

## 2. Objektų rinkinys

Susipažinsite su:

- masyvo struktūra ir masyvo tipo kintamuoju, kuriame bus saugomas objektų rinkinys;
- CSV (comma separated values) tekstiniais failais, kuriuose saugomi objektų duomenys ir į kuriuos surašomi skaičiavimų rezultatai;
- sumos, kiekio skaičiavimo algoritmais;
- didžiausios (mažiausios) reikšmės paieškos algoritmu.
- C# dinaminio masyvu, kuriame bus saugomi objektų rinkiniai (**pažengusiems tik peržiūrai**);
- C# Lambda išraiškomis, kurios supaprastina duomenų rinkinių apdorojimo operacijas (**pažengusiems tik peržiūrai**);

Visi programiniai kodai, kurie pateikiami šiame praktiniame darbe pateikti šioje saugykloje:

<https://bitbucket.org/blaztoma/lab1/downloads>

Naudokite juos, jei nesiseka savarankiškai išspręsti uždavinių.

### 2.1. Pažintis su objektų rinkiniu

- Masyvas objektų rinkiniui saugoti.
- Duomenų skaitymas iš tekstinio failo.
- Sumos, kiekio ir aritmetinio vidurkio skaičiavimo algoritmų taikymas.
- Mažiausios reikšmės paieška.

**Užduotis.** Gyvūnų registras.

Gyvūnų duomenų bazėje registruojami šunys. Žinomas kiekvieno *šuns vardas, mikroschemos identifikatorius, svoris (kg.), amžius (metais), veislė, šeimininko pavardė, telefono numeris, paskutinio skiepavimo data ir agresyvumo požymis*.

Parašykite programą, kuri:

- Suskaičiuotų ir atspausdintų agresyvių šunų kiekį;
- Surastų visus nurodytos veislės šunis;
- Surastų nurodytos veislės seniausius šunis;
- Nustatytų šunis, kuriuos reikia skiepyti (vakcinos galiojimo trukmė - 1 metai).

**Pradiniai duomenys ir rezultatai.**

Duomenys surašyti tekstiname faile `L1Data.csv`. Kiekvienoje eilutėje užrašytas: *šuns vardas, mikroschemos identifikatorius, svoris, amžius, šeimininko pavardė, telefono numeris, paskutinio skiepavimo data ir agresyvumo požymis*. Naudojamas skyriklis – kabliataškis. Realiojo skaičiaus trupmeninė dalis atskiriama kableliu – tokie kompiuterio nustatymai.

Pradiniai duomenys
Reksas;123;20,15;7;Buldogas;Jonas Jonaitis;867424992;2015-07-24;TRUE
Bimas;456;25,20;8;Taksas;Petras Petraitis;866612845;2014-12-01;FALSE
Haris;456;15,20;10;Taksas;Rimas Rimaitis;866612395;2013-12-01;FALSE
Pifas;456;7,10;8;Taksas;Antanas Antanaitis;866612344;2013-02-01;FALSE
Rikis;456;9,8;9;Spanielis;Simas Simaitis;866612329;2014-10-01;FALSE

Rezultatai surašomi tekstiname faile `L1Results.csv`. Pradiniai duomenys spausdinami lentele bei išsaugojami CSV formatu.

**Programos kūrimo eiga.**

- Sukuriama klasė gyvūnų (kol kas - šunų) duomenims saugoti.
- Sukuriamos klasės savybės (*Properties*), skirtos duomenų paėmimui ir atnaujinimui.
- Paruošiamas duomenų failas.
- Paskelbiamas objektų rinkinys.
- Parašomas duomenų skaitymo iš failo metodas, kuris suteikia rinkinio objektams reikšmes.

- Parašomas nurodytos veislės šunų kiekio skaičiavimo metodas.
- Parašomas seniausio šuns suradimo metodas.

### 🔗Pirmas žingsnis.

- Sukurkite sprendimą (*solution*) pavadinimu **Lab1**.
- Sukurkite jame projektą pavadinimu **Lab1.Step1**.
- Sukurkite klasę, skirtą šuns duomenims saugoti. Šuns klasei sukurkite atskirą klasę (failą) Dog.cs.

```
using System;

namespace Lab1.Step1
{
    class Dog
    {
        private static int VaccinationDuration = 1;

        public string Name { get; set; }
        public int ChipId { get; set; }
        public double Weight { get; set; }
        public int Age { get; set; }
        public string Breed { get; set; }
        public string Owner { get; set; }
        public string Phone { get; set; }
        public DateTime VaccinationDate { get; set; }
        public bool Aggressive { get; set; }

        public Dog()
        {
        }

        public Dog(string name, int chipId, double weight, int age, string breed,
            string owner, string phone, DateTime vaccinationDate, bool aggressive)
        {
            Name = name;
            ChipId = chipId;
            Weight = weight;
            Age = age;
            Breed = breed;
            Owner = owner;
            Phone = phone;
            VaccinationDate = vaccinationDate;
            Aggressive = aggressive;
        }

        public bool isVaccinationExpired()
        {
            return
                VaccinationDate.AddYears(VaccinationDuration).CompareTo(DateTime.Now) <
                0;
        }
    }
}
```

- Klasė parengta, galima pradėti ją naudoti. Klasė yra šablonas, pagal kurį konstruojami objektai. C# kalba turi kelis būdus objektų sukūrimui.
- **Objekto sukūrimas**
  - **1 būdas.** Panaudojame klasės konstruktorių su šuns savybių parametrais. Šis būdas įpareigoja laikytis griežtų taisyklių, t.y. nurodyti visus reikalaujamus parametrus ir tinkama seka:

```
static void Main(string[] args)
{
```

```

Dog dog = new Dog("Bimas", 1234, 24, 5, "taksas", "Antanas
                Kavaliauskas", "+37061485555", new DateTime(2015, 7, 24),
                true);

Console.WriteLine("{0, -10} {1, 5:d} {2, 5:f} {3, 5:d} {4, -16} {5, -
                12} {6, 8:Y} {7}", dog.Name, dog.ChipId, dog.Weight, dog.Age,
                dog.Owner, dog.Phone, dog.VaccinationDate, dog.Aggressive);
Console.Read();
}

```

- **2 būdas (Pažengusiems).** Panaudojame klasės konstruktorių be parametrų, savybes nustatant apibrėžimo dalyje. Šis būdas leidžia lanksčiai nusakyti žinomas savybes, nenurodant kitų:

```

static void Main(string[] args)
{
    Dog dog2 = new Dog()
    {
        Name = "Reksas",
        ChipId = 1235,
        Weight = 20,
        Age = 8,
        Breed = "Buldogas",
        Owner = "Petras Petrauskas",
        Phone = "+37061485556",
        VaccinationDate = new DateTime(2015, 8, 24),
        Aggressive = true
    };

    Console.WriteLine("{0, -10} {1, 5:d} {2, 5:f} {3, 5:d} {4, -16} {5, -
        12} {6, 8:Y} {7}", dog2.Name, dog2.ChipId, dog2.Weight, dog2.Age,
        dog2.Owner, dog2.Phone, dog2.VaccinationDate, dog2.Aggressive);
    Console.Read();
}

```

- **3 būdas (Pažengusiems).** Sukuriame anoniminių objektą nenaudodami klasės. Šis būdas naudojamas kai reikia turėti laikiną rezultatų objektą (pvz. naudojant LINQ), tai yra kai objektas turi tokį trumpą gyvavimo laikotarpį, kad dėl jo neapsimoka kurti klasės:

```

static void Main(string[] args)
{

    var dog3 = new
    {
        Name = "Haris",
        ChipId = 1236,
        Weight = 10,
        Age = 3,
        Breed = "Taksas",
        Owner = "Simas Simaitis",
        Phone = "+37061485556",
        VaccinationDate = new DateTime(2015, 8, 4),
        Aggressive = true
    };

    Console.WriteLine("{0, -10} {1, 5:d} {2, 5:f} {3, 5:d} {4, -16} {5,
        -12} {6, 8:Y} {7}", dog3.Name, dog3.ChipId, dog3.Weight,
        dog3.Age, dog3.Owner, dog3.Phone, dog3.VaccinationDate,
        dog3.Aggressive);
    Console.Read();
}

```

- Įvykdysite programą. Ekrane turėtumėte matyti užregistruoto šuns duomenis.

**Antras žingsnis.**

- Paruoškite duomenų failą.
- Pašalinkite eksperimentų su objektų sukūrimu kodą Main() metode.
- Parašykite rinkinio (masyvo) maksimalų galimą dydį apibrėžiančią konstantą:  
`public const int MaxNumberOfDogs = 50;`
- Sukurkite skaitliuką, kuriame bus saugomas užregistruotų šunų kiekis:  
`private static int dogCount = 0;`
- Sukurkite objektų (šunų) rinkinį: `Dog[] dogs = new Dog[MaxNumberOfDogs];`
- Parašykite duomenų skaitymo iš failo metodą. Norint skaityti duomenis iš failo, programos pradžioje būtina įtraukti System.IO biblioteką:

```
using System.IO;
```

```
/// <summary>
/// Read dog data from CSV file
/// </summary>
/// <param name="dogs">Returns array of dogs</param>
/// <param name="count">Returns count of the dogs</param>
private static void ReadDogData(out Dog[] dogs, out int dogCount)
{
    dogCount = 0;
    dogs = new Dog[MaxNumberOfDogs];

    using (StreamReader reader = new StreamReader(@"L1Data.csv"))
    {
        string line = null;
        while (null != (line = reader.ReadLine()))
        {
            string[] values = line.Split(';');
            string name = values[0];
            int chipId = int.Parse(values[1]);
            double weight = Convert.ToDouble(values[2]);
            int age = int.Parse(values[3]);
            string breed = values[4];
            string owner = values[5];
            string phone = values[6];
            DateTime vaccinationDate = DateTime.Parse(values[7]);
            bool aggressive = bool.Parse(values[8]);

            Dog dog = new Dog(name, chipId, weight, age, breed, owner, phone,
vaccinationDate, aggressive);

            dogs[dogCount++] = dog;
        }
    }
}
```

- Parašykite duomenų spausdinimo faile metodą.

```
/// <summary>
/// Saves dog data to CSV type file
/// </summary>
/// <param name="dogs">Array of dogs</param>
/// <param name="count">Count of the dogs</param>
static void SaveReportToFile(Dog[] dogs, int count)
{
    using (StreamWriter writer = new StreamWriter(@"L1Results.csv"))
    {
        writer.WriteLine("Vardas;MikroId;Svoris;Amžius;Savininkas;Tel.
Nr.;Vakcinacija;Agresyvumas");
        for (int i = 0; i < count; i++)
        {
```

```

        writer.WriteLine("{0};{1};{2};{3};{4};{5};{6};{7}", dogs[i].Name,
            dogs[i].ChipId, dogs[i].Weight, dogs[i].Age, dogs[i].Owner,
            dogs[i].Phone, dogs[i].VaccinationDate, dogs[i].Aggressive);
    }
}

```

- Parašykite duomenų spausdinimo konsolėje metodą tam, kad būtų galima stebėti duomenis nenaudojant failų.

```

/// <summary>
/// Show dog data in the Console
/// </summary>
/// <param name="dogs">Array of dogs</param>
/// <param name="count">Count of the dogs</param>
static void PrintDogsToConsole(Dog[] dogs, int count)
{
    for (int i = 0; i < count; i++)
    {
        Console.WriteLine("Vardas: {0}\nMikroschemas ID: {1}\nSvoris: {2}\nŪgis: {3}\nSavininka: {4}\nTelefonas: {5}\nVakcinacijos data: {6}\nAgresyvus: {7}\n", dogs[i].Name, dogs[i].ChipId, dogs[i].Weight, dogs[i].Age, dogs[i].Owner, dogs[i].Phone, dogs[i].VaccinationDate, dogs[i].Aggressive);
    }
}

```

- Papildykite Main() metodą kreipiniais į parašytus metodus

```

using System;
using System.IO;
using System.Linq;

namespace Lab1.Step1
{
    class Program
    {
        public const int MaxNumberOfDogs = 50;

        static void Main(string[] args)
        {
            Dog[] dogs;
            int dogCount;

            ReadDogData(out dogs, out dogCount);
            SaveReportToFile(dogs, dogCount);
        }
    }
}

```

- Įvykdysite programą. Atidarykite sugeneruotą csv formato rezultatų failą su Excel programa.

