

Super Bowl-döntők¹

A Super Bowl egy évente megrendezésre kerülő amerikaifutball-mérkőzés, ahol eldől, hogy ki lesz a National Football League (NFL) bajnoka. A rendezvénynek az Egyesült Államok egyik városa ad otthont, melyet 3-4 évvel korábban választanak ki. Ebben a feladatban a Super Bowl- döntők adataival kell feladatokat megoldania. Megoldásában vegye figyelembe a következőket:

- *A képernyőre írást igénylő feladatok eredményének megjelenítése előtt írja a képernyőre a feladat sorszámát (például: 3. feladat)!*
- *Az egyes feladatokban a kiírásokat a minta szerint készítse el!*
- *Az ékezetmentes kiírások is elfogadottak.*
- *Az azonosítókat kis- és nagybetűkkel is kezdheti.*
- *A program megírásakor az állományban lévő adatok helyes szerkezetét nem kell ellenőriznie, feltételezheti, hogy a rendelkezésre álló adatok a leírtaknak megfelelnek.*
- *A megoldását úgy készítse el, hogy az azonos szerkezetű, de tetszőleges bemeneti adatok mellett is helyes eredményt adjon.*

Az UTF-8 kódolású SuperBowl.txt forrásállomány tartalmazza soronként a döntők adatait, melyeket pontosvesszővel választottuk el:

```
Ssz;Dátum;Győztes;Eredmény;Vesztes;Helyszín;VárosÁllam;Nézőszám
...
II.;1968.01.14.;Green Bay Packers;33-14;Oakland Raiders;Orange Bowl;Miami, Florida;75546
III.;1969.01.12.;New York Jets;16-7;Baltimore Colts;Orange Bowl;Miami, Florida;75389
...
```

- Ssz: A döntő sorszáma római számmal
- Dátum: A döntő ideje
- Győztes: A győztes csapat neve
- Eredmény: A döntő végeredménye (győztes pontjai-vesztes pontjai)
- Vesztes: A vesztes csapat neve
- Helyszín: A döntő helyszíne (stadion)
- VárosÁllam: A döntő városa és a város állama vesszővel elválasztva
- Nézőszám: A nézők száma

1. Készítsen **konzolos alkalmazást** a következő feladatok megoldására, melynek projektjét SuperBowl néven mentse el!
2. Forráskódjában tegye elérhetővé a java.txt vagy a csharp.txt állományból a RomaiSorszam osztályt definiáló kódrészletet! **class Program** elé kell bemásolni
3. Olvassa be a SuperBowl.txt állományban lévő adatokat, és tárolja el egy olyan adatszerkezetben, ami a további feladatok megoldására alkalmas! Az állományban legfeljebb 100 sor lehet. A sorszámok tárolásánál felhasználhatja az előző feladatban elérhetővé tett osztályt is.
4. Határozza meg és írja ki a képernyőre, hogy hány döntő adatai találhatóak a forrásállományban!

¹ Forrás: https://hu.wikipedia.org/wiki/Super_Bowl_győztesek_listája

5. Határozza meg és írja ki a képernyőre, hogy a döntők során mennyi volt az átlagos pontkülönbség a két csapat között! Az átlagot egy tizedesjegyre kerekítve jelenítse meg!
6. Keresse meg, hogy melyik döntőn volt a legtöbb néző! Feltételezheti, hogy nem alakult ki holtverseny. A döntő adatait jelenítse meg a minta szerint! A döntő sorszámát arab sorszámmal jelenítse meg a `RomaiSorszam` osztálypéldány megfelelő kódtagjának a felhasználásával! A csapatnevek mögött jelenjen meg a döntőben elért pontszámuk is!
7. Készítsen új UTF-8 kódolású állományt `SuperBowlNew.txt` néven a kiadott minta szerint! Az állomány első sora a mezőneveket tartalmazza! A döntők számozásánál arab sorszámmal használjon! A győztes és vesztes csapatok nevei után zárójelben jelenítse meg, hogy a csapat hányadik Super Bowl szereplése volt a mérkőzés! (Tipp a megoldáshoz: ezt a részfeladatot úgy is megoldhatja, hogy elsőként az állomány sorait egy szöveges típusú listában tárolja, és az új sor hozzáadása előtt megszámolja, hogy az új sorba írandó csapatok nevei hányszor szerepeltek korábban ebben a listában, majd a teljes lista elkészülte után írja annak tartalmát az állományba.)

