

## Fuvar

A következő feladatban 2016-os chicagói taxis fuvarozások adatait tartalmazó szöveges állományt kell feldolgoznia. A megoldás során vegye figyelembe a következőket!

- A képernyőre írást igénylő részfeladatok eredményének megjelenítése előtt írja a képernyőre a feladat sorszámát (például: 3. feladat:)!
- Az egyes feladatokban a kiírásokat a minta szerint készítse el!
- Az ékezetmentes kiírás is elfogadott.
- A program megírásakor a fájlban lévő adatok helyes szerkezetét nem kell ellenőriznie, feltételezheti, hogy a rendelkezésre álló adatok a leírtaknak megfelelnek.
- Megoldását úgy készítse el, hogy az azonos szerkezetű, de tetszőleges bemeneti adatok mellett is helyes eredményt adjon!

1. A feladat megoldásához hozzon létre grafikus vagy konzolalkalmazást (projektet) Fuvar azonosítóval!
2. Az UTF-8 kódolású fuvar.csv állomány tartalmazza a fuvarok adatait. A feladatok megoldása előtt tanulmányozza az állomány szerkezetét! Minden adatsorhoz rendre a következő mezők tartoznak:

- taxi azonosítója (egész szám, pl.: 8192)
- ● indulás időpontja (időbélyegző, melyben minden adat előnullázott, akár szöveggént is kezelhető, pl.: 2016-12-02 07:45:00)
- ● az utazás időtartama(egész szám, az adatok másodpercben értendők, pl. 900)
- ● a megtett távolság (valós szám, az adatok mérföldben értendők, pl. 1,5)
- ● viteldíj (valós szám, az adatok dollárban értendők, pl. 7,5)
- ● borraivaló (valós szám, az adatok dollárban értendők, pl. 4, 15)
- ● a fizetés módja (szöveges, pl. „bankkártya”)

Az adatokat pontosvessző választja el egymástól. Ügyeljen arra, hogy a fájl első sora az adatok fejlécét tartalmazza! Olvassa be a fuvar.csv állományban található adatokat és tárolja el egy megfelelően megválasztott adatszerkezetben!

3. Határozza meg és írja ki a képernyőre a minta szerint, hogy hány utazás került feljegyzésre az állományban!
4. Határozza meg és írja ki a képernyőre a minta szerint, hogy a 6185-ös azonosítójú taxisnak mennyi volt a bevétele, és ez hány fuvarból állt! Feltételezheti, hogy van ilyen azonosítójú taxis.
5. Programjával határozza meg az állomány adataiból a fizetési módokat, majd osszesítse, hogy az egyes fizetési módokat hányszor választották az utak során! Ezeket az eredményeket a minta szerint írja a képernyőre! A kiírás során a fizetési módok sorrendje bármilyen lehet.
6. Határozza meg és írja ki a képernyőre a minta szerint, hogy összesen hány km-t tettek meg a taxisok (1 mérföld 1,6 km)! Az eredményt két tizedesjegyre kerekítve jelenítse meg a képernyőn!
7. Határozza meg és írja ki a képernyőre a minta szerint az időben leghosszabb fuvar adatait! Feltételezheti, hogy nem alakult ki holtverseny.
8. Hozzon létre hibak.txt néven egy UTF-8 kódolású szöveges állományt, ami tartalmazza azokat az adatokat, amelyek esetében hiba van az eredeti állományban! Hibás sornak tekintjük azokat az eseteket, amelyekben az utazás időtartama és a viteldíj egy nullánál nagyobb érték, de a hozzá tartozó megtett távolság értéke nulla. A sorok indulási időpont szerint növekvő rendben legyenek az állományban! A hibak.txt állomány szerkezete egyezzen meg a fuvar.csv állomány szerkezetével!

## Minta

```
3. feladat: 1859 fuvar
4. feladat: 4 fuvar alatt: 33,75 $
5. feladat:
    bankkártya: 793 fuvar
    készpénz: 1050 fuvar
    vitatott: 4 fuvar
    ingyenes: 10 fuvar
    ismeretlen: 2 fuvar
6. feladat: 8099,95 km
7. feladat: Leghosszabb
    Fuvar hossza: 9120 másodperc
    Taxi azonosító: 2559
    Megtett távolság: 56,16 km
    Viteldíj: 88,25 $
8. feladat: hibak.txt
```

