

A feladat során egy telekocsi szolgáltatást végző cég hirdetőinek és utasainak párosítását kell elvégeznie. A hirdetők egy adott útvonalon (induló- és célállomás) vállalják, hogy az autójukban a szabad férőhely erejéig utasokat visznek el.

A feladat megoldása során két UTF-8 kódolású fájlal kell dolgoznia: `autok.csv` és `igenyek.csv`.

Mindkét fájl első sora fejléc adatokat tartalmaz.

Az `autok.csv` fájl minden sora egy-egy hirdető adatát tartalmazza pontosvesszővel elválasztva.

```
Indulás;Cél;Rendszám;Telefonszám;Férőhely
Veszprém;Budapest;NB-95-38;438866905;4
Szekszárd;Győr;XY-84-28;312734682;6
Nyíregyháza;Budapest;AM-75-44;648097594;4
Kaposvár;Kecskemét;YL-50-15;559731734;4
Győr;Budapest;IZ-64-34;616492230;6
Budapest;Debrecen;BO-54-23;713833803;4
```

Az adatok egymás után: indulóváros, célváros. Ezeket követi a rendszám és a sofőr telefonszáma. A sorban az utolsó adat a férőhelyek száma.

Az `igenyek.csv` fájl minden sora egy-egy felhasználói igényt tartalmaz. Az adatok egymás után – ahogy a fejlécben is látszik – az igénylő azonosítója, az induló- és célállomás, valamint az utasok száma:

```
Azonosító;Indulás;Cél;Személyek
F82953;Debrecen;Szeged;4
C44366;Debrecen;Budapest;2
C63068;Szekszárd;Budapest;1
Y47760;Székesfehérvár;Debrecen;1
M64723;Zalaegerszeg;Budapest;1
C98061;Miskolc;Budapest;2
J64779;Eger;Budapest;3
Y87769;Miskolc;Székesfehérvár;2
H21355;Salgótarján;Nyíregyháza;3
```

A megoldás során vegye figyelembe a következőket:

- Az ékezetmentes kiírás is elfogadott.
- A feladat jobb megértése érdekében tanulmányozza a mintákat is!

Készítsen konzolos vagy grafikus programot Telekocsi néven, amely az alábbi feladatokat oldja meg!

1. Olvassa be és tárolja el az `autok.csv` fájl tartalmát!
2. Írja ki, hogy hány hirdető adatát tartalmazta az első feladatban beolvasott fájl!
3. Határozza meg és írja ki a képernyőre, hogy Budapestről Miskolcra összesen hány férőhelyet hirdettek a sofőrök!
4. Határozza meg és írja ki, hogy melyik volt az az útvonal (induló- és célállomás), amelyhez a legtöbb férőhelyet ajánlották fel a hirdetők!
5. Az `igenyek.csv` fájl beérkezési sorrendben tartalmazza az utazási igényeket. Az igények feldolgozása beérkezési sorrendben történik. Olvassa be az igényeket és a beérkezési sorrendnek megfelelően határozza meg, hogy melyek azok az igények, amelyekhez lehet hirdetést (sofőrt) találni! A találatokat a mintának megfelelően írja ki a képernyőre!

6. Készítse el a minta szerint az utasuzenetek.txt fájlt, amely tartalmazza az egyes igényekre adott választ! Az igénylő sikeres párosítás esetén megkapja az autó rendszámát és a sofőr telefonszámát, sikertelen párosítás esetén egy „Sajnos nem sikerült autót találni” üzenetet kap. **Minta**

2. feladat:

500 Autós hirdet fuvart

3. feladat:

összesen 7 férőhelyet hirdettek a sofőrök Budapestről Miskolcra

4. feladat:

A legtöbb férőhelyet (13) Miskolc-Nyíregyháza útvonalon ajánlottak fel

4. feladat(2):