### 📍 Trečias žingsnis.

- Parašykite metodą, skaičiuojantį kiek registre yra agresyvių nurodyto amžiaus šunų

```

/// <summary>
/// Count aggressive dogs of the specific age
/// </summary>
/// <param name="dogs">dog array</param>
/// <param name="age">age of the dogs to take into account</param>
/// <returns></returns>
private static int CountAggressive(Dog[] dogs, int dogCount, int age)
{
    int counter = 0;
}

```

```

    for (int i = 0; i < dogCount; i++)
    {
        if (dogs[i].Aggressive && (dogs[i].Age == age))
        {
            counter++;
        }
    }

    return counter;
}

```

- Papildykite `Main()` metodą kreipiniu į metodą `countAggressive` ir gautų rezultatų spausdinimo sakiniu:

```

Console.WriteLine("Kurio amžiaus šunis skaičiuoti?");
int age = int.Parse(Console.ReadLine());
Console.WriteLine("Agresyvių šunų kiekis: " + CountAggressive(dogs, dogCount, age));

```

- Įvykdykite programą ir patikrinkite gautus rezultatus.

#### 📌 Ketvirtas žingsnis.

- Parašykite metodą, grąžinantį visus nurodytos veislės šunis ir išveskite į konsolę jų vardus

```

/// <summary>
/// Filter dogs by its breed
/// </summary>
/// <param name="dogs">Array of dogs to filter from</param>
/// <param name="dogCount">Count of the dogs</param>
/// <param name="breed">Specific breed to look for</param>
/// <param name="filteredDogs">Returns filtered array of dogs</param>
/// <param name="filteredDogsCount">Count of the dogs in the filtered array</param>
private static void FilterByBreed(Dog[] dogs, int dogCount, string breed, out Dog[]
                                filteredDogs, out int filteredDogsCount)
{
    filteredDogs = new Dog[MaxNumberOfDogs];
    filteredDogsCount = 0;

    for (int i = 0; i < dogCount; i++)
    {
        if (dogs[i].Breed == breed)
        {
            filteredDogs[filteredDogsCount++] = dogs[i];
        }
    }
}

```

- Papildykite `Main()` metodą kreipiniu į metodą ir gautų rezultatų spausdinimo sakiniu:

```

Console.WriteLine("Kurios veislės šunis filtruoti?");
string breed = Console.ReadLine();

Dog[] filteredByBreed;
int filteredDogsCount;

FilterByBreed(dogs, dogCount, breed, out filteredByBreed, out
              filteredDogsCount);
for (int i = 0; i < filteredDogsCount; i++)
{
    Console.WriteLine(filteredByBreed[i].Name);
}
Console.WriteLine();

```

- Įvykdysite programą ir patikrinsite gautus rezultatus.

### Penktas žingsnis.

- Parašysite metodą, kuris suranda nurodytos veislės seniausio šuns amžių ir parašysite metodą, kuris suranda nurodytos veislės seniausius šunis (metodas turi suformuoti seniausių šunų masyvą):

```

/// <summary>
/// Find oldest dogs of the specific breed
/// </summary>
/// <param name="dogs">Array of dogs to filter from</param>
/// <param name="dogCount">Count of the dogs</param>
/// <param name="breed">Specific breed to look for</param>
private static int FindOldestAge(Dog[] dogs, int dogCount, string breed)
{
    Dog[] filteredDogs;
    int filteredDogsCount;
    FilterByBreed(dogs, dogCount, breed, out filteredDogs, out filteredDogsCount);
    int maxAge = 0;

    for (int i = 0; i < filteredDogsCount; i++)
    {
        if (filteredDogs[i].Age > maxAge)
        {
            maxAge = filteredDogs[i].Age;
        }
    }

    return maxAge;
}

/// <summary>
/// Find oldest dogs of the specific breed
/// </summary>
/// <param name="dogs">Array of dogs to filter from</param>
/// <param name="dogCount">Count of the dogs</param>
/// <param name="oldestDogs">Returns array of oldest dogs</param>
/// <param name="oldestDogsCount">Returns count of the oldest dogs</param>
/// <param name="age">Age of the dogs to look for</param>
private static void findOldestDogs(Dog[] dogs, int dogCount, out Dog[] oldestDogs,
    out int oldestDogsCount, int age)
{
    oldestDogs = new Dog[MaxNumberOfDogs];
    oldestDogsCount = 0;

    for (int i = 0; i < dogCount; i++)
    {
        if (dogs[i].Age == age)
        {
            oldestDogs[oldestDogsCount++] = dogs[i];
        }
    }
}

```

- Papildysite metodą Main() likusiais veiksmiais:

```

Console.WriteLine("Kurios veislės seniausius šunis surasti?");
breed = Console.ReadLine();

Dog[] oldestDogs;
int oldestDogCount;

FindOldestDogs(dogs, dogCount, out oldestDogs, out oldestDogCount,
    FindOldestAge(dogs, dogCount, breed));
PrintDogsToConsole(oldestDogs, oldestDogCount);

```

- Įvykdysite programą ir patikrinkite gautus rezultatus.

### Šeštasis žingsnis.

- Parašysite metodą, kuris suranda visas skirtingas veisles:

```

/// <summary>
/// Find all different breeds existing in dog array
/// </summary>
/// <param name="dogs">Array of dogs</param>
/// <param name="dogCount">Count of the dogs</param>
/// <param name="breeds">Returns array of the breeds</param>
/// <param name="breedCount">Returns count of the breeds</param>
private static void GetBreeds(Dog[] dogs, int dogCount, out string[] breeds, out int
                                breedCount)
{
    breeds = new string[MaxNumberOfDogs];
    breedCount = 0;

    for (int i = 0; i < dogCount; i++)
    {
        if (!breeds.Contains(dogs[i].Breed))
        {
            breeds[breedCount++] = dogs[i].Breed;
        }
    }
}

```

- Patys papildysite pagrindinę programą taip, kad šis metodas būtų teisingai iškvičiamas, o jo rezultatai spausdinami konsolėje.

### Septintasis žingsnis.

- Pagrindinėje programoje parašysite kodą, kuris į konsolę pateiktų visus šunis, kuriems baigėsi vakcinacijos galiojimo laikas:

```

Console.WriteLine("Vaccination expired:");
for (int i = 0; i < dogCount; i++)
{
    if (dogs[i].isVaccinationExpired())
    {
        Console.WriteLine(String.Format("Dog name: {0}, Owner: {1}, Phone: {2}",
            dogs[i].Name, dogs[i].Owner, dogs[i].Phone));
    }
}

```

- Išnagrinėkite Dog klasėje esantį metodą, kuriuo pasinaudojote: isVaccinationExpired()

### Programos papildymas.

- Sukurkite metodą, kuris nustatytų, kuri veislė registre yra populiariausia.
- Sukurkite metodą, kuris sudarytų masyvą šunų, vakcinuotų paskutiniais metais.

### 1 savarankiško darbo užduotis

Žinoma, kiek kuris turistinės grupės narys turi pinigų (eurai, centai). Duomenys tekstiniam faile. Kiekvienas narys bendroms grupės išlaidoms skiria ketvirtadalį turimų pinigų. Kiek pinigų bus iš viso surinkta bendroms grupės išlaidoms? Kuris grupės narys skyrė daugiausiai pinigų grupės išlaidoms? Nurodykite skirtą maksimalų kiekį ir grupės narių, skyrusių šį kiekį, vardus ir pavardes.



## 2.2. Dinaminio masyvo naudojimas veiksams su objektų rinkiniu (pažengusiems tik peržiūrai)

- **List** naudojimas 2.1 poskyrio uždavinių sprendimui.

Pirmoje dalyje naudojote statinius masyvus objektų rinkinių saugojimui bei operacijų su jais atlikimui. Pagrindinė statinių masyvų problema – iš anksto apibrėžiamas masyvo dydis. Masyvo dydžio problemą galima spręsti lanksčiau, jei naudojamas dinaminis masyvas.

Pažiūrėkite, kaip anksčiau spręstas uždavinys realizuojamas **List** dinaminio masyvu.

### 📌 Pirmas žingsnis.

- Sukurkite sprendime projektą **Lab1.Step2**.
- Nukopijuokite **Dog.cs** klasę iš **Lab1.Step1** projekto į **Lab1.Step2**.
- **Dog.cs** klasėje pakeiskite namespace reikšmę į **Lab1.Step2**.
- Nukopijuokite duomenų failą **L1Data.csv** iš **Lab1.Step1** projekto į **Lab1.Step2**.
- Nukopijuokite **Lab1.Step1** pagrindinės programos metodus į **Lab1.Step2**.

