

## Bérek 2020 Programozásfeladat

Ebben a feladatban egy cég dolgozóinak 2020-as adatai<sup>1</sup> állnak rendelkezésünkre, melyekkel programozási feladatokat kell megoldania.

A feladat megoldása 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 ékezetmentes azonosítók és kiírások is elfogadottak.
4. Az azonosítókat kis- és nagybetűkkel is kezdheti.
5. A program megírásakor az állományban lévő adatok helyes szerkezetét nem kell ellenőriznie, feltételezheti, hogy a rendelkezésre álló adatok a leírtaknak megfelelnek.
6. Megoldását úgy készítse el, hogy az azonos szerkezetű, de tetszőleges bemeneti adatok mellett is helyes eredményt adjon!

A berek2020.txt UTF-8 kódolású forrásállomány soraiban egy-egy dolgozó adatait tároltuk a következő sorrendben: •neve, például: Beri Dániel •neme: nő vagy férfi •a részleg, ahol dolgozik, például: beszerzés •a belépés éve, például: 1979 •a dolgozó bére (fizetése), például: 222 943

Az állomány első sora a mezőneveket tartalmazza, az adatokat pontosvesszővel választottuk el:

**Név;Neme;Részleg;Belépés;Bér**

**Beri Dániel;férfi;beszerzés;1979;222943**

**Csavar Pista;férfi;pénzügy;1995;234074**

1. Készítsen grafikus vagy konzolalkalmazást (projektet) a következő feladatok megoldásához, amelynek projektjét **berek2020** néven mentse el!
2. Olvassa be a **berek2020.txt** állomány sorait és tárolja az adatokat egy olyan összetett adatszerkezetben, amely használatával a további feladatok megoldhatók! Ügyeljen arra, hogy az állomány első sora az adatok fejlécét tartalmazza!
3. Határozza meg és írja ki a képernyőre, hogy hány dolgozó adatai találhatóak a forrásállományban!
4. Határozza meg és írja ki a képernyőre a 2020-as átlagbéreket! Az eredményt ezer forintban, egy tizedesjegyre kerekítve jelenítse meg!
5. Kérje be egy részleg nevét a felhasználótól a minta szerint!
6. Az előző feladatban megadott részlegben keresse meg és írja ki a legnagyobb bérrel (fizetéssel) rendelkező dolgozó adatait! Ha a megadott részleg nem létezik a cégnél, akkor a „A megadott részleg nem létezik a cégnél!” feliratot jelenítse meg! Feltételezheti, hogy nem alakult ki „holtverseny” egy-egy részlegben dolgozók fizetése között!
7. Készítsen statisztikát az egyes részlegeken dolgozók számáról! A részlegek kiírásának sorrendje tetszőleges!

Képernyőminta:

---

<sup>1</sup> Adatok forrása: <https://www.komal.hu/feladat?a=honap&h=201901&t=inf&l=hu>

```

3. Feladat: Dolgozók száma: 170 fő
4. Feladat: Bérek átlaga: 250,3 eFt
5. Feladat: Kérem a részleg nevét: beszerzés
6. Feladat: A legtöbbet kereső dolgozó a megadott részlegen
    Név: Czecei Zsolt
    Nem: férfi
    Belépés: 1981
    Bér: 452 042 Forint
7. Feladat: Statisztika
    beszerzés - 17 fő
    pénzügy - 36 fő
    asztalosműhely - 18 fő
    értékesítés - 16 fő
    lakatosműhely - 15 fő
    karbantartás - 26 fő
    szerelőműhely - 30 fő
    személyzeti - 12 fő

```

Képernyőminta, ha a megadott részleg nem létezik:

```

3. Feladat: Dolgozók száma: 170 fő
4. Feladat: Bérek átlaga: 250,3 eFt
5. Feladat: Kérem a részleg nevét: logisztika
6. Feladat: A megadott részleg nem létezik a cégnél!
7. Feladat: Statisztika
    beszerzés - 17 fő
    pénzügy - 36 fő
    asztalosműhely - 18 fő
    értékesítés - 16 fő
    lakatosműhely - 15 fő
    karbantartás - 26 fő
    szerelőműhely - 30 fő
    személyzeti - 12 fő

```

Megoldás:

```

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

namespace berek2020
{
    class Dolgozok
    {
        public Dolgozok(string sor)
        {
            string[] sorelemek = sor.Split(';');
            this.Nev = sorelemek[0];
            this.Nem = sorelemek[1];
            this.Reszleg = sorelemek[2];
            this.Belepés = Convert.ToInt32(sorelemek[3]);
            this.Ber = Convert.ToInt32(sorelemek[4]);
        }
        //Név;Neme;Részleg;Belépés;Bér
        public string Nev { get; set; }
        public string Nem { get; set; }
        public string Reszleg { get; set; }
        public int Belepés { get; set; }
        public int Ber { get; set; }
    }
    class Program
    {
        /*A berek2020.txt UTF-8 kódolású forrásállomány soraiban egy-egy dolgozó adatait tároltuk a
        következő sorrendben:
        * •neve, például: Beri Dániel