A legtöbb férőhelyet (13) Miskolc-Nyíregyháza útvonalon ajánlottak fel

5. feladat

Z27646=>IPC-445

C61870=>IKS-792

Z45414=>IBW-589

N96245=>NIY-208

S75260=>HBN-630

A29833=>POD-644

D48838=>JGH-785

S92468=>NDS-678

B18683=>NCP-785

P39083=>GEG-407

P39083=>KPF-878

B83376=>IYK-163

R58111=>HVB-511

V96337=>GWY-070

A47162=>IYM-692

D95888=>KNG-671

E58233=>MCL-235

E58233=>LFO-735

S58138=>NGX-631

M39256=>RLW-854

M34983=>RUT-774

M70541=>OLO-593

P11906=>NUE-592

B76976=>QOV-634

K93365=>OFP-181

B97080=>GWR-067

6. feladat

B34019:Sajnos nem sikerült autót találni

S91006:Sajnos nem sikerült autót találni

G28666:Sajnos nem sikerült autót találni

B97693:Sajnos nem sikerült autót találni

Z27646: Rendszám: IPC-445, Telefonszám: 70/718-1826

B87545:Sajnos nem sikerült autót találni

G80209:Sajnos nem sikerült autót találni

P91717:Sajnos nem sikerült autót találni

P88916:Sajnos nem sikerült autót találni

M15988:Sajnos nem sikerült autót találni

V65758:Sajnos nem sikerült autót találni

M32636:Sajnos nem sikerült autót találni

C61870: Rendszám: IKS-792, Telefonszám: 30/639-1689

Z94138:Sajnos nem sikerült autót találni

Z45414: Rendszám: IBW-589, Telefonszám: 20/838-8612

N69508:Sajnos nem sikerült autót találni

V24713:Sajnos nem sikerült autót találni

B46609:Sajnos nem sikerült autót találni

E16189:Sajnos nem sikerült autót találni

N96245: Rendszám: NIY-208, Telefonszám: 70/882-9613

P19188:Sajnos nem sikerült autót találni

Minta a 6. feladathoz (utasuzenetek.txt)

```

B34019:Sajnos nem sikerült autót találni
S91006:Sajnos nem sikerült autót találni
G28666:Sajnos nem sikerült autót találni
B97693:Sajnos nem sikerült autót találni
Z27646: Rendszám: IPC-445, Telefonszám: 70/718-1826
B87545:Sajnos nem sikerült autót találni
G80209:Sajnos nem sikerült autót találni
P91717:Sajnos nem sikerült autót találni
P88916:Sajnos nem sikerült autót találni
M15988:Sajnos nem sikerült autót találni
V65758:Sajnos nem sikerült autót találni
M32636:Sajnos nem sikerült autót találni
C61870: Rendszám: IKS-792, Telefonszám: 30/639-1689
Z94138:Sajnos nem sikerült autót találni
Z45414: Rendszám: IBW-589, Telefonszám: 20/838-8612
N69508:Sajnos nem sikerült autót találni
V24713:Sajnos nem sikerült autót találni
B46609:Sajnos nem sikerült autót találni
E16189:Sajnos nem sikerült autót találni
N96245: Rendszám: NIY-208, Telefonszám: 70/882-9613
P19188:Sajnos nem sikerült autót találni
B76358:Sajnos nem sikerült autót találni
V92261:Sajnos nem sikerült autót találni
S77157:Sajnos nem sikerült autót találni
K17960:Sajnos nem sikerült autót találni
B60481:Sajnos nem sikerült autót találni
B45738:Sajnos nem sikerült autót találni
C59246:Sajnos nem sikerült autót találni
B85008:Sajnos nem sikerült autót találni
S75260: Rendszám: HBN-630, Telefonszám: 30/262-1788

```

Megoldás:

```

using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```
namespace Telekocsi
```

```

{
    /*autok.csv
    * Indulás;Cél;Rendszám;Telefonszám;Férőhely
    Eger;Salgótarján;PQA-209;30/771-8574;5
    Debrecen;Csíkszereda;HLH-090;20/379-3412;2*/
    class Autok
    {
        public string indulas { get; set; }
        public string cel { get; set; }
        public string indcel { get; set; }
        public string rendszam { get; set; }
        public string telefonszam { get; set; }
        public int ferohely { get; set; }
        public Autok(string sor)
        {
            string[] sorelemek=sor.Split(';');//Eger;Salgótarján;PQA-209;30/771-8574;5
            this.indulas= sorelemek[0];
            this.cel= sorelemek[1];
            this.indcel = indulas + "-" + cel;//Eger-Salgótarján
            this.rendszam= sorelemek[2];