### 📌 Antras žingsnis.

- Modifikuokite metodą `ReadDogData()` taip, kad jis naudotų **List** masyvą:

```
private static List<Dog> ReadDogData()
{
    List<Dog> dogs = new List<Dog>();

    using (StreamReader reader = new StreamReader(@"L1Data.csv"))
    {
        string line = null;
        while (null != (line = reader.ReadLine()))
        {
            string[] values = line.Split(';');
            string name = values[0];
            int chipId = int.Parse(values[1]);
            double weight = Convert.ToDouble(values[2]);
            int age = int.Parse(values[3]);
            string breed = values[4];
            string owner = values[5];
            string phone = values[6];
            DateTime vaccinationDate = DateTime.Parse(values[7]);
            bool aggressive = bool.Parse(values[8]);

            Dog dog = new Dog(name, chipId, weight, age, breed, owner, phone,
                              vaccinationDate, aggressive);

            dogs.Add(dog);
        }
    }

    return dogs;
}
```

- Tą patį padarykite su išsaugojimo į failą metodu.

```
private static void SaveReportToFile(List<Dog> dogs)
{
    using (StreamWriter writer = new StreamWriter(@"L1Results.csv"))
    {
        writer.WriteLine("Vardas;MikroId;Svoris;Amžius;Savininkas;Tel.
Nr.;Vakcinacija;Agresyvumas");
        foreach (Dog dog in dogs)
        {

```

```

        writer.WriteLine("{0};{1};{2};{3};{4};{5};{6};{7}", dog.Name,
            dog.ChipId, dog.Weight, dog.Age, dog.Owner, dog.Phone,
            dog.VaccinationDate, dog.Aggressive);
    }
}

```

- Modifikuokite duomenų pateikimo konsolėje metodą.

```

static void PrintDogsToConsole(List<Dog> dogs)
{
    foreach (Dog dog in dogs)
    {
        Console.WriteLine("Vardas: {0}\nMikroschemas ID: {1}\nSvoris: {2}\nŪgis: {3}\nSavininka: {4}\nTelefonas: {5}\nVakcinacijos data: {6}\nAgresyvus: {7}\n", dog.Name, dog.ChipId, dog.Weight, dog.Age, dog.Owner, dog.Phone, dog.VaccinationDate, dog.Aggressive);
    }
}

```

- Papildykite `Main()` metodą šunų rinkinio sukūrimu ir kreipiniais į parašytus metodus. Atkreipkite dėmesį, jog tapo nebereikalingi masyvų skaitliukai.

```

using System;
using System.IO;
using System.Linq;

namespace Lab1.Step1
{
    class Program
    {
        static void Main(string[] args)
        {
            List<Dog> dogs = ReadDogData();
            dogs = ReadDogData();
            SaveReportToFile(dogs);
        }
    }
}

```

- Įvykdysite programą. Atidarykite sugeneruotą `csv` formato rezultatų failą su Excel programa.

### 🕒 Trečias žingsnis.

- Modifikuokite metodą, skaičiuojantį, kiek registre yra agresyvių nurodyto amžiaus šunų:

```

private static int countAggressive(List<Dog> dogs, int age)
{
    int counter = 0;
    foreach (Dog dog in dogs)
    {
        if ((dog.Aggressive.Equals(true)) && (dog.Age == age))
        {
            counter++;
        }
    }

    return counter;
}

```

- Papildykite `Main()` metodą kreipiniu į metodą `countAggressive` ir gautų rezultatų spausdinimo sakiniu:

```

Console.WriteLine("Kurio amžiaus šunis skaičiuoti?");
int age = int.Parse(Console.ReadLine());
Console.WriteLine("Number of aggressive dogs: " + countAggressive(dogs, age));

```

- Įvykdysite programą ir patikrinkite gautus rezultatus.

#### 🕒 Ketvirtas žingsnis.

- Modifikuokite metodą, grąžinantį visus nurodytos veislės šunis ir išveskite į konsolę jų vardus

```

private static List<Dog> filterByBreed(List<Dog> dogs, string breed)
{
    List<Dog> filtered = new List<Dog>();

    foreach (Dog dog in dogs)
    {
        if (breed.Equals(dog.Breed))
        {
            filtered.Add(dog);
        }
    }

    return filtered;
}

```

- Papildykite Main() metodą kreipiniu į metodą ir gautų rezultatų spausdinimo sakiniu:

```

Console.WriteLine("Kurios veislės šunis filtruoti?");
string breedToFilter = Console.ReadLine();
List<Dog> filteredByBreed = filterByBreed(dogs, breedToFilter);
foreach(Dog dog in filteredByBreed)
{
    Console.WriteLine(dog.Name);
}

```

- Įvykdysite programą ir patikrinkite gautus rezultatus.

#### 🕒 Penktas žingsnis.

- Modifikuokite metodą, kuris suranda nurodytos veislės seniausio šuns amžių ir modifikuokite metodą, kuris suranda nurodytos veislės seniausius šunis:

```

private static int findOldestAge(List<Dog> dogs, string breed)
{
    int maxAge = 0;

    foreach (Dog dog in dogs)
    {
        if (dog.Age > maxAge)
        {
            maxAge = dog.Age;
        }
    }

    return maxAge;
}
private static List<Dog> findOldestDogs(List<Dog> dogs, string breed, int age)
{
    List<Dog> oldestDogs = new List<Dog>();

    foreach (Dog dog in dogs)
    {
        if (dog.Age == age)

```

```

        {
            oldestDogs.Add(dog);
        }
    }

    return oldestDogs;
}

```

- Papildykite metodą `Main()` likusiais veiksmais:

```

Console.WriteLine("Kurios veislės šunis skaičiuoti?");
breedToFilter = Console.ReadLine();
List<Dog> oldestDogs = findOldestDogs(dogs, breedToFilter, findOldestAge(dogs,
breedToFilter));
PrintDogsToConsole(oldestDogs);

```

- Įvykdysite programą ir patikrinkite gautus rezultatus.

#### Šeštas žingsnis.

- Modifikuokite metodą, kuris suranda visas skirtingas veisles:

```

private static List<string> getBreeds(List<Dog> dogs)
{
    List<string> breeds = new List<string>();

    foreach(Dog dog in dogs)
    {
        if (!breeds.Contains(dog.Breed))
        {
            breeds.Add(dog.Breed);
        }
    }

    return breeds;
}

```

- Patys papildykite pagrindinę programą taip, kad šis metodas būtų teisingai iškviečiamas, o jo rezultatai atspausdinami konsolėje.

#### Septintas žingsnis.

- Pagrindinėje programoje parašykite algoritmą, kuris į konsolę pateiktų visus šunis, kuriems baigėsi vakcinacijos galiojimo laikas:

```

Console.WriteLine("Vakcinacijos galiojimas baigėsi:");
foreach (Dog dog in dogs)
{
    if (dog.isVaccinationExpired())
    {
        Console.WriteLine(String.Format("Vardas: {0}, Šeimininkas: {1},
        Telefonas: {2} ", dog.Name, dog.Owner, dog.Phone));
    }
}

```

## 2.3. Dinaminio masyvo ir Lambda išraiškų naudojimas veiksams su objektų rinkiniu (pažengusiems tik peržiūrai)

Antroje dalyje jūs patobulinsite programą taip, kad ji būtų paprastesnė ir lengviau skaitoma bei prižiūrima. Apžvelgta struktūra dažnai naudojama praktikoje. Bet pateiktus uždavinius galima spręsti dar paprastesniu būdu – panaudojant Lambda išraiškas.

Pažiūrėkite, kaip anksčiau spręstas uždavinys realizuojamas **Lambda** išraiškomis.

### 🕒 Pirmas žingsnis.

- Sukurkite sprendime projektą **Lab1.Step3**.
- Nukopijuokite **Dog.cs** klasę iš **Lab1.Step2** projekto į **Lab1.Step3**.
- **Dog.cs** klasėje pakeiskite namespace reikšmę į **Lab1.Step3**.
- Nukopijuokite duomenų failą **L1Data.csv** iš **Lab1.Step2** projekto į **Lab1.Step3**.
- Nukopijuokite **Lab1.Step2** pagrindinės programos metodus į **Lab1.Step3**.
- **Lambda** išraiškos bus naudojamos kartu su jau nagrinėta **List** masyvu, todėl modifikuosime tik tuos metodus, kurie keitėsi.

### 🕒 Antras žingsnis.

- Modifikuokite metodą, skaičiuojantį, kiek registre yra agresyvių nurodyto amžiaus šunų

```
private static int countAggressive(List<Dog> dogs, int Age)
{
    return dogs.Where(o => (o.Aggressive.Equals(true))&&(o.Age == Age)).Count();
}
```

- Įvykdysite programą ir patikrinkite gautus rezultatus.

### 🕒 Ketvirtas žingsnis.

- Modifikuokite metodą, grąžinantį visus nurodytos veislės šunis ir išveskite į konsolę jų vardus

```
private static List<Dog> filterByBreed(List<Dog> dogs, string breed)
{
    return dogs.FindAll(o => o.Breed.Equals(breed)).ToList();
}
```

- Įvykdysite programą ir patikrinkite gautus rezultatus.

### 🕒 Penktas žingsnis.

- Modifikuokite metodą, kuris suranda nurodytos veislės seniausius šunis (metodas turi suformuoti seniausių šunų masyvą):

```
private static List<Dog> findOldestDogs(List<Dog> dogs, string breed)
{
    List<Dog> filteredDogs = dogs.Where(o => o.Breed == breed).ToList();
    int maxAge = filteredDogs.Max(o => o.Age);
    return filteredDogs.Where(o => o.Age == maxAge).ToList();
}
```

- Įvykdysite programą ir patikrinkite gautus rezultatus.

### 🕒 Šeštasis žingsnis.

- Modifikuokite metodą, kuris suranda visas skirtingas veisles:

```
private static List<string> getBreeds(List<Dog> dogs)
{
    return dogs.Select(o => o.Breed).Distinct().ToList();
}
```

## 2.4. Kontroliniai klausimai

- Kiek ir kokių kintamųjų reikia, norint aprašyti duomenų masyvą arba objektų rinkinį?
- Kuo skiriasi reikšmių sumos radimo algoritmas nuo aritmetinio vidurkio skaičiavimo algoritmo ir kas juose yra bendra?
- Kokia turi būti pradinė kintamojo, skirto didžiausiai reikšmei masyve (rinkinyje) rasti, reikšmė?
- Kuo skiriasi mažiausios reikšmės radimo algoritmas nuo algoritmo, skirto rasti mažiausią reikšmę turintį objektą rinkinyje?
- Kaip surasti, kiek objektų rinkinyje yra objektų, turinčių vienodą didžiausią reikšmę?
- Duotas masyvo aprašas: `double[] A = new double[35];`  
Kuris iš masyvo elemento panaudojimo atvejų yra teisingas? Kodėl kiti neteisingi?
  - `double x = A[0]+3.5;`
  - `A[3.5] = 2.5;`
  - `A[0] = A[35]+2;`
  - `A[7] = "2.5";`
- Duotas masyvo aprašas: `double[] A = new double[35];` Kintamajame S reikia sukaupti teigiamų masyvo A elementų sumą.  
Kokia turi būti kintamojo s pradinė reikšmė?
  - 0
  - 1
  - bet koks teigiamas skaičius
  - pradinės reikšmės priskirti nereikia
- Duotas masyvo aprašas: `double[] A = new double[35];` Reikia rasti masyvo A didžiausio elemento indeksą.  
Kokia turi būti kintamojo, kuriame saugosime ieškomą indeksą, pradinė reikšmė? Kodėl netinka kiti atvejai
  - bet kurio elemento indeksas
  - bet kurio elemento reikšmė
  - bet koks teigiamas skaičius
  - bet koks neigiamas skaičius
- Sveikųjų skaičių masyve A(n) yra teigiamų, neigiamų ir nulinių reikšmių. Didžiausios reikšmės paieškos algoritmas toks:
 

```
int did = A[0];
for (int i = 0; i < n; i++)
    if (A[i] > did) did = A[i];
```

 Reikia surasti mažiausią teigiamą reikšmę.  
Kokia turi būti paieškos rezultato kintamojo did pradinė reikšmė? Kurie variantai tinkami, o kurie ne ir kodėl ne?
  - bet kuri masyvo teigiama reikšmė;
  - bet kuri masyvo reikšmė, tačiau reikia palyginimo sąlygą papildyti tikrinimu, ar (A[i] > 0)
  - gali būti A[0], tačiau reikia palyginimo sąlygą papildyti tikrinimu, ar (A[i] > 0)
  - bet kokia teigiama reikšmė.
- Ką matysime ekrane, atlikus tokius veiksmus:
 

```
int Suma(int [] A, int pr, int ga, int h)
{
    int s = 0;
    for (int i = pr; i < ga; s += A[i], i +=h);
    return s;
}

int D[8] = {5, 4, -3, 8, 0, 0, 0, 0};
int B[8] = {6, 5, -9, -4, 3, 0, 0, 0};
Console.WriteLine("{0} {1}, Suma(D, 0, 4, 2), Suma(B, 1, 5, 2));
```

  - 2 1
  - 14 1
  - 2 0
  - Programos fragmente yra klaidų.

