

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 taxinak 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 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.

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: 8099,95km
7. feladat: Leghosszabb fuvar:
    Fuvar hossza: 9120 másodperc
    Taxi azonosító: 2559
    Megtett távolság: 35,1 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;készpénz
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;készpénz
6535;2016-12-01 21:30:00;840;0;8;0;készpénz
710;2016-12-01 21:45:00;360;0;6,25;0;készpénz
2643;2016-12-01 23:45:00;180;0;4,25;0;készpénz
7939;2016-12-02 04:15:00;300;0;5,75;0;készpénz
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;készpénz
6197;2016-12-02 13:30:00;60;0;3,25;0;készpénz
4044;2016-12-02 13:45:00;960;0;11;0;készpénz
1026;2016-12-02 15:45:00;300;0;5,5;0;készpénz
3773;2016-12-02 16:00:00;420;0;6;0;készpénz
569;2016-12-02 17:00:00;180;0;4,5;2;bankkártya
...
```

Megoldás:

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

namespace Fuvar
{
    class Fuvarok
    {
        public Fuvarok(string sor)
        {
            string[] sorelemek = sor.Split(';');
            this.Taxi_id = sorelemek[0];
            this.Indulas = Convert.ToDateTime(sorelemek[1]);
            this.Idotartam = Convert.ToInt32(sorelemek[2]);
            this.Tavolsag = Convert.ToDouble(sorelemek[3]);
            this.Viteldij = Convert.ToDouble(sorelemek[4]);
            this.Borravalo = Convert.ToDouble(sorelemek[5]);
            this.Fizetesimod = sorelemek[6];
        }
        //taxi_id;indulas;idotartam;tavolsag;viteldij;borravalo;fizetes_modja
    }
}
```

```

//5240;2016-12-15 23:45:00;900;2,5;10,75;2,45;bankkártya
public string Taxi_id { get; set; }
public DateTime Indulas { get; set; }
public int Idotartam { get; set; } //sec
public double Tavolsag { get; set; } //mérföld
public double Viteldij { get; set; } //dollár
public double Borravalo { get; set; } //dollár
public string Fizetesimod { get; set; }
}
class Program
{
    public static List<Fuvarok> adatok = new List<Fuvarok>();
    static void Main(string[] args)
    {
        StreamReader olvas = new StreamReader("fuvar.csv", Encoding.UTF8);
        string fejlac = olvas.ReadLine();
        while(!olvas.EndOfStream)
        {
            adatok.Add(new Fuvarok(olvas.ReadLine()));
        }
        int i, j;
        int adatokszama = adatok.Count;
        /*Console.WriteLine("taxi_id indulas idotartam tavolsag
viteldij borravalo fizetes_modja");
for(i=0;i<adatokszama;i++)
    Console.WriteLine("{0,-7}{1,-20}{2,-7}{3,-7}{4,-7}{5,-7}{6}",
        adatok[i].Taxi_id, adatok[i].Indulas, adatok[i].Idotartam,
adatok[i].Tavolsag, adatok[i].Viteldij, adatok[i].Borravalo, adatok[i].Fizetesimod);*/

        //3. Határozza meg és írja ki a képernyőre a minta szerint, hogy hany utazás
került feljegyzésre az állományban
        Console.WriteLine("3. feladat: {0} fuvar",adatokszama);

        //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.
        double bevetel = 0;
        int fuvarokszama = 0;
        double osszeskm = 0; //6. feladat
        for (i = 0; i < adatokszama; i++)
        {
            osszeskm += (adatok[i].Tavolsag * 1.6); //6. feladat
            if (adatok[i].Taxi_id == "6185")
            {
                fuvarokszama++;
                bevetel += adatok[i].Viteldij + adatok[i].Borravalo;
            }
        }

        Console.WriteLine("4. feladat: {0} fuvar alatt {1} $", fuvarokszama, bevetel);

        //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.*/
        Console.WriteLine("5. feladat:");
        adatok.GroupBy(x => x.Fizetesimod).ToList().ForEach(x =>
Console.WriteLine("\t{0}: {1} fuvar", x.Key, x.Count()));

        //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!*/
        Console.WriteLine("6. feladat: {0} km", Math.Round(osszeskm, 2));
        //7. Határozza meg és írja ki a képernyőre a minta szerint az időben
leghosszabb fuvar adatait!
        * Feltételezheti, hogy nem alakult ki holtverseny.*/
        int max = adatok[0].Idotartam;

```

```

int maxi = 0;
for (i = 0; i < adatokszama; i++)
    if(adatok[i].Idotartam>max)
    {
        max = adatok[i].Idotartam;
        maxi = i;
    }
Console.WriteLine("7. feladat: Leghosszabb fuvar:");
Console.WriteLine("\tFuvar hossza: {0} másodperc",max);
Console.WriteLine("\tTaxi azonosító: {0}", adatok[maxi].Taxi_id);
Console.WriteLine("\tMegtett távolság: {0} km", adatok[maxi].Tavolsag*1.6);
Console.WriteLine("\tViteldíj: {0} $", adatok[maxi].Viteldij+
adatok[maxi].Borravalo);

/*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!*/
List<Fuvarok> hibalista = new List<Fuvarok>();
for (i = 0; i < adatokszama; i++)
    if (adatok[i].Idotartam > 0 && adatok[i].Viteldij > 0 &&
adatok[i].Tavolsag == 0)
        hibalista.Add(adatok[i]);
List<Fuvarok> sorbarendezt = hibalista.OrderBy(x=>x.Indulas).ToList();
FileStream fnev = new FileStream("hibak.txt", FileMode.Create);
StreamWriter fajlbairo = new StreamWriter(fnev);

fajlbairo.WriteLine("taxi_id;indulas;idotartam;tavolsag;viteldij;borravalo;fizetes_modja"
);
for(i=0;i<sortbarendezt.Count;i++)
{
    fajlbairo.Write("{0};", sortbarendezt[i].Taxi_id);
    fajlbairo.Write("{0};", sortbarendezt[i].Indulas);
    fajlbairo.Write("{0};", sortbarendezt[i].Idotartam);
    fajlbairo.Write("{0};", sortbarendezt[i].Tavolsag);
    fajlbairo.Write("{0};", sortbarendezt[i].Viteldij);
    fajlbairo.Write("{0};", sortbarendezt[i].Borravalo);
    fajlbairo.WriteLine("{0};", sortbarendezt[i].Fizetesimod);
}
fajlbairo.Close();
fnev.Close();
Console.WriteLine("8. feladat: hibak.txt");
    Console.ReadKey();
}
}
}

```