

5. Teksto analizė ir redagavimas

Susipažinsite su:

- Klasių `string` ir `char` metodais bei savybėmis.
- Žodžių išskyrimu.
- Teksto analizės ir redagavimo elementais.
- Reguliariosiomis išraiškomis.

5.1. 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 | | | | |
|--|----|--|---|---|
| Draugas draugui sako: | | | | |
| - Ką pasakytum, jei sutiktum moterį, kuri visada būtų švelni ir maloni, viską tau atleistų, tave dievintų. | | | | |
| - Tokių nebūna. | | | | |
| - Būna. Reiktų pasakyti: | | | | |
| - Labas, mama! | | | | |
| Rezultatai | | | | |
| a | 20 | | A | 0 |
| b | 3 | | B | 1 |
| c | 0 | | C | 0 |
| d | 3 | | D | 1 |
| e | 8 | | E | 0 |
| f | 0 | | F | 0 |
| g | 2 | | G | 0 |
| h | 0 | | H | 0 |
| i | 15 | | I | 0 |
| j | 1 | | J | 0 |
| k | 8 | | K | 1 |
| l | 3 | | L | 1 |
| m | 6 | | M | 0 |
| n | 6 | | N | 0 |
| o | 4 | | O | 0 |
| p | 2 | | P | 0 |
| q | 0 | | Q | 0 |
| r | 5 | | R | 1 |
| s | 9 | | S | 0 |
| t | 12 | | T | 1 |
| u | 8 | | U | 0 |
| v | 5 | | V | 0 |
| w | 0 | | W | 0 |
| x | 0 | | X | 0 |
| y | 2 | | Y | 0 |
| z | 0 | | Z | 0 |

Programos kūrimo eiga.

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

U Pirmas žingsnis.

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

```
//-----
/** Klasė, skirta skaičiuoti raidžių dažnius. */
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 Main() metodą, kuris įvestų eilutę iš ekrano, ir parašykite metodą, kuris atspausdintų suskaičiuotus dažnius:

```

//-----
static void Main(string[] args)
{
    const string CFd = "..\\..\\Duomenys.txt";
    const string CFr = "..\\..\\Rezultatai.txt";
    RaidziuDazniai eil = new RaidziuDazniai();
    Console.WriteLine("Įveskite eilutę iš mažųjų ir didžiųjų raidžių");
    string line = Console.ReadLine();
    eil.eil = line;
    eil.kiek();
    Spausdinti(CFr, eil);
}
//-----
/** 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 `Main()` metodą kreipiniu į parašytą metodą, panaikinkite kodą, kuris įvesdavo vieną eilutę iš ekrano.
- Patikrinkite, kaip dirba programa.
- Patikrinkite 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.

1 savarankiško darbo užduotis (nebūtina atlikti).

Lietuviškuose tekstuose naudojamos lietuviškos abėcėlės raidės, kurių nėra lotyniškoje abėcėlėje. Rašydami programos papildymą įvertinkite, kad lietuviškos raidės simbolių lentelėje nėra surašytos eilės tvarka ir jų kodai viršija 256. Nustatykite lietuviškų raidžių kodus.

5.2. Teksto eilučių šalinimas

- 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>Aš atverčiau knygą kur pakliuvo. Raidžių formos man buvo niekur nematytos. Puslapiai apšiurę, raidės neryškios, tekstas ėjo dviem skiltimis. Šriftas buvo tankus ir padalytas į pastraipas. Ir mažas paveikslėlis, toks, koks pasitaiko žodynuose: plunksna nupieštas inkaras, bet lyg negrabia vaiko ranka.</p> <p>Jorge Luis Borges. Smėlio knyga. Vilnius, 2006.</p> |
| Rezultatai |
| <p>Aš atverčiau knygą kur pakliuvo. Raidžių formos man buvo niekur nematytos. Šriftas buvo tankus ir padalytas į pastraipas. Ir mažas paveikslėlis, toks, koks pasitaiko žodynuose: plunksna nupieštas inkaras, bet lyg negrabia vaiko ranka.</p> <p>Jorge Luis Borges. Smėlio knyga. Vilnius, 2006.</p> |

Programos kūrimo eiga.

- Parašoma klasė, kurioje saugomas tekstas kaip eilučių rinkinys.
- Parašomas klasės metodas, kuris tekste pašalina nurodytą eilutę. Šalinimas vykdomas perrašant duotą tekstą į naują eilutę. Šalinamoji eilutė neperrašoma.
- Parašomas metodas, kuris suranda ilgiausios teksto eilutės numerį.
- Parašomas metodas, kuris iš vieno failo perrašo tekstą į kitą failą be ilgiausios eilutės.
- Parašomas `Main()` metodas, kuris iškviečia anksčiau minėtus metodus.

U Pirmas žingsnis.

- Parašykite metodą ilgiausios teksto eilutės numerio suradimui:

```
//-----
/** Suranda ilgiausios eilutės numerį.
 * @param fv - duomenų failo vardas
 * @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++;
    }
}
//-----
```

U Antras žingsnis.

- Parašykite metodą teksto perrašymui iš vieno failo į kitą 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++;
        }
    }
}
//-----
```

- Parašykite Main() metodą:

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

- Patikrinkite, kaip dirba programa. Ekrane turi būti matomas skaičius. Tai ilgiausios eilutės numeris. Duotam duomenų pavyzdžiui tai turi būti 3. Rezultatų faile turite matyti duoto teksto kopiją be ilgiausios eilutės.

Programos patikrinimas.

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

Programos pakeitimas.

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

5.3. 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 masyvą fd >> pav >> tip; s1.Dėti (pav, tip); if(!fd.eof() && (ns - 1 < Grybai::CMax)) grybai[ns++] = s1; // įrašo naują elementą else yra = false; } fd.close(); grybai.Dėti(ns); }</pre> |
| Galutiniai rezultatai |
| <pre>void DuomenysInternet(Grybai & grybai) { ifstream fd(u2); int ns = 0; bool yra = true; while(!fd.eof() && yra) { fd >> pav >> tip; s1.Dėti (pav, tip); if(!fd.eof() && (ns - 1 < Grybai::CMax)) grybai[ns++] = s1; else yra = false; } fd.close(); grybai.Dėti(ns); }</pre> |
| Analizės rezultatai |
| <pre>// string pav, tip; // GrybasInfo s1; while(!fd.eof() && yra) { // kol yra duomenų ir jie telpa į masyvą grybai[ns++] = s1; // įrašo naują elementą</pre> |

Programos kūrimo eiga.

- Parašomas metodas, kuri įveda teksto eilutes, formuoja rezultatų ir analizės failus.
- Parašomas metodas, kuris eilutėje ieško komentarų. Jei randa, pašalina

Pirmas žingsnis.

- Parašykite metodą, kuris įveda teksto eilutes, jas analizuoja, formuoja analizės ir rezultatų failus

```
//-----
/** Skaito, analizuoja ir rašo į skirtingus 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);
            }
        }
    }
}
//-----
```

🔹 Antras žingsnis.

- Parašykite metodą, kuris ieško eilutėje komentarų ir juos pašalina

```
//-----
/** Pašalina iš eilutės komentarus ir gražina požymį, ar šalino.
 * @param line - eilutė su komentarais
 * @param nauja - eilutė be komentarų */
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;
}
//-----
```

🔹 Trečias žingsnis.

- Parašykite Main() metodą:

```
//-----
static void Main(string[] args)
{
    const string CFd = "..\\..\\Duomenys.txt";
    const string CFr = "..\\..\\Rezultatai.txt";
    const string CFa = "..\\..\\Analyze.txt";
    Apdoroti(CFd, CFr, CFa);
}
//-----
```

- Patikrinkite, kaip veikia programa.

2 savarankiško darbo užduotis.

Duotame C# programos tekste pašalinkite visus komentarus (/* */ , //)

5.4. Žodžių išskyrimas eilutėje

- Žodžių išskyrimas eilutėje, panaudojant `string` klasės metodus.
- Žodžių išskyrimas eilutėje, panaudojant `Regex` klasės metodus.

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

| Pradiniai duomenys | Rezultatai |
|---|--------------------|
| V. M. Putinas Margi sakalai Lydėdami gėstančią žarą vėlai Pakilo į dangų;;, margi sakalai. Paniekinę žemės vylingus sapnus, Padangėje ištiesė,,; savo sparnus. Ir tarė margieji: negrišim į žemę, Kol josios kalnai ir pakalnės aptemę. ... | Žodžių skaičius: 6 |

Programos kūrimo eiga.

- Sukuriamas metodas, kuris skaito tekstą iš failo ir skaido jį į eilutes. Žodžių skyrikliai metodui perduodama per parametą.
- Sukuriamas metodas, kuris skaido eilutes į žodžius ir skaičiuoja žodžių, kurių pirma ir paskutinė raidės sutampa, kiekį.
- Tokia pati programos struktūra taikoma tiek `string` metodų naudojimo atveju, tiek `Regex` metodų naudojimo atveju.

➊ Pirmas žingsnis.

- Sukurkite metodą teksto skaitymui ir skaidymui į eilutes:

```
//-----
/** Skaito failą ir analizuoja 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;
}
//-----
```

➋ Antras žingsnis.

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

```
//-----
/** Skaido eilutę į žodžius ir analizuoja 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`:

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

U Trečias žingsnis.

- Parašykite metodus, naudodami `Regex` klasę. Šiai klasei įtraukti reikia papildomos direktyvos:

```
using System.Text.RegularExpressions;

//-----
static void Main(string[] args)
{
    const string CFd = "..\\..\\Duomenys.txt";
    string skirt = "[\\s,.;!()?\\-]+"; // skyrikliai tarp žodžių
    Console.WriteLine("Sutampančių žodžių {0, 3:d}", Apdoroti(CFd, skirt));
}
//-----
/** Skaity failą ir analizuoja eilutes.
    @param fv - duomenų failo vardas
    @param skyrikliai - žodžių skyrikliai */
static int Apdoroti(string fv, string 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;
}
//-----
/** Skaity eilutę į žodžius ir analizuoja žodžius.
    @param eilute - duomenų eilutė
    @param skyrikliai - žodžių skyrikliai */
static int Zodziai(string eilute, string skyrikliai)
{
    string[] parts = Regex.Split(eilute, skyrikliai);
    int sutampa = 0;
    foreach (string zodis in parts)
        if (zodis.Length > 0) // yra tuščių žodžių eilučių pabaigoje
            if (zodis[0] == zodis[zodis.Length - 1])
                sutampa++;
    return sutampa;
}
//-----
```

Programos papildymas.

- Papildykite metodus abeiose programose tam atvejui, kai žodis prasideda ir pasibaigia skirtingo dydžio (kodo) raidėmis (pvz.: 'A' ir 'a'), tačiau jas reikėtų laikyti sutampančiomis.

3 savarankiško darbo užduotis (nebūtina atlikti).

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

5.5. Eilutės konstravimas

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

🔗Pirmas žingsnis.

- Sukurkite metodą darbui su viena eilute:

```
//-----
/** Analizuoja vieną eilutę.
 * @param fd - analizuojama eilutė
 * @param fr - rezultatų failo vardas
 * @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 line = fd;
    {
        StringBuilder nauja = new StringBuilder();
        Zodziai(line, skyrikliai, vardas, pavarde, nauja);
        Console.WriteLine(nauja);
    }
}
//-----
```

🔗Antras žingsnis.

- Sukurkite metodą darbui su viena eilute:

```
//-----
/** Ieško eilutėje žodžių ir konstruoja naują 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 eilute:

```

//-----
static void Main(string[] args)
{
    const string CFd = "..\\..\\Duomenys.txt";
    const string CFr = "..\\..\\Rezultatai.txt";
    string skyr = " .,!?;:()\\t'";
    string vardas = "Arvydas";
    string pavarde = "Sabonis";
    Console.WriteLine("Įveskite eilutę");
    string fv;
    fv = Console.ReadLine();
    Apdoroti(fv, CFr, skyr, vardas, pavarde);
}
//-----

```

- Ekrane turite matyti metodo darbo rezultatus:
- Atlikite keletą įvedimo variantų, palikdami vieną žodį (vardą), daug vienodų žodžių (vardų) ir pan. Patikrinkite, ar gerai dirba sudarytas metodas.

🔹 Trečias žingsnis.

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

```

//-----
/** Skaito failą ir analizuoja eilutes.
    @param fd - duomenų failo vardas
    @param fr - rezultatų failo vardas
    @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:

```

//-----
static void Main(string[] args)

```

```

{
    const string CFd = "..\\..\\Duomenys.txt";
    const string CFr = "..\\..\\Rezultatai.txt";
    string skyr = " .,!?;:()\\t'";
    string vardas = "Arvydas";
    string pavarde = "Sabonis";
    Apdoroti(CFd, CFr, skyr, vardas, pavarde);
}
//-----

```

Programos papildymas.

- Papildykite metodą `Zodziai()` taip, kad papildomas žodis būtų įrašomas, paliekant tarp žodžių tarpelį.

4 savarankiško darbo užduotis.

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

5.6. Analizės failo sukūrimas

- Analizės failo, kuris padėtų sekti teksto keitimo eigą, 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 rytą | | | |
| <p>Anksti Kūčių rytą šeimininkė budina savo vyrą:</p> <p>- Eik greičiau, saulei netekėjus, kur dalgės kabo, ištverk dalges. Dalges padėk po stogu, o dalgiakočius sudėk svirnan.</p> <p>Šeimininkė ieško kubilų lanko, kad būtų visai apskritas, nepertrūkės niekur. Tą lanką neša vištų tvartan, vidury tvarto paguldo. O tada šeimininkė skuba tvartan prie kodžio, kur būna žirniai supilti. Šeimininkė tuos žirnius semia negailėdama didžiulį gorčių, kad visos vištos prilestų lig soties. Šeimininkė pila tuos žirnius tan kubilų lankan, kad nei vienas žirnis nebūt už kubilų lanko - kad vištos visos dėtų kiaušinius vienan daiktan, nemėtytų kiaušinių. Berdama žirnius tan lankan, šeimininkė garsiai sako vištom:</p> <p>- Žiūrėkite, kad nei vieno kiaušinio, nei vieno niekur nepamestut, visus vienon vieton dėkite!</p> <p>Na ir visos vištos šeimininkės įsakymą vykdo.</p> | | | |
| Analizė | | | |
| ----- | | | |
| Ilgiausias žodis | Pradžia | Ilgis | |
| ----- | | | |
| Kūčių | 0 | 5 | |
| šeimininkė | 20 | 10 | |
| netekėjus | 25 | 9 | |
| dalgiakočius | 18 | 12 | |
| nepertrūkės | 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 rytą | | | |
| Anksti Kūčių rytą šmnk budina savo vyrą: | | | |

- Eik greičiau, saulei ntkjs, kur dalgės kabo, ištverk dalges. Dalges padėk po stogu, o dlgekės sudėk svirnan.