## 2.5. Užduotys

- Krepšinio rinktinė.** Artėja Europos vyrų krepšinio čempionatas. Turite į rinktinės stovyklą pakviestų kandidatų sąrašą. Duomenų faile pateikiama informacija pakviestus krepšininkus: vardas, pavardė, gimimo data, ūgis, pozicija, klubas, požymis „pakviestas“, požymis „kapitonas“ (true, false).
  - Raskite vyriausią kandidatą, ir ekrane atspausdinkite jo vardą, pavardę bei amžių.
  - Raskite visus puolėjus, ir ekrane atspausdinkite jų vardus, pavardes bei ūgį.
  - Sudarykite į rinktinę pakviestų krepšininkų sąrašą, visus jų duomenis įrašykite į failą „Rinktinė.csv“.
  - Sudarykite sąrašą klubų, kuriuose žaidė kandidatai į rinktinę, ir įrašykite į failą „Klubai.csv“.
- Automobilių parkas.** Turite duomenis apie UAB „Žaibas“ priklausančius automobilius. Duomenų faile pateikta ši informacija: valstybinis numeris, gamintojas, modelis, pagaminimo metai ir mėnuo, techninės apžiūros galiojimo data, kuras, vidutinės kuro sąnaudos (100km).
  - Raskite, kurio gamintojo automobilių daugiausia, ekrane atspausdinkite gamintojo pavadinimą, bei automobilių kiekį.
  - Sudarykite visų „Volvo“ markės automobilių sąrašą, ekrane atspausdinkite automobilio valstybinį numerį, modelį, bei pagaminimo metus.
  - Sudarykite sąrašą automobilių, senesnių nei 10 metų, į failą „Senienos.csv“ surašykite visus jų duomenis.
  - Sudarykite automobilių, kuriems jau pasibaigęs techninės apžiūros galiojimas, arba liko mažiau nei mėnuo, sąrašą. Į failą „Apžiūra.csv“ įrašykite automobilio gamintoją, modelį, valstybinį numerį, techninės apžiūros galiojimo datą. Jei techninė apžiūra nebegalioja, atitinkamoje eilutėje įrašykite žodį „SKUBIAI“.
- Studentų atstovybė.** Kiekvienais metais studentų atstovybė atsinaujina – vieni studentai ją palieka, jų vietą užima nauji. Turite dabartinių studentų atstovybės narių sąrašą. Duomenų faile pateikta informacija apie studentus: pavardė, vardas, gimimo data, studento pažymėjimo numeris, kursas, telefono numeris, požymis „fuksas“ (šįmet į atstovybę įstojęs studentas).
  - Suskaičiuokite, kiek yra „fuksų“, rezultatą atspausdinkite ekrane.
  - Raskite, kurį mėnesį bus švenčiama daugiausiai gimtadienių. Ekrane atspausdinkite mėnesio pavadinimą, bei visų tą mėnesį gimtadienius švėsančių vardus, pavardes ir gimimo dienas.
  - Sudarykite visų pirmakursių sąrašą, į failą „Pirmakursiai.csv“ įrašykite visus jų duomenis.
  - Sudarykite visų ketvirtakursių sąrašą, į failą „Ketvirtakursiai.csv“ įrašykite visus jų duomenis.
- Biblioteka.** Turite KTU bibliotekos turimų knygų sąrašą. Duomenų faile pateikta informacija apie knygas: ISBN, pavadinimas, autorius(-iai), tipas, leidykla, išleidimo metai, puslapių skaičius.
  - Raskite storiąsias (daugiausiai psl. turinčią knygą), ekrane atspausdinkite jos pavadinimą ir autorių(-ius).
  - Raskite 5 knygas, išleistas 2014m., ekrane atspausdinkite knygų ISBN, pavadinimus bei autorius.
  - Raskite visas knygas, išleistas leidyklos „Technologija“, visus duomenis apie šiais knygas įrašykite į failą „Technologija.csv“.
  - Sudarykite visų autorių sąrašą, surašykite juos į failą „Autoriai.csv“.
- Proto mūšis.** Studentų atstovybė organizuoja žaidimą „Protų mūšis“. Turite žaidimui paruoštus klausimus. Duomenų faile pateikiama ši informacija: tema, sudėtingumas, klausimo autorius, klausimo tekstas, 4 atsakymo variantai, teisingas atsakymas, balai.
  - Raskite, kokios temos klausimai sunkiausi (lyginkite sudėtingumo vidurkius), temos pavadinimą atspausdinkite ekrane.
  - Raskite, kas sukūrė daugiausiai klausimų, autoriaus vardą bei klausimų kiekį atspausdinkite ekrane.
  - Sudarykite du klausimų rinkinius, kurių kiekvienas turėtų po 5 atsitiktiniu būdu parinktus klausimus. Į failus „Klausimai1.csv“ ir „Klausimai2.csv“ įrašykite klausimų temas, tekstus ir balų skaičius.
- Nekilnojamo turto agentūra.** Turite duomenis apie šiuo metu Kaune parduodamus namus. Duomenų faile pateikta ši informacija: mikrorajonas, gatvė, namo numeris, tipas, pastatymo metai, plotas, kambarių skaičius.

- Raskite seniausią namą, ekrane atspausdinkite jo amžių, adresą, tipą ir plotą.
  - Raskite, kurioje gatvėje daugiausiai parduodamų namų, ekrane atspausdinkite gatvės pavadinimą ir parduodamų namų kiekį.
  - Sudarykite visų mūrinių namų, kurių plotas didesnis nei 100 kv.m., sąrašą, į failą „M100.csv“ įrašykite visus duomenis apie šiuos namus.
  - Sudarykite visų blokinių namų, kurių plotas didesnis nei 150 kv.m., sąrašą, į failą „B150.csv“ įrašykite visus duomenis apie šiuos namus.
7. **WCG turnyras.** Kaune vyksta atrankinis kompiuterinio žaidimo „League of Legends“ turnyras. Žaidime dvi penkių žaidėjų komandos kovoja tarpusavyje valdydamos skirtingus čempionus, siekdamos sunaikinti priešininkų bazės gilumoje stovintį „Nexus“. Duomenų faile pateikta informacija apie pirmo rato dalyvius ir jų rezultatus: vardas, pavardė, komanda, pozicija, čempionas, sunaikinimai(K), dalyvavimai sunaikinimuose(A).
- Raskite žaidėją, pademonstravusį geriausią asmeninį rezultatą. Palyginimui naudokite vadinamąjį KA rodiklį (K+A). Ekrane atspausdinkite jo vardą, pavardę, komandos pavadinimą, poziciją bei naudotą čempioną.
  - Raskite, kuris čempionas buvo naudotas „universaliausiai“ (daugiausiai skirtingų pozicijų). Ekrane atspausdinkite čempiono pavadinimą, bei kokiose pozicijose jis buvo naudotas.
  - Sudarykite „Top“ pozicijoje žaidusių žaidėjų sąrašą, į failą „Top.csv“ įrašykite žaidėjų komandos pavadinimus, pavardes, vardus, naudoto čempiono pavadinimą.
8. **Turistų informacijos centras.** Turite turistų informacijos centro pateiktus duomenis apie Lietuvoje veikiančius muziejus. Duomenų faile pateikta ši informacija: pavadinimas, miestas, tipas, 5 darbo dienos, bilieto kaina, požymis „turi gidą“.
- Suskaičiuokite, kiek muziejų Kaune turi gidus, rezultatą atspausdinkite ekrane.
  - Raskite, kokio tipo muziejus galima aplankyti Vilniuje trečiadieniais, ir atspausdinkite muziejų tipus ekrane.
  - Sudarykite Kauno muziejų tipų sąrašą, įrašykite muziejų tipus į failą „Kaunas.csv“.
  - Sudarykite Vilniaus muziejų, kuriuos galima aplankyti nemokamai, sąrašą, į failą „Nemokami.csv“ įrašykite muziejaus tipą ir pavadinimą. Jei muziejus dirba tik šeštadieniais ir sekmadieniais, atitinkamoje eilutėje įrašykite „TIK SAVAITGALIAIS“.
9. **IMDB.** Turite iš IMDB „ištrauktą“ filmų sąrašą. Duomenų faile pateikta informacija apie filmus: filmo pavadinimas, leidimo metai, žanras, kino studija, režisierius, 2 aktoriai, pajamos.
- Raskite pelningiausią 2014m. filmą, ekrane atspausdinkite šio filmo pavadinimą, režisierių, bei kiek filmas uždirbo.
  - Raskite daugiausiai filmų pastačiusį režisierių, ekrane atspausdinkite jo pavardę.
  - Sudarykite filmų, kuriuose vaidino N. Cage, sąrašą, į failą „Cage.csv“ įrašykite filmų pavadinimus, leidimo metus bei kino studijos pavadinimus.
  - Sudarykite filmų žanrų sąrašą, įrašykite juos į failą „Žanrai.csv“.
10. **Buitinės technikos parduotuvė.** Apsigyvenote bendrabutyje, tačiau kambaryje trūksta svarbiausio daikto – šaldytuvo. Turite duomenis apie įmonėje „Viskas Namams“ parduodamus šaldytuvus. Duomenų faile pateikta ši informacija: gamintojas, modelis, talpa, energijos klasė, montavimo tipas, spalva, požymis „turi šaldiklį“, kaina, aukštis, plotis, gylis.
- Suskaičiuokite, kiek skirtingų „Siemens“ šaldytuvų modelių yra sąraše, rezultatą atspausdinkite ekrane.
  - Raskite pigiausią pastatomą šaldytuvą, kurio talpa 80 litrų ar didesnė, ekrane atspausdinkite jo gamintoją, modelį, talpą ir kainą.
  - Sudarykite laisvai pastatomų šaldytuvų, kurių talpa 80 litrų ar didesnė, sąrašą, į failą „Pastatomi80.csv“ įrašykite visus duomenis apie šiuos šaldytuvus.
  - Sudarykite visų šaldytuvų gamintojų sąrašą, gamintojų pavadinimus įrašykite į failą „Gamintojai.csv“.
11. **Juvelyrikos parduotuvė.** Turite UAB „Blizgučiai“ parduodamų žiedų sąrašą. Duomenų faile pateikta informacija apie žiedus: gamintojas, pavadinimas, metalas, svoris, dydis, praba, kaina.
- Raskite brangiausią žiedą, ekrane atspausdinkite jo pavadinimą, metalą, skersmenį, svorį ir prabą.



