvienos eilutės trumpiausią žodį, kuris prasideda balse. Jei yra keli trumpiausi žodžiai, tuomet reikia pašalinti juos visus. Skyrikliai nešalinami.

U4-3. Žodžių poros

Tekstiniame faile pateiktas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės Skyrikliai žinomi. Kiek yra žodžių porų, kuriose vieno žodžio paskutinė raidė sutampa su sekančio žodžio pirmaja raide? Kiekvienoje eilutėje raskite ilgiausią (daugiausia simbolių) iš šių porų ir perkelkite jos antrajį žodį į eilutės pradžią kartu su už jo esančiais skyrikliais. Jei yra kelios ilgiausios poros, nagrinėkite tik pirmą iš jų.

U4-4. Trumpiausias žodis, pasibaigiantis balse

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Pašalinkite kiekvienos eilutės trumpiausią žodį, kuris pasibaigia balse. Jei yra keli trumpiausi žodžiai, tuomet reikia pašalinti juos visus. Skyrikliai nešalinami.

U4-5. Nelyginis simbolių skaičius

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Pašalinkite visus žodžius, sudarytus iš nelyginio simbolių skaičiaus. Kartu šalinami ir visi skyrikliai už jų.

U4-6. Lyginis nesikartojančių žodžių skaičius

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Eilutėse, kuriose yra lyginis nesikartojančių žodžių skaičius n, n / 2 žodį pakeiskite žodžiu „darbas“.

U4-7. Lyginis simbolių skaičius

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Pašalinkite visus žodžius, sudarytus iš lyginio simbolių skaičiaus. Kartu šalinami ir visi skyrikliai už jų.

U4-8. Ilgesni žodžiai

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Kiekvienos eilutės pradžioje parašykite skaičių, kiek eilutėje yra žodžių, ilgesniu už pirmajį eilutės žodį. Už skaičiaus įrašykite tarpo simbolį. Teksto pabaigoje naujoje eilutėje parašykite eilutės, kurioje daugiausiai žodžių, kopiją. Jei yra kelios eilutės su vienodu didžiausiu žodžių skaičiumi, reikia parašyti jų visų kopijas.

U4-9. Vienodi simboliai

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Už kiekvieno žodžio skliausteliuose užrašyti to žodžio ilgį ir kiek vienodų simbolių yra tame žodyje. Tarp skaičių įrašykite tarpo simbolį. Didžiosios ir mažosios raidės laikomos vienodais simboliais.

U4-10. Skaitmenys

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Kiekvienoje eilutėje po vieną (pirmą) ilgiausią žodį, sudarytą tik iš skaitmenų, jei toks yra, kartu su už jo esančiais skyrikliais perkelkite į eilutės pradžią.

U4-11. Skirtingos raidės

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Kiek yra kiekvienoje eilutėje žodžių, kurie baigiasi ir prasideda ta pačia raide. Kuris iš šių žodžių turi daugiausiai skirtingų raidžių? Perkelkite šį žodį kartu su už jo esančiais skyrikliais į eilutės pradžią. Jei yra keli daugiausiai skirtingų raidžių turintys žodžiai, imkite pirmą iš jų.

U4-12. Ilgiausio žodžio simboliai

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. I kiekvienos eilutės pabaigą perkelkite visus žodžius, turinčius bent vieną ilgiausio eilutės žodžio simbolį, išskyrus jį patį, kartu su už jų esančiais skyrikliais. Jei yra keli ilgiausiai žodžiai, imkite pirmą iš jų.

U4-13. Dažniausiai pasikartojantis simbolis

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Kiekvienos eilutės kiekviename žodyje raskite dažniausiai (>1) pasikartojantį simbolį.

Trumpiausių eilutės žodžių, turintį vieną iš tokiu simboliu, kartu su už jo esančiais skyrikliais, pašalinkite iš eilutės. Jei yra keli trumpiausi žodžiai, imkite pirmą iš jų.

U4-14. Eilučių išlyginimas

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Išlyginti teksto dešinijį kraštą pagal ilgiausią eilutę. Iterpiamus tarpus tarp žodžių paskirstyti tolygiai. Tekstą pertvarkyti taip, kad žodis, einantis po taško, prasidėtų didžiaja raide. Didžiąją raide privalo prasidėti ir pirmasis teksto žodis.

U4-15. Balsės

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. I kiekvienos eilutės pradžią perkelkite po vieną žodį, turintį bent vieną ilgiausio eilutės žodžio balsę, jei toks yra, išskyrus jį patį, kartu su už jo esančiais skyrikliais. Jei yra keli ilgiausi žodžiai, imkite pirmą iš jų.

U4-16. Skirtingi simboliai

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Pašalinkite iš teksto žodžius, sudarytus tik iš skirtingų simbolų. Žodžiai šalinami kartu su už jo esančiais skyrikliais.

U4-17. Trumpiausias žodis

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. I kiekvienos eilutės pradžią perkelkite po vieną žodį, turintį bent vieną trumpiausio eilutės žodžio, kuris yra netrumpesnis už nurodytą ilgi, balsę, jei toks yra, išskyrus jį patį. Kelkite kartu su už jo esančiais skyrikliais. Jei yra keli trumpiausi žodžiai, kurie yra netrumpesni už nurodytą ilgi, imkite pirmą iš jų.