Šeimininkė ieško kubilų lanko, kad būtų visai apskritas, nprtrks niekur. Tą lanką neša vištų tvartan, vidury tvarto paguldo. O tada šmnk skuba tvartant prie kodžio, kur būna žirniai supilti. Šmnk tuos žirnius semia ngldm didžiulį gorčių, kad visos vištos prilestų lig soties. Šeimininkė pila tuos žrns tan kubilų lankan, kad nei vienas žirnis nebūt už kubilų lanko - kad vištos visos dėtų kšns vienan daiktan, nemėtytų kiaušinių. Berdama žirnius tan lankan, šmnk garsiai sako vištom:

- Žiūrėkite, kad nei vieno kiaušinio, nei vieno niekur nrmstt, visus vnn vieton dėkite!

Na ir visos vištos šmnks įsakymą vykdo.

Programos kūrimo eiga.

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

➊ 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;
}
```

➋ Antras žingsnis.

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

```
//-----
/** Iš žodžio pašalina bales ir grąžina rezultatą per vardą.
    @param eilute - žodis su balsėmis
    @param balsės - abėcėlės balsės */
static StringBuilder BeBalsiu(string eilute, string balsės)
{
    StringBuilder nauja = new StringBuilder();
    for (int i = 0; i < eilute.Length; i++)
        if (balsės.IndexOf(eilute[i]) == -1)
            nauja.Append(eilute[i]);
    return nauja;
}
```

➌ Trečias žingsnis.

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

```
//-----
/** Skaito failą, analizuoja eilutes, kuria rezultatų ir analizės
failus.
    @param fd - duomenų failo vardas
    @param fr - rezultatų failo vardas
    @param fa - analizės failo vardas
    @param skyrikliai - žodžių skyrikliai
    @param balsės - abėcėlės balsės */
```

```

static void Apdoroti(string fd, string fr, string fa, char[] skyrikliai,
string bales)
{
    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, bales).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);
                }
            }
        }
    }
}
//-----

```

- Analizės faile rezultatus kaupiame lentelėje. Lentelės linijų brėžimui skelbiame string kintamąjį su konstruktoriumi:

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

- Toks paskelbimas užpildo eilutės kintamąjį 38 brūkšneliais. Privalumai yra tokie:
 - neriekia brėžti ilgos brūkšninės linijos;
 - kintamąjį paskelbėme vieną kartą, o galėsime naudoti tiek kartų, kiek reikia.

🔗 Ketvirtas žingsnis.

- Parašykite pagrindinį metodą:

```

//-----
static void Main(string[] args)
{
    const string CFd = "..\\..\\Duomenys.txt";
    const string CFr = "..\\..\\Rezultatai.txt";
    const string CFa = "..\\..\\Analyze.txt";
    const string bales = "AEIYOUaeiouAaEeĖėİiŲųŪū";
    char[] skyrikliai = {' ', '.', ',', '!', '?', ':', ';', '(', ')', '\t'};
    Apdoroti(CFd, CFr, CFa, skyrikliai, bales);
}
//-----

```

5.7. HTML teksto analizė ir redagavimas

Visi programiniai kodai, kurie pateikiami šiame poskyryje, yra šioje saugykloje:

<https://bitbucket.org/blaztoma/lab4/downloads>

Naudokite juos, jei nesiseka savarankiškai išspręsti uždavinių.

Užduotis.

Bankrutavus konkurentų įmonei Klaipėdoje, buvo nuspręsta perimti jų veiklą. Deja, buvo įmanoma prieiti tik prie bankrutavusios įmonės tinklapyje skelbiamų duomenų, todėl prireikė HTML formatu saugomus duomenis konvertuoti į CSV formatą. Taip pat pateiktuose duomenyse pastebėta klaidų.

Išnagrinėjus duomenis pateiktus tinklalapyje paaiškėjo, kad:

- kai kurie vardai (tiek gyvūnų, tiek jų savininkų) prasideda mažąja raide.
- vieni telefonų numeriai prasideda skaičiais 8, kiti: +370.
- kai kurie telefonų numeriai buvo įvesti neteisingai (yra per daug arba per mažai skaičių, kai kuriuose numeriuose yra raidžių).
- Duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas bold, italic ar underline formatavimas).

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV kartu ištaisydamas aptinkamas klaidas.

Parašykite tokią programą, kuri atliktų tokius veiksmus:

- nuskaitytų tinklalapio failą Klaipeda.html ir jame esančius duomenis konvertuotų į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
- suvienodintų telefonų numerių pateikimo būdą (pertvarkyti eilutes taip, kad visų telefonų numeriai prasidėtų +370, tai yra, jei numeris prasideda 8 – pakeisti į +370);
- surastų duomenų eilutes, kuriose neteisingi telefonų numeriai (telefonų numeriai turi prasidėti +, susidėti iš 11 simbolių tarp kurių neturi būti raidžių) ir jas išvesti į failą KlaidingiDuomenys.csv;
- Pašalintų iš duomenų eilučių tas, kuriose yra neteisingi telefonų numeriai ir išvesti teisingus duomenis į failą Klaipeda.csv.
- Pavaizduotų atrinktų teisingų duomenų sąrašą html formatu.

Pradiniai duomenys ir rezultatai.

Html formato faile pateikiamas filialo pavadinimas ir šie laukai: gyvūno tipas, vardas, mikroschemos identifikatorius, veislė, savininko vardas, savininko telefonas, paskutinio vakcinavimo data, agresyvumo požymis (jei gyvūnas yra šuo). Visi šie duomenys gaubiami html formato žymėmis.

| Pradiniai duomenys |
|--|
| <p>Failas Klaipeda.html:</p> <pre> <!DOCTYPE html> <html> <head> <title>Gyvūnų registras</title> <meta charset="UTF-8"> <style type="text/css"> .tg {border-collapse:collapse;border-spacing:0;} .tg td{font-family:Arial, sans-serif;font-size:14px;padding:10px 5px;border- style:solid;border-width:1px;overflow:hidden;word-break:normal;} .tg th{font-family:Arial, sans-serif;font-size:14px;font- weight:normal;padding:10px 5px;border-style:solid;border- width:1px;overflow:hidden;word-break:normal;} .tg .tg-amwm{font-weight:bold;text-align:center;vertical-align:top} .tg .tg-yw4l{vertical-align:top} </style> </head> <body> <table class="tg"> <tr> <th class="tg-amwm" colspan="8">Klaipėda</th> </tr> <tr> <td class="tg-yw4l">D</td> <td class="tg-yw4l">Haris</td> <td class="tg-yw4l">351</td> <td class="tg-yw4l">Buldogas</td> </pre> |

```

        <td class="tg-yw4l">mikas</td>
        <td class="tg-yw4l">867424992</td>
        <td class="tg-yw4l">2015-07-24</td>
        <td class="tg-yw4l">TRUE</td>
    </tr>
    <tr>
        <td class="tg-yw4l">C</td>
        <td class="tg-yw4l">Micius</td>
        <td class="tg-yw4l">972</td>
        <td class="tg-yw4l">Siamo</td>
        <td class="tg-yw4l">Eduardas</td>
        <td class="tg-yw4l">865412654</td>
        <td class="tg-yw4l">2014-11-24</td>
        <td class="tg-yw4l"></td>
    </tr>
    <tr>
        <td class="tg-yw4l">D</td>
        <td class="tg-yw4l">maksas</td>
        <td class="tg-yw4l">359</td>
        <td class="tg-yw4l">Taksas</td>
        <td class="tg-yw4l">kęstas</td>
        <td class="tg-yw4l">86661a345</td>
        <td class="tg-yw4l">2014-12-01</td>
        <td class="tg-yw4l">FALSE</td>
    </tr>
    <tr>
        <td class="tg-yw4l">C</td>
        <td class="tg-yw4l">Rikę</td>
        <td class="tg-yw4l">558</td>
        <td class="tg-yw4l">Siamo</td>
        <td class="tg-yw4l">jurga</td>
        <td class="tg-yw4l">+37066612345</td>
        <td class="tg-yw4l">2014-12-05</td>
        <td class="tg-yw4l"></td>
    </tr>
    <tr>
        <td class="tg-yw4l">C</td>
        <td class="tg-yw4l">Meilutę</td>
        <td class="tg-yw4l">778</td>
        <td class="tg-yw4l">Rusų męlynoji</td>
        <td class="tg-yw4l">Kristina</td>
        <td class="tg-yw4l">861253226</td>
        <td class="tg-yw4l">2012-05-11</td>
        <td class="tg-yw4l"></td>
    </tr>
    <tr>
        <td class="tg-yw4l">C</td>
        <td class="tg-yw4l">Šuva</td>
        <td class="tg-yw4l">889</td>
        <td class="tg-yw4l">Bengalijos</td>
        <td class="tg-yw4l">Sigitas</td>
        <td class="tg-yw4l">867892544</td>
        <td class="tg-yw4l">2013-07-21</td>
        <td class="tg-yw4l"></td>
    </tr>

```

```

<tr>
  <td class="tg-yw4l">C</td>
  <td class="tg-yw4l">Tigras</td>
  <td class="tg-yw4l">665</td>
  <td class="tg-yw4l">Britų trumpapalukis</td>
  <td class="tg-yw4l">Rasa</td>
  <td class="tg-yw4l">+37061588345</td>
  <td class="tg-yw4l">2010-11-04</td>
  <td class="tg-yw4l"></td>
</tr>
<tr>
  <td class="tg-yw4l">D</td>
  <td class="tg-yw4l">Miledi</td>
  <td class="tg-yw4l">779</td>
  <td class="tg-yw4l">Jorkšyro terjeras</td>
  <td class="tg-yw4l">simas</td>
  <td class="tg-yw4l">867897315</td>
  <td class="tg-yw4l">2010-11-14</td>
  <td class="tg-yw4l">FALSE</td>
</tr>
<tr>
  <td class="tg-yw4l">D</td>
  <td class="tg-yw4l">Ferdinandas</td>
  <td class="tg-yw4l">797</td>
  <td class="tg-yw4l">Sibiro haskis</td>
  <td class="tg-yw4l">arūnas</td>
  <td class="tg-yw4l">8687990321</td>
  <td class="tg-yw4l">2015-01-02</td>
  <td class="tg-yw4l">FALSE</td>
</tr>
<tr>
  <td class="tg-yw4l">D</td>
  <td class="tg-yw4l">Rubinis</td>
  <td class="tg-yw4l">898</td>
  <td class="tg-yw4l">Sibiro haskis</td>
  <td class="tg-yw4l">Angela</td>
  <td class="tg-yw4l">+370687990321</td>
  <td class="tg-yw4l">2015-01-02</td>
  <td class="tg-yw4l">FALSE</td>
</tr>
<tr>
  <td class="tg-yw4l">D</td>
  <td class="tg-yw4l">Meškis</td>
  <td class="tg-yw4l">223</td>
  <td class="tg-yw4l">Laika</td>
  <td class="tg-yw4l">Miglė</td>
  <td class="tg-yw4l">869811065</td>
  <td class="tg-yw4l">2012-04-12</td>
  <td class="tg-yw4l">TRUE</td>
</tr>
<tr>
  <td class="tg-yw4l">D</td>
  <td class="tg-yw4l">Bumsis</td>
  <td class="tg-yw4l">241</td>
  <td class="tg-yw4l">Biglis</td>

```



```
<td class="tg-yw4l">saulius</td>
<td class="tg-yw4l">861489752</td>
<td class="tg-yw4l">2011-03-25</td>
<td class="tg-yw4l">FALSE</td>
</tr>
</table>
</body>
</html>
```

Rezultatai

Failas **Klaipeda.csv**:

D;Harris;351;Buldogas;Mikas;+37067424992;2015-07-24;TRUE;
C;Miccius;972;Siamo;Eduardas;+37065412654;2014-11-24;;
C;Rikė;558;Siamo;Jurga;+37066612345;2014-12-05;;
C;Meilutė;778;Rusų mėlynoji;Kristina;+37061253226;2012-05-11;;
C;Šuva;889;Bengalijos;Sigitas;+37067892544;2013-07-21;;
C;Tigras;665;Britų trumpapalukis;Rasa;+37061588345;2010-11-04;;
D;Miledi;779;Jorkšyro terjeras;Simas;+37067897315;2010-11-14;FALSE;
D;Meškis;223;Laika;Miglė;+37069811065;2012-04-12;TRUE;
D;Bumsis;241;Biglis;Saulius;+37061489752;2011-03-25;FALSE;

Failas **KlaidingiDuomenys.csv**:

D;Maksas;359;Taksas;Kęstas;+3706661a345;2014-12-01;FALSE;
D;Ferdinandas;797;Sibiro haskis;Arūnas;+370687990321;2015-01-02;FALSE;
D;Rubinis;898;Sibiro haskis;Angela;+370687990321;2015-01-02;FALSE;

Programas kūrimo eiga.

- Sukuriamas naujas sprendimas **Lab4**, o jame naujas projektas **Lab4.Converter**. Jei norite – galite konverterio projektą prijungti prie **Lab3** sprendimo projektų.
- Prie projekto pridedamas failas **Klaipeda.html**, kuris pateiktas aukščiau esančioje lentelėje.
- Parašomas metodas duomenų nuskaitymui ir jų pavertimui į CSV formato eilutes.
- Parašomas metodas perteklinių html žymių pašalinimui.
- Parašomas metodas vardų tikrinimui ir koregavimui.
- Parašomas metodas telefonų numerių koregavimui.
- Parašomas metodas neteisingų duomenų aptikimui ir išvedimui į failą **KlaidingiDuomenys.csv**.
- Parašomas metodas eilučių su neteisingais duomenimis šalinimui ir išvedimui į failą **Klaipeda.csv**.
- Parašomas metodas **csv** formato eilučių konvertavimui į **html** formatą. Rezultatai išvedami į **Klaipeda2.html**.

↓ Pirmas žingsnis.

- Sukurkite naują sprendimą **Lab4**, o jame naują projektą **Lab4.Converter**.
- Projekto (**Lab4.Converter**) kataloge sukurkite failą **Klaipeda.html** ir užpildykite duotu kodu. Tai padarę pamėginkite atidaryti šį failą naršykle ir įsitikinti, jog jį rodo teisingai.
- Naujai sukurto projekto pagrindinėje programoje (**program.cs**) sukurkime metodą duomenų nuskaitymui ir jų pavertimui į CSV formato eilutes:

[illegible]