- Raskite, kokios prabos žiedų, parduotuvėje daugiausia, ekrane atspausdinkite prabą bei žiedų kiekį.
  - Sudarykite balto aukso žiedų, pigesnių nei 300 eurų, sąrašą, į failą „BA300.csv“ įrašykite visus duomenis apie šiuos žiedus.
  - Raskite 3 žiedus, kurių kaina tarp 300 ir 500 eurų, į failą įrašykite visus duomenis apie šiuos žiedus.
12. **Kompiuterinis žaidimas.** Kuriate „fantasy“ tipo kompiuterinį žaidimą. Duomenų faile turite informacija apie žaidimo herojus: vardas, rasė, klasė, gyvybės taškai, mana, žalos taškai, gynybos taškai, jėga, vikrumas, intelektas, ypatinga galia.
- Raskite, kuris herojus pasižymi geriausiomis charakteristikomis (jėgos, vikrumo ir intelekto suma didžiausia), ekrane atspausdinkite herojaus vardą, rasę, klasę bei charakteristikas.
  - Raskite, kokios rasės herojų daugiausia. Ekrane atspausdinkite rasės pavadinimą, bei visų tos rasės herojų vardus.
  - Sudarykite visų elfų sąrašą, faile „Elfai.csv“ įrašykite visus jų duomenis.
  - Sudarykite herojų, kurių gyvybės taškai bent 100, o gynybos taškai – bent 30, sąrašą. Į failą „Tankai.csv“ įrašykite herojų vardus, klases, rases bei ypatingas galias.
13. **Krepšinio rinktinė.** Artėja Europos vyrų krepšinio čempionatas. Turime į rinktinės stovyklą pakviestų kandidatų sąrašą. Duomenų faile pateikiama informacija pakviestus krepšininkus: vardas, pavardė, gimimo data, ūgis, pozicija, klubas, požymis „pakviestas“, požymis „kapitonas“ (true, false).
- Raskite jauniausią į rinktinę pakviestą krepšininką, ekrane atspausdinkite jo vardą, pavardę, amžių ir poziciją.
  - Raskite krepšininkus, žaidusius Kauno „Žalgiryje“, ekrane atspausdinkite jų vardus, pavardes bei pozicijas.
  - Krepšininkai mėgsta švęsti gimtadienius. Sudarykite sąrašą krepšininkų, kurie švęs gimtadienius pasirengimo krepšinio čempionatui metu (liepos 20d. – rugsėjo 3d.), į failą „Gimtadieniai.csv“ įrašykite krepšininkų vardus, pavardes bei gimimo mėnesį ir dieną.
  - Sudarykite sąrašą krepšininkų, kurių ūgis – 2 metrai ir daugiau, į failą „Aukštaūgiai.csv“ įrašykite krepšininkų vardus, pavardes ir ūgį.
14. **Automobilių parkas.** Turime duomenis apie UAB „Žaibas“ priklausančius automobilius. Duomenų faile pateikta ši informacija: valstybinis numeris, gamintojas, modelis, pagaminimo metai ir mėnuo, techninės apžiūros galiojimo data, kuras, vidutinės kuro sąnaudos (100km).
- Raskite seniausią automobilį, atspausdinkite jo gamintoją, modelį, valstybinius numerius ir amžių.
  - Sudarykite visų benzinu varomų automobilių sąrašą, ekrane atspausdinkite automobilio valstybinį numerį, gamintoją, modelį, bei pagaminimo metus.
  - Sudarykite sąrašą automobilių, kurių kuro sąnaudos mažesnės nei 6l/100km, į failą „Ekonomiški.csv“ surašykite visus jų duomenis.
  - Sudarykite sąrašą automobilių, kurių techninės apžiūros galiojimas baigsis per artimiausius 3 mėnesius. Į failą „Apžiūra.csv“ įrašykite šių automobilių gamintojus, modelius, valstybinius numerius, techninės apžiūros galiojimo pabaigą.
15. **Studentų atstovybė.** Kiekvienais metais studentų atstovybė atsinaujina – vieni studentai ją palieka, jų vietą užima nauji. Dabartinių studentų atstovybės narių sąrašą. Duomenų faile pateikta informacija apie studentus: pavardė, vardas, gimimo data, studento pažymėjimo numeris, kursas, telefono numeris, požymis „fuksas“ (šįmet į atstovybę įstojęs studentas).
- Raskite vyriausią studentų atstovybės narį, ekrane atspausdinkite jo vardą, pavardę ir amžių.
  - Raskite, kurio kurso studentų atstovybėje daugiausia, ekrane atspausdinkite kurso numerį, bei to kurso studentų kiekį.
  - Sudarykite visų antro ir aukštesnių kursų studentų sąrašą, į failą „Senbuviai.csv“ įrašykite visus jų duomenis.
  - Sudarykite visų „fuksų“ sąrašą, į failą „Fuksai.csv“ įrašykite visus jų duomenis.
16. **Biblioteka.** Turite KTU bibliotekos turimų knygų sąrašą. Duomenų faile pateikta informacija apie knygas: ISBN, pavadinimas, autorius(-iai), tipas, leidykla, išleidimo metai, puslapių skaičius.
- Raskite autorių, kurio knygų bibliotekoje daugiausia, ekrane atspausdinkite jo pavardę.
  - Suskaičiuokite, kiek yra knygų, senesnių nei 2 metų, rezultatą atspausdinkite ekrane.

- Sudarykite visų leidyklų, kurių knygos yra bibliotekoje, sąrašą. Į failą „Leidyklos.csv“ surašykite leidyklų pavadinimus.
  - Sudarykite visų knygų, kurių puslapių kaičius didesnis nei 400, sąrašą, surašykite šių knygų pavadinimus ir puslapių skaičių į failą „StorosKnygos.csv“.
17. **Proto mūšis.** Studentų atstovybė organizuoja žaidimą „Protų mūšis“. Turite žaidimui paruoštus klausimus. Duomenų faile pateikiama ši informacija: tema, sudėtingumas, klausimo autorius, klausimo tekstas, 4 atsakymo variantai, teisingas atsakymas, balai.
- Raskite, kiek yra I, II ir III sudėtingumo lygio klausimų, rezultatus atspausdinkite ekrane.
  - Sudarykite visų klausimų temų sąrašą, surašykite temas į failą „Temos.csv“.
  - Sudarykite 2 klausimų rinkinius, kurių kiekvienas turėtų po 4 skirtingų temų klausimus. Į failus „Klausimai.csv“ įrašykite klausimų temas, tekstus ir balų skaičių.
18. **Nekilnojamo turto agentūra.** Turite duomenis apie šiuo metu Kaune parduodamus namus. Duomenų faile pateikta ši informacija: mikrorajonas, gatvė, namo numeris, tipas, pastatymo metai, plotas, kambarių skaičius.
- Raskite ar „Saulėtekio“ mikrorajone parduodamas nors vienas namas?
  - Raskite, kokio tipo namų daugiausia, ekrane atspausdinkite namo tipą, ir to tipo namų skaičių.
  - Sudarykite visų mikrorajonų, kuriuose šiuo metu pardavinėjami namai, sąrašą, mikrorajonų pavadinimus surašykite faile „Mikrorajonai.csv“.
  - Sudarykite visų medinių namų sąrašą, į rezultatų failą „Mediniai.csv“ surašykite namo adresą, plotą ir kambarių skaičių.
19. **WCG turnyras.** Kaune vyksta atrankinis kompiuterinio žaidimo „League of Legends“ turnyras. Žaidime dvi penkių žaidėjų komandos kovoja tarpusavyje valdydamos skirtingus čempionus, siekdamos sunaikinti priešininkų bazės gilumoje stovintį „Nexus“. Duomenų faile pateikta informacija apie pirmo rato dalyvius ir jų rezultatus: vardas, pavardė, komanda, pozicija, čempionas, sunaikinimai(K), dalyvavimai sunaikinimuose(A).
- Raskite aktyviausią „Jungle“ pozicijoje žaidusį žaidėją. Žaidėjo aktyvumui palyginti naudokite nužudymų ir dalyvavimų nužudymuose sumą (K+A).
  - Raskite geriausiai bendradarbiavusią komandą (didžiausia bendra dalyvavimų nužudymuose suma), komandos pavadinimą atspausdinkite ekrane.
  - Sudarykite panaudotų čempionų sąrašą, čempionų pavadinimus surašykite į failą „Čempionai.csv“.
20. **Turistų informacijos centras.** Turite turistų informacijos centro pateiktus duomenis apie Lietuvoje veikiančius muziejus. Duomenų faile pateikta ši informacija: pavadinimas, miestas, tipas, 5 darbo dienos, bilietai kaina, požymis „turi gidą“.
- Raskite, ar yra muziejų, kurie yra nemokami ir turi gidus, ekrane atspausdinkite „Taip“ arba „Ne“.
  - Raskite, kuriame mieste galima nemokamai aplankyti daugiausiai muziejų penktadienį, ekrane atspausdinkite miesto pavadinimą ir tokių muziejų kiekį.
  - Sudarykite muziejų, nesančių didžiuosiuose Lietuvos miestuose (Vilniuje, Kaune, Klaipėdoje) sąrašą, į failą „Atrinkti.csv“ surašykite visus šių muziejų duomenis.
  - Sudarykite visų dailės muziejų sąrašą, į failą „Dailė.csv“ įrašykite muziejų pavadinimus, miestus, darbo dienas, ir įėjimo kainą.

21. **IMDB.** Turite iš IMDB „ištrauktą“ filmų sąrašą. Duomenų faile pateikta informacija apie filmus: filmo pavadinimas, leidimo metai, žanras, kino studija, režisierius, 2 aktoriai, pajamos.
- Raskite filmus, kuriuose vaidino T. Cruise ir N. Kidman, ekrane atspausdinkite filmo pavadinimus ir leidimo metus.
  - Raskite, kiek sąraše esančių filmų sukurti kino studijos „Warner Bros“, rezultata atspausdinkite ekrane.
  - Sudarykite visų aktorių sąrašą, aktorių pavardes surašykite į failą „Aktoriai.csv“.
  - Raskite 5 filmus, uždirbusius daugiau nei 500mln., į failą „500.csv“ atspausdinkite visus duomenis apie šiuos filmus.
22. **Buitinės technikos parduotuvė.** Apsigyvenote bendrabutyje, tačiau kambaryje trūksta svarbiausio daikto – šaldytuvo. Turite duomenis apie įmonėje „Viskas Namams“ parduodamus šaldytuvus. Duomenų faile pateikta ši informacija: gamintojas, modelis, talpa, energijos klasė, montavimo tipas, spalva, požymis „turi šaldiklį“, kaina.
- Raskite, kokių skirtingų talpų šaldytuvus galima įsigyti, šaldytuvų talpų sąrašą atspausdinkite ekrane.
  - Raskite pigiausią pastatomą šaldytuvą, turintį šaldiklį, ekrane atspausdinkite jo gamintoją, modelį, talpą ir kainą.
  - Sudarykite šaldytuvų, kurių plotis nuo 52 iki 56 cm, sąrašą, į failą „Tilps.csv“ įrašykite visus duomenis apie šiuos šaldytuvus.
  - Sudarykite „Bosch“ gaminamų šaldytuvų sąrašą, į failą „Bosch.csv“ įrašykite visą informaciją apie šiuos šaldytuvus.
23. **Juvelyrikos parduotuvė.** Turite UAB „Blizgučiai“ parduodamų žiedų sąrašą. Duomenų faile pateikta informacija apie žiedus: gamintojas, pavadinimas, metalas, svoris, dydis, praba, kaina.
- Raskite sunkiausią žiedą, ekrane atspausdinkite jo pavadinimą, metalą, skersmenį, svorį ir prabą.
  - Raskite, kiek žiedų yra aukščiausios prabos, rezultata atspausdinkite ekrane.
  - Sudarykite visų metalų, iš kurių pagaminti žiedai, sąrašą. Metalų pavadinimus surašykite į failą „Metalai.csv“.
  - Sudarykite 12-13 dydžio žiedų, kurių kaina iki 300 eurų, sąrašą. Į failą „Žiedai.csv“ įrašykite žiedų dydžius, metalus iš kurio jie pagaminti, prabas, svorius ir kainas.
24. **Kompiuterinis žaidimas.** Kuriate „fantasy“ tipo kompiuterinį žaidimą. Duomenų faile turite informacija apie žaidimo herojus: vardas, rasė, klasė, gyvybės taškai, mana, žalos taškai, gynybos taškai, jėga, vikrumas, intelektas, ypatinga galia.
- Raskite daugiausiai gyvybės taškų turintį herojų, ekrane atspausdinkite jo vardą, rasę, klasę ir gyvybės taškų kiekį.
  - Raskite žaidėją, kurio gynybos ir žalos taškų skirtumas yra mažiausias. Atspausdinkite informaciją apie žaidėją į ekraną.
  - Sudarykite visų herojų klasių sąrašą, klasių pavadinimus įrašykite į failą „Klasės.csv“.
  - Herojai turi būti kuo skirtingesni. Raskite herojus, turinčius identiškas charakteristikas (jėga, vikrumas, intelektas). Į failą „Pasikartojantys.csv“ surašykite visus duomenis apie juos.