U4-18. Teksto formato keitimas

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Duotas tekstas, prasideda žodžiu "Begin" ir baigiasi žodžiu "End". Jie užima atskiras eilutes. Likusį tekstą sudaro tokio formato eilutės: **žodis1 = žodis2 (žodis3, žodis4,..., žodisn)**. Tokio formato tekstas gali užimti keletą eilučių. Pakeiskite šį teksto formatą į tokį: žodis2 (žodis1, žodis3, žodis4,..., žodisn). Atspausdinkite eilučių numerius, kur tokio formato sakinys užima keletą eilučių.

U4-19. Tituliniai žodžiai

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. I atskirą failą surašykite žodžius, kurie prasideda didžiaja raide, o iš senojo failo juos pašalinkite kartu su skyrikliais už jų. Jei naujame faile netelpa į eilutę (eilutės ilgis - 81 simbolis), formuokite naują eilutę. Skyrikliai tarp žodžių naujame faile yra kablelis ir taškas.

U4-20. Skaičių sumos

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Ar yra žodžių, kuriuos sudaro tik skaitmenys? Jei yra, raskite šią skaitmenų seką atitinkančių skaitmenų sumą ir ją išrašykite vietoj pradinės skaitmenų sekos.

U4-21. Didžiosios raidės

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Ilgiausią eilutės žodį iš tų žodžių, kurie prasideda didžiaja raide, perkelkite į eilutės pradžią kartu su už jo esančiais skyrikliais. Jei yra keli, kelkite paskutini.

U4-22. Žodžių poros

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Kiek yra žodžių porų, kuriose vieno žodžio paskutinė raidė sutampa su tolimesnio žodžio pirmaja raide? Kiekvienoje eilutėje raskite ilgiausią (daugiausia simbolų) iš šių porų ir perkelkite jos

pirmajį žodį kartu su jo už esančiais skyrikliais į eilutės pabaigą. Jei ilgiausios poros yra kelios, imkite pirmą iš jų.

U4-23. „Didesnės“ raidės

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Kokių žodžių eilutėje daugiau: tų, kurių pirmoji raidė “didesnė” už paskutinią ar tų, kurių pirmoji raidė “mažesnė” už paskutinią? Kiekvienos eilutės atsakymą išrašykite į naują failą. Duotame faile pašalinkite žodžius, kurių pirmosios dvi raidės sutampa su dviem paskutinėmis raidėmis. Skyrikliai nešalinami.

U4-24. Baziniai žodžiai

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Kitame faile duotas bazinių žodžių sąrašas, kur žodžiai surašyti stulpeliu. Kiekvienoje eilutėje raskite žodį, turintį daugiausiai skirtingų simbolių ir pašalinkite jį iš eilutės kartu su už jo esančiais skyrikliais. Bazinių žodžių nenagrinėkite. Jei yra keli žodžiai, turintys daugiausiai skirtingu simbolių, imkite pirmą iš jų.

U4-25. Lyginis žodžių simbolių skaičius

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Eilutės žodžiuose, kuriuose simbolių skaičius yra lyginis, balses pakeiskite didžiosiomis raidėmis. Konsolėje atspausdinkite paskutinę eilutę, kurioje daugiausiai žodžių su lyginiu simbolių skaičiumi. Visas failas į masyvą netelpa.

U4-26. Žodžių balsių ir priebalsių sutapimas

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Eilutės žodžius, kuriuose balsių skaičius lygus priebalsių skaičiui, užrašykite didžiosiomis raidėmis. Raskite pirmą eilutę, kurioje buvo daugiausiai tokų žodžių sutapimų. Ją atspausdinkite konsolėje. Visas failas į masyvą netelpa.

U4-27. Skirtingi simboliai

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Po kiekvieno žodžio skliausteliuose užrašykite to žodžio ilgį ir kiek skirtingu simbolių yra tame žodyje. Skaičius skirkite tarpo simboliu. Raskite pirmą daugiausiai skirtingu simbolių turintį žodį ir kurioje eilutėje jis yra. Surastus dydžius atspausdinkite konsolėje. Visas failas į masyvą netelpa.

U4-28. Vienodi simboliai

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Pašalinkite iš teksto žodžius, kurie turi nurodytą skaičių n vienodų simbolių „s“. Skyrikliai nešalinami. Raskite paskutinę eilutę, iš kurios buvo pašalinta daugiausiai tokų žodžių. Ją atspausdinkite konsolėje. Visas failas į masyvą netelpa.

U4-29. Ilgiausi žodžiai

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Pašalinkite iš kiekvienos eilutės po du ilgiausius žodžius su už jų esančiais skyrikliais. Pašalintų žodžių vietą pažymėkite žvaigždute (*).

U4-30. Trumpiausi žodžiai

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Kiekvienoje eilutės du trumpiausius žodžius, ne trumpesnius kaip du simboliai, dubliuoti, tarp dubliuotų žodžių išrašant minuso ženkla (-).

U4-31. Ilgiausių žodžių perkėlimas

Tekstiniame faile pateikiamas tekstas. Žodžiai iš eilutės į kitą eilutę nekeliami. Žodžiai eilutėse skiriami skyrikliais. Skyrikliai gali būti eilutės pradžioje bei gale, gali būti tuščios eilutės. Skyrikliai žinomi. Kiekvienos eilutės pirmą ilgiausią žodį su už jo esančiais skyrikliais perkelkite į eilutės pradžią. Perkelto žodžio vietoje išrašykite žvaigždutę (*).