Konzolos alkalmazás minta:

```
4. feladat: Döntők száma: 55
5. feladat: Átlagos pontkülönbség: 14,1
6. feladat: Legmagasabb nézőszám a döntők során:
    Sorszám (dátum): 14. (1980.01.20.)
    Győztes csapat: Pittsburgh Steelers, szerzett pontok: 31
    Vesztes csapat: Los Angeles Rams, szerzett pontok: 19
    Helyszín: Rose Bowl
    Város, állam: Pasadena, Kalifornia
    Nézőszám: 103985
```

SuperBowlNew.txt állomány minta:

```
Ssz;Dátum;Győztes;Eredmény;Vesztes;Nézőszám
1.;1967.01.15.;Green Bay Packers (1);35-10;Kansas City Chiefs (1);61946
2.;1968.01.14.;Green Bay Packers (2);33-14;Oakland Raiders (1);75546
3.;1969.01.12.;New York Jets (1);16-7;Baltimore Colts (1);75389
4.;1970.01.11.;Kansas City Chiefs (2);23-7;Minnesota Vikings (1);80562
5.;1971.01.17.;Baltimore Colts (2);16-13;Dallas Cowboys (1);79204
6.;1972.01.16.;Dallas Cowboys (2);24-3;Miami Dolphins (1);81023
...
52.;2018.02.04.;Philadelphia Eagles (3);41-33;New England Patriots (10);67612
53.;2019.02.03.;New England Patriots (11);13-3;Los Angeles Rams (2);70081
54.;2020.02.02.;Kansas City Chiefs (3);31-20;San Francisco 49ers (7);62417
55.;2021.02.07.;Tampa Bay Buccaneers (2);31-9;Kansas City Chiefs (4);24835
```

Megoldás:

```
using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
/*1. Készítsen konzolos alkalmazást a következő feladatok megoldására, melynek projektjét
SuperBowl néven mentse el! */
namespace SuperBowl
{
```

```

/*Ssz A döntő sorszama római számmal
Datum: A döntő ideje
Gyoztes: A gyoztes csapat neve
Eredmeny: A döntő végeredménye (gyoztes pontjai-vesztes pontjai)
Vesztes: A vesztes csapat neve
Helyszin: A döntő helyszíne (stadion)
VarosAllam: A döntő városa és a város állama vesszővel elválasztva
Nézőszám: A nézők száma
*
* Ssz;Datum;Gyoztes;Eredmeny;Vesztes;Helyszin;VarosAllam;Nézőszám
I.;1967.01.15.;Green Bay Packers;35-10;Kansas City Chiefs;Los Angeles Memorial Coliseum;Los
Angeles, Kalifornia;61946*/

```

```

class super
{
    public RomaiSorszam ssz { get; private set; }
    public string datum { get; set; }
    public string gyoztes { get; set; }
    public string eredmeny { get; set; }
    public string vesztes { get; set; }
    public string helyszin { get; set; }
    public string varosallam { get; set; }
    public int nezoszam { get; set; }
    public int gyoztespont { get; set; }
    public int vesztespont { get; set; }
    public super(string sor)
    {
        string[] sorelemek = sor.Split(';');
        this.ssz = new RomaiSorszam(sorelemek[0]);
        this.datum = sorelemek[1];
        this.gyoztes =sorelemek[2];
        this.eredmeny =sorelemek[3];
        gyoztespont= Convert.ToInt32(eredmeny.Split('-')[0]);
        vesztespont= Convert.ToInt32(eredmeny.Split('-')[1]);
        this.vesztes =sorelemek[4];
        this.helyszin =sorelemek[5];
        this.varosallam =sorelemek[6];
        this.nezoszam = Convert.ToInt32(sorelemek[7]);
    }
}

```

```

/*2. Forráskódjában tegye elérhetővé a java.txt vagy a csharp.txt állományból a
* RomaiSorszam osztályt definiáló kódrészletet! */

```

```

class RomaiSorszam
{
    public string RomaiSsz { get; private set; }

    private static Dictionary<char, int> RomaiMap = new Dictionary<char, int>()
    {
        {'I', 1}, {'V', 5}, {'X', 10}, {'L', 50}, {'C', 100}, {'D', 500}, {'M', 1000}
    };

    public string ArabSsz
    {
        get
        {
            int ertek = 0;
            string romaiSzam = RomaiSsz.TrimEnd('.');
            for (int i = 0; i < romaiSzam.Length; i++)
            {
                if (i + 1 < romaiSzam.Length &&
                    RomaiMap[romaiSzam[i]] < RomaiMap[romaiSzam[i + 1]])
                {
                    ertek -= RomaiMap[romaiSzam[i]];
                }
                else
                {
                    ertek += RomaiMap[romaiSzam[i]];
                }
            }
            return $"{ertek}.";
        }
    }
}