## 25. Parduotuvė.

Parduotuvėje visos prekės turi pavadinimą ir kainą. Siekdama padidinti pardavimus, parduotuvė siūlo pirkti kelias to paties tipo prekes už mažesnę kainą. Pavyzdžiui, viena šokolado plytelė kainuoja 2.99. Perkant 2 to paties tipo plyteles kaina 2,49 už plytelę, o perkant 4 plyteles – 1,99.

Padėkite pirkėjui sudaryti savo pirkinį krepšelį taip, kad jis sutaupyti kuo daugiau pinigų, tačiau neviršytų maksimalios apsipirkimui skiriamos sumos.

Pradiniai duomenys duoti dvejuose tekstiniuose failuose:

- Faile `Prekės.csv` surašytos parduodamų prekių kainos: prekės pavadinimas, siūlomų prekių vienetų skaičius, prekės vieneto kaina, kaina už 2 prekes, kaina už 4 prekes.
- Faile `Pirkėjas.csv` surašytos pirkėjo norimos įsigyti prekės: prekės pavadinimas, pageidaujamas kiekis (jei prekei nuolaidos nėra), didžiausias pageidaujamas kiekis (jei yra nuolaida).

Į krepšelį prekės dedamos faile `Pirkėjas.csv` nurodyta tvarka. Pirkėjo turimų pinigų suma įvedama klaviatūra.

Rezultatus spausdinkite rezultatų faile `Krepšelis.txt` ir parodykite ekrane tokiu pavidalu: prekės pavadinimas, perkamų vienetų skaičius, tarpinė suma. Pabaigoje spausdinkite bendrą sumą ir pirkėjo sutaupytų pinigų sumą.

## 26. Birža.

Akcijų biržoje kiekviena akcija turi savo pavadinimą ir kainą. Birža pradeda darbą 10 val. ir baigia 14 val. Tarkime, kad makleris darbo dienos pradžioje turi jam skirtą pinigų sumą ir gali prekiauti tik vieno pavadinimo akcijomis. Siekdamas uždirbti daugiau pinigų, makleris perka akcijas, jei jų kaina padidėja ir parduoda, jei kaina sumažėja lyginant su ankstesne kaina.

Raskite makleriui likusių pinigų sumą darbo dienos pabaigoje ir jo atliktų pirkimo–pardavimo operacijų skaičių.

- Maklerio pradiniai duomenys – akcijos pavadinimas, turimų pinigų suma ir vienu metu perkamų akcijų skaičius – įvedami klaviatūra.
- Akcijų kainos surašytos faile `Kursai.csv` – laikas (minutėmis nuo darbo dienos pradžios), pavadinimas, siūlomų pirkti arba parduoti akcijų skaičius, kaina. Jei akcijos parduodamos, jų skaičius pateikiamas su minuso ženklu.

Rezultatus spausdinkite faile `Operacijos.txt` – maklerio atliktos operacijos laikas, operacijos tipas („pirko“ arba „pardavė“), akcijos kaina, makleriui likusių akcijų skaičius ir likusių pinigų suma. Pabaigoje spausdinkite akcijos pavadinimą, atliktų operacijų skaičių ir maklerio per dieną uždirbtų/prarastų pinigų sumą.

## 27. Tvarkaraštis.

Mokslo metų pradžioje mokykloje sudaromas pamokų tvarkaraštis. Kūno kultūros pamokos į tvarkaraštį įtraukiamos vėliausiai, nes jos vyksta dviem klasėms vienu metu. Įtraukti kūno kultūros pamoką į tvarkaraštį nelengva, nes du kūno kultūros mokytojai tuo metu turi būti laisvi.

Raskite, kada dviem klasėms gali vykti dvi kūno kultūros pamokos.

Pradiniai duomenys pateikiami trijuose tekstiniuose failuose:

- Faile `Klasės.csv` surašyti dviejų klasių tvarkaraščiai: klasė, diena, kelinta pamoka, dalykas. Jei kai kurios pamokos vykti negali, tuomet dalykas žymimas  $\times$  simboliu.
- Faile `Mokytojai.csv` surašytas kūno kultūros mokytojų užimtumas: diena, kelinta pamoka, užimtumas (jei mokytojas turi pamoką, rašomas 1, jei ne – 0, jei mokytojas negali dirbti nurodytu laiku –  $\times$ ).
- Faile `Pamokos.csv` nurodytas tinkamiausias kūno kultūros pamokoms laikas: diena, kelinta pamoka.

Rezultatus spausdinkite rezultatų faile `Tvarkaraštis.txt` ir parodykite ekrane tokiu pavidalu: klasė, diena, kelinta pamoka, dalykas. Pabaigoje spausdinkite kūno kultūros mokytojų tvarkaraštį.

## 28. Biblioteka.

Studentas rašo referatą. Bibliotekoje jis ieško reikalingos knygos. Biblioteka turi ne vieną ieškomos knygos egzempliorių, tačiau visos knygos šiuo metu yra išduotos skaitytojams. Studentui reikalingos knygos laukia ir daugiau žmonių, kurie knygą užsakė anksčiau ir turi pirmenybę. Biblioteka pasiūlė studentui užsisakyti knygą iš kitų bibliotekų, tačiau jam reikės mokėti už knygos pristatymą.

Raskite, kaip studentas galėtų greičiausiai ir pigiausiai gauti reikalingą knygą.

Pradiniai duomenys pateikiami trijuose tekstiniuose failuose:

- Faile `Skaitytojai.csv` surašyta informacija apie bibliotekos skaitytojus išduotas knygas: skaitytojo pavardė, knygos išdavimo data, kiek dienų galima skaityti knygą.
- Faile `Pageidavimai.csv` surašyta informacija apie bibliotekos skaitytojus, laukiančius knygos: skaitytojo pavardė, knygos užsakymo data, kelioms dienoms skaitytojas pageidauja knygos.
- Faile `Bibliotekos.csv` surašyti kitų bibliotekų pasiūlymai: bibliotekos pavadinimas, per kelias dienas pristatoma užsakyta knyga ir knygos pristatymo mokestis.

Studento pradiniai duomenys – apsilankymo bibliotekoje data, laiko intervalas, per kurį jis turi gauti reikalingą knygą, ir mokestis, kurį studentas gali mokėti už knygos pristatymą – įvedami klaviatūra.

Rezultatus spausdinkite faile `Knygos.txt` – studentui tinkami knygos gavimo variantai: (skaitytojas arba biblioteka, iš kur studentas galėtų gauti knygą, gavimo data, pristatymo kaina). Pabaigoje spausdinkite studentui tinkamiausius knygos gavimo variantus: per trumpiausią laiką ir už mažiausią kainą.

## 29. Tarpininkas.

Elektroninėje parduotuvėje pirkėjai ir pardavėjai gali skelbti savo pageidaujamus pirkti arba parduodamas prekes ir jų kainas. Prekybos tarpininkas peržiūri siūlymus ir sudaro užsakymus. Užsakymas susieja vieno pirkėjo pasiūlymą su vienu ar daugiau pardavėjų pasiūlymais. Užsakymą galima sudaryti tik jeigu galima rasti visą pirkėjo prašomų prekių kiekį, o jų pardavimo kainų ir tarpininko atlyginimo suma yra ne didesnė nei pirkėjo siūloma kaina. Pirkimo ir pardavimo kainų skirtumas sudaro tarpininko atlyginimą, kuris negali būti mažesnis nei nurodytas (įvedamas klaviatūra). Sudarydamas užsakymą, tarpininkas kiekvienam pirkimo pasiūlymui iš eilės peržiūri pardavimo pasiūlymus ir tinkamus pasiūlymus įtraukia į užsakymą tol, kol užsakymas yra sudaromas. Parašykite programą, kuri padėtų tarpininkui sudaryti užsakymą.

Pradiniai duomenys pateikiami trijuose failuose:

- Faile `Perku.csv` surašyti pirkėjų pasiūlymai: pirkėjo kodas, perkamos prekės pavadinimas, kiekis, siūloma kaina už vienetą.
- Faile `Parduodu.csv` surašyti pardavėjų pasiūlymai: pardavėjo kodas, parduodamos prekės pavadinimas, kiekis, prašoma kaina už vienetą.
- Faile `Klientai.csv` surašyti pirkėjų ir pardavėjų duomenys: kodas, pavardė ir vardas, adresas, telefonas.

Sudarytus užsakymus spausdinkite rezultatų faile `Užsakymai.txt` lentelę ir parodykite ekrane kiekvienam užsakymui tokiu pavidalu: pirkėjo duomenys, perkamos prekės pavadinimas, kiekis, užsakymo suma, tarpininko atlyginimas, toliau kiekvienam užsakymą vykdančiam pardavėjui atskirose eilutėse – pardavėjo duomenys, parduodamas kiekis, tarpinė suma. Pabaigoje spausdinkite visų tarpininko uždirbtų pinigų sumą.

## 30. Dėžės.

Sandėlyje prekių pakavimui paruoštos dėžės. Į vieną dėžę galima dėti ne daugiau prekių (kg) nei leidžiama dėžės apkrova. Pakuodamas sandėlininkas ima pirmą iš eilės prekę ir deda į pirmą iš eilės dėžę, jei neviršijami dėžės apkrovos reikalavimai. Parašykite programą, kuri padėtų sandėlininkui sudėti prekes į dėžes.

