

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 hirdetés fuvart

3. feladat  
Összesen 7 férőhelyet hirdettek az autósok Budapestről Miskolcra

4. feladat  
A legtöbb férőhelyet (13-t) a Miskolc-Nyíregyháza útvonalon ajánlották fel a hirdetőik

5. feladat  
Z27646=>IPC-445  
Z45414=>IBW-589  
S75260=>HBN-630  
A29833=>POD-644  
S92468=>NDS-678  
P39083=>GEG-407  
P39083=>KPF-878  
B83376=>IYK-163  
V96337=>GWY-070  
A47162=>IYM-692  
E58233=>MCL-235  
E58233=>LFO-735  
S58138=>NGX-631  
M39256=>RLW-854  
M34983=>RUT-774  
P11906=>NUE-592  
B76976=>OQV-634  
B97080=>GWR-067

### Minta a 6. feladathoz (utasuzenetek.txt)

ß34019: 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: GCJ-707, Telefonszám: 20/931-7703  
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: Sajnos nem sikerült autót találni  
Z94138: Sajnos nem sikerült autót találni  
Z45414: Rendszám: PGR-744, Telefonszám: 20/320-8689  
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: Sajnos nem sikerült autót találni  
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: LDN-259, Telefonszám: 70/610-7459  
L52932: Sajnos nem sikerült autót találni  
B96261: Sajnos nem sikerült autót találni  
P54027: Sajnos nem sikerült autót találni  
B45472: Sajnos nem sikerült autót találni  
K81132: Sajnos nem sikerült autót találni  
B39510: Sajnos nem sikerült autót találni  
A29833: Rendszám: JLC-242, Telefonszám: 30/538-6940  
P92957: Sajnos nem sikerült autót találni  
N30895: Sajnos nem sikerült autót találni  
C50428: Sajnos nem sikerült autót találni

### **Megoldás:**

using System;

```

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

namespace telekocsi
{
    class Program
    {
        struct auto//Készítsen összetett változót az adatok tárolására!
        {
            public string indulas;
            public string cel;
            public string rendszam;
            public string telefonszam;
            public int ferohely;
        }
        static auto[] adatok = new auto[1000];//Az állományban legfeljebb 1000 sor lehet.
        struct igeny//Készítsen összetett változót az adatok tárolására!
        {
            public string azonosito;
            public string indulas;
            public string cel;
            public int személyek;
            public string rendszam;
            public string telefonszam;
            public int ok;
        }
        static igeny[] igenyek = new igeny[500];//Az állományban legfeljebb 500 sor lehet.
        static void Main(string[] args)
        {
            string[] fajlbol = File.ReadAllLines("autok.csv");
            int sorokszama = 0;//sorok száma a fájlban
            int i, j;//ciklusváltozó
            for (int k = 1; k < fajlbol.Count(); k++)//Ügyeljen arra, hogy az állomány első sora az
            adatok fejlécét tartalmazza!
            {
                string[] egysordarabolva = fajlbol[k].Split(';');//Az adatokat pontosvessző
                választja el.
                adatok[sorokszama].indulas = egysordarabolva[0];
                adatok[sorokszama].cel = egysordarabolva[1];
                adatok[sorokszama].rendszam = egysordarabolva[2];
                adatok[sorokszama].telefonszam = egysordarabolva[3];
                adatok[sorokszama].ferohely = Convert.ToInt32(egysordarabolva[4]);
                sorokszama++;
            }
            int autokszama = sorokszama;

            /*Console.WriteLine("Az adatok listája fájlból");

            Console.WriteLine(" Indulás        Cél            Rendszám      Telefonszám   Férőhely");
            for (i = 0; i < autokszama; i++)
            {
                Console.WriteLine("{0,-15} {1,-15} {2,-10} {3,-15} {4} ", adatok[i].indulas,
                adatok[i].cel, adatok[i].rendszam, adatok[i].telefonszam, adatok[i].ferohely);
            }*/
            string[] fajlbol2 = File.ReadAllLines("igenyek.csv");
            sorokszama = 0;//sorok száma a fájlban
            for (int k = 1; k < fajlbol2.Count(); k++)//Ügyeljen arra, hogy az állomány első sora az
            adatok fejlécét tartalmazza!
            {
                string[] egysordarabolva = fajlbol2[k].Split(';');//Az adatokat pontosvessző
                választja el.
                igenyek[sorokszama].azonosito = egysordarabolva[0];
                igenyek[sorokszama].indulas = egysordarabolva[1];
                igenyek[sorokszama].cel = egysordarabolva[2];
                igenyek[sorokszama].szemelyek = Convert.ToInt32(egysordarabolva[3]);
                sorokszama++;
            }
        }
    }
}

