

Az Amerikai Egyesült Államok elnökei

Az Amerikai Egyesült Államok elnökeinek adatait tartalmazza az *elnokok.txt* állomány. Az állomány minden sora egy-egy elnök adatait tárolja. Az adatokat | karakter választja el egymástól.

Minden kiírást igénylő feladat elé írja ki a feladat sorszámát!

Pl..

George Washington|1789|1797|Pártonkívüli|1732|1799

A sorokban lévő adatok rendre a következők:

- Az elnök neve. Pl.: George Washington
- Elnökségének kezdete Pl.: 1789
- Elnökségének vége Pl.: 1797
- Pártja Pl.: Pártonkívüli
- Születési éve Pl.: 1732
- Halálának éve Pl.: 1799 Ha az elnök még él, a mező értéke 0.

1. Készítsen programot a következő feladatok megoldására, amelynek a forráskódját *elnokok* néven mentse el!
2. Olvassa be az *elnokok.txt* állományban lévő adatokat és tárolja el úgy, hogy a további feladatok megoldására alkalmasak legyenek! Írja ki a képernyőre, hogy hány elnök adatait tartalmazza az állomány!
3. Hozzon létre egy függvényt, amely paraméterként megkapja az elnök születési évét, valamint beiktatásának évét, és visszaadja, hogy hány évesen lett elnök!
4. Írja ki annak az elnöknek a nevét, születési évét és beiktatásának évét, aki a legfiatalabb volt elnökségének kezdetén! (Lehetőleg a 3. feladat függvényének segítségével oldja meg a feladatot!)
5. Hozzon létre egy függvényt Vege néven, amely paraméterként megkapja az elnök halálának évét, valamint mandátumának végét, és igaz értéket ad vissza, amennyiben az elnök mandátuma a halálával ért véget, hamis értéket különben. (Az elnökségének éve megegyezik a halálának évével.)
6. Írja ki, hogy hány olyan elnöke volt eddig az Amerikai Egyesült Államoknak, akinek mandátuma a halálával ért véget!
7. Egy elnök maximum 3 cikluson át lehet folyamatosan hatalmon. Volt-e olyan demokrata elnök, aki kitöltötte a 3 ciklust? (Egy elnöki ciklus 4 évig tart.) Ha igen, mi volt a neve és mettől meddig volt elnök?
8. Hozza létre a republikanus.txt állományt! Írja ki az állományba az összes republikánus elnök adatát! Az állomány szerkezete egyezzen meg az eredeti állományéval!

Minta a kiírást igénylő feladatokhoz (a mintában szereplő adatok helytelenek):

2. feladat.

Az Amerikai Egyesült Államok 50 elnöke

4. feladat

A legfiatalabb elnök neve: George Washington

A legfiatalabb elnök született: 1789

A legfiatalabb elnököt beiktatták: 1932

6. feladat

12 elnök hunyt el

7. feladat

3 ciklust töltött ki: George Washington

Minta:

```
2. feladat.
    Az Amerikai Egyesült Államok 44 elnöke
4. feladat
    A legfiatalabb elnök neve: Theodore Roosevelt
    A legfiatalabb elnök született: 1858
    A legfiatalabb elnököt beiktatták: 1901
6. feladat
    9 elnök hunyt el
7. feladat
    3 ciklust töltött ki: George Washington
```

Megoldás:

```
using System;
```

```
using System.IO;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading.Tasks;
```

```
namespace elnokok
```

```
{
    class Program
    {
```

```
        /*Minden kiírást igénylő feladat elé írja ki a feladat sorszámát!
```

```
Pl..
```

```
George Washington|1789|1797|Pártonkívüli|1732|1799
```

```
A sorokban lévő adatok rendre a következők:
```

- Az elnök neve. Pl.: George Washington
- Elnökségének kezdete Pl.: 1789
- Elnökségének vége Pl.: 1797
- Pártja Pl.: Pártonkívüli
- Születési éve Pl.: 1732
- Halálának éve Pl.: 1799 Ha az elnök még él, a mező értéke 0.