```

string[] csvLines = new string[tr_matches.Count];
//pirmoje eilutėje ieškome antraščių
MatchCollection th_matches = Regex.Matches(tr_matches[0].Value, thPattern,
                                         RegexOptions.Singleline);

//iname pirmąją antraštę
string title = th_matches[0].Groups[1].Value;
csvLines[0] = title;

//iname per likusias eilutes
for (int i = 1; i < tr_matches.Count; i++)
{
    StringBuilder line = new StringBuilder();
    MatchCollection td_matches = Regex.Matches(tr_matches[i].Value, tdPattern,
                                                RegexOptions.Singleline);

    for (int j = 0; j < td_matches.Count; j++)//iname per stulpelius
    {
        //nuskaitom langelio reikšmę ir dedam į eilutę, po jos dedam skyriklį
        line.Append(td_matches[j].Groups[1].Value).Append(csvSeparator);
    }

    csvLines[i] = line.ToString();
}

return csvLines;
}

```

- Šis metodas naudoja reguliariąsias išraiškas. Šių išraiškų šablonai yra nusakyti konstantose **tablePattern**, **trPattern**, **thPattern**, **tdPattern**. Šias konstantas ir kitus reikalingus kintamuosius aprašome programos klasės atributų skiltyje (kodas pateiktas toliau).
- Main() metode iškviečiame ConvertToCsv metodą ir konsolėje pateikiame konvertuotų duomenų CSV eilutes:

```

using System;
using System.IO;
using System.Text;
using System.Text.RegularExpressions;

namespace Lab4.Converter
{
    class Program
    {
        const string htmlDataFile = @"Klaipeda.html";
        const string csvDataFile = @"Klaipeda.csv";
        const string wrongDataFile = @"KlaidingiDuomenys.csv";
        const string htmlOutputFile = @"Klaipeda2.html";

        const string tablePattern = "<table.*?>(.*?)</table>";
        const string thPattern = "<th.*?>(.*?)</th>";
        const string trPattern = "<tr>(.*?)</tr>";
        const string tdPattern = "<td.*?>(.*?)</td>";

        const string csvSeparator = ";";

        static void Main(string[] args)
        {
            string[] csvLines = ConvertToCsv(htmlDataFile);

            foreach (string line in csvLines)
            {
                Console.WriteLine(line);
            }

            Console.Read();
        }
    }
}

```

```
}
```

```
....
```

- Atkreipkite dėmesį, kad darbu su failais ir tekstais reikės šių bibliotekų:


```
using System.IO;
using System.Text;
using System.Text.RegularExpressions;
```
- Išmėginkite programą. Kartu su dėstytojų ir kolegų pagalba išsiaiškinkite čia naudotų reguliariųjų išraiškų veikimą.

🕒 Antras žingsnis.

- Išmėginę programą pastebėsite, jog kai kur liko **html** žymių, tokių kaip ****, **<i>** ir panašių. Jas pašalinsime iš eilučių tam, kad liktų tik gryni duomenys. Tuo tikslu sukurkime metodą **RemoveTags**:

```
private static string[] RemoveTags(string[] csvLines)
{
    for(int i = 0; i<csvLines.Length;i++)
    {
        csvLines[i] = Regex.Replace(csvLines[i], "<.*?>", string.Empty);
    }
    return csvLines;
}
```

- Reguliarioji išraiška "**<.*?>**" pašalina viską kas yra tarp atidarančių ir uždarančių html simbolių (**<** ir **>**).
- Patikrinkite kaip veikia programa.

🕒 Trečias žingsnis.

- Sukursime metodą **FixNames**, kuris pirmą vardo raidę pavers didžiąja, jei vardas prasideda mažąja.
- Lig šiol dirbome su eilutėmis, o dabar mums iš visos CSV formato eilutė reikės išgauti gyvūno ir savininko vardų laukų reikšmes. Tai padarysime eilutės metodo **Split** pagalba išskaidydami eilutę į masyvą ir nurodysime nagrinėjamo lauko indeksą (numerį).
- Kad būtų paprasčiau atrinkti laukus – panaudosime **enum** tipą. Jis leidžia skaičius aprašyti vardinėmis reikšmėmis. Išnagrinėkite žemiau esantį kodą ir išsiaiškinkite kaip naudojamas šis tipas – tai jums pravers ateityje.

```
private static string[] FixNames(string[] csvLines, int fieldToCorrect)
{
    for (int i = 1; i < csvLines.Length; i++)
    {
        string[] stringFields = csvLines[i].Split(csvSeparator[0]);
        if(Char.IsLower(stringFields[fieldToCorrect][0]))
        {
            stringFields[fieldToCorrect] = stringFields[fieldToCorrect].Substring(0,
            1).ToUpper() + stringFields[fieldToCorrect].Substring(1).ToLower();
            csvLines[i] = string.Join(csvSeparator, stringFields);
        }
    }
    return csvLines;
}
```

- Papildykite **Main()** metodą kreipiniu į parašytą metodą ir patikrinkite programos veikimą išvesdami rezultatus į konsolę.
- Žemiau pateikiamos kodo iškarpas, parodančios kaip panaudojamas **enum** tipas ir iškviečiamas metodas gyvūnų ir žmonių vardų koregavimui:

```

{
    class Program
    {
        const string htmlDataFile = @"Klaipeda.html";
        const string csvDataFile = @"Klaipeda.csv";
        const string wrongDataFile = @"KlaidingiDuomenys.csv";

        const string tablePattern = "<table.*?>(.*?)</table>";
        const string thPattern = "<th.*?>(.*?)</th>";
        const string trPattern = "<tr>(.*?)</tr>";
        const string tdPattern = "<td.*?>(.*?)</td>";

        const string csvSeparator = ";";

        enum Fields { AnimalType, AnimalName, AnimalChip, AnimalBreed, OwnerName,
                     OwnerPhone, VaccinationDate, AgressiveState };

        static void Main(string[] args)
        {
            string[] csvLines = ConvertToCsv(htmlDataFile);

            csvLines = RemoveTags(csvLines);

            csvLines = FixNames(csvLines, (int)Fields.AnimalName);
            csvLines = FixNames(csvLines, (int)Fields.OwnerName);
        }
    }
}

```

...

U Ketvirtas žingsnis.

- Sukursime metodą **FixPhoneNumbers** klaidingų telefonų numerių aptikimui ir koregavimui. Šis metodas galės pakoreguoti tik tuos numerius, kurie prasideda skaičiumi 8 (paversdamas į +370). Jis naudos atskirą metodą numerio patikrinimui.
- Telefono laukui nurodyti panaudosime jau aptartą **enum** tipą.
- Telefono numerio patikrinimui sukursime atskirą metodą **isPhoneNumber**, kuriame surašysime užduotyje pateiktas taisykles:

```

public static bool IsPhoneNumber(string number)
{
    if(number[0] != '+') return false;
    if (number.Length != 12) return false;

    for(int i = 1; i < number.Length; i++)
    {
        if((number[i] < '0') || (number[i] > '9')) return false;
    }
    return true;
}

```

- Sukurdami tokį metodą mes pasitreniravome analizuoti eilutę po vieną simbolį. Kita vertus, C# ir daugelis kitų klaidų teikia reguliariųjų išraiškų mechanizmus, kurie supaprastina darbą. Jei išmokote analizuoti eilutę paraidžiui, galite šį metodą perrašyti tokiu būdu:

```

public static bool IsPhoneNumber(string number)
{
    return Regex.Match(number, @"^(?=[0-9]{11})$").Success;
}

```

- Dabar parašysime metodą, kuris naudodamas aprašytą metodą patikrins ar telefono numeris teisingas ir mėgins jį pakoreguoti jei ne:

```

private static string[] FixPhoneNumbers(string[] csvLines, int fieldToCorrect)
{
    for (int i = 1; i < csvLines.Length; i++)

```

```

    {
        string[] stringFields = csvLines[i].Split(csvSeparator[0]);
        if (!IsPhoneNumber(stringFields[fieldToCorrect]))
        {
            if (stringFields[fieldToCorrect][0] == '8')
            {
                stringFields[fieldToCorrect] = "+370" +
                    stringFields[fieldToCorrect].Substring(1);
                csvLines[i] = string.Join(csvSeparator, stringFields);
            }
        }
    }
    return csvLines;
}

```

- Papildykite `Main()` metodą kreipiniu į parašytą metodą ir patikrinkite programos veikimą išvesdami rezultatus į konsolę:

```
csvLines = FixPhoneNumbers(csvLines, (int)Fields.OwnerPhone);
```

🔗 Penktas žingsnis.

- Išrašysime visas eilutes su klaidingais duomenimis į failą `KlaidingiDuomenys.csv` tam, kad vėliau juos būtų galima pakoreguoti.
- Tuo tikslu sukursime du metodus: eilučių su klaidingais duomenimis aptikimo metodą (`GetLinesWithWrongNumbers`) ir duomenų išrašymo į failą metodą (`WriteLinesToFile`).
- `GetLinesWithWrongNumbers` metodas naudoja jau aptartą **IsPhoneNumber** metodą klaidingų numerių aptikimui:

```

private static string[] GetLinesWithWrongNumbers(string[] csvLines, int
                                                fieldToCheck, out int wrongCount)
{
    string[] wrongDataLines = new string[csvLines.Length];

    wrongCount = 0;
    for (int i = 1; i < csvLines.Length; i++)
    {
        string[] stringFields = csvLines[i].Split(csvSeparator[0]);
        if (!IsPhoneNumber(stringFields[fieldToCheck]))
        {
            wrongDataLines[wrongCount++] = csvLines[i];
        }
    }
    return wrongDataLines;
}

```

- Išrašymo į failą metodą sukursime tokį, kad jis galėtų išvesti bet kokius eilučių duomenis:

```

private static void WriteLinesToFile(string dataFile, string[] data, int dataCount)
{
    using (var fileHandle = File.CreateText(dataFile))
    {
        for (int i = 0; i < dataCount; i++)
        {
            fileHandle.WriteLine(data[i]);
        }
    }
}

```

- Telieta iškviešti šiuos metodus pagrindiniame programos metode.:

```
int wrongCount;
```

```
string[] wrongDataLines = GetLinesWithWrongNumbers(csvLines,
                                                    (int)Fields.OwnerPhone, out wrongCount);
WriteLinesToFile(wrongDataFile, wrongDataLines, wrongCount);
```

- Įvykdysite programą ir patikrinkite kokius rezultatus gaunate faile **KlaidingiDuomenys.csv**. Šis failas bus jūsų **bin/Debug** ar **bin/Release** kataloge.

🔗 Šeštasis žingsnis.

- Pašalinsime visas eilutes, kurios turi neteisingus telefonų numerius. Tuo tikslu sukursime metodą **RemoveWrongDataLines**, kuris naudos jau aptiktas klaidingas eilutes ir suformuos naują eilučių masę, neturinčią klaidingųjų:

```
private static string[] RemoveWrongDataLines(string[] csvLines, string[]
                                              wrongDataLines, int wrongCount, out int correctCount)
{
    string[] correctDataLines = new string[csvLines.Length];

    correctCount = 0;
    for (int i = 1; i < csvLines.Length; i++)
    {
        bool foundEqual = false;
        for (int j = 0; j < wrongCount; j++)
        {
            if(csvLines[i].Equals(wrongDataLines[j]))
            {
                foundEqual = true;
            }
        }
        if(!foundEqual) correctDataLines[correctCount++] = csvLines[i];
    }
    return correctDataLines;
}
```

- Išrašymo į failą metodas jau sukurtas, todėl pakanka iškviešti šiuos metodus pagrindiniame programos metode:

```
int correctCount;
string[] correctDataLines = RemoveWrongDataLines(csvLines, wrongDataLines,
                                                  wrongCount, out correctCount);
WriteLinesToFile(csvDataFile, correctDataLines, correctCount);
```

- Įvykdysite programą ir patikrinkite kokius rezultatus gaunate faile **Klaipeda.csv**. Šis failas bus jūsų **bin/Debug** ar **bin/Release** kataloge.

🔗 Septintasis žingsnis.

- Duomenų pateikimui html formatu naudosime šablonus. Tai toks turinio generavimo metodas, kai turime etaloninę html dokumento struktūrą, kurioje sužymėta – kur ir kurių kintamųjų reikšmėmis turi būti pakeistos tam tikros dokumento vietos.
- Panagrinėkite Klaipeda.html failo struktūrą ir pastebėkite, kad jis susideda iš 3 dalių:
 - Dokumento ir lentelės antraštinės dalies;
 - Lentelės eilučių;
 - Lentelės ir dokumento pabaigos.
- Kinta tik lentelės eilutės, kurioms mums ir reiks priskirti suformuotus duomenis.
- Tai įvertinę sukuriame 3 šablonų failus:

| |
|-------------------------------------|
| Failas HeaderTemplate.html : |
|-------------------------------------|

```

<!DOCTYPE html>
<html>
<head>
  <title>Gyvūnų registras</title>
  <meta charset="UTF-8">
<style type="text/css">
.tg {border-collapse:collapse;border-spacing:0;}
.tg td{font-family:Arial, sans-serif;font-size:14px;padding:10px 5px;border-
style:solid;border-width:1px;overflow:hidden;word-break:normal;}
.tg th{font-family:Arial, sans-serif;font-size:14px;font-
weight:normal;padding:10px 5px;border-style:solid;border-
width:1px;overflow:hidden;word-break:normal;}
.tg .tg-amwm{font-weight:bold;text-align:center;vertical-align:top}
.tg .tg-yw4l{vertical-align:top}
</style>
</head>
<body>
<table class="tg">
  <tr>
    <th class="tg-amwm" colspan="8">Klaipėda</th>
  </tr>
  <tr>
    <td class="tg-yw4l">{AnimalType}</td>
    <td class="tg-yw4l">{AnimalName}</td>
    <td class="tg-yw4l">{AnimalChip}</td>
    <td class="tg-yw4l">{Breed}</td>
    <td class="tg-yw4l">{OwnerName}</td>
    <td class="tg-yw4l">{OwnerPhone}</td>
    <td class="tg-yw4l">{VaccinationDate}</td>
    <td class="tg-yw4l">{AggressiveState}</td>
  </tr>
</table>
</body>
</html>

```

Failas **DataLineTemplate.html**:

```

<tr>
  <td class="tg-yw4l">{AnimalType}</td>
  <td class="tg-yw4l">{AnimalName}</td>
  <td class="tg-yw4l">{AnimalChip}</td>
  <td class="tg-yw4l">{Breed}</td>
  <td class="tg-yw4l">{OwnerName}</td>
  <td class="tg-yw4l">{OwnerPhone}</td>
  <td class="tg-yw4l">{VaccinationDate}</td>
  <td class="tg-yw4l">{AggressiveState}</td>
</tr>

```

Failas **FooterTemplate.html**:

```

</table>
</body>
</html>

```

- Sukurkite šiuos tris failus projekto kataloge ir pažymėkite, kad programavimo aplinka juos nukopijuotų į vykdymo katalogą (pažymėkite failą, savybių lange suraskite „Copy to Output Directory“ ir parinkite „Copy allways“).
- Atkreipkite dėmesį kaip sužymėjome **DataLineTemplate.html** šabloną: tarp {} skliaustelių nurodėme kokį kintamąjį reikia įrašyti toje vietoje.
- Esame pasiruošę parašyti html eilutės suformavimo metodą, kuris veiks tokiu būdu:
 - Nuskaitys HeaderTemplate.html failą ir prijungs prie formuojamos html eilutės;
 - Nuskaitys DataLineTemplate.html failą į atskirą kintamąjį ir naudos suformuojant kiekvieną duomenų eilutę;
 - Nuskaitys FooterTemplate.html failą ir prijungs prie formuojamos html eilutės.
- Metodas **ConvertToHtml**:

```

private static string ConvertToHtml(string[] data, int dataCount)
{
  string headerContent = File.ReadAllText(@"HeaderTemplate.html");
  string footerContent = File.ReadAllText(@"FooterTemplate.html");
  string dataLineContent = File.ReadAllText(@"DataLineTemplate.html");

```

```

        StringBuilder htmlContent = new StringBuilder();

        htmlContent.Append(headerContent);

        for (int i = 0; i < dataCount; i++)
        {
            string[] lineFields = data[i].Split(csvSeparator[0]);
            string processedLine = dataLineContent;
            processedLine = processedLine.Replace("{AnimalType}",
                lineFields[(int)Fields.AnimalType]);
            processedLine = processedLine.Replace("{AnimalName}",
                lineFields[(int)Fields.AnimalName]);
            processedLine = processedLine.Replace("{AnimalChip}",
                lineFields[(int)Fields.AnimalChip]);
            processedLine = processedLine.Replace("{Breed}",
                lineFields[(int)Fields.AnimalBreed]);
            processedLine = processedLine.Replace("{OwnerName}",
                lineFields[(int)Fields.OwnerName]);
            processedLine = processedLine.Replace("{OwnerPhone}",
                lineFields[(int)Fields.OwnerPhone]);
            processedLine = processedLine.Replace("{VaccinationDate}",
                lineFields[(int)Fields.VaccinationDate]);
            processedLine = processedLine.Replace("{AggressiveState}",
                lineFields[(int)Fields.AgressiveState]);

            htmlContent.Append(processedLine);
        }

        htmlContent.Append(footerContent);

        return htmlContent.ToString();
    }

```

- Dabar iškviečiame šį metodą pagrindiniame metode pateikdami eilučių su teisingais duomenimis masyvą:

```

string htmlContent = ConvertToHtml(correctDataLines, correctCount);
using (var fileHandle = File.CreateText(htmlOutputFile))
{
    fileHandle.WriteLine(htmlContent);
}

```

- Įvykdysite programą ir patikrinkite kokius rezultatus gaunate faile **Klaipeda2.html**. Šis failas bus jūsų **bin/Debug** ar **bin/Release** kataloge. Atidarykite jį naršyklėje.

5.8. Kontroliniai klausimai

1. Ką vadiname žodžiu?
2. Kokiam tikslui naudojamas metodas `IndexOf`?
3. Kaip reikia kreiptis į eilutę, norint išrinkti vieną eilutės simbolį?
4. Kaip užrašomas `string` konstruktorius, kuris leidžia užpildyti eilutę nurodytu simboliu?
5. Kokiam tikslui naudojamas `string` eilutės `Split` metodas?
6. Kokiam tikslui `string` eilutės `Split` metode naudojamas parametras `StringSplitOptions.RemoveEmptyEntries`?
7. Koks turi būti skyriklių tipas `string` klasės `Split` metodui?
8. Koks turi būti skyriklių tipas `Regex` klasės `Split` metodui?
9. Ar gali metodai `Remove`, `Insert`, `Replace` pakeistą eilutę priskirti buvusiai eilutei? Atsakymą pakomentuokite.
10. Kada reikia naudoti `StringBuilder` klasę? O kodėl tuo tikslu negalima panaudoti `string` klasės?

5.9. Užduotys

U5-1. Unikalus žodžiai

Dviejuose tekstiniuose failuose `Knyga1.txt` ir `Knyga2.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma ir abejuose failuose yra ta pati. Analizuojant tekstus, didžiosios ir mažosios raidės nesvarbios.

Raskite, spausdinkite faile `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- unikalių žodžių, kurie yra tik faile `Knyga1.txt`, bet nėra faile `Knyga2.txt`, skaičių ir tokių žodžių sąrašą (ne daugiau nei 10 žodžių), surikiuotą pagal pasikartojimo skaičių;
- unikalių žodžių, kurie yra abejuose failuose, skaičių ir tokių žodžių sąrašą (ne daugiau nei 10 žodžių), surikiuotą pagal pasikartojimo skaičių;

Spausdinkite faile `ManoKnyga.txt` apjungtą tekstą, sudarytą pagal tokias taisykles:

- kopijuojamas pirmojo failo tekstas tol, kol sutinkamas pirmasis antrojo failo žodis arba pasiekama failo pabaiga;
- kopijuojamas antrojo failo tekstas tol, kol sutinkamas pirmasis nenukopijuotas pirmojo failo žodis arba pasiekama failo pabaiga;
- kartojama tol, kol pasiekama abiejų failų pabaiga.

U5-2. Lygiavimas

Tekstiniame faile `Knyga.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma. Raskite, spausdinkite faile `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- unikalių žodžių skaičių ir tokių žodžių sąrašą (ne daugiau nei 10 žodžių), surikiuotą pagal pasikartojimo skaičių;
- ilgiausių žodžių sąrašą (ne daugiau nei 10 žodžių) ir jų pasikartojimo skaičių;

Reikia teksto žodžius sulygiuoti, kad kiekvienos eilutės kiekvienas žodis prasidėtų fiksuotoje toje pačioje pozicijoje. Galima įterpti tik minimalų būtiną tarpų skaičių. Galima šalinti kelis iš eilės einančius vienodus skyriklius, paliekant tik vieną jų atstovą. Įterpimo ir šalinimo taisyklės taikome, siekdami gauti lygiuotą minimalų tekstą. Šalinimo taisyklės netaikome, jei nėra poreikio. Pradinio teksto eilutės ilgis neviršija 80 simbolių.

Spausdinkite faile `ManoKnyga.txt` pertvarkytą tekstą pagal tokias taisykles:

- kiekvienos eilutės pirmasis žodis turi prasidėti pozicijoje $p_1=1$.
- antrasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_2 , tokioje, kad kiekvienos eilutės pirmasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_2-2 arba p_2-2 .
- trečiasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_3 , tokioje, kad kiekvienos eilutės antrasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_3-2 arba p_3-2 .
- ir t.t.

U5-3. Ilgiausi žodžiai

Dviejuose tekstiniuose failuose `Knyga1.txt` ir `Knyga2.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma ir abejuose failuose yra ta pati. Analizuojant tekstus, didžiosios ir mažosios raidės nesvarbios.

Raskite, spausdinkite faile `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- ilgiausių žodžių, kuris yra abejuose failuose, sąrašą (ne daugiau nei 10 žodžių) ir jų pasikartojimo skaičių kiekviename iš failų;
- ilgiausių žodžių, kurie yra tik faile `Knyga1.txt`, bet nėra faile `Knyga2.txt`, sąrašą (ne daugiau nei 10 žodžių) ir jų pasikartojimo skaičių;

Spausdinkite faile `ManoKnyga.txt` apjungtą tekstą, sudarytą pagal tokias taisykles:

- kopijuojamas pirmojo failo tekstas tol, kol sutinkamas pirmasis antrojo failo žodis arba pasiekama failo pabaiga;
 - kopijuojamas antrojo failo tekstas tol, kol sutinkamas pirmasis nenukopijuotas pirmojo failo žodis arba pasiekama failo pabaiga;
 - kartojama tol, kol pasiekama abiejų failų pabaiga.
-

U5-4. Teksto fragmentai

Tekstiniame faile `Knyga.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma. Raskite, spausdinkite faile `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- unikalių žodžių, prasidedančių nurodyta raide, sąrašą (ne daugiau nei 10 žodžių) ir jų pasikartojimo skaičių;
- ilgiausią teksto fragmentą, sudarytą iš žodžių, kur žodžio paskutinė raidė sutampa su kito žodžio pirmąja raide (tarp didžiųjų ir mažųjų raidžių skirtumo nedaryti) ir juos skiriančių skyriklių, bei jo pradžios ir pabaigos eilutės numerius;

Reikia teksto žodžius sulygiuoti, kad kiekvienos eilutės kiekvienas žodis prasidėtų fiksuotoje toje pačioje pozicijoje. Galima įterpti tik minimalų būtiną tarpų skaičių. Galima šalinti kelis iš eilės einančius vienodus skyriklius, paliekant tik vieną jų atstovą. Įterpimo ir šalinimo taisyklės taikome, siekdami gauti lygiuotą minimalų tekstą. Šalinimo taisyklės netaikome, jei nėra poreikio. Pradinio teksto eilutės ilgis neviršija 80 simbolių.

Spausdinkite faile `ManoKnyga.txt` pertvarkytą tekstą pagal tokias taisykles:

- kiekvienos eilutės pirmasis žodis turi prasidėti pozicijoje $p_1=1$.
- antrasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_2 , tokioje, kad kiekvienos eilutės pirmasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_2-2 arba p_2-2 .
- trečiasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_3 , tokioje, kad kiekvienos eilutės antrasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_3-2 arba p_3-2 .
- ir t.t.

U5-5. Ilgiausias sakiny

Dviejuose tekstiniuose failuose `Knyga1.txt` ir `Knyga2.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma ir abejuose failuose yra ta pati. Analizuojant tekstus, didžiosios ir mažosios raidės nesvarbios.

Raskite, spausdinkite faile `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- unikalių žodžių, kurie yra tik faile `Knyga1.txt`, bet nėra faile `Knyga2.txt`, skaičių ir tokių žodžių sąrašą (ne daugiau nei 10 žodžių), surikiuotą pagal pasikartojimo skaičių;
- ilgiausią sakinį, jo ilgį (simboliais ir žodžiais) ir vietą (sakinio pradžios eilutės numerį) pirmame ir antrame faile.

Spausdinkite faile `ManoKnyga.txt` apjungtą tekstą, sudarytą pagal tokias taisykles:

- kopijuojamas pirmojo failo tekstas tol, kol sutinkamas pirmasis antrojo failo žodis arba pasiekama failo pabaiga;
- kopijuojamas antrojo failo tekstas tol, kol sutinkamas pirmasis nenukopijuotas pirmojo failo žodis arba pasiekama failo pabaiga;
- kartojama tol, kol pasiekama abiejų failų pabaiga.

U5-6. Skaitmenys

Tekstiniame faile `Knyga.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma. Raskite, spausdinkite faile `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- ilgiausią sakinį, jo ilgį (simboliais ir žodžiais) ir vietą (sakinio pradžios eilutės numerį).
- Žodžių, kuriuos sudaro tik skaitmenys, kiekį. Suskaičiuokite tokių skaičių bendrą sumą.

Reikia teksto žodžius sulygiuoti, kad kiekvienos eilutės kiekvienas žodis prasidėtų fiksuotoje toje pačioje pozicijoje. Galima įterpti tik minimalų būtiną tarpų skaičių. Galima šalinti kelis iš eilės einančius vienodus skyriklius, paliekant tik vieną jų atstovą. Įterpimo ir šalinimo taisyklės taikome, siekdami gauti lygiuotą minimalų tekstą. Šalinimo taisyklės netaikome, jei nėra poreikio. Pradinio teksto eilutės ilgis neviršija 80 simbolių.

Spausdinkite faile `ManoKnyga.txt` pertvarkytą tekstą pagal tokias taisykles:

- kiekvienos eilutės pirmasis žodis turi prasidėti pozicijoje $p_1=1$.
- antrasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_2 , tokioje, kad kiekvienos eilutės pirmasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_2-2 arba p_2-2 .

- trečiasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_3 , tokioje, kad kiekvienos eilutės antrasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_3-2 arba p_3-2 .
- ir t.t.

U5-7. Pasikartojimai

Dviejuose tekstiniuose failuose `Knyga1.txt` ir `Knyga2.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma ir abejuose failuose yra ta pati. Analizuojant tekstus, didžiosios ir mažosios raidės nesvarbios.

Raskite, spausdinkite failu `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- ilgiausių žodžių, kurie yra tik faile `Knyga1.txt`, bet nėra faile `Knyga2.txt`, sąrašą (ne daugiau nei 10 žodžių) ir jų pasikartojimo skaičių;
- ilgiausią teksto fragmentą, sudarytą iš žodžių ir juos skiriančių skyriklių, kuris yra abejuose failuose ir jo eilutės numerius pirmame ir antrame faile;

Spausdinkite failu `ManoKnyga.txt` apjungtą tekstą, sudarytą pagal tokias taisykles:

- kopijuojamas pirmojo failo tekstas tol, kol sutinkamas pirmasis antrojo failo žodis arba pasiekama failo pabaiga;
- kopijuojamas antrojo failo tekstas tol, kol sutinkamas pirmasis nenukopijuotas pirmojo failo žodis arba pasiekama failo pabaiga;
- kartojama tol, kol pasiekama abiejų failų pabaiga.

U5-8. Skaičių suma

Tekstiniame faile `Knyga.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma. Raskite, spausdinkite failu `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- ilgiausią teksto fragmentą, sudarytą iš žodžių, kur žodžio paskutinė raidė sutampa su kito žodžio pirmąja raide (tarp didžiųjų ir mažųjų raidžių skirtumo nedaryti) ir juos skiriančių skyriklių, bei jo eilutės numerius;
- Žodžių, kuriuos sudaro tik skaitmenys, kiekį. Suskaičiuokite tokių skaičių bendrą sumą.

Reikia teksto žodžius sulygiuoti, kad kiekvienos eilutės kiekvienas žodis prasidėtų fiksuotoje toje pačioje pozicijoje. Galima įterpti tik minimalų būtiną tarpų skaičių. Galima šalinti kelis iš eilės einančius vienodus skyriklius, paliekant tik vieną jų atstovą. Įterpimo ir šalinimo taisyklės taikome, siekdami gauti lygiuotą minimalų tekstą. Šalinimo taisyklės netaikome, jei nėra poreikio. Pradinio teksto eilutės ilgis neviršija 80 simbolių.

Spausdinkite failu `ManoKnyga.txt` pertvarkytą tekstą pagal tokias taisykles:

- kiekvienos eilutės pirmasis žodis turi prasidėti pozicijoje $p_1=1$.
- antrasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_2 , tokioje, kad kiekvienos eilutės pirmasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_2-2 arba p_2-2 .
- trečiasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_3 , tokioje, kad kiekvienos eilutės antrasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_3-2 arba p_3-2 .
- ir t.t.

U5-9. Failo kopija

Dviejuose tekstiniuose failuose `Knyga1.txt` ir `Knyga2.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma ir abejuose failuose yra ta pati. Analizuojant tekstus, didžiosios ir mažosios raidės nesvarbios.

Raskite, spausdinkite failu `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- ilgiausių žodžių, kuris yra abejuose failuose, sąrašą (ne daugiau nei 10 žodžių) ir jų pasikartojimo skaičių kiekviename iš failų;
- ilgiausią sakinį, jo ilgį (simboliais ir žodžiais) ir vietą (sakinio pradžios eilutės numerį) pirmame ir antrame faile.

Spausdinkite failu `ManoKnyga.txt` apjungtą tekstą, sudarytą pagal tokias taisykles:

- kopijuojamas pirmojo failo tekstas tol, kol sutinkamas pirmasis antrojo failo žodis arba pasiekama failo pabaiga;
- kopijuojamas antrojo failo tekstas tol, kol sutinkamas pirmasis nenukopijuotas pirmojo failo žodis arba pasiekama failo pabaiga;
- kartojama tol, kol pasiekama abiejų failų pabaiga.

U5-10. Trumpiausi žodžiai

Tekstiniame faile `Knyga.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma. Raskite, spausdinkite faile `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- trumpiausių žodžių, bet ne trumpesnių nei 3 simboliai, sąrašą (ne daugiau nei 10 žodžių) ir jų pasikartojimo skaičių;
- ilgiausią teksto fragmentą, sudarytą iš žodžių, kur žodžio paskutinė raidė sutampa su kito žodžio pirmąja raide (tarp didžiųjų ir mažųjų raidžių skirtumo nedaryti) ir juos skiriančių skyriklių, bei jo eilutės numerius;

Reikia teksto žodžius sulygiuoti, kad kiekvienos eilutės kiekvienas žodis prasidėtų fiksuotoje toje pačioje pozicijoje. Galima įterpti tik minimalų būtiną tarpų skaičių. Galima šalinti kelis iš eilės einančius vienodus skyriklius, paliekant tik vieną jų atstovą. Įterpimo ir šalinimo taisyklės taikome, siekdami gauti lygiuotą minimalų tekstą. Šalinimo taisyklės netaikome, jei nėra poreikio. Pradinio teksto eilutės ilgis neviršija 80 simbolių.

Spausdinkite faile `ManoKnyga.txt` pertvarkytą tekstą pagal tokias taisykles:

- kiekvienos eilutės pirmasis žodis turi prasidėti pozicijoje $p_1=1$.
- antrasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_2 , tokioje, kad kiekvienos eilutės pirmasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_2-2 arba p_2-2 .
- trečiasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_3 , tokioje, kad kiekvienos eilutės antrasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_3-2 arba p_3-2 .
- ir t.t.

U5-11. Trumpiausias sakiny

Dviejuose tekstiniuose failuose `Knyga1.txt` ir `Knyga2.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma ir abejuose failuose yra ta pati. Analizuojant tekstus, didžiosios ir mažosios raidės nesvarbios.

Raskite, spausdinkite faile `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- ilgiausių žodžių, kurie yra tik faile `Knyga1.txt`, bet nėra faile `Knyga2.txt`, sąrašą (ne daugiau nei 10 žodžių) ir jų pasikartojimo skaičių;
- trumpiausią sakinį, bet ne trumpesnį, nei iš 3 žodžių, jo ilgį (simboliais ir žodžiais) ir vietą (sakinio pradžios eilutės numerį) pirmame ir antrame faile.

Spausdinkite faile `ManoKnyga.txt` apjungtą tekstą, sudarytą pagal tokias taisykles:

- kopijuojamas pirmojo failo tekstas tol, kol sutinkamas pirmasis antrojo failo žodis arba pasiekama failo pabaiga;
- kopijuojamas antrojo failo tekstas tol, kol sutinkamas pirmasis nenukopijuotas pirmojo failo žodis arba pasiekama failo pabaiga;
- kartojama tol, kol pasiekama abiejų failų pabaiga.

U5-12. Pasikartojantys žodžiai

Tekstiniame faile `Knyga.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma. Raskite, spausdinkite faile `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- Nurodytą kiekį dažniausiai pasikartojančių žodžių (ne daugiau nei 10 žodžių), surikiuotą pagal pasikartojimo skaičių;
- ilgiausią sakinį, jo ilgį (simboliais ir žodžiais) ir vietą (sakinio pradžios eilutės numerį).

Reikia teksto žodžius sulygiuoti, kad kiekvienos eilutės kiekvienas žodis prasidėtų fiksuotoje toje pačioje pozicijoje. Galima įterpti tik minimalų būtiną tarpų skaičių. Galima šalinti kelis iš eilės einančius vienodus

skyriklius, paliekant tik vieną jų atstovą. Įterpimo ir šalinimo taisyklės taikome, siekdami gauti lygiuotą minimalų tekstą. Šalinimo taisyklės netaikome, jei nėra poreikio. Pradinio teksto eilutės ilgis neviršija 80 simbolių.

Spausdinkite failę `ManoKnyga.txt` pertvarkytą tekstą pagal tokias taisykles:

- kiekvienos eilutės pirmasis žodis turi prasidėti pozicijoje $p_1=1$.
- antrasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_2 , tokioje, kad kiekvienos eilutės pirmasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_2-2 arba p_2-2 .
- trečiasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_3 , tokioje, kad kiekvienos eilutės antrasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_3-2 arba p_3-2 .
- ir t.t.

U5-13. Unikalus žodžiai

Dviejuose tekstiniuose failuose `Knyga1.txt` ir `Knyga2.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma ir abejuose failuose yra ta pati. Analizuojant tekstus, didžiosios ir mažosios raidės nesvarbios.

