

Programozás

Snooker világranglista

A snooker játékban különböző szempontok szerint világranglistákat vezetnek. Ebben a feladatban egy programot kell készítenie, melyben a 2019. 10. 20-án aktuális² pénzdíjas világranglistával kell dolgoznia.

A megoldás során vegye figyelembe a következőket:

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

A `snooker.txt` UTF-8 kódolású forrásállomány soraiban a következő adatokat találja:

```
Helyezés;Nev;Ország;Nyeremeny
52;Akani Sunny;Thaiföld;118500
7;Allen Mark;Észak-Írország;681000
72;Anda Zhang;Kína;44750
76;Astley John;Anglia;40000
73;Baird Sam;Anglia;44750
...
```

Az állomány sorai a versenyzők neve szerinti ábécérendben tárolja a versenyző helyezését a ranglistán, nevét, országát és az elmúlt időszakban elnyert pénzdíjak összegét angol fontban. Az állomány első sora az adatok fejlécét tartalmazza. Az adatokat pontosvesszővel választottuk el.

1. Készítsen konzolalkalmazást (projektet) a következő feladatok megoldásához, amelynek forráskódját `Snooker` néven mentse el!
2. Olvassa be a `snooker.txt` állomány sorait és tárolja az adatokat egy olyan összetett adatszerkezetben (pl. vektor, lista stb.), amely használatával a további feladatok megoldhatók! Ügyeljen arra, hogy az állomány első sora a mezőneveket tartalmazza!
3. Határozza meg és írja ki a képernyőre a minta szerint, hogy hány versenyző szerepel a világranglistán!
4. Határozza meg, hogy a ranglistán szereplő versenyzők átlagosan mekkora bevételre tettek szert az elmúlt időszakban! Az eredményt két tizedesjegyre kerekítve jelenítse meg a minta szerint!

5. Határozza meg és írja ki a képernyőre a minta szerint a legjobban kereső **kínai** játékos adatait! Feltételezheti, hogy legalább egy kínai versenyző volt, és nem alakult ki holtverseny közöttük. A nyeresmény összegét forintba jelenítse meg! Az átszámoláshoz 380 Ft-os angol font árfolyammal dolgozzon!
6. Határozza meg, hogy a világranglistán található-e **norvég** játékos!
7. Készítsen statisztikát országok szerinti csoportosításban a versenyzők számáról! Csak azok az országok jelenjenek meg a minta szerint, amelyekből több mint négy versenyző szerepel a világranglistán!

Minta kimenet:

```

3. feladat: A világranglistán 100 versenyző szerepel
4. feladat: A versenyzők átlagosan 183373,50 fontot kerestek
5. feladat: A legjobban kereső kínai versenyző:
    Helyezés: 17
    Név: Yan Bingtao
    Ország: Kína
    Nyeresmény összege: 108 300 000 Ft
6. feladat: A versenyzők között van norvég versenyző.
7. feladat: Statisztika
    Kína - 20 fő
    Anglia - 47 fő
    Wales - 10 fő
    Skócia - 6 fő

```

Megoldás:

```

using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace snooker
{
    class Versenyzok
    {
        public Versenyzok(string sor)
        {
            string[] soreslemek = sor.Split(';');
            this.Helyezes = Convert.ToInt32(soreslemek[0]);
            this.Nev = soreslemek[1];
            this.Orszag = soreslemek[2];
            this.Nyeresmeny = Convert.ToInt32(soreslemek[3]);
        }
        //Helyezes;Nev;Orszag;Nyeresmeny
        public int Helyezes { get; set; }
        public string Nev { get; set; }
        public string Orszag { get; set; }
        public int Nyeresmeny { get; set; }
    }
    class Program
    {
        public static List<Versenyzok> versenyzoadatok = new List<Versenyzok>();
        static void Main(string[] args)
        {
            StreamReader olvas = new StreamReader("snooker.txt", Encoding.UTF8);
            string fejléc = olvas.ReadLine();//ha van fejléc
            while (!olvas.EndOfStream)
            {
                versenyzoadatok.Add(new Versenyzok(olvas.ReadLine()));
            }
        }
    }
}

```

```

int i;//ciklusváltozó
int versenyzokszama = versenyzoadatok.Count;
/*Console.WriteLine("Helyezés   Nev                               Ország           Nyeremeny");
for (i = 0; i < versenyzokszama; i++)
{
    Console.WriteLine("{0,-4}{1,-25}{2,-25}{3}",
        versenyzoadatok[i].Helyezés, versenyzoadatok[i].Nev, versenyzoadatok[i].Ország,
        versenyzoadatok[i].Nyeremeny);
}*/
Console.WriteLine("3. feladat: A világranglistán {0} versenyző szerepel",
    versenyzokszama);
double osszesnyeremeny = 0;
for (i = 0; i < versenyzokszama; i++)
{
    osszesnyeremeny += versenyzoadatok[i].Nyeremeny;
}
Console.WriteLine("4. feladat: A versenyzők átlagosan {0:### ##} fontot
kerestek",Math.Round(osszesnyeremeny/ versenyzokszama, 2));
double max = versenyzoadatok[0].Nyeremeny;
int maxi = 0;
for (i = 1; i < versenyzokszama; i++)
{
    if (versenyzoadatok[i].Ország == "Kína")
    {
        if (versenyzoadatok[i].Nyeremeny > max)
        {
            max = versenyzoadatok[i].Nyeremeny;
            maxi = i;
        }
    }
}
Console.WriteLine("5. feladat: A legjobban kereső kínai versenyző:");
Console.WriteLine("\tNév: {0}", versenyzoadatok[maxi].Helyezés);
Console.WriteLine("\tOrszág: {0}", versenyzoadatok[maxi].Ország);
Console.WriteLine("\tNyeremény összege: {0:### ##} Ft",
    versenyzoadatok[maxi].Nyeremeny * 380);
bool van = false;
i = 0;
while(i< versenyzokszama && versenyzoadatok[i].Ország != "Norvégia")
{
    i++;
}
van = i < versenyzokszama ? true : false;
if (van)
{
    Console.WriteLine("6. feladat: A versenyzők között van norvég versenyző {0}. adat",i);
}
else
{
    Console.WriteLine("6. feladat: A versenyzők között nincs norvég versenyző");
}
Console.WriteLine("7. feladat: Statisztika ");
versenyzoadatok.GroupBy(x => x.Ország).Where(x => x.Count() > 4).ToList().ForEach(x =>
    Console.WriteLine("\t{0}-{1} fő", x.Key, x.Count()));
/*Console.WriteLine("Sorbarendezett adatok nyeremény szerint csökkenő sorrendbe");
List<Versenyzok> sorbarendezett = versenyzoadatok.OrderByDescending(x =>
    x.Nyeremeny).ToList();
for(i=0;i<sorted.Count;i++)
{
    Console.WriteLine("{0,-4}{1,-25}{2,-25}{3}",
        sorted[i].Helyezés, sorted[i].Nev, sorted[i].Ország,
        sorted[i].Nyeremeny);
}*/
Console.ReadKey();
}
}
}

```