Pradiniai duomenys pateikiami dvejuose failuose:

- Faile `Dėžės.csv` surašytos turimos dėžės: dėžės numeris, didžiausia leistina dėžės apkrova (kg).
- Faile `Prekės.csv` surašytos turimos prekės: prekės pavadinimas, svoris (kg), vienetų skaičius.

Pakavimo nurodymus spausdinkite rezultatų faile `Pakavimas.txt` lentelę ir parodykite ekrane tokiu pavidalu: dėžės numeris, pakrautų prekių skaičius, bendras svoris. Pabaigoje spausdinkite bendrą dėžių skaičių, supakuotų prekių skaičių ir pakrovimui neišnaudotą svorį.

## 31. Gyvūnai.

Zoologijos sode gauta maisto gyvūnams siunta. Parašykite programą, kuri padėtų prižiūrėtojų išnešioti maistą gyvūnams. Prižiūrėtojas ima iš eilės maisto produktus ir pagal gyvūnų sąrašą išnešioja gyvūnams. Narve gali būti vienas arba daugiau to paties tipo gyvūnų. Raskite, kiek gyvūnų dienos pabaigoje buvo sotūs ir kiek liko alkani (gyvūnas lieka alkanas, jei negauna viso dienai reikalingo maisto davinio).

Pradiniai duomenys pateikiami dvejuose failuose:

- Faile `Maistas.csv` surašyti gauti maisto produktai: produkto pavadinimas, kiekis (kg), galiojimo data.
- Faile `Gyvūnai.csv` surašyti gyvūnai ir jų maisto poreikis vienai dienai: narvo numeris, gyvūno pavadinimas, gyvūnų skaičius, produkto pavadinimas, reikiamas maisto kiekis vienam gyvūnui.

Rezultatus spausdinkite faile `Porcijos.txt`: narvo numeris, pavadinimas, reikalingas produkto kiekis, likusių alkanų gyvūnų skaičius. Pabaigoje spausdinkite bendrą likusių alkanų gyvūnų skaičių ir išdalinto maisto kiekį (kg).

### 32. Sultys.

Sulčių gamykloje reikia išpilstyti pagamintas vaisių sultis į įvairios talpos butelius. Dalis butelių yra sandėlyje, o dalį reikia užsakyti. Sultys pilstomos ta tvarka, kuria yra pagamintos ir išpilstomos į butelius ta tvarka, kuria jie yra surašyti pradinių duomenų faile. Padėkite vadybininkui sudaryti gamyklai reikalingų butelių užsakymą.

Pradiniai duomenys pateikiami dvejuose failuose:

- Faile `Sultys.csv` surašytos gaminamos sultys: pavadinimas, tūris (l), tūrio vieneto savikaina.
- Faile `Sandėlis.csv` surašyti turimi buteliai: butelio kodas, talpa (l), vienetų skaičius, butelio kaina.

Rezultatus spausdinkite faile `Užsakymas.txt`: butelio kodas, reikalingų vienetų skaičius. Pabaigoje spausdinkite viso užsakymo kainą ir gamyklos nuostolius (patirtus dėl pagamintų, bet neišpilstytų sulčių), jei užsakymas nebūtų įvykdytas.

### 33. Protų mūšis.

Studentai rengiasi protų mūšio turnyrui. Ekspertų komisija parengė įvairių sričių (matematika, fizika, informatika, muzika, dailė, literatūra, sportas) protų mūšio klausimus, kurie komandoms bus pateikiami ta tvarka, kuria surašyti pradinių duomenų faile. Atlikus klausimų analizę pasirodė, kad kai kurių sričių klausimų trūksta. Parašykite programą, kuri nustatytų, kiek ir kokių sričių klausimų turi parengti ekspertų komisija protų mūšio turnyrui.

Pradiniai duomenys pateikiami trijuose failuose:

- Faile `Komandos.csv` surašyti duomenys apie komandas: pavadinimas, kokios srities klausimas turi būti pateiktas komandai, klausimo sunkumas (S – sunkus, VS – vidutinio sunkumo, L – lengvas).
- Faile `Žaidimas.csv` surašyti duomenys apie klausimus, kurie turės būti pateikti komandoms: kokios srities klausimas, klausimo sunkumas, koks klausimų skaičius turės būti pateiktas vienai žaidime dalyvaujančiai komandai.
- Faile `Klausimai.csv` surašyti parengti klausimai: kokios srities klausimas, klausimo sunkumas, parengtų klausimų skaičius.

Rezultatus spausdinkite faile `Papildomi.txt`: klausimo sritis, klausimo sunkumas, kiek papildomų klausimų reikia parengti. Pabaigoje spausdinkite visų turimų parengti klausimų skaičių ir skaičių pagal sunkumą.

### 34. Šachmatai.

Universitete rengiamas greitųjų šachmatų turnyras. Šachmatų klubas turi šachmatų komplektus, kuriuose sudaro dviejų spalvų (juodos ir baltos) figūros: kiekvienos spalvos po 8 pėstininkus (P), 2 žirgus (Ž), 2 rikius (R), 2 bokštus (B), 1 valdovę (V) ir 1 karalių (K). Kai kuriuose komplektuose trūksta figūrų. Šachmatų klubo narių buvo paprašyta atnešti iš namų turimus nenaudojamus šachmatų komplektus (juose gali trūkti figūrų), kad būtų galima sėkmingai surengti turnyrą. Parenkite programą, nustatančią, ar turnyras įvyks.

Pradiniai duomenys pateikiami dvejuose failuose:

- Faile `Komplektai.csv` surašyti duomenys apie turimus komplektus: komplekto eilės numeris, figūrų pavadinimai, kiekiai ir spalva.
- Faile `Parama.csv` surašyti duomenys apie klubo narių atneštus komplektus: rėmėjo vardas, atneštų figūrų pavadinimai, kiekiai ir spalva.

Turnyre dalyvaujančių žmonių skaičius įvedamas klaviatūra.

Rezultatus spausdinkite faile `Turnyras.txt`: pilno komplekto numeris, kokiomis figūromis komplektas buvo papildytas (pavadinimai, kiekiai ir spalva). Pabaigoje spausdinkite, kokių ir kiek atneštų figūrų buvo panaudota.

### 35. Laimingos valandos.

Parduotuvėje, norint padidinti pardavimų apimtį, skelbiamos laimingos valandos. Šeima stengiasi produktus pirkti laimingomis valandomis, nes tuomet taikomos iki 30 procentų nuolaidos. Parašykite programą, kuri suformuotų parduotuvės lankymo grafiką, kad šeima sutaupytų kuo daugiau pinigų.

Pradiniai duomenys pateikiami dviejuose failuose:

- Faile `Parduotuve.csv` surašyti duomenys apie laimingas valandas: valanda, prekės pavadinimas, prekių kiekis, prekės kaina ir nuolaida.
- Faile `Krepšelis.csv` surašyti duomenys apie šeimos prekių krepšelį: prekės pavadinimas, kiek prekių reikia, galima už prekes mokėti suma.

Rezultatus spausdinkite faile `Valandos.txt`: valanda, prekės pavadinimas, įsigytų prekių kiekis, už prekes sumokėtų pinigų suma. Pabaigoje spausdinkite, kiek pinigų sutaupė šeima įsigydama prekes laimingomis valandomis.

### 36. Planai.

Mobiliųjų telefonų operatorius, siekdamas išlaikyti lojalius klientus, pasiūlė naujų pokalbių planų. Šeima nori išsiaiškinti, kuriuo planu naudotis labiausiai apsimoka visiems šeimos nariams. Reikia palyginti operatoriaus siūlomus planus su dabartinėmis šeimos išlaidomis už telekomunikacijų paslaugas.

Pradiniai duomenys pateikiami dviejuose failuose:

- Faile `Skambučiai.csv` surašyti duomenys apie šeimos narių pokalbius: vardas, į kokį tinklą skambinta (jei pokalbis vyko tame pačiame tinkle – 1, jei kitame – 0), kiek laiko kalbėta, kiek sumokėta už paslaugas.
- Faile `Planai.csv` surašyti duomenys apie operatoriaus siūlomus planus: plano pavadinimas, fiksuotas mėnesinis mokestis, nemokamų minučių skaičius operatoriaus tinkle, minutės kaina tame pačiame tinkle, nemokamų minučių skaičius kituose tinkluose, minutės kaina kituose tinkluose.

Rezultatus spausdinkite faile `Tinkamiausi.txt`: plano pavadinimas, kiek nemokamų minučių šeimos nariai išnaudotų savame tinkle ir kituose tinkluose, kokią pinigų sumą mokėtų kiekvienas šeimos narys už paslaugas. Pabaigoje spausdinkite, kiek pinigų sutaupytų šeima, pasirinkusi tinkamiausią planą.

### 37. Liftas.

Pastate yra liftas, kurio keliamoji galia kilogramais yra žinoma. Žinomas pastato aukštų skaičius. Darbuotojai renkasi į darbą ir visi, išskyrus 1-3 aukštų darbuotojus, naudojami liftu. Padėkite vadybininkui sudaryti darbuotojų kėlimosi liftu eilę, siekiant, kad visi darbuotojai atsidurtų savo aukšte kaip galint per trumpesnę laiko tarpą. Visi darbuotojai prie lifto susirinko vienu metu.

Pradiniai duomenys pateikiami dviejuose failuose:

- Faile `Liftas.csv` yra užrašyta lifto keliamoji galia kilogramais ir kiek daugiausiai gali žmonių tilpti lifte. Taip pat yra užrašyta, kiek laiko vienetų liftas pasiekia kiekvieną aukštą, skaičiuojant nuo apačios. Lifto stovėjimo laikas (kiek trunka sustojimas ir darbuotojų išlipimas) tarpiniuose aukštuose yra taip pat duotas. Liftas vienodai greitai juda tiek į viršų, tiek į apačią.
- Faile `Darbuotojas.csv` surašytos darbuotojų pavardės ir vardai, svoriai kilogramais ir aukštas, kuriame darbuotojas dirba.

Rezultatus spausdinkite faile `Užsakymas.txt`: darbuotojo pavardė ir vardas, svoris, aukštas, kuriame jis dirba, ir per kiek laiko jis pateks į savo aukštą. Pabaigoje spausdinkite kiek laiko vienetų užtruko visų darbuotojų pristatymas į darbo aukštus ir kiek kartų liftas kėlė darbuotojus.

### 38. E-paštas.

Registruojama elektroninio pašto perdavimo į kitus serverius informacija. Žinomas sąsajos su serveriais internetu greitis. Reikia sudaryti sąrašą serverių, į kuriuos nebuvo perduota nei vieno brito, kurios nors valandos bėgyje. Turėdami perdavimo pradžios laiką, galite suskaičiuoti, kada perdavimas buvo baigtas. Jeigu perdavimas baigtas kitoje valandoje, tai joje taip pat buvo perduota informacija.

Pradiniai duomenys pateikiami dvejuose failuose:

- Faile `Laiškai.csv` yra vienos paros persiunčiamų laiškų sąrašas. Kiekvienoje eilutėje yra užrašytas laiško išsiuntimo laikas, siuntėjas ir gavėjas (pašto adresai), serverio, į kurį perduodamas laiškas, pavadinimas, laiško dydis baitais.
- Faile `Serveris.csv` duomenys apie serverius. Kiekvienoje eilutėje yra užrašytas serverio vardas ir greitis (baitais / s).