Raskite, spausdinkite failę `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- unikalių žodžių, kurie yra tik failę `Knyga1.txt`, bet nėra failę `Knyga2.txt`, skaičių ir tokių žodžių sąrašą (ne daugiau nei 10 žodžių), surikiuotą pagal pasikartojimo skaičių;
- unikalių žodžių, kurie yra abejuose failuose, skaičių ir tokių žodžių sąrašą (ne daugiau nei 10 žodžių), surikiuotą pagal pasikartojimo skaičių;

Spausdinkite failę `ManoKnyga.txt` apjungtą tekstą, sudarytą pagal tokias taisykles:

- kopijuojamas pirmojo failo tekstas tol, kol sutinkamas pirmasis antrojo failo žodis arba pasiekama failo pabaiga;
- kopijuojamas antrojo failo tekstas tol, kol sutinkamas pirmasis nenukopijuotas pirmojo failo žodis arba pasiekama failo pabaiga;
- kartojama tol, kol pasiekama abiejų failų pabaiga.

U5-14. Lygiavimas

Tekstiniame failę `Knyga.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma.

Raskite, spausdinkite failę `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- unikalių žodžių skaičių ir tokių žodžių sąrašą (ne daugiau nei 10 žodžių), surikiuotą pagal pasikartojimo skaičių;
- ilgiausių žodžių sąrašą (ne daugiau nei 10 žodžių) ir jų pasikartojimo skaičių;

Reikia teksto žodžius sulygiuoti, kad kiekvienos eilutės kiekvienas žodis prasidėtų fiksuotoje toje pačioje pozicijoje. Galima įterpti tik minimalų būtiną tarpų skaičių. Galima šalinti kelis iš eilės einančius vienodus skyriklius, paliekant tik vieną jų atstovą. Įterpimo ir šalinimo taisyklės taikome, siekdami gauti lygiuotą minimalų tekstą. Šalinimo taisyklės netaikome, jei nėra poreikio. Pradinio teksto eilutės ilgis neviršija 80 simbolių.

Spausdinkite failę `ManoKnyga.txt` pertvarkytą tekstą pagal tokias taisykles:

- kiekvienos eilutės pirmasis žodis turi prasidėti pozicijoje $p_1=1$.
- antrasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_2 , tokioje, kad kiekvienos eilutės pirmasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_2-2 arba p_2-2 .
- trečiasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_3 , tokioje, kad kiekvienos eilutės antrasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_3-2 arba p_3-2 .
- ir t.t.

U5-15. Ilgiausi žodžiai

Dviejuose tekstiniuose failuose `Knyga1.txt` ir `Knyga2.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma ir abejuose failuose yra ta pati. Analizuojant tekstus, didžiosios ir mažosios raidės nesvarbios.

Raskite, spausdinkite failę `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- ilgiausių žodžių, kuris yra abejuose failuose, sąrašą (ne daugiau nei 10 žodžių) ir jų pasikartojimo skaičių kiekviename iš failų;
- ilgiausių žodžių, kurie yra tik faile `Knyga1.txt`, bet nėra faile `Knyga2.txt`, sąrašą (ne daugiau nei 10 žodžių) ir jų pasikartojimo skaičių;

Spausdinkite faile `ManoKnyga.txt` apjungtą tekstą, sudarytą pagal tokias taisykles:

- kopijuojamas pirmojo failo tekstas tol, kol sutinkamas pirmasis antrojo failo žodis arba pasiekama failo pabaiga;
- kopijuojamas antrojo failo tekstas tol, kol sutinkamas pirmasis nenukopijuotas pirmojo failo žodis arba pasiekama failo pabaiga;
- kartojama tol, kol pasiekama abiejų failų pabaiga.

U5-16. Teksto fragmentai

Tekstiniame faile `Knyga.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma. Raskite, spausdinkite faile `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- unikalių žodžių, prasidedančių nurodyta raide, sąrašą (ne daugiau nei 10 žodžių) ir jų pasikartojimo skaičių;
- ilgiausią teksto fragmentą, sudarytą iš žodžių, kur žodžio paskutinė raidė sutampa su kito žodžio pirmąja raide (tarp didžiųjų ir mažųjų raidžių skirtumo nedaryti) ir juos skiriančių skyriklių, bei jo pradžios ir pabaigos eilutės numerius;

Reikia teksto žodžius sulygiuoti, kad kiekvienos eilutės kiekvienas žodis prasidėtų fiksuotoje toje pačioje pozicijoje. Galima įterpti tik minimalų būtiną tarpų skaičių. Galima šalinti kelis iš eilės einančius vienodus skyriklius, paliekant tik vieną jų atstovą. Įterpimo ir šalinimo taisyklės taikome, siekdami gauti lygiuotą minimalų tekstą. Šalinimo taisyklės netaikome, jei nėra poreikio. Pradinio teksto eilutės ilgis neviršija 80 simbolių.

Spausdinkite faile `ManoKnyga.txt` pertvarkytą tekstą pagal tokias taisykles:

- kiekvienos eilutės pirmasis žodis turi prasidėti pozicijoje $p_1=1$.
- antrasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_2 , tokioje, kad kiekvienos eilutės pirmasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_2-2 arba p_2-2 .
- trečiasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_3 , tokioje, kad kiekvienos eilutės antrasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_3-2 arba p_3-2 .
- ir t.t.

U5-17. Ilgiausias sakiny

Dviejuose tekstiniuose failuose `Knyga1.txt` ir `Knyga2.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma ir abejuose failuose yra ta pati. Analizuojant tekstus, didžiosios ir mažosios raidės nesvarbios.

Raskite, spausdinkite faile `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- unikalių žodžių, kurie yra tik faile `Knyga1.txt`, bet nėra faile `Knyga2.txt`, skaičių ir tokių žodžių sąrašą (ne daugiau nei 10 žodžių), surikiuotą pagal pasikartojimo skaičių;
- ilgiausią sakinį, jo ilgį (simboliais ir žodžiais) ir vietą (sakinio pradžios eilutės numerį) pirmame ir antrame faile.

Spausdinkite faile `ManoKnyga.txt` apjungtą tekstą, sudarytą pagal tokias taisykles:

- kopijuojamas pirmojo failo tekstas tol, kol sutinkamas pirmasis antrojo failo žodis arba pasiekama failo pabaiga;
- kopijuojamas antrojo failo tekstas tol, kol sutinkamas pirmasis nenukopijuotas pirmojo failo žodis arba pasiekama failo pabaiga;
- kartojama tol, kol pasiekama abiejų failų pabaiga.

U5-18. Skaitmenys

Tekstiniame faile `Knyga.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma. Raskite, spausdinkite faile `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- ilgiausią sakinį, jo ilgį (simboliais ir žodžiais) ir vietą (sakinio pradžios eilutės numerį).
- Žodžių, kuriuos sudaro tik skaitmenys, kiekį. Suskaičiuokite tokių skaičių bendrą sumą.

Reikia teksto žodžius sulygiuoti, kad kiekvienos eilutės kiekvienas žodis prasidėtų fiksuotoje toje pačioje pozicijoje. Galima įterpti tik minimalų būtiną tarpų skaičių. Galima šalinti kelis iš eilės einančius vienodus skyriklius, paliekant tik vieną jų atstovą. Įterpimo ir šalinimo taisyklės taikome, siekdami gauti lygiuotą minimalų tekstą. Šalinimo taisyklės netaikome, jei nėra poreikio. Pradinio teksto eilutės ilgis neviršija 80 simbolių.

Spausdinkite faile `ManoKnyga.txt` pertvarkytą tekstą pagal tokias taisykles:

- kiekvienos eilutės pirmasis žodis turi prasidėti pozicijoje $p_1=1$.
- antrasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_2 , tokioje, kad kiekvienos eilutės pirmasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_2-2 arba p_2-2 .
- trečiasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_3 , tokioje, kad kiekvienos eilutės antrasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_3-2 arba p_3-2 .
- ir t.t.

U5-19. Pasikartojimai

Dviejuose tekstiniuose failuose `Knyga1.txt` ir `Knyga2.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma ir abejuose failuose yra ta pati. Analizuojant tekstus, didžiosios ir mažosios raidės nesvarbios.

Raskite, spausdinkite faile `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- ilgiausių žodžių, kurie yra tik faile `Knyga1.txt`, bet nėra faile `Knyga2.txt`, sąrašą (ne daugiau nei 10 žodžių) ir jų pasikartojimo skaičių;
- ilgiausią teksto fragmentą, sudarytą iš žodžių ir juos skiriančių skyriklių, kuris yra abejuose failuose ir jo eilutės numerius pirmame ir antrame faile;

Spausdinkite faile `ManoKnyga.txt` apjungtą tekstą, sudarytą pagal tokias taisykles:

- kopijuojamas pirmojo failo tekstas tol, kol sutinkamas pirmasis antrojo failo žodis arba pasiekama failo pabaiga;
- kopijuojamas antrojo failo tekstas tol, kol sutinkamas pirmasis nenukopijuotas pirmojo failo žodis arba pasiekama failo pabaiga;
- kartojama tol, kol pasiekama abiejų failų pabaiga.

U5-20. Skaičių suma

Tekstiniame faile `Knyga.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma. Raskite, spausdinkite faile `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- ilgiausią teksto fragmentą, sudarytą iš žodžių, kur žodžio paskutinė raidė sutampa su kito žodžio pirmąja raide (tarp didžiųjų ir mažųjų raidžių skirtumo nedaryti) ir juos skiriančių skyriklių, bei jo eilutės numerius;
- Žodžių, kuriuos sudaro tik skaitmenys, kiekį. Suskaičiuokite tokių skaičių bendrą sumą.

Reikia teksto žodžius sulygiuoti, kad kiekvienos eilutės kiekvienas žodis prasidėtų fiksuotoje toje pačioje pozicijoje. Galima įterpti tik minimalų būtiną tarpų skaičių. Galima šalinti kelis iš eilės einančius vienodus skyriklius, paliekant tik vieną jų atstovą. Įterpimo ir šalinimo taisyklės taikome, siekdami gauti lygiuotą minimalų tekstą. Šalinimo taisyklės netaikome, jei nėra poreikio. Pradinio teksto eilutės ilgis neviršija 80 simbolių.

Spausdinkite faile `ManoKnyga.txt` pertvarkytą tekstą pagal tokias taisykles:

- kiekvienos eilutės pirmasis žodis turi prasidėti pozicijoje $p_1=1$.
- antrasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_2 , tokioje, kad kiekvienos eilutės pirmasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_2-2 arba p_2-2 .
- trečiasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_3 , tokioje, kad kiekvienos eilutės antrasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_3-2 arba p_3-2 .

- ir t.t.

U5-21. Failo kopija

Dviejuose tekstiniuose failuose `Knyga1.txt` ir `Knyga2.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma ir abejuose failuose yra ta pati. Analizuojant tekstus, didžiosios ir mažosios raidės nesvarbios.

Raskite, spausdinkite faile `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- ilgiausių žodžių, kuris yra abejuose failuose, sąrašą (ne daugiau nei 10 žodžių) ir jų pasikartojimo skaičių kiekviename iš failų;
- ilgiausią sakinį, jo ilgį (simboliais ir žodžiais) ir vietą (sakinio pradžios eilutės numerį) pirmame ir antrame faile.

Spausdinkite faile `ManoKnyga.txt` apjungtą tekstą, sudarytą pagal tokias taisykles:

- kopijuojamas pirmojo failo tekstas tol, kol sutinkamas pirmasis antrojo failo žodis arba pasiekama failo pabaiga;
- kopijuojamas antrojo failo tekstas tol, kol sutinkamas pirmasis nenukopijuotas pirmojo failo žodis arba pasiekama failo pabaiga;
- kartojama tol, kol pasiekama abiejų failų pabaiga.

U5-22. Trumpiausi žodžiai

Tekstiniame faile `Knyga.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma. Raskite, spausdinkite faile `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- trumpiausių žodžių, bet ne trumpesnių nei 3 simboliai, sąrašą (ne daugiau nei 10 žodžių) ir jų pasikartojimo skaičių;
- ilgiausią teksto fragmentą, sudarytą iš žodžių, kur žodžio paskutinė raidė sutampa su kito žodžio pirmąja raide (tarp didžiųjų ir mažųjų raidžių skirtumo nedaryti) ir juos skiriančių skyriklių, bei jo eilutės numerius;

Reikia teksto žodžius sulygiuoti, kad kiekvienos eilutės kiekvienas žodis prasidėtų fiksuotoje toje pačioje pozicijoje. Galima įterpti tik minimalų būtiną tarpų skaičių. Galima šalinti kelis iš eilės einančius vienodus skyriklius, paliekant tik vieną jų atstovą. Įterpimo ir šalinimo taisyklės taikome, siekdami gauti lygiuotą minimalų tekstą. Šalinimo taisyklės netaikome, jei nėra poreikio. Pradinio teksto eilutės ilgis neviršija 80 simbolių.

Spausdinkite faile `ManoKnyga.txt` pertvarkytą tekstą pagal tokias taisykles:

- kiekvienos eilutės pirmasis žodis turi prasidėti pozicijoje $p_1=1$.
- antrasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_2 , tokioje, kad kiekvienos eilutės pirmasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_2-2 arba p_2-2 .
- trečiasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_3 , tokioje, kad kiekvienos eilutės antrasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_3-2 arba p_3-2 .
- ir t.t.

U5-23. Trumpiausias sakiny

Dviejuose tekstiniuose failuose `Knyga1.txt` ir `Knyga2.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma ir abejuose failuose yra ta pati. Analizuojant tekstus, didžiosios ir mažosios raidės nesvarbios.

Raskite, spausdinkite faile `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- ilgiausių žodžių, kurie yra tik faile `Knyga1.txt`, bet nėra faile `Knyga2.txt`, sąrašą (ne daugiau nei 10 žodžių) ir jų pasikartojimo skaičių;
- trumpiausią sakinį, bet ne trumpesnį, nei iš 3 žodžių, jo ilgį (simboliais ir žodžiais) ir vietą (sakinio pradžios eilutės numerį) pirmame ir antrame faile.

Spausdinkite faile `ManoKnyga.txt` apjungtą tekstą, sudarytą pagal tokias taisykles:

- kopijuojamas pirmojo failo tekstas tol, kol sutinkamas pirmasis antrojo failo žodis arba pasiekama failo pabaiga;
- kopijuojamas antrojo failo tekstas tol, kol sutinkamas pirmasis nenukopijuotas pirmojo failo žodis arba pasiekama failo pabaiga;
- kartojama tol, kol pasiekama abiejų failų pabaiga.

U5-24. Pasikartojantys žodžiai

Tekstiniame faile `Knyga.txt` duotas tekstas sudarytas iš žodžių, atskirtų skyrikliais. Skyriklių aibė žinoma. Raskite, spausdinkite faile `Analizė.txt` ir išveskite ekrane teksto analizės rezultatus:

- Nurodytą kiekį dažniausiai pasikartojančių žodžių (ne daugiau nei 10 žodžių), surikiuotą pagal pasikartojimo skaičių;
- ilgiausią sakinį, jo ilgį (simboliais ir žodžiais) ir vietą (sakinio pradžios eilutės numerį).

Reikia teksto žodžius sulygiuoti, kad kiekvienos eilutės kiekvienas žodis prasidėtų fiksuotoje toje pačioje pozicijoje. Galima įterpti tik minimalų būtiną tarpų skaičių. Galima šalinti kelis iš eilės einančius vienodus skyriklius, paliekant tik vieną jų atstovą. Įterpimo ir šalinimo taisyklės taikome, siekdami gauti lygiuotą minimalų tekstą. Šalinimo taisyklės netaikome, jei nėra poreikio. Pradinio teksto eilutės ilgis neviršija 80 simbolių.

Spausdinkite faile `ManoKnyga.txt` pertvarkytą tekstą pagal tokias taisykles:

- kiekvienos eilutės pirmasis žodis turi prasidėti pozicijoje $p_1=1$.
- antrasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_2 , tokioje, kad kiekvienos eilutės pirmasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_2-2 arba p_2-2 .
- trečiasis kiekvienos eilutės žodis turi prasidėti minimalioje galimoje pozicijoje p_3 , tokioje, kad kiekvienos eilutės antrasis žodis kartu su už jo esančiais skyrikliais baigiasi iki p_3-2 arba p_3-2 .
- ir t.t.

U5-25. Krepšinio rinktinė.

Daug krepšinio rinktinės kandidatų ir kitų narių nebuvo įtraukti į rinktinės narių sąrašą. Duomenys apie juos prieinami rinktinės puslapyje. Jums reikia įtraukti juos į atitinkamų metų rinktinės narių sąrašus, todėl reikia konvertuoti HTML formatu saugomus duomenis į CSV formatą. Puslapyje pateiktuose duomenyse pastebėta klaidų.

Išnagrinėjus duomenis, pateiktus tinklalapyje, paaiškėjo, kad:

- kai kurių narių vardai ir pavardės prasideda mažąja raide;
- kai kurių krepšininkų pozicijos pateiktos skaičiais 1, 2, 3, 4 bei 5, kur 1 – įžaidėjas, 2 – atakuojantis gynėjas, 3 – lengvasis krašto puolėjas, 4 – sunkusis krašto puolėjas, 5 – vidurio puolėjas;
- kai kurių krepšininkų ūgio reikšmė pateikta klaidingai. Ūgio reikšmė pateikiama centimetrais, ją visada sudaro 3 skaitmenys (pvz. 214 – tai 2 m 14 cm). Duomenyse pastebėta reikšmių, turinčių daugiau arba mažiau skaitmenų nei reikalinga nusakyti ūgiui;
- duomenys apie narius pateikiami nevienodai (kai kur pritaikytas *Bold*, *Italic* ar *Underline* formatavimas).

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytą tinklalapio failą (pradinius duomenis galite susidaryti pagal pratybą 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuotų į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (``, `<i>`, `<u>` ar bet kurias kitas);
- krepšininko poziciją, užrašytą skaičiumi, pakeistų tekstu: 1 – įžaidėjas, 2 – atakuojantis gynėjas, 3 – lengvasis krašto puolėjas, 4 – sunkusis krašto puolėjas, 5 – vidurio puolėjas;