```

```

        this.telefonszam= sorelemek[3];
        this.ferohely= Convert.ToInt32(sorelemek[4]);
    }
}
/*igenyek.csv
 * Azonosító;Indulás;Cél;Személyek
B34019;Brassó;Gyula;3
S91006;Sopron;Balassagyarmat;3*/
class Igenyek
{
    public string azonosito { get; set; }
    public string indulas { get; set; }
    public string cel { get; set; }
    public int személyek { get; set; }
    public Igenyek(string sor)
    {
        string[] sorelemek=sor.Split(';');//B34019;Brassó;Gyula;3
        this.azonosito= sorelemek[0];
        this.indulas = sorelemek[1];
        this.cel = sorelemek[2];
        this.szemelyek= Convert.ToInt32(sorelemek[3]);
    }
}
internal class Program
{
    public static List<Autok> autoadatok=new List<Autok>();
    public static List<Igenyek> igenyadatok=new List<Igenyek>();
    static void Main(string[] args)
    {
        string[] fajl = File.ReadAllLines("autok.csv", Encoding.UTF8);
        for (int i = 1; i < fajl.Count(); i++)//i=1 van fejléc
        {
            autoadatok.Add(new Autok(fajl[i]));
        }
        int autoadatokdb = autoadatok.Count();
        fajl = File.ReadAllLines("igenyek.csv", Encoding.UTF8);
        for (int i = 1; i < fajl.Count(); i++)//i=1 van fejléc
        {
            igenyadatok.Add(new Igenyek(fajl[i]));
        }
        int igenyekdb = igenyadatok.Count();
        Console.WriteLine("2. feladat:\n\t{0} Autós hirdet fuvar", autoadatokdb);
        //Console.WriteLine("Igenyek száma: {0}",igenyekdb);
        //3. Határozza meg és írja ki a képernyőre,
        //hogy Budapestről Miskolcra összesen hány férőhelyet hirdettek a sofőrök!
        int osszesferohelyeksza = 0;
        for (int i = 1; i < autoadatokdb; i++)
            if (autoadatok[i].indulas == "Budapest" && autoadatok[i].cel == "Miskolc")
                osszesferohelyeksza += autoadatok[i].ferohely;
        Console.WriteLine("3. feladat:\n\tösszesen {0} férőhelyet hirdettek a sofőrök
Budapestről Miskolcra",osszesferohelyeksza);
        /*osszesferohelyeksza = 0;
        foreach (var l in autoadatok)
        {
            if(l.indulas=="Budapest" && l.cel=="Miskolc") osszesferohelyeksza +=
l.ferohely;
        }
        Console.WriteLine("3. feladat:\n\tösszesen {0} férőhelyet hirdettek a sofőrök
Budapestről Miskolcra", osszesferohelyeksza);
        */
        /*4. Határozza meg és írja ki, hogy melyik volt az az útvonal
        * (induló- és célállomás),
        * amelyhez a legtöbb férőhelyet ajánlották fel a hirdető! */
        Console.WriteLine("4. feladat:");
        /*autoadatok.GroupBy(x => x.indcel, x => x.ferohely).OrderByDescending(x =>
x.Sum()).Where(x => x.Sum()>=13).
        ToList().ForEach(x => Console.WriteLine("\tA legtöbb férőhelyet ({0}) {1}
útvonalon ajánlottak fel", x.Sum(), x.Key));*/
        List<string> legtobbferohely = new List<string>();