Rezultatus spausdinkite faile `Gauti.txt`: serveris, gautų laiškų skaičius, laiškų bendras dydis baitais, kiek laiko truko laiškų perdavimas, kiek laiko serveris negavo nei vieno laiško. Laiškai išsiunčiami vos tik jie yra gaunami. Sąraše neturi būti tų serverių, kuriems nebuvo persiųstas nei vienas laiškas.

### 39. Šeimos gydytojas.

Šeimos gydytojas priima ligonius: išklauso nusiskundimus, užpildo medicininius dokumentus, išrašo receptus. Vienu metu gydytojas bendrauja tik su vienu pacientu. Tačiau tuo pačiu metu kabinete gali būti dar du tos pačios lyties pacientai: vienas nusirengia, o kitas po apžiūros apsirengia. Visi pacientai iš anksto registravosi. Padėkite vadybininkui sudaryti pacientų eilę, siekiant, kad visi pacientai būtų aptarnauti kaip galint per trumpesnę laiko tarpą.

Pradiniai duomenys pateikiami dvejuose failuose:

- Faile `Gydytojas.csv` yra užrašyta, kiek laiko reikia dokumentų užpildymui ir kiek vieno recepto išrašymui. Taip pat yra užrašyta, kiek laiko vienetų trunka paciento apžiūra pagal ligos pobūdį (apžiūros trukmė priklauso nuo ligos tipo) ir keli receptai bus išrašomi.
- Faile `Pacientas.csv` užrašyta kiekvieno paciento pavardė ir vardas, lytis, liga, kiek laiko užtruks nusirengimas ir apsirengimas.

Rezultatus spausdinkite faile `Eilė.txt`: paciento pavardė ir vardas, liga, receptų skaičius, kiek laiko užtruks gydytojo kabinete. Pabaigoje spausdinkite kiek laiko vienetų užtruko visų ligonių aptarnavimas.

### 40. Krepšinis.

Žinomi krepšininkų kvalifikaciniai duomenys. Taip pat žinomos komandos, kuriose tie krepšininkai žaidžia. Reikia sudaryti nurodytos pozicijos (įvedama klaviatūra) naudingiausių žaidėjų nurodyto kiekio (įvedama klaviatūra) sąrašą. Naudingiausias žaidėjas tas, kuris pelnė daugiausiai taškų, žaidė mažiausiai minučių ir padarė mažiausiai klaidų.

- Faile `Krepšis.csv` mače parodyti žaidėjų duomenys: komandos pavadinimas, krepšininko pavardė, vardas, žaistų minučių skaičius, pelnytų taškų skaičius, padarytų klaidų skaičius.
- Faile `Komandos.csv` yra žaidėjų sąrašas: komandos pavadinimas, krepšininko pavardė, vardas, žaidimo pozicija (puolėjas, gynėjas, centras).

Rezultatus spausdinkite faile `Pozicija.txt` spausdinkite atrinktų žaidėjų duomenis: žaidėjo pavardė ir vardas, žaistų minučių skaičius, pelnytų taškų skaičius, padarytų klaidų skaičius. Pabaigoje spausdinkite kiek laiko vienetų užtruko visų ligonių aptarnavimas.

### 41. Išrinktieji.

Darbo valandomis seimo nariai priiminėja rinkėjus seimo priimamajame. Žinomi seimo narių priėmimo laikai. Priėmimai vyksta visą darbo dieną. Vienu metu gal būti ne vienas seimo narys. Rinkėją konsultuoja bet kuris seimo narys. Jeigu visi seimo nariai tuo metu užimti, tenka laukti. Gali būti situacija, kad rinkėjas nebus priimtas atvykimo dienoje. Sudaryti seimo narių sąrašą, kuriame būtų informacija apie vidutinę vienos konsultacijos trukmę, vidutinį apsilankančių rinkėjų skaičių per dieną.

- Faile `Priima.csv` yra priimtų lankytojų sąrašas. Kiekvienoje eilutėje yra: lankytojo pavardė ir vardas, data ir atvykimo laikas, konsultacijos trukmė.
- Faile `Budi.csv` yra seimo narių budėjimo grafikas. Kiekvienoje eilutėje yra: seimo nario pavardė ir vardas, budėjimo priimamajame laikas (pradžią, pabaigą).

Rezultatų faile `Konsultavo.txt` spausdinti: seimo nario pavardė ir vardą, vidutinę vienos konsultacijos trukmę, vidutinis apsilankančių rinkėjų skaičius per dieną. Pabaigoje spausdinti nekonsultuotų rinkėjų skaičių.



#### 42. Telefonai.

Žinomi fiksuoto ryšio telefonų abonentai. Firma kai kuriuos telefonų numerius keičia naujais. Sudaryti sąrašą abonentų, kurių telefonų numeriai pasikeitė.

- Faile `Gyventojai.csv` yra miestelio telefonų abonentų sąrašas. Kiekvienoje eilutėje yra: abonto pavardė ir vardas, adresas, telefono numeris.
- Faile `Firma.csv` yra keičiamų telefonų numerių sąrašas. Kiekvienoje eilutėje yra užrašytas: senas telefono numeris ir naujas telefono numeris.

Rezultatų faile `GyventojaiN.txt` spausdinti pakeistų telefono abonentų sąrašą: pavardė ir vardas, adresas., naujas telefono nr.

#### 43. Prenumerata.

Žinomi prenumeratorių užsakyti dienraščiai. Žinomi duomenys apie dienraščius. Sudaryti užsakymų suvestinę.

- Faile `Gyventojai.csv` yra miestelio prenumeratorių užsakymų sąrašas. Kiekvienoje eilutėje yra: pavardė ir vardas, dienraščio pavadinimas, užsakymo pradžios mėnuo ir keliems mėnesiams užsako. Jeigu gyventojas užsako kelis dienraščius arba kelis kartus per metus, tuomet jam yra skiriamos kelios eilutės ir nebūtinai iš eilės.
- Faile `Dienraščiai.csv` yra duomenys apie leidinius. Kiekvienoje eilutėje yra: leidinio pavadinimas, vieno mėnesio kaina.

Rezultatų faile `Spauda.txt` spausdinti suvestinį sąrašą už visus metus: leidinio pavadinimas, bendras užsakytų mėnesių skaičius, kaina ir pajamos. Pabaigoje spausdinti kiek gyventojai išleido pinigų spaudai per metus.

#### 44. Gėlės.

Studentas nori, kad prie Agnės kompiuterio visa laiką būtų pamerkta nenuvystančių gėlių puokštė. Parašykite programą, kuri sudarytų žydinčių gėlių grafikus.

- Faile `Gėlės.csv` yra gėlių sąrašas: pavadinimas, data, kada parašyta gėlė, kiek dienų žydi, kiek dienų nenuvysta pamerkta, kiek tų gėlių yra.
- Faile `Tinka.csv` yra surašytos gėlių poros, kurios negali būti vienoje puokštėje dėl biologinių ir socialinių priežasčių: pavadinimas, pavadinimas.

Rezultatų faile `Puokštė.txt` spausdinkite gėlių, tinkamų puokštei, sąrašą: pavadinimas, pamerkimo data, nuvytymo data. Puokštės gėlių skaičius, sudarymo pradžios ir pabaigos datos nurodomos klaviatūra. Sąrašo pabaigoje spausdinkite, kiek dienų puokštė bus nepilna.

#### 45. Transportas.

Studentas nori, iš stotelės A nuvažiuoti į stotelę B. Parašykite programą, kuri rekomenduojamų viešojo transporto maršrutų sąrašą be persėdimų ir tik su vienu persėdimu.

- Faile `Stotelės.csv` kiekvienos miestelio stotelės informacinės lentelės duomenys. Viena eilutė vienai stotelei: pavadinimas, maršrutų, kurie stoja toje stotelėje, numeriai.
- Faile `Maršrutai.csv` yra surašyti duomenys apie maršrutus: pavadinimas, stotelių skaičius, stotelių pavadinimai važiavimo krypties seka.

Rezultatų faile `Važiuok.txt` spausdinkite rekomendacijas studentui: maršruto numeris, kiek stotelių važiuoti, stotelių pavadinimai, persėsti į kurį maršrutą, jeigu reikia. Tokiu atveju, sekanti eilutė turi būti persėdimo duomenys. Pradinė ir galinė stotelės nurodomos klaviatūra. Turi būti pranešimas, jeigu studentas negali nuvažiuoti.

#### 46. Produktai.

Studentams reikia maitintis, todėl Zuikio parduotuvėje visai dienai perkami produktai, įvertinant jų kaloringumą ir reikalingą kiekį. Parašykite programą, kuri pateiktų rekomenduojamų produktų sąrašą.

- Faile `Produktas.csv` yra produktų sąrašas: pavadinimas, kaina, vieneto kaloringumas, turimas kiekis.
- Faile `Studentas.csv` yra studentų sąrašas: pavardė ir vardas, turimas pinigų skaičius, svoris.

Rezultatų faile `Maistas.txt` spausdinkite kiekvienam studentui rekomenduojamų produktų sąrašą: pavardė ir vardas, produkto pavadinimas, perkamas kiekis, mokama pinigų suma, kalorijų skaičius. Studentui galima skirti kelias eilutes, jeigu vieno kurio nors produkto jau nepakako. Perkamų produktų suma negali viršyti studento turimų pinigų sumos ir paros būtinų kalorijų skaičiaus  $\pm 10$  proc.

#### 47. Kelionė.

Studentas per atostogas važiuoja į namus. Parašykite programą, kuri pateiktų rekomenduojamų maršrutų sąrašą.

- Faile `Maršrutai.csv` yra maršrutų sąrašas: pavadinimas, sustojimų skaičius, miestų vardai, kuriuose sustoja autobusas.
- Faile yra `Kelionė.csv` yra autobusų sąrašas: maršruto pavadinimas, autobuso numeris, išvykimo laikas, dar nenupirktų bilietų skaičius.

Rezultatų faile `Atostogos.txt` spausdinkite studentui rekomenduojamų maršrutų sąrašą: pavadinimas, autobuso numeris, išvykimo laikas, sustojimų skaičius iki kelionės tikslo. Išvykimo miestas ir kelionės pabaigos miestas nurodomi klaviatūra.

#### 48. Valiuta.

Studentai per atostogas lankosi įvairiose užsienio šalyse. Parašykite programą, kuri padėtų studentams likusią užsienio valiutą pasikeisti į savo šalies valiutą.

- Faile `Sudentas.csv` yra studentų turimų užsienio valiuta pinigų sumos: studento pavardė ir vardas, studento šalies pavadinimas, kiek šalių aplankė, aplankytų šalių pavadinimai ir turima pinigų suma ta valiuta.
- Faile `Valiuta.csv` yra valiutų atitikimai: šalis, pinigų suma, šalis į kurią konvertuojama valiuta, konvertavimo suma.

Rezultatų faile `Pinigai.txt` spausdinkite, kiek kuris studentas gaus pinigų, konvertavęs visus turimus užsienio šalių pinigus į savo šalies: pavardė ir vardas, studento šalis, pinigų suma.