Hibak.txt állomány sorai:

```

taxi_id;indulas;idotartam;tavolsag;viteldij;borravalo;fizetes_modja
6065;2016-12-01 02:30:00;60;0;8;0;bankkártya
5243;2016-12-01 09:15:00;480;0;6,75;0;késpénz
5114;2016-12-01 09:30:00;420;0;6;2;bankkártya
2118;2016-12-01 20:45:00;780;0;11,75;0;bankkártya
4531;2016-12-01 21:15:00;540;0;7,25;0;késpénz
6535;2016-12-01 21:30:00;840;0;8;0;késpénz
710;2016-12-01 21:45:00;360;0;6,25;0;késpénz
2643;2016-12-01 23:45:00;180;0;4,25;0;késpénz
7939;2016-12-02 04:15:00;300;0;5,75;0;késpénz
2863;2016-12-02 07:45:00;360;0;6,75;1;bankkártya
5400;2016-12-02 09:30:00;780;0;9,25;0;késpénz
6197;2016-12-02 13:30:00;60;0;3,25;0;késpénz
4044;2016-12-02 13:45:00;960;0;11;0;késpénz
1026;2016-12-02 15:45:00;300;0;5,5;0;késpénz
3773;2016-12-02 16:00:00;420;0;6;0;késpénz
569;2016-12-02 17:00:00;180;0;4,5;2;bankkártya
6078;2016-12-02 17:00:00;540;0;7;2;bankkártya
3449;2016-12-02 17:15:00;60;0;3,25;0;késpénz
4611;2016-12-02 19:00:00;600;0;7,5;0;késpénz
6562;2016-12-02 20:00:00;240;0;5,25;0;késpénz
1901;2016-12-02 21:45:00;60;0;3,25;0;késpénz

```

## Megoldás:

```

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

namespace fuvar
{
    class Program
    {
        struct fuvarozas//Készítsen összetett változót az adatok tárolására!
        {
            public int taxiid;
            public string indulas;
            public int idotartam;
            public double tavolsag;
            public double viteldij;
            public double borravalo;
            public string fizetesmod;
        }

        static fuvarozas[] adatok = new fuvarozas[2000];//Az állományban legfeljebb 2000 sor lehet.
        struct fuvarozas2//Készítsen összetett változót az adatok tárolására!
        {
            public int taxiid;
            public string indulas;
            public int idotartam;
            public double tavolsag;
            public double viteldij;
            public double borravalo;
            public string fizetesmod;
        }

        static fuvarozas2[] adatok2 = new fuvarozas2[2000];
        static void Main(string[] args)
        {
            string[] fajlbol = File.ReadAllLines("fuvar.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].taxiid = Convert.ToInt32(egysordarabolva[0]);
                adatok[sorokszama].indulas = egysordarabolva[1];
                adatok[sorokszama].idotartam = Convert.ToInt32(egysordarabolva[2]);
                adatok[sorokszama].tavolsag = Convert.ToDouble(egysordarabolva[3]);
                adatok[sorokszama].viteldij = Convert.ToDouble(egysordarabolva[4]);
                adatok[sorokszama].borravalo = Convert.ToDouble(egysordarabolva[5]);
                adatok[sorokszama].fizetesmod = egysordarabolva[6];
                sorokszama++;
            }
            int fuvarokszama = sorokszama;

