

## Magyarország hegyei

A következő feladatban Magyarország legmagasabb hegyeinek adataival kell dolgoznia. 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) HegyekMo azonosítóval!

2. Az UTF-8 kódolású hegyekMo.txt állomány Magyarország legmagasabb hegyeinek adatait tartalmazza a következő minta szerint (forrás: wikipedia.hu):

Hegycsúcs neve;Hegység;Magasság

Ágasvár;Mátra;789

Bálvány;Bükk-vidék;956

Büszkés-hegy;Bükk-vidék;952

Cserepes-kő;Bükk-vidék;823

Az állományban a hegycsúcs nevét, a hegység megnevezését és a hegycsúcs magasságát (méter) tároltuk. Az adatokat pontosvessző választja el. Olvassa be a hegyekMo.txt állományban lévő adatokat és tárolja el egy olyan adatszerkezetben, ami a további feladatok megoldására alkalmas! A fájlban legfeljebb 1000 sor lehet! Ügyeljen arra, hogy az állomány első sora az adatok fejlécét tartalmazza!

3. Határozza meg és írja ki a képernyőre a minta szerint, hogy hány hegy található az állományban!

4. Határozza meg és írja ki a képernyőre a minta szerint az állományban található hegyek átlagmagasságát!

5. Határozza meg és írja ki a képernyőre a minta szerint a legmagasabb hegy adatait! Feltételezheti, hogy nem alakult ki holtverseny.

6. Kérjen be a felhasználótól egy magasságértéket! A bevitt adatot nem kell ellenőriznie. Döntse el, hogy található-e a megadott értéknél magasabb hegycsúcs! A keresést ne folytassa, ha a választ meg tudja adni! A képernyőre írást a minta szerint végezze!

7. Határozza meg és írja ki a képernyőre a minta szerint azoknak a hegycsúcsoknak a számát, amelyek 3000 lábnál magasabbak! Az átváltáshoz az  $1\text{ m} = 3.280839895\text{ láb}$  értékkel dolgozzon!

8. Készítsen statisztikát hegységek szerint a hegycsúcsok számáról! A megoldást úgy készítse el, hogy az inputállományba később más hegységek is bekerülhetnek! A képernyőre írást a minta szerint végezze!

9. A bukk-vidék.txt állományba írja ki azoknak a hegycsúcsoknak nevét és magasságát a minta szerint, amelyek a Bükk-vidéken magasodnak! Az állomány első sora az adatok fejlécét tartalmazza! A magasságokat egy tizedesjegyre kerekítve, lábban kell kiírnia. Az átváltáshoz az  $1\text{ m} = 3.280839895\text{ láb}$  értékkel dolgozzon!

### Minta:

```
3. feladat: Hegycsúcsok száma: 100 db
4. feladat: Hegycsúcsok átlagos magassága: 872,66 m
5. feladat: A legmagasabb hegycsúcs adatai:
    Név: Kékes
    Hegység: Mátra
    Magasság: 1014 m
6. feladat: Kérek egy magasságot: 1000
    Van 1000 m-nél magasabb hegycsúcs a Kékes
7. feladat: 3000 lábnál magasabb hegycsúcsok száma: 29
8. feladat: Hegység statisztika
    Mátra: 25 db
    Bükk-vidék: 60 db
    Börzsöny: 9 db
    Zempléni-hegység: 4 db
    Kőszegi-hegység: 2 db
9. feladat: bukk-vidék.txt
    Bálvány;3136,5
    Büszkés-hegy;3123,4
    Cserepes-kő;2700,1
    Csikorgó;2552,5
    Esztea-fő;2614,8
    Fekete-Sár-bérc;3051,2
    ...
```

### Megoldás:

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

namespace hegyekmo
{
    class Program
    {
        /*2. Az UTF-8 kódolású hegyekMo.txt állomány Magyarország legmagasabb hegyeinek adatait
        tartalmazza
        * a következő minta szerint (forrás: wikipedia.hu):
        Hegycsúcs neve;Hegység;Magasság
        Ágasvár;Mátra;789
        Bálvány;Bükk-vidék;956
        Büszkés-hegy;Bükk-vidék;952
        Cserepes-kő;Bükk-vidék;823
        Az állományban a hegycsúcs nevét, a hegység megnevezését és a hegycsúcs magasságát (méter) tároltuk.
        Az adatokat pontosvessző választja el. Olvassa be a hegyekMo.txt állományban lévő adatokat és
        tárolja el egy olyan adatszerkezetben,
        ami a további feladatok megoldására alkalmas! A fájlban legfeljebb 1000 sor lehet!
        Ügyeljen arra, hogy az állomány első sora az adatok fejlécét tartalmazza!
        */

        struct hegyek//Készítsen összetett változót az adatok tárolására!
        {
            public string hegycsucsnev;
            public string hegysegnev;
            public int magassag;
        }
        static hegyek[] adatok = new hegyek[1000];//Az állományban legfeljebb 1000 sor lehet.
        static void Main(string[] args)
        {
            string[] fajlbol = File.ReadAllLines("hegyekMo.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].hegycsucsnev = egysordarabolva[0];
                adatok[sorokszama].hegysegnev = egysordarabolva[1];
                adatok[sorokszama].magassag =Convert.ToInt32(egysordarabolva[2]);
                sorokszama++;
            }
            int hegyekszama = sorokszama;