```

```

        autoadatok.GroupBy(x => x.indcel, x => x.ferohely).OrderByDescending(x =>
x.Sum()).ToList().ForEach(x =>legtobbferohely.Add( "("+ x.Sum().ToString()+") "+x.Key ));
        Console.WriteLine("\tA legtöbb férőhelyet {0} útvonalon ajánlottak
fel", legtobbferohely.First());

List<string> utvonalaklista = new List<string>();
autoadatok.GroupBy(x=>x.indcel).ToList().ForEach(x=>utvonalaklista.Add(x.Key));
/*foreach (string s in utvonalaklista)
    Console.WriteLine(s);*/
int utvonalakdb = utvonalaklista.Count();
int[] osszferohely=new int[utvonalakdb];
for (int i = 0; i < autoadatokdb; i++)
{
    for (int j = 0; j < utvonalakdb; j++)
    {
        if (utvonalaklista[j] == autoadatok[i].indcel)
            osszferohely[j] += autoadatok[i].ferohely;
    }
}
int maxferohely = osszferohely[0];
int maxsorszam = 0;
for (int i = 0; i < utvonalakdb; i++)
{
    if(osszferohely[i]>maxferohely)
    {
        maxferohely= osszferohely[i];
        maxsorszam = i;
    }
}
Console.WriteLine("4. feladat(2):\n\tA legtöbb férőhelyet ({0}) {1} útvonalon
ajánlottak fel",
    maxferohely,utvonalaklista[maxsorszam]);
/*5. Az igények.csv fájl beérkezési sorrendben tartalmazza az utazási igényeket.
* Az igények feldolgozása beérkezési sorrendben történik.
* Olvassa be az igényeket és a beérkezési sorrendnek megfelelően határozza meg,
* hogy melyek azok az igények, amelyekhez lehet hirdetést (sofőrt) találni!
* A találatokat a mintának megfelelően írja ki a képernyőre! */
Console.WriteLine("5. feladat");
for (int i = 0; i < igényekdb; i++)
{
    for (int j = 0; j < autoadatokdb; j++)
    {
        if(igényadatok[i].indulas==autoadatok[j].indulas
            && igényadatok[i].cel== autoadatok[j].cel
            && igényadatok[i].szemelyek<= autoadatok[j].ferohely)
            Console.WriteLine("{0}=>{1}", igényadatok[i].azonosito,
autoadatok[j].rendszam);
    }
}
/*6. Készítse el a minta szerint az utasuzenetek.txt fájlt,
* amely tartalmazza az egyes igényekre adott választ!
* Az igénylő sikeres párosítás esetén megkapja az autó rendszámát és a sofőr
telefonszámát,
    sikertelen párosítás esetén egy „Sajnos nem sikerült autót találni” üzenetet
kap.
*/
Console.WriteLine("6. feladat");
FileStream f = new FileStream("utasuzenetek.txt", FileMode.Create);
StreamWriter fajlbairo=new StreamWriter(f);
bool ok = false;
for (int i = 0; i < igényekdb; i++)
{
    ok = false;
    for (int j = 0; j < autoadatokdb; j++)
    {
        if (igényadatok[i].indulas == autoadatok[j].indulas
            && igényadatok[i].cel == autoadatok[j].cel
            && igényadatok[i].szemelyek <= autoadatok[j].ferohely)
        {

```

```

        ok = true;
        fajlbairo.WriteLine("{0}: Rendszám: {1}, Telefonszám: {2}",
            igényadatok[i].azonosito, autoadatok[j].rendszám,
            autoadatok[j].telefonszam);
        Console.WriteLine("{0}: Rendszám: {1}, Telefonszám: {2}",
            igényadatok[i].azonosito, autoadatok[j].rendszám,
            autoadatok[j].telefonszam);
    }
    }
    if(!ok)
    {
        fajlbairo.WriteLine("{0}:Sajnos nem sikerült autót
találni", igényadatok[i].azonosito);
        Console.WriteLine("{0}:Sajnos nem sikerült autót
találni", igényadatok[i].azonosito);
    }

    }
    fajlbairo.Close();
    f.Close();

    Console.ReadKey();
}
}
}

```