- surastų duomenų eilutes, kuriose neteisingai įvesta ūgio reikšmė. Ūgio reikšmė nurodoma centimetrais, ją sudaro 3 skaitmenys, ji gali prasidėti tik 1 arba 2. Neteisingas eilutes surašytų į failą `KlaidingiDuomenys.csv`;
- pašalintų iš duomenų eilučių tas, kuriose yra neteisinga ūgio reikšmė, ir išvestų teisingus duomenis į failą `Rinktinė.csv`.

U5-26. Automobilių parkas.

Įmonė UAB „Žaibas“ perėmė bankrutavusių konkurentų verslą ir jų turimus automobilius. Duomenys apie turimus automobilių modelius prieinami tik per bankrutavusios įmonės interneto puslapį, todėl reikia konvertuoti HTML formatu pateiktus duomenis į CSV formatą. Puslapyje pateiktuose duomenyse pastebėta klaidų.

Išnagrinėjus duomenis, pateiktus tinklalapyje, paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas *Bold*, *Italic* ar *Underline* formatavimas);
- ne visos raidės valstybiniuose automobilių numeriuose yra didžiosios;
- kai kurie valstybiniai automobilių numeriai nėra validūs, t.y. netenkina numeriams keliamų reikalavimų, pagal kuriuos jie turi susidėti iš 6 simbolių, kur pirmi 3 simboliai yra lotyniškos raidės, o likę 3 – skaičiai.

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytų tinklalapio failą (pradinius duomenis galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuotų į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
- valstybiniame automobilio numeryje esančias mažąsias raides pakeistų į didžiąsias;
- patikrintų, ar valstybinis automobilio numeris yra validus: numeris turi susidaryti iš 6 simbolių, kur 3 pirmieji simboliai yra lotyniškos raidės, o likę 3 – skaičiai. Neteisingas eilutes surašytų į failą `KlaidingiDuomenys.csv`;
- pašalintų iš duomenų eilučių tas, kuriose nurodytas nevalidus valstybinis automobilio numeris, ir išvestų teisingus duomenis į failą `Automobiliai.csv`.

U5-27. Studentų atstovybė.

Norite papildyti atstovybės narių informaciją iš studentų atstovybės interneto puslapio, todėl reikia konvertuoti HTML formatu pateiktus duomenis į CSV formatą. Puslapyje pateiktuose duomenyse pastebėta klaidų.

Išnagrinėjus duomenis, pateiktus tinklalapyje, paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas *Bold*, *Italic* ar *Underline* formatavimas);
- kai kurių narių vardai ir pavardės prasideda mažąja raide;
- vieni telefonų numeriai prasideda skaičiais 8, kiti: +370;
- kai kurios gimimo datos įvestos neteisingai. Datos formato pavyzdys – 1995-08-30. Kai kurios datos įvestos be brūkšnių, pvz. 19950830. Taip pat kai kurios datos turi per daug arba per mažai skaičių.

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytų tinklalapio failą (pradinius duomenis galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuotų į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
- gimimo datas, kurios pateiktos be brūkšnių, konvertuotų į datas su brūkšniais (pvz. 19950830 -> 1995-08-30);
- patikrintų, ar gimimo data yra validi, t.y. atitinka formatą ir turi tiek simbolių, kiek reikia. Neteisingas duomenų eilutes surašytų į failą `KlaidingiDuomenys.csv`;
- pašalintų iš duomenų eilučių tas, kuriose nurodyta nevalidi gimimo data, ir išvestų teisingus duomenis į failą `Studentai.csv`.

U5-28. Biblioteka.

Bibliotekos buvo papildytos naujomis knygomis. Užsisakius knygas gavote užsakytų knygų sąrašą, pateiktą užsakovo interneto svetainės puslapyje. Jums reikia informaciją apie knygas konvertuoti iš HTML į CSV

formatą. Tiekėjo pateiktos užsakytų knygų atskaitos formatai neatitinka jūsų naudojamo duomenų saugojimo formato.

Išnagrinėjus duomenis, pateiktus tinklalapyje, paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas *Bold*, *Italic* ar *Underline* formatavimas);
- ISBN kodas tiekėjo suvestinėje pateikiamas kaip 13 iš eilės einančių skaitmenų seka. Jūsų sistemoje dedami brūkšniai atitinkamose ISBN kodo vietose (pvz. tiekėjo ISBN kodas – 9786090211267, jūsų ISBN kodas – 978-6-09-021126-7);
- kai kur ISBN kodas nėra validus – jis turi daugiau arba mažiau nei 13 skaitmenų.

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytų tinklalapio failą (pradinius duomenis galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuoti į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
- konvertuotų ISBN kodo formatą taip, kad būtų tinkamas jūsų sistemai (pvz. 9786090211267 → 978-6-09-021126-7);
- duomenų eilutes, kuriose ISBN kodas nėra validus (turi daugiau arba mažiau nei 13 skaitmenų arba turi raidžių) surašyti į failą `KlaidingiDuomenys.csv`;
- pašalintų iš duomenų eilučių tas, kuriose nurodytas nevalidus ISBN kodas, ir išvestų teisingus duomenis į failą `Knygos.csv`.

U5-29. Proto mūšis.

Studentų atstovybė aptiko interneto puslapį, kuriame pateikta daug įdomių klausimų. Klausimų kopijavimas iš puslapio užimtų daug laiko, todėl buvo nuspręsta informaciją apie klausimus, pateiktą HTML formatu, konvertuoti į CSV formatą. Pateiktoje informacijoje apie klausimus pastebėta formato neatitikimų ir klaidų.

Išnagrinėjus duomenis, pateiktus tinklalapyje, paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas *Bold*, *Italic* ar *Underline* formatavimas);
- kai kurie puslapio klausimai turi daugiau arba mažiau nei 4 atsakymo variantus;
- kai kurie autorių vardai prasideda mažąja raide.

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytų tinklalapio failą (pradinius duomenis galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuoti į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
- patikrintų, ar klausimas turi daugiau nei 4 atsakymo variantus, ir perteklinius variantus pašalintų (prieš tai patikrinant, ar nėra šalinamas teisingo atsakymo variantas);
- patikrintų, ar klausimas turi mažiau nei 4 atsakymo variantus. Jei taip, tokius klausimus surašyti į failą `TrūkstaAtsakymoVariantų.csv`;
- vardus, prasidedančius mažąja raide, pakeistų į vardus, prasidedančius didžiąja raide;
- pašalintų klausimus, kuriems trūksta atsakymo variantų, ir likusius klausimus surašyti į failą `TinkamiKlausimai.csv`.

U5-30. Nekilnojamojo turto agentūra.

Gavus vartotojų sutikimą, buvo nuspręsta nekilnojamojo turto agentūrų duomenis papildyti duomenimis iš privačių skelbimų interneto svetainės, todėl prireikė svetainėje pateiktus duomenis konvertuoti iš HTML formato į CSV formatą. Puslapyje pateiktoje nekilnojamojo turto informacijoje pastebėta formato neatitikimų ir klaidų.

Išnagrinėjus duomenis, pateiktus tinklalapyje, paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas *Bold*, *Italic* ar *Underline* formatavimas);
- duomenys apie gatvę, buto numerį ir aukštą pateikiami kaip viena eilutė (pvz. K. Baršausko g. 88-6);
- kai kuriems butams nenurodytas buto numeris.

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytų tinklalapio failą (pradinius duomenys galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuotų į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
- adreso duomenis išskaidytų į atskirus duomenų laukus (pvz. K. Baršausko g. 88-6 – gatvė: K. Baršausko, namo numeris: 88, buto numeris: 6);
- saugant gatvės pavadinimą, išmesti iš jo užrašą „g.“ arba „gatvė“ (pvz. K. Baršausko g. → K. Baršausko);
- butų duomenų eilutes, kurioms trūksta buto numerio, surašytų į failą `TrūkstaDuomenų.csv`;
- pašalintų duomenų eilutes, kurioms trūksta duomenų ir likusias eilutes surašytų į duomenų failą `Nekilnojamasturtas.csv`.

U5-31. WCG turnyras.

Turite ketvirto turnyro duomenis interneto puslapyje. Kiekvienas turnyro dalyvis turi pateikti savo el. pašto adresą, todėl klasėje „Žaidėjas“ atsiranda papildomas duomenų laukas *email*. Duomenis, pateiktus interneto puslapyje, reikia konvertuoti iš HTML formato į CSV formatą. Pastebėta, kad puslapyje pateiktoje informacijoje duomenys nėra visiškai patikimi.

Išnagrinėjus duomenis, pateiktus tinklalapyje, paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas *Bold*, *Italic* ar *Underline* formatavimas);
- el. pašto adresai nurodyti nekorektiškai, t.y. neatitinka formato (pvz. neturi @ simbolio, pabaigoje neturi taško ir/ar domeno vardo).

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytų tinklalapio failą (pradinius duomenys galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuotų į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
- patikrintų, ar el. pašto adresas yra validus ir tenkina šablono reikalavimus;
- duomenų eilutes su nevalidžiais el. pašto adresais surašyti į failą `KlaidingiDuomenys.csv`;
- pašalintų duomenų eilutes su nevalidžiais el. pašto adresais ir likusius duomenis surašytų į failą `Žaidėjai.csv`.

U5-32. Turistų informacijos centras.

Turizmo informacijos centras pradės teikti informaciją apie viso Baltijos šalių regiono muziejus. Didelė dalis informacijos apie kitų šalių muziejus pateikiama internetinėse svetainėse, todėl norint perkelti duomenis iš šių svetainių, reikia konvertuoti HTML formato duomenis į CSV formatą. Muziejai, kurie teikia gido paslaugą, taip pat nurodo ir jo el. pašto adresą, todėl duomenyse atsiranda papildomas laukas *email*. Pastebėta, kad puslapiuose pateiktoje informacijoje duomenys nėra visiškai patikimi.

Išnagrinėjus duomenis, pateiktus tinklalapyje, paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas *Bold*, *Italic* ar *Underline* formatavimas);
- el. pašto adresai nurodyti nekorektiškai, t.y. neatitinka formato (pvz. neturi @ simbolio, pabaigoje neturi taško ir/ar domeno vardo).

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytą tinklalapio failą (pradinius duomenis galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuoti į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
- patikrintų, ar el. pašto adresas yra validus ir tenkina šablono reikalavimus;
- duomenų eilutes su nevalidžiais el. pašto adresais surašytų į failą `KlaidingiDuomenys.csv`;
- pašalintų duomenų eilutes su nevalidžiais el. pašto adresais ir likusius duomenis surašytų į failą `Muziejai.csv`.

U5-33. IMDb.

Rodote įdomų filmų sąrašą, kurį pateikia kino mėgėjas savo internetinėje svetainėje. Nusprendėte papildyti filmų sąrašą filmais iš jo svetainės, todėl jums reikia konvertuoti filmų sąrašą iš HTML į CSV formatą. Pastebėjote, kad kai kurių aktorių pavardės pateiktos lietuviškai. Taip pat pateiktuose duomenyse pastebėjote ir daugiau netikslumų.

Išnagrinėjus duomenis, pateiktus tinklalapyje, paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas *Bold*, *Italic* ar *Underline* formatavimas);
- aktoriaus Nikolo Keidžo vardas vienur užrašomas lietuviškai (Nikolas Keidžas), kitur – angliškai (Nicolas Cage);
- režisieriaus lauke pirmiau pateikiama pavardė, o po tarpelio vardas.

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytą tinklalapio failą (pradinius duomenis galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuoti į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
- suvienodintų informaciją, pateikiamą apie aktorių Nikolą Keidžą; turi likti arba tik angliški vardai ir pavardės, arba tik lietuviški;
- sukeistų duomenis režisieriaus lauke taip, kad pirmoje vietoje būtų vardas, po to pavardė;
- leistų rasti filmus pagal tam tikrą frazę pavadinime. Surastus filmus saugotų paieškos frazės vardo CSV faile.

U5-34. Buitinės technikos parduotuvė.

Buvo nuspręsta realizuoti funkciją, kuri leistų palyginti buitinių prietaisų kainas konkurentų parduotuvėse. Duomenys apie konkurentų prekes yra pateikiami jų interneto svetainėse, todėl reikia konvertuoti HTML formato duomenis į CSV formatą. Pastebėta, kad puslapiuose pateiktoje informacijoje duomenys yra šiek tiek iškraipyti.

Išnagrinėjus duomenis, pateiktus tinklalapyje, paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas *Bold*, *Italic* ar *Underline* formatavimas);
- duomenys apie šaldytuvų matmenis pateikiami skirtingais vienetais (tinklalapyje – mm ir ml, jūsų sistemoje – cm ir l);
- duomenys apie virdulių talpą pateikiami skirtingais vienetais (tinklalapyje – ml, jūsų sistemoje – l).

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytą tinklalapio failą (pradinius duomenis galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuoti į CSV formato eilutes;
 - iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
 - milimetrais pateikiamus šaldytuvų matmenis paverstų į cm ir prie atitinkamo dydžio prirašytų vienetus (pvz. 1000 – „100 cm“);
-

- mililitrais pateikiamų virdulių talpas paverstų į l ir prie atitinkamo dydžio prirašytų vienetus (pvz. 1500 – „1,5 l“);
- gautas duomenų eilutes surašytų į failą `Gaminiai.csv`.

U5-35. Juvelyrikos parduotuvė.

Reikia importuoti juvelyrinių gaminių informaciją iš internetinio tinklalapio, todėl prirėkė konvertuoti HTML formato duomenis į CSV formatą. Pastebėta, kad puslapyje pateiktoje informacijoje duomenys yra šiek tiek iškraipyti.

Išnagrinėjus duomenis pateiktus tinklalapyje paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas bold, italic ar underline formatavimas);

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytų tinklalapio failą (pradinius duomenys galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuotų į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
- iš aukso pagamintuose juvelyriniuose gaminiuose reikia patikslinti aukso pavadinimą pagal prabą. Jei aukso praba yra mažesnė nei 500 (imtinai), tokio aukso pavadinimą reikia pakeisti į „Pigus juvelyrinis auksas“. Jei praba tarp 501 ir 900 (imtinai), aukso pavadinimas – „Juvelyrinis auksas“. Jei praba tarp 901 ir 999 (imtinai), aukso pavadinimas – „Brangus juvelyrinis“. Jeigu praba lygi 1000, tokį auksą reikia pervadinti į „Chemiškai grynas auksas“;
- aukso gaminius, kurie pagaminti iš pigaus juvelyrinio aukso, surašytų į failą `PigūsGaminiai.csv`;
- aukso gaminius, kurie pagaminti iš pigaus juvelyrinio aukso, pašalinti, o likusius gaminius surašyti į failą `JuvelyriniaiGaminiai.csv`.

U5-36. Kompiuterinis žaidimas.

Kai kurių žaidimo herojų informacija pateikiama internetinėje svetainėje. Reikia konvertuoti HTML formato duomenis į CSV formatą. Pastebėta, kad puslapyje pateiktoje informacijoje duomenys yra šiek tiek iškraipyti.

Išnagrinėjus duomenis pateiktus tinklalapyje paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas bold, italic ar underline formatavimas);

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytų tinklalapio failą (pradinius duomenys galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuotų į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
- iš aukso pagamintuose juvelyriniuose gaminiuose reikia patikslinti aukso pavadinimą pagal prabą. Jei aukso praba yra mažesnė nei 500 (imtinai), tokio aukso pavadinimą reikia pakeisti į „Pigus juvelyrinis auksas“. Jei praba tarp 501 ir 900 (imtinai), aukso pavadinimas – „Juvelyrinis auksas“. Jei praba tarp 901 ir 999 (imtinai), aukso pavadinimas – „Brangus juvelyrinis“. Jeigu praba lygi 1000, tokį auksą reikia pervadinti į „Chemiškai grynas auksas“;
- aukso gaminius, kurie pagaminti iš pigaus juvelyrinio aukso, surašytų į failą `PigūsGaminiai.csv`;
- aukso gaminius, kurie pagaminti iš pigaus juvelyrinio aukso, pašalinti, o likusius gaminius surašyti į failą `JuvelyriniaiGaminiai.csv`.

U5-37. Krepšinio rinktinė.

Daug krepšinio rinktinės kandidatų ir kitų narių nebuvo įtraukti į rinktinės narių sąrašą. Duomenys apie juos prieinami rinktinės puslapyje. Jums reikia įtraukti juos į atitinkamų metų rinktinės narių sąrašus, todėl reikia

konvertuoti HTML formatu saugomus duomenis į CSV formatą. Puslapyje pateiktuose duomenyse pastebėta klaidų.

Išnagrinėjus duomenis, pateiktus tinklalapyje, paaiškėjo, kad:

- kai kurių narių vardai ir pavardės prasideda mažąja raide;
- kai kurių krepšininkų pozicijos pateiktos skaičiais 1, 2, 3, 4 bei 5, kur 1 – įžaidėjas, 2 – atakuojantis gynėjas, 3 – lengvasis krašto puolėjas, 4 – sunkusis krašto puolėjas, 5 – vidurio puolėjas;
- kai kurios gimimo datos įvestos neteisingai. Datos formato pavyzdys – 1995-08-30. Kai kurios datos įvestos be brūkšnių, pvz. 19950830. Taip pat kai kurios datos turi per daug arba per mažai skaičių;