```

- \* •neme: nő vagy férfi
- \* •a részleg, ahol dolgozik, például: beszerzés
- \* •a belépés éve, például: 1979
- \* •a dolgozó bére (fizetése), például: 222 943\*/

```

public static List<Dolgozok> dolgozoadatok = new List<Dolgozok>();
static void Main(string[] args)
{
    StreamReader olvas = new StreamReader("berek2020.txt", Encoding.UTF8);
    string fejléc = olvas.ReadLine();//ha van fejléc
    while (!olvas.EndOfStream)
    {
        dolgozoadatok.Add(new Dolgozok(olvas.ReadLine()));
    }
    int i;//ciklusváltozó

    int adatokszama = dolgozoadatok.Count;
    /*2. Olvassa be a berek2020.txt állomány sorait és tárolja az adatokat egy olyan
összetett adatszerkezetben,
    * amely használatával a további feladatok megoldhatók!
    * Ügyeljen arra, hogy az állomány első sora az adatok fejlécét tartalmazza!*/

    string bekertReszleg = "";
    double atlagber = 0;//átlagbér

    for (i=0;i< adatokszama; i++)
        Console.WriteLine("{0,-20}{1,-8}{2,-15}{3,-10}{4,-15}",
            dolgozoadatok[i].Nev, dolgozoadatok[i].Nem, dolgozoadatok[i].Reszleg,
dolgozoadatok[i].Belepes, dolgozoadatok[i].Ber);

    //3. Határozza meg és írja ki a képernyőre, hogy hány dolgozó adatai találhatóak a
forrásállományban!
    Console.WriteLine("3. Feladat: Dolgozók száma: {0} fő", adatokszama);

    //4. Határozza meg és írja ki a képernyőre a 2020-as átlagbéreket!
//Az eredményt ezer forintban, egy tizedesjegyre kerekítve jelenítse meg!
    for (i = 0; i < adatokszama; i++)
        atlagber += dolgozoadatok[i].Ber;
    atlagber = atlagber / adatokszama;
    Console.WriteLine("4. Feladat: Bérek átlaga: {0:00.0} eFt", atlagber / 1000);

    //5. Kérje be egy részleg nevét a felhasználótól a minta szerint!
    Console.Write("5. Feladat: Kérem a részleg nevét: ");
    bekertReszleg = Console.ReadLine();

    /*6. Az előző feladatban megadott részlegben keresse meg és írja ki a legnagyobb
bérrel (fizetéssel)
    * rendelkező dolgozó adatait! Ha a megadott részleg nem létezik a cégnél,
    * akkor a „A megadott részleg nem létezik a cégnél!” feliratot jelenítse meg!
    * Feltételezheti, hogy nem alakult ki „holtverseny” egy-egy részlegben dolgozók fizetése
között!*/

    //max kiválasztás
    int max = dolgozoadatok[0].Ber;
    int maxi = 0;
    bool ok = false;//A megadott részleg nem létezik a cégnél!
    for (i = 0; i < adatokszama; i++)
    {
        if(dolgozoadatok[i].Reszleg== bekertReszleg)
        {
            ok = true;//A megadott részleg létezik a cégnél!
            if (dolgozoadatok[i].Ber>max)
            {
                max = dolgozoadatok[i].Ber;
                maxi = i;
            }
        }
    }
}

```

```
}
if(ok)
{
    Console.WriteLine("6. Feladat: A legtöbbet kereső dolgozó a megadott részlegen");
    Console.WriteLine("\tNév: {0}", dolgozoatok[maxi].Nev);
    Console.WriteLine("\tNem: {0}", dolgozoatok[maxi].Nem);
    Console.WriteLine("\tBelépés: {0}", dolgozoatok[maxi].Belepes);
    Console.WriteLine("\tBér: {0:### ###} Forint", dolgozoatok[maxi].Ber);
}
else
    Console.WriteLine("6. Feladat: A megadott részleg nem létezik a cégnél!");

//7. Készítsen statisztikát az egyes részlegeken dolgozók számáról! A részlegek
kiírásának sorrendje tetszőleges!
Console.WriteLine("7. Feladat: Statisztika");
dolgozoatok.GroupBy(x => x.Reszleg).ToList().ForEach(x => Console.WriteLine("\t{0}-{1}
fő", x.Key, x.Count()));

Console.ReadKey();
}
}
```