```

```

    }
}

public RomaiSorszam(string romaiSsz)
{
    RomaiSsz = romaiSsz.ToUpper();
}
}

class Program
{
    public static List<super> adatok = new List<super>();
    static void Main(string[] args)
    {
        /*3.  Olvassa be a SuperBowl.txt állományban lévő adatokat, és tárolja el egy
        olyan adatszerkezetben,
        * ami a további feladatok megoldására alkalmas!
        * Az állományban legfeljebb 100 sor lehet. A sorszámok tárolásánál
        felhasználhatja az előző feladatban
        * elérhetővé tett osztályt is. */
        string[] fajl = File.ReadAllLines("SuperBowl.txt", Encoding.UTF8);
        for (int i = 1; i < fajl.Length; i++)
        {
            adatok.Add(new super(fajl[i])); //listához adom az adatokat
        }
        int merkozesekszama = adatok.Count();
        /*4.  Határozza meg és írja ki a képernyőre, hogy hány döntő adatai találhatóak a
        forrásállományban!
        Console.WriteLine("4. feladat: Döntők száma: {0}", merkozesekszama);
        /*5.  Határozza meg és írja ki a képernyőre, hogy a döntők során
        //mennyi volt az átlagos pontkülönbség a két csapat között!
        //Az átlagot egy tizedesjegyre kerekítve jelenítse meg!
        double atlag = 0;
        for (int i = 0; i < merkozesekszama; i++)
        {
            atlag+=adatok[i].gyoztespont- adatok[i].vesztespont;
        }
        Console.WriteLine("5. feladat: Átlagos pontkülönbség:
        {0}", Math.Round(atlag/merkozesekszama, 1));
        /* 6.Keresse meg, hogy melyik döntőn volt a legtöbb néző!
        * Feltételezheti, hogy nem alakult ki holtverseny.
        * A döntő adatait jelenítse meg a minta szerint!
        * A döntő sorszámát arab sorszámmal jelenítse meg
        * a RomaiSorszam osztálypéldány megfelelő kódtagjának a felhasználásával!
        * A csapatnevek mögött jelenjen meg a döntőben elért pontszámuk is !*/
        List<super> sorbarendeztett = adatok.OrderByDescending(x=>x.nezoszam).ToList();
        //sorbarendezem az adatokat nézőszám szerint csökkenő sorrendbe
        Console.WriteLine("6. feladat: Legmagasabb nézőszám a döntők során"); //az első
        (0.) adatot kell kiírni
        Console.WriteLine("\tSorszám (dátum): {0} ({1})", sorbarendeztett[0].ssz.ArabSsz,
        sorbarendeztett[0].datum);
        Console.WriteLine("\tGyőztes csapat: {0}, szerzett pontok:
        {1}", sorbarendeztett[0].gyoztes, sorbarendeztett[0].gyoztespont);
        Console.WriteLine("\tVesztes csapat: {0}, szerzett pontok:
        {1}", sorbarendeztett[0].vesztes, sorbarendeztett[0].vesztespont);
        Console.WriteLine("\tHelyszín: {0}", sorbarendeztett[0].helyszin);
        Console.WriteLine("\tVáros, állam: {0}", sorbarendeztett[0].varosallam);
        Console.WriteLine("\tNézőszám: {0}", sorbarendeztett[0].nezoszam);

        /*7.  Készítsen új UTF-8 kódolású állományt SuperBowlNew.txt néven a kiadott
        minta szerint!
        * Az állomány első sora a mezőneveket tartalmazza!
        * A döntők számozásánál arab sorszámokat használjon!
        * A győztes és vesztes csapatok nevei után zárójelben jelenítse meg,
        * hogy a csapat hányadik Super Bowl szereplése volt a mérkőzés!
        * (Tipp a megoldáshoz: ezt a részfeladatot úgy is megoldhatja,
        * hogy elsőként az állomány sorait egy szöveges típusú listában tárolja,
        * és az új sor hozzáadása előtt megszámozolja, hogy az új sorba írandó
        * csapatok nevei hányszor szerepeltek korábban ebben a listában,

```

```

        * majd a teljes lista elkészülte után írja annak tartalmát az állományba.) */

        FileStream fnev = new FileStream("SuperBowlNew.txt",
        FileMode.Create); //létrehozásra megnyitjuk a fájlt
        StreamWriter fajlbairo = new StreamWriter(fnev,
        System.Text.Encoding.UTF8); //fájlbaíráshoz
        List<string> ki = new List<string>(); //listában gyűjtjük az adatokat
        ki.Add("Ssz;Dátum;Győztes;Eredmény;Vesztes;Nézőszám"); //első sor
        foreach (var d in adatok) //adatokon végigmegyünk egyesével
        {
            int gyoztesDb = 0;
            int vesztesDb = 0;
            foreach (var sor in ki) //gyűjtött adatokon végigmegyünk
            {
                if (sor.Contains(d.gyoztes)) gyoztesDb++; //ha szerepelt már korábban a
                győztes csapat
                if (sor.Contains(d.vesztes)) vesztesDb++; //ha szerepelt már korábban a
                vesztes csapat
            }
            string gyoztescsapat = d.gyoztes + "(" + (gyoztesDb + 1).ToString() + ")";
            string vesztescsapat = d.vesztes + "(" + (vesztesDb + 1).ToString() + ")";
            ki.Add(d.ssz.ArabSsz + ";" + d.datum + ";" + gyoztescsapat + ";" + d.eredmeny
            + ";" + vesztescsapat + ";" + d.nezoszam);
        }
        foreach (var d in ki) fajlbairo.WriteLine(d);
        fajlbairo.Close(); //fájl lezárása
        fnev.Close();

        Console.ReadKey();
    }
}

```