```

//3. Határozza meg és írja ki a képernyőre a minta szerint, hogy hany utazás került feljegyzésre az állományban!

```
Console.WriteLine("3. feladat: {0} fuvar", fuvarokszama);
//4. Határozza meg és írja ki a képernyőre a minta szerint,
//hogy a 6185-ös azonosítójú taxinak mennyi volt a bevétele,
//és ez hány fuvarból állt! Feltételezheti, hogy van ilyen azonosítójú taxis.
//összegzés tétele, megszámlálás tétele
double osszesen = 0;
int db = 0;
for (i = 0; i < fuvarokszama; i++)
{
    if (adatok[i].taxiid == 6185)
    {
        osszesen += adatok[i].viteldij;
        db++;
    }
}
Console.WriteLine("4. feladat: {0} fuvar alatt: {1} $", db,osszesen);
/*5. Programjával határozza meg az állomány adataiból a fizetési módokat,
* majd összesítse, hogy az egyes fizetési módokat hányszor választották az utak során!
* Ezeket az eredményeket a minta szerint írja a képernyőre!
* A kiírás során a fizetési módok sorrendje bármilyen lehet.*/
//adott egy sorozat, határozzuk meg hány különböző eleme van és gyűjtsük ki egy tömbbe
int kulonbozoelemekszama = 0;
string[] fizetesimodok = new string[100];
int[] fizetesszam = new int[100];
for (i = 0; i < fuvarokszama; i++)
{
    j = 0;
    while ((j <= kulonbozoelemekszama) && (adatok[i].fizetesmod != fizetesimodok[j]))
    {
        j++;
    }
    if (j > kulonbozoelemekszama)
    {
        kulonbozoelemekszama++;
        fizetesimodok[kulonbozoelemekszama] = adatok[i].fizetesmod;
    }
}
//megszámlálás tétele
for (i = 0; i < fuvarokszama; i++)
{
    for (int k = 1; k <= kulonbozoelemekszama; k++)
    {
        if (fizetesimodok[k] == adatok[i].fizetesmod) fizetesszam[k]++;
    }
}
Console.WriteLine("5. feladat:");
for (i = 1; i <= kulonbozoelemekszama; i++)
    Console.WriteLine("\t{0}: {1} fuvar ", fizetesimodok[i], fizetesszam[i]);
```

```

/*6. Határozza meg és írja ki a képernyőre a minta szerint,
 * hogy összesen hány km-t tettek meg a taxikok (1 mérföld : 1,6 km)!
 * Az eredményt két tizedesjegyre kerekítve jelenítse meg a képernyőn!*/
//összegzés tétele
double osszesmerfold = 0;
for (i = 0; i < fuvarokszama; i++)
{
    osszesmerfold += adatok[i].tavolsag;
}
Console.WriteLine("6. feladat: {0} km",Math.Round(osszesmerfold*1.6,2));
//7. Határozza meg és írja ki a képernyőre a minta szerint az időben leghosszabb fuvar adatait!
//Feltételezheti, hogy nem alakult ki holtverseny.
//maximumkiválasztás tétele
int maxfuvar= adatok[0].idotartam;//feltételezem, hogy az első a legnagyobb
int maxi = 0;//sorszám
for (i = 0; i < fuvarokszama; i++)
{
    if (adatok[i].idotartam > maxfuvar)
    {
        maxfuvar = adatok[i].idotartam;
        maxi = i;
    }
}
Console.WriteLine("7. feladat: Leghosszabb\n\tFuvar hossza: {0} másodperc", adatok[maxi].idotartam);
Console.WriteLine("\tTaxi azonosító: {0}", adatok[maxi].taxiid);
Console.WriteLine("\tMegtett távolság: {0} km", adatok[maxi].tavolsag*1.6);
Console.WriteLine("\tViteldíj: {0} $", adatok[maxi].viteldij);

/*8. Hozzon létre hibak.txt néven egy UTF-8 kódolású szöveges állományt,
 * ami tartalmazza azokat az adatokat, amelyek esetében hiba van az eredeti állományban!
 * Hibás sornak tekintjük azokat az eseteket, amelyekben az utazás időtartama és a viteldíj egy nullánál nagyobb érték,
 * de a hozzá tartozó megtett távolság értéke nulla.
 * A sorok indulási időpont szerint növekvő rendben legyenek az állományban!
 * A hibak.txt állomány szerkezete egyezzen meg a fuvar.csv állomány szerkezetével!*/
Console.WriteLine("8. feladat: hibak.txt");
//hibás adatok kiválogatása
int idb = 0;
for (i = 0; i < fuvarokszama; i++)
{
    if (adatok[i].idotartam>0 && adatok[i].viteldij > 0 && adatok[i].tavolsag == 0)
    {
        adatok2[idb].taxiid= adatok[i].taxiid;
        adatok2[idb].indulas= adatok[i].indulas;
        adatok2[idb].idotartam= adatok[i].idotartam;
        adatok2[idb].tavolsag= adatok[i].tavolsag;
        adatok2[idb].viteldij= adatok[i].viteldij;
        adatok2[idb].borravallo= adatok[i].borravallo;
        adatok2[idb].fizetesmod= adatok[i].fizetesmod;
        idb++;
    }
}
int hibakszama = idb;
//sorbarendezés min kiválasztással

int mini = 0;
string min;
for (i = 0; i < hibakszama; i++)
{
    mini = i;
    min = adatok2[i].indulas;
    for (j = i; j < hibakszama; j++)
    {
        int x = String.Compare(adatok2[j].indulas, min);
        if (x < 0)
        {
            mini = j;
            min = adatok2[j].indulas;
        }
    }
    //csere
    var s = adatok2[i];
    adatok2[i] = adatok2[mini];
    adatok2[mini] = s;
}
}

```

```
FileStream fnev = new FileStream("hibak.txt", FileMode.Create);
StreamWriter fajlbairo = new StreamWriter(fnev);
fajlbairo.WriteLine("taxi_id;indulas;idotartam;tavolsag;viteldij;borravaló;fizetes_modja");
for (i = 0; i < hibak.szama; i++)
{
    fajlbairo.Write("{0};", adatok2[i].taxiid);
    fajlbairo.Write("{0};", adatok2[i].indulas);
    fajlbairo.Write("{0};", adatok2[i].idotartam);
    fajlbairo.Write("{0};", adatok2[i].tavolsag);
    fajlbairo.Write("{0};", adatok2[i].viteldij);
    fajlbairo.Write("{0};", adatok2[i].borravaló);
    fajlbairo.Write("{0};", adatok2[i].fizetesmod);
    fajlbairo.WriteLine("\n");//sortörés
}
fajlbairo.Close();
fnev.Close();

Console.ReadKey();
}
```