- duomenys apie narius pateikiami nevienodai (kai kur pritaikytas *Bold*, *Italic* ar *Underline* formatavimas).

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytą tinklalapio failą (pradinius duomenys galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuotų į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
- gimimo datas, kurios pateiktos be brūkšnių, konvertuotų į datas su brūkšniais (pvz. 19950830 -> 1995-08-30);
- patikrintų, ar gimimo data yra valid; t.y. ar ji atitinka formatą ir turi tiek simbolių, kiek reikia (6 simboliai). Neteisingas duomenų eilutes surašytų į failą *KlaidingiDuomenys.csv*;
- pašalintų iš duomenų eilučių tas, kuriose yra neteisinga ūgio reikšmė, ir išvestų teisingus duomenis į failą *Rinktinė.csv*.

U5-38. Automobilių parkas.

Įmonė UAB „Žaibas“ perėmė bankrutavusių konkurentų verslą ir jų turimus automobilius. Duomenys apie turimus automobilių modelius prieinami tik per bankrutavusios įmonės interneto puslapį, todėl reikia konvertuoti HTML formatu pateiktus duomenis į CSV formatą. Puslapyje pateiktuose duomenyse pastebėta klaidų.

Išnagrinėjus duomenis, pateiktus tinklalapyje, paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas *Bold*, *Italic* ar *Underline* formatavimas);
- ne visos raidės valstybiniuose automobilių numeriuose yra didžiosios;
- kai kurios pagaminimo datos įvestos neteisingai. Datos formato pavyzdys – 1995-08. Kai kurios datos įvestos be brūkšnių, pvz. 199508. Taip pat kai kurios datos turi per daug arba per mažai skaičių.

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytą tinklalapio failą (pradinius duomenys galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuotų į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
- valstybiniame automobilio numeryje esančias mažąsias raides pakeistų į didžiąsias;
- pagaminimo metus ir mėnesį, kurie pateikti be brūkšnių, konvertuoti į datas su brūkšniais (pvz. 199508 → 1995-08);
- jeigu data pateikta ne tik su mėnesiu, bet ir su diena, dieną panaikintų (1995-08-15 → 1995-08).
- duomenų laukus, kuriuose pagaminimo data pateikta be mėnesio, surašyti į failą *NepilniDuomenys.csv*;
- pašalintų iš duomenų eilučių tas, kuriose nurodytas nevalidus valstybinis automobilio numeris, ir išvestų teisingus duomenis į failą *Automobiliai.csv*.

U5-39. Studentų atstovybė.

Norite papildyti atstovybės narių informaciją iš studentų atstovybės interneto puslapio, todėl reikia konvertuoti HTML formatu pateiktus duomenis į CSV formatą. Puslapyje pateiktuose duomenyse pastebėta klaidų.

Išnagrinėjus duomenis, pateiktus tinklalapyje, paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas *Bold*, *Italic* ar *Underline* formatavimas);
- kai kurių narių vardai ir pavardės prasideda mažąja raide;
- vieni telefonų numeriai prasideda skaičiais 8, kiti: +370;
- kai kurie studentų pažymėjimų numeriai nėra validūs. KTU studentų pažymėjimo numeriai prasideda raidėmis KTU, po kurių seka 8 skaitmenys (pvz. KTU11115900). Kai kurie duomenyse pateikti studentų pažymėjimo numeriai netenkina šių reikalavimų ir turi per daug ar per mažai skaičių arba prasideda ne tais simboliais.

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytą tinklalapio failą (pradinius duomenys galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuotų į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
- gimimo datas, kurios pateiktos be brūkšnių, konvertuotų į datas su brūkšniais (pvz. 19950830 -> 1995-08-30);
- patikrintų, ar studento pažymėjimo numeris atitinka formatą. Duomenų eilutes, kuriose formatas netinkamas, surašytų į failą `KlaidingiDuomenys.csv`;
- pašalintų iš duomenų eilučių tas, kuriose nurodyta nevalidi gimimo data, ir išvestų teisingus duomenis į failą `Studentai.csv`.

U5-40. Biblioteka.

Bibliotekos buvo papildytos naujomis knygomis. Užsisakius knygas gavote užsakytų knygų sąrašą, pateiktą užsakovo interneto svetainės puslapyje. Jums reikia informaciją apie knygas konvertuoti iš HTML į CSV formatą. Tiekėjo pateiktos užsakytų knygų atskaitos formatas neatitinka jūsų naudojamo duomenų saugojimo formato.

Išnagrinėjus duomenis, pateiktus tinklalapyje, paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas *Bold*, *Italic* ar *Underline* formatavimas);
- tiekėjo pateiktame knygų sąrašė ties autoriaus įrašu pirmiausiai nurodyta pavardė, o po to - vardas. Jūsų duomenyse pirmiausiai saugomas vardas, o po to pavardė;
- kai kuriose knygose nurodyta autoriaus pavardė ir vardo raidė.

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytą tinklalapio failą (pradinius duomenys galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuotų į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
- autoriaus informaciją pakeistų taip, kad pirmoje vietoje būtų vardas, o tada pavardė;
- duomenų eilutes, kuriose yra pateikta tik autoriaus vardo pirma raidė, surašytų į failą `NepilnasAutoriausVardas.csv`;
- pašalintų iš duomenų eilučių tas, kuriose nurodytas nevalidus ISBN kodas, ir išvestų teisingus duomenis į failą `Knygos.csv`.

U5-41. Proto mūšis.

Studentų atstovybė aptiko interneto puslapį, kuriame pateikta daug įdomių klausimų. Klausimų kopijavimas iš puslapio užimtų daug laiko, todėl buvo nuspręsta informaciją apie klausimus, pateiktą HTML formatu, konvertuoti į CSV formatą. Pateiktoje informacijoje apie klausimus pastebėta formato neatitikimų ir klaidų.

Išnagrinėjus duomenis, pateiktus tinklalapyje, paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas *Bold*, *Italic* ar *Underline* formatavimas);
- kai kurie puslapio klausimai turi daugiau arba mažiau nei 4 atsakymo variantus;

- kai kurie autorių vardai prasideda mažąja raide.

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytų tinklalapio failą (pradinius duomenys galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuotų į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
- patikrintų, ar klausimas turi daugiau nei 4 atsakymo variantus, ir perteklinius variantus pašalintų (prieš tai patikrinant, ar nėra šalinamas teisingo atsakymo variantas);
- patikrintų, ar klausimas turi mažiau nei 4 atsakymo variantus. Jei taip, tokius klausimus surašytų į failą `TrūkstaAtskaymoVariantų.csv`;
- vardus, prasidedančius mažąja raide, pakeistų į vardus, prasidedančius didžiąja raide;
- leistų klausimų paiešką (klausimų pavadinimuose) pagal nurodytą frazę; paieškos rezultatus saugotų CSV faile.

U5-42. Nekilnojamojo turto agentūra.

Gavus vartotojų sutikimą, buvo nuspręsta nekilnojamojo turto agentūrų duomenis papildyti duomenimis iš privačių skelbimų interneto svetainės, todėl prireikė svetainėje pateiktus duomenis konvertuoti iš HTML formato į CSV formatą. Puslapyje pateiktoje nekilnojamojo turto informacijoje pastebėta formato neatitikimų ir klaidų.

Išnagrinėjus duomenis, pateiktus tinklalapyje, paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas *Bold*, *Italic* ar *Underline* formatavimas);
- duomenys apie gatvę, buto numerį ir aukštą pateikiami kaip viena eilutė (pvz. K. Baršausko g. 88-6);
- kai kuriems butams nenurodytas buto numeris.

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytų tinklalapio failą (pradinius duomenys galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuotų į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
- adreso duomenis išskaidytų į atskirus duomenų laukus (pvz. K. Baršausko g. 88-6 – gatvė: K. Baršausko, namo numeris: 88, buto numeris: 6);
- saugant gatvės pavadinimą, išmesti iš jo užrašą „g.“ arba „gatvė“ (pvz. K. Baršausko g. → K. Baršausko);
- butų duomenų eilutes, kurioms trūksta buto numerio, surašytų į failą `TrūkstaDuomenų.csv`;
- pašalintų duomenų eilutes, kurioms trūksta duomenų ir likusias eilutes surašytų į duomenų failą `NekilnojamasisTurtas.csv`.

U5-43. WCG turnyras.

Turite ketvirto turnyro duomenis interneto puslapyje. Kiekvienas turnyro dalyvis turi pateikti savo el. pašto adresą, todėl klasėje „Žaidėjas“ atsiranda papildomas duomenų laukas *email*. Duomenis, pateiktus interneto puslapyje, reikia konvertuoti iš HTML formato į CSV formatą. Pastebėta, kad puslapyje pateiktoje informacijoje duomenys nėra visiškai patikimi.

Išnagrinėjus duomenis, pateiktus tinklalapyje, paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas *Bold*, *Italic* ar *Underline* formatavimas);
- el. pašto adresai nurodyti nekorektiškai, t.y. neatitinka formato (pvz. neturi @ simbolio, pabaigoje neturi taško ir/ar domeno vardo).

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytų tinklalapio failą (pradinius duomenys galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuoti į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
- patikrintų, ar el. pašto adresai yra validūs ir tenkina šablono reikalavimus;
- duomenų eilutes su nevalidžiais el. pašto adresais surašyti į failą `KlaidingiDuomenys.csv`;
- pašalintų duomenų eilutes su nevalidžiais el. pašto adresais ir likusius duomenis surašyti į failą `Žaidėjai.csv`.

U5-44. Turistų informacijos centras.

Turizmo informacijos centras pradės teikti informaciją apie viso Baltijos šalių regiono muziejus. Didelė dalis informacijos apie kitų šalių muziejus pateikiama internetinėse svetainėse, todėl norint perkelti duomenis iš šių svetainių, reikia konvertuoti HTML formato duomenis į CSV formatą. Muziejai, kurie teikia gido paslaugą, taip pat nurodo ir jo el. pašto adresą, todėl duomenyse atsiranda papildomas laukas *email*. Pastebėta, kad puslapiuose pateiktoje informacijoje duomenys nėra visiškai patikimi.

Išnagrinėjus duomenis, pateiktus tinklalapyje, paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas *Bold*, *Italic* ar *Underline* formatavimas);
- el. pašto adresai nurodyti nekorektiškai, t.y. neatitinka formato (pvz. neturi @ simbolio, pabaigoje neturi taško ir/ar domeno vardo).

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytų tinklalapio failą (pradinius duomenys galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuoti į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
- patikrintų, ar el. pašto adresai yra validūs ir tenkina šablono reikalavimus;
- duomenų eilutes su nevalidžiais el. pašto adresais surašyti į failą `KlaidingiDuomenys.csv`;
- pašalintų duomenų eilutes su nevalidžiais el. pašto adresais ir likusius duomenis surašyti į failą `Muziejai.csv`.

U5-45. IMDb.

Rodote įdomų filmų sąrašą, kurį pateikia kino mėgėjas savo internetinėje svetainėje. Nusprendėte papildyti filmų sąrašą filmais iš jo svetainės, todėl jums reikia konvertuoti filmų sąrašą iš HTML į CSV formatą. Pastebėjote, kad kai kurių aktorių pavardės pateiktos lietuviškai. Taip pat pateiktuose duomenyse pastebėjote ir daugiau netikslumų.

Išnagrinėjus duomenis, pateiktus tinklalapyje, paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas *Bold*, *Italic* ar *Underline* formatavimas);
- aktoriaus Tomo Kruzo vardas vienur užrašomas lietuviškai (Tomas Kruzas), kitur – angliškai (Tom Cruise);
- režisieriaus lauke pirmiau pateikiama pavardė, o po tarpelio vardas.

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytų tinklalapio failą (pradinius duomenys galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuoti į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
- suvienodintų informaciją, pateikiamą apie aktorių Nikolą Keidžą; turi likti arba tik angliški vardai ir pavardės, arba tik lietuviški;
- sukeistų duomenis režisieriaus lauke taip, kad pirmoje vietoje būtų vardas, po to pavardė;
- leistų rasti filmus pagal tam tikrą frazę pavadinime. Surastus filmus saugotų paieškos frazės vardo CSV faile.

U5-46. Buitinės technikos parduotuvė.

Buvo nuspręsta realizuoti funkciją, kuri leistų palyginti buitinių prietaisų kainas konkurentų parduotuvėse. Duomenys apie konkurentų prekes yra pateikiami jų interneto svetainėse, todėl reikia konvertuoti HTML formato duomenis į CSV formatą. Pastebėta, kad puslapiuose pateiktoje informacijoje duomenys yra šiek tiek iškraipyti.

Išnagrinėjus duomenis, pateiktus tinklalapyje, paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas *Bold*, *Italic* ar *Underline* formatavimas);
- duomenys apie šaldytuvų matmenis pateikiami skirtingais vienetais (tinklalapyje – mm ir ml, jūsų sistemoje – cm ir l);
- duomenys apie virdulių talpą pateikiami skirtingais vienetais (tinklalapyje – ml, jūsų sistemoje – l);
- kai kur energijos klasė nurodyta netinkamai. Energijos klasė žymima A+++, A++, A+, A, B, C, D, E, F arba G simboliais.

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytą tinklalapio failą (pradinius duomenys galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuotų į CSV formato eilutes;
- iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
- milimetrais pateikiamus šaldytuvų matmenis paverstų į cm ir prie atitinkamo dydžio prirašytų vienetus (pvz. 1000 – „100 cm“);
- mililitrais pateikiamų virdulių talpas paverstų į l ir prie atitinkamo dydžio prirašytų vienetus (pvz. 1500 – „1,5 l“);
- patikrintų, ar energijos klasė nurodyta tinkamai. Duomenų eilutes, kuriose nurodyta netinkama energijos klasė, surašytų į failą `NetinkamiDuomenys.csv`.
- pašalintų duomenų eilutes, kuriose nurodyta neteisinga energijos klasė, ir likusias eilutes surašytų į failą `Gaminiai.csv`.

U5-47. Juvelyrikos parduotuvė.

Reikia importuoti juvelyrinių gaminių informaciją iš internetinio tinklalapio, todėl prireikė konvertuoti HTML formato duomenis į CSV formatą. Pastebėta, kad puslapyje pateiktoje informacijoje duomenys yra šiek tiek iškraipyti.

Išnagrinėjus duomenis pateiktus tinklalapyje paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas **bold**, *italic* ar underline formatavimas);

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytą tinklalapio failą (pradinius duomenys galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šabloną) ir jame esančius duomenis konvertuotų į CSV formato eilutes;
 - iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
 - iš aukso pagamintuose juvelyriniuose gaminiuose reikia patikslinti aukso pavadinimą pagal prabą. Jei aukso praba yra mažesnė nei 500 (imtinai), tokio aukso pavadinimą reikia pakeisti į „Pigus juvelyrinis auksas“. Jei praba tarp 501 ir 900 (imtinai), aukso pavadinimas – „Juvelyrinis auksas“. Jei praba tarp 901 ir 999 (imtinai), aukso pavadinimas – „Brangus juvelyrinis“. Jeigu praba lygi 1000, tokį auksą reikia pervadinti į „Chemiškai grynas auksas“;
 - aukso gaminius, kurie pagaminti iš pigaus juvelyrinio aukso, surašytų į failą `PigūsGaminiai.csv`;
 - aukso gaminius, kurie pagaminti iš pigaus juvelyrinio aukso, pašalinti, o likusius gaminius surašyti į failą `JuvelyriniaiGaminiai.csv`.
-

U5-48. Kompiuterinis žaidimas.

Kai kurių žaidimo herojų informacija pateikiama internetinėje svetainėje. Reikia konvertuoti HTML formato duomenis į CSV formatą. Pastebėta, kad puslapyje pateiktoje informacijoje duomenys yra šiek tiek iškreipti.

Išnagrinėjus duomenis pateiktus tinklalapyje paaiškėjo, kad:

- duomenys lentelėje pateikiami nevienodai (kai kur pritaikytas bold, italic ar underline formatavimas);

Nuspręsta parašyti pagalbinę programą (konverterį), kuri konvertuotų duomenis iš HTML formato į CSV formatą, kartu ištaisydama aptinkamas klaidas.

Parašykite programą, kuri atliktų tokius veiksmus:

- nuskaitytą tinklalapio failą (pradinius duomenys galite susidaryti pagal pratybų 5.7. skyriuje pateiktą šablono) ir jame esančius duomenis konvertuoti į CSV formato eilutes;
 - iš duomenų eilučių pašalintų formatavimo žymes (, <i>, <u> ar bet kurias kitas);
 - jeigu herojaus jėgos, vikrumo ir intelekto laukų vertės yra didesnės nei 20, prie šio veikėjo klasės pavadinimo pridedama frazė „I lygio“. Jeigu visų laukų vertės didesnės nei 60 – pridedama frazė „II lygio“, jeigu visų laukų vertės didesnės nei 90 – pridedama frazė „III lygio“;
 - skirtingo lygio herojus surašytų į atskirus failus: `LygisI.csv`, `LygisII.csv` ir `LygisIII.csv`.
-