```

```

}
int igenyekszama = sorokszama;

/*Console.WriteLine("Az adatok listája fájlból");

Console.WriteLine(" Azonosító      Indulás      Cél      Személyek");
for (i = 0; i < igenyekszama; i++)
{
    Console.WriteLine("{0,-15} {1,-15} {2,-15} {3} ", igenyek[i].azonosito,
igenyek[i].indulas, igenyek[i].cel, igenyek[i].szemelyek);
}*/
//2. Írja ki, hogy hány hirdető adatát tartalmazta az első feladatban beolvasott
fájl!
Console.WriteLine("2. feladat\n\t{0} autós hirdet fuvart", autokszama);
//3. Határozza meg és írja ki a képernyőre, hogy Budapestről Miskolcra összesen hány
férőhelyet hirdettek a sofőrök!
//összegzés tétele
int osszesferohely = 0;
for (i = 0; i < autokszama; i++)
{
    if(adatok[i].indulas=="Budapest" && adatok[i].cel == "Miskolc")
    {
        osszesferohely += adatok[i].ferohely;
    }
}
Console.WriteLine("3. feladat\n\tÖsszesen {0} férőhelyet hirdettek az autósok
Budapestről Miskolcra", osszesferohely);
//4. Határozza meg és írja ki, hogy melyik volt az az útvonal (induló- és
célállomás),
//amelyhez a legtöbb férőhelyet ajánlották fel a hirdető!
//adott egy sorozat, határozzuk meg hány különböző eleme van és gyűjtsük ki egy tömbbe
int kulonbozoelemekszama = 0;
string[] indcel = new string[1000];//különböző útvonalak kiválogatása (indulas-cel)
int[] ferohelyosszesen = new int[1000];
for (i = 0; i < 1000; i++) ferohelyosszesen[i] = 0;
for (i = 0; i < autokszama; i++)
{
    j = 0;
    while ((j <= kulonbozoelemekszama) && (adatok[i].indulas+"-"+adatok[i].cel !=
indcel[j]))
    {
        j++;
    }
    if (j > kulonbozoelemekszama)
    {
        kulonbozoelemekszama++;
        indcel[kulonbozoelemekszama] = adatok[i].indulas + "-" + adatok[i].cel;
    }
}
/*for (i = 1; i <= kulonbozoelemekszama; i++)
    Console.WriteLine("\t{0} útvonal ", indcel[i]);*/
//megszámlálás tétele
for (i = 0; i < autokszama; i++)
{
    for (j = 0; j < kulonbozoelemekszama; j++)
    {
        if(indcel[j]== adatok[i].indulas + "-" + adatok[i].cel)
        {
            ferohelyosszesen[j] += adatok[i].ferohely;
        }
    }
}
//maximumkiválasztás tétele
int max = ferohelyosszesen[0];
int maxi = 0;
for (j = 0; j < kulonbozoelemekszama; j++)
{
    if (ferohelyosszesen[j] > max)

```

```

        {
            max=ferohelyosszesen[j];
            maxi = j;
        }
    }
    Console.WriteLine("4. feladat\n\tA legtöbb férőhelyet ({0}-t) a {1} útvonalon ajánlották
fel a hirdető", ferohelyosszesen[maxi],indcel[maxi]);
/*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 (i = 0; i < igyekszama; i++)
{
    for (j = 0; j < autokszama; j++)
    {
        if(igenyek[i].indulas==adatok[j].indulas && igenyek[i].cel == adatok[j].cel &&
igenyek[i].szemelyek < adatok[j].ferohely)
        {
            Console.WriteLine("{0}=>{1}",igenyek[i].azonosito,adatok[j].rendsza);
            igenyek[i].telefonszam = adatok[j].telefonszam;
            igenyek[i].rendsza = adatok[j].rendsza;
            igenyek[i].ok = 1;
        }
    }
}
/*6. Készítse el a minta szerint az utasuzenetek.txt fájl,
* 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. */
//kiválogatás
Console.WriteLine("6. feladat");
FileStream fnev = new FileStream("utasuzenetek.txt", FileMode.Create);
StreamWriter fajlbairo = new StreamWriter(fnev);
for (i = 0; i < igyekszama; i++)
{
    if (igenyek[i].ok==1)
    {
        fajlbairo.WriteLine("{0}: Rendszám: {1}, Telefonszám: {2}",
igenyek[i].azonosito, adatok[i].rendsza,adatok[i].telefonszam);
        Console.WriteLine("{0}: Rendszám: {1}, Telefonszám: {2}",
igenyek[i].azonosito, igenyek[i].rendsza, igenyek[i].telefonszam);
    }
    else
    {
        fajlbairo.WriteLine("{0}: Sajnos nem sikerült autót
találni",igenyek[i].azonosito);
        Console.WriteLine("{0}: Sajnos nem sikerült autót találni",
igenyek[i].azonosito);
    }
}

    }

    fajlbairo.Close();
    fnev.Close();

    Console.ReadKey();
}
}
}

```