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

            Console.WriteLine("Hegycsúcs neve           Hegység           Magasság");
            for (i = 0; i < hegyekszama; i++)
            {
                Console.WriteLine("{0,-25} {1,-20} {2}", adatok[i].hegycsucsnev,
                adatok[i].hegysegnev, adatok[i].magassag);
            }*/
            //3. Határozza meg és írja ki a képernyőre a minta szerint, hogy hány hegy található az
            állományban!
            Console.WriteLine("3. feladat: Hegycsúcsok száma: {0} db", hegyekszama);
            //4.Határozza meg és írja ki a képernyőre a minta szerint az állományban található
            hegyek átlagmagasságát!
            //összegzés tétele
            double atlagmagassag = 0;
            for (i = 0; i < hegyekszama; i++)
            {
                atlagmagassag += adatok[i].magassag;
            }
        }
    }
}
```

```

Console.WriteLine("4. feladat: Hegycsúcsok átlagos magassága: {0} m",
atlagmagassag/hegyekszama);

//5. Határozza meg és írja ki a képernyőre a minta szerint a legmagasabb hegy adatait!
//Feltételezheti, hogy nem alakult ki holtverseny.
//maximumkiválasztás tétele
int max = adatok[0].magassag;
int maxi = 0;
for (i = 0; i < hegyekszama; i++)
{
    if (adatok[i].magassag > max)
    {
        max = adatok[i].magassag;
        maxi = i;
    }
}

Console.WriteLine("5. feladat: A legmagasabb hegycsúcs adatai:\n\tNév: {0}\n\tHegység:
{1}\n\tMagasság: {2} m", adatok[maxi].hegycsucsnev, adatok[maxi].hegysegnev, adatok[maxi].magassag);
/*6. Kérjen be a felhasználótól egy magasságértéket!
* A bevitt adatot nem kell ellenőriznie.
* Döntse el, hogy a található-e a megadott értéknél magasabb hegycsúcs!
* A keresést ne folytassa, ha a választ meg tudja adni!
* A képernyőre írást a minta szerint végezze!*/
//keresés tétele
int keresetmagassag;
Console.Write("6. feladat: Kérek egy magasságot: ");
keresetmagassag = Convert.ToInt32(Console.ReadLine());
bool van;
i = 0;
while ((i<hegyekszama) && !(adatok[i].magassag > keresetmagassag))
{
    i++;
}
van = i < hegyekszama ? true : false;
if (van)
{
    Console.WriteLine("\tVan {0} m-nél magasabb hegycsúcs a
{1}", keresetmagassag, adatok[i].hegycsucsnev);
}

else
{
    Console.WriteLine("\tNincs {0} m-nél magasabb hegycsúcs", keresetmagassag);
}

//7.Határozza meg és írja ki a képernyőre a minta szerint azoknak a hegycsúcsoknak a
számát,
//amelyek 3000 lábnál magasabbak!Az átváltáshoz az 1 m = 3.280839895 láb értékkel
dolgozzon!
//megszámolás tétele
int db = 0;
for (i = 0; i < hegyekszama; i++)
{
    if (adatok[i].magassag* 3.280839895 > 3000)
    {
        db++;
    }
}
Console.WriteLine("7. feladat: 3000 lábnál magasabb hegycsúcsok száma: {0}", db);
/*8. Készítsen statisztikát hegységek szerint a hegycsúcsok számáról!
* A megoldást úgy készítse el, hogy az inputállományba később más hegységek is
bekerülhetnek!
* A képernyőre írást a minta szerint végezze! */
//adott egy sorozat, határozzuk meg hány különböző eleme van és gyűjtsük ki egy tömbbe
int kulonbozoelemekszama = 0;
string[] hegysegnevek = new string[100];
int[] hegysegekdb = new int[100];
for (i = 0; i < 100; i++) hegysegekdb[i] = 0;//adatok nullázása

```

```

for (i = 0; i < hegyekszama; i++)
{
    j = 0;
    while ((j <= kulonbozoelemekszama) && (adatok[i].hegysegnev != hegysegnevek[j]))
    {
        j++;
    }
    if (j > kulonbozoelemekszama)
    {
        kulonbozoelemekszama++;
        hegysegnevek[kulonbozoelemekszama] = adatok[i].hegysegnev;
    }
}
//megszámolás tétele
for (i = 0; i < hegyekszama; i++)
{
    for (int k = 1; k <= kulonbozoelemekszama; k++)
    {
        if (hegysegnevek[k] == adatok[i].hegysegnev) hegysegekdb[k]++;
    }
}
Console.WriteLine("8. feladat: Hegység statisztika");
for (i = 1; i <= kulonbozoelemekszama; i++)
    Console.WriteLine("\t{0}: {1} db ", hegysegnevek[i], hegysegekdb[i]);
/*9. A bukk-vidék.txt állományba írja ki azoknak a hegycsúcsoknak nevét és magasságát a
minta szerint,
* amelyek a Bükk-vidéken magasodnak! Az állomány első sora az adatok fejlécét
tartalmazza!
* A magasságokat egy tizedesjegyre kerekítve, lábban kell kiírnia.
* Az átváltáshoz az 1 m = 3.280839895 láb értékkel dolgozzon! */
Console.WriteLine("9. feladat: bukk-vidék.txt");
FileStream fnev = new FileStream("bukk-vidék.txt", FileMode.Create);
StreamWriter fajlbairo = new StreamWriter(fnev);
double magassaglab;
fajlbairo.WriteLine("Hegycsúcs neve;Magasság láb");
for (i = 1; i <= hegyekszama; i++)
{
    if(adatok[i].hegysegnev== "Bükk-vidék")
    {
        magassaglab =Math.Round((adatok[i].magassag * 3.280839895),1);
        fajlbairo.Write("{0};", adatok[i].hegycsucsnev);
        fajlbairo.Write("{0}", magassaglab);
        Console.Write("\t{0};", adatok[i].hegycsucsnev);
        Console.WriteLine("{0}", magassaglab);
        fajlbairo.WriteLine("\n");//sortörés
    }
}
fajlbairo.Close();
fnev.Close();

Console.ReadKey();

}
}
}

```