1. Készítsen programot a következő feladatok megoldására, amelynek a forráskódját elnokok néven mentse el!

2. Olvassa be az elnokok.txt állományban lévő adatokat és tárolja el úgy, hogy a további feladatok megoldására alkalmasak legyenek! Írja ki a képernyőre, hogy hány elnök adatait tartalmazza az állomány!

```
*/
```

```
    struct elnok
```

```
    {
        public string nev;
        public int kezdet;
        public int veg;
        public string part;
        public int szul;
        public int halal;
    }
```

```
    static elnok[] adatok = new elnok[100];
```

```
    /*3. Hozzon létre egy függvényt,
```

```

    * amely paraméterként megkapja az elnök születési évét,
    * valamit beiktatásának évét, és visszaadja, hogy hány évesen lett elnök!*/
static int sajátfuggveny(int szez,int beev)
{
    int hanyevesenlettelnok = beev - szez;
    return hanyevesenlettelnok;
}
/*5. Hozzon létre egy függvényt Vege néven, amely paraméterként megkapja az elnök halálának
évét,
    * valamit mandátumának végét, és igaz értéket ad vissza,
    * amennyiben az elnök mandátuma a halálával ért véget, hamis értéket különben.
    * (Az elnökségének éve megegyezik a halálának évével.)*/
static bool Vege(int halev, int mandveg)
{
    if (halev==mandveg)
        return true;
    else
        return false;
}
static void Main(string[] args)
{
    /*2. Olvassa be az elnokok.txt állományban lévő adatokat és tárolja el úgy,
    * hogy a további feladatok megoldására alkalmasak legyenek!
    * Írja ki a képernyőre, hogy hány elnök adatait tartalmazza az állomány!*/
    string[] fajlbol = File.ReadAllLines("elnokok.txt");

    int sorokszama = 0;//sorok száma a fájlban
    int i;//ciklusváltozó
    for (int k = 0; k < fajlbol.Count(); k++)
    {
        string[] egysordarabolva = fajlbol[k].Split('|');
        adatok[sorokszama].nev = egysordarabolva[0];
        adatok[sorokszama].kezdet = Convert.ToInt32(egysordarabolva[1]);
        adatok[sorokszama].veg = Convert.ToInt32(egysordarabolva[2]);
        adatok[sorokszama].part = egysordarabolva[3];
        adatok[sorokszama].szul = Convert.ToInt32(egysordarabolva[4]);
        adatok[sorokszama].halal = Convert.ToInt32(egysordarabolva[5]);
        sorokszama++;
    }

    Console.WriteLine("Az elnökök listája fájlból");
    int elnokokszama = sorokszama;
    Console.WriteLine("név                kezdet                vég                párt
szül    halál ");//adatok kiírása táblázatosan (nem volt feladat)
    for (i = 0; i < elnokokszama; i++)
    {
        Console.WriteLine("{0,-30} {1,-10} {2,-10} {3,-20} {4,-10} {5}", adatok[i].nev,
        adatok[i].kezdet, adatok[i].veg, adatok[i].part, adatok[i].szul, adatok[i].halal);
    }
    Console.WriteLine("2. feladat.\n\tAz Amerikai Egyesült Államok {0} elnöke",
    elnokokszama);

    /*4. Írja ki annak az elnöknek a nevét, születési évét és beiktatásának évét,
    * aki a legfiatalabb volt elnökségének kezdetén! */
    int legfiatalabb = sajátfuggveny(adatok[0].szul, adatok[0].kezdet);
    int mini = 0;
    for (i = 1; i < elnokokszama; i++)
    {
        if(legfiatalabb > sajátfuggveny(adatok[i].szul, adatok[i].kezdet))
        {
            legfiatalabb = sajátfuggveny(adatok[i].szul, adatok[i].kezdet);
            mini = i;
        }
    }
    Console.WriteLine("4. feladat");
    Console.WriteLine("\t A legfiatalabb elnök neve: {0}", adatok[mini].nev);
    Console.WriteLine("\t A legfiatalabb elnök született: {0}", adatok[mini].szul);
    Console.WriteLine("\t A legfiatalabb elnököt beiktatták: {0}", adatok[mini].kezdet);

```

```

/*6. Írja ki, hogy hány olyan elnöke volt eddig az Amerikai Egyesült Államoknak,
 * akinek mandátuma a halálával ért véget!*/
int db = 0;
for (i = 0; i < elnokokszama; i++)
{
    if(Vege(adatok[i].halal, adatok[i].veg))
    {
        db++;
    }
}
Console.WriteLine("6. feladat\n\t{0} elnök hunyt el",db);

/*7. Egy elnök maximum 3 cikluson át lehet folyamatosan hatalmon.
 * Volt-e olyan demokrata elnök, aki kitöltötte a 3 ciklust?
 * (Egy elnöki ciklus 4 évig tart.) Ha igen, mi volt a neve és mettől meddig volt
elnök?*/
i = 0;
while(i<elnokokszama && adatok[i].veg- adatok[i].kezdet == 12)
{
    i++;
}
if (i < elnokokszama)
{
    Console.WriteLine("7. feladat\n\t3 ciklust töltött ki: {0}", adatok[i].nev);
}
else
{
    Console.WriteLine("7.feladat\n\tNem volt olyan demokrata elnök, aki kitöltötte a 3
ciklust");
}

/*8. Hozza létre a republikanus.txt állományt!
 * Írja ki az állományba az összes republikánus elnök adatát!
 * Az állomány szerkezete egyezzen meg az eredeti állományéval!*/

FileStream fnev = new FileStream("republikanus.txt", FileMode.Create);
StreamWriter fajlbairo = new StreamWriter(fnev);
for (i = 0; i < elnokokszama; i++)
{
    if (adatok[i].part == "Republikánus")
    {
        fajlbairo.Write("{0};", adatok[i].nev);
        fajlbairo.Write("{0};", adatok[i].kezdet);
        fajlbairo.Write("{0};", adatok[i].veg);
        fajlbairo.Write("{0};", adatok[i].part);
        fajlbairo.Write("{0};", adatok[i].szul);
        fajlbairo.WriteLine("{0}", adatok[i].halal);
    }
}
fajlbairo.Close();
fnev.Close();
Console.ReadKey();
}
}
}

```