

Egy Opel gyűjtő adatbázisba rendezte az ismert Opel típusok adatait. Legfeljebb 2000 autóról vannak információi, melyeket az opel.txt UTF-8 kódolású, tabulátorral tagolt állományban tárolt el. Készítsen programot Opelek néven, melynek segítségével kezelheti ezeket az adatokat.

Az állomány minden sora egy-egy autó legfontosabb adatait tartalmazza tabulátorral elválasztva egymástól.

pl: Admiral B 2.8 1969 5 2784 0 1495

Az adatok sorrendben a következők:

- A típus neve. Pl.: Admiral B 2.8
- Melyik évben kezdték el gyártani. Pl. 1969
- Az ülések száma. Pl.: 5
- A motor hengerűrtartalma cm³-ben. Pl.: 2784
- Gyorsulás. Az adat azt mutatja, hogy az autó hány másodperc alatt képes elérni 0-ról 100 km/h sebességet. Pl.: 8,8 Amennyiben az adat nem ismert, 0 szerepel az adatbázisban.
- Az autó önsúlya kg-ban. Pl: 1495 Amennyiben az adat nem ismert, 0 szerepel az adatbázisban.

1. Olvassa be az adatokat egy megfelelő adatszerkezetbe! Jelenítse meg a képernyőn a legutolsó sorban szereplő autó adatait a minta szerint!

1. feladat:

Az utolsó autó típusa: Zafira B (facelift 2008) 1.7 DTR EcoFLEX 2010-es, 5 üléses modell.

2. Hány éves(ek) az adatbázisban szereplő legidősebb autó(k)? Írja ki a képernyőre a minta szerint!

2. feladat:

Az Opel már 50 éve gyárt autót.

3. Sportautónak azokat a gépjárműveket tekintjük, melyek 2 ülésesek és legalább 2000 cm³ a motor hengerűrtartalma. Készítsen logikai értékkel visszatérő függvényt vagy metódust *Sportauto* néven, mely igaz értékkel tér vissza, ha az autó sportautó, hamissal egyébként. Ügyeljen arra, hogy a függvény minden szükséges paramétert megkapjon!

4. Van-e olyan sportautó a listában, amelyet 2010 után gyártottak. Ha van, jelenítse meg egy ilyen Opel adatait a képernyőn a minta szerint. Ha nincs, akkor a „Nincs 2010 utáni sportautó a listánkban.” feliratot írja ki a monitorra! Amennyiben lehetséges, alkalmazza a *Sportauto* függvényt vagy metódust!

4. feladat:

2011-ben gyártott sportautó a Corsa van D (Facelift 2011) 1.2 Twinport Ecotec

5. Átlagosan hány másodperc alatt érik el a 100 km/h sebességet az 5 üléses Opelek? **Ügyeljen rá, hogy a számításban ne szerepeljenek azok a modellek, melyek gyorsulását nem ismerjük!** Az eredményt kerekítse 2 tizedes pontosságra!

5. feladat:

Átlagos gyorsulás: 11,56 s

6. Rendezze az adatokat a gyártás éve szerint növekvő sorrendbe!

7. Készítse el az opel2000.csv állományt, melybe mentse el minden 2000 után gyártott Opel adatát! Az állomány minden sora egy-egy autó adatait tartalmazza pontosvesszővel (;) elválasztva! Ahol nem ismerjük a jármű súlyát vagy a gyorsulást, ott 0 helyett az állományban NaN szerepeljen!

Minta:

1. feladat:

Az utolsó autó típusa: Zafira B (facelift 2008) 1.7 DTR EcoFLEX
2010 - es, 5 üléses modell.

2. feladat:

Az Opel már 55 éve gyárt autót.

4. feladat:

2013 - ben gyártott sportautó a Insignia Hatchback (facelift 2013) OPC 2.8 V6 AWD Turbo Ecotec

5. feladat:

Átlagos gyorsulás: 12,31 s

opel2000.csv - Jegyzetfőm

Fájl Szerkesztés Formátum Nézet Súgó

Vectra C Caravan (facelift 2005) 3.0 V6 CDTI;2005;5;2958;9,1;1615

Vectra C CC (facelift 2005) 2.0i 16V Turbo;2005;5;1998;9,1;1455

Signum (facelift 2005) 3.0 V6 CDTI;2005;5;2958;9,2;1670

Astra H GTC 1.8i;2005;5;1796;10,1;1210

Vectra C Caravan (facelift 2005) 2.2i 16V DIRECT;2005;5;2198;10,1;1550

Astra H Caravan 1.3 CDTI;2005;5;1248;10,14;1280

Vectra C Caravan (facelift 2005) 1.9 CDTI;2005;5;1910;10,3;1625

Astra H GTC 1.8i Automatic;2005;5;1796;11,2;1210

Vectra C Caravan (facelift 2005) 1.8i 16V;2005;5;1796;11,2;1425

Astra H GTC 1.6i 16V;2005;5;1598;12,2;1145

Astra H GTC 1.7 CDTI;2005;5;1686;12,2;1215

Signum (facelift 2005) 2.2i 16V ECOTEC Automatic;2005;5;2198;12,2;1535

Vectra C Caravan (facelift 2005) 1.9 CDTI;2005;5;1910;12,2;1625

Astra H TwinTop 1.8i ECOTEC;2006;4;1796;NaN;1400

Vivaro A (facelift 2006) 2.0 CDTI DPF;2006;9;1995;NaN;1764

```

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

namespace opel
{
    internal class Opelek
    {
        /*  

        □ A típus neve. Pl.: Admiral B 2.8  

        □ Melyik évben kezdték el gyártani. Pl. 1969  

        □ Az ülések száma. Pl.: 5  

        □ A motor hengerűrtartalma cm3-ben. Pl.: 2784  

        □ Gyorsulás. Az adat azt mutatja, hogy az autó hány másodperc alatt képes elérni 0-ról 100  

        km/h sebességet. Pl.: 8,8 Amennyiben az adat nem ismert, 0 szerepel az adatbázisban.  

        □ Az autó önsúlya kg-ban. Pl: 1495 Amennyiben az adat nem ismert, 0 szerepel az  

        adatbázisban.*/
        public string tipus { get; set; }
        public int kezdet { get; set; }
        public int ulesek { get; set; }
        public int henger { get; set; }
        public double gyors { get; set; }
        public int suly { get; set; }
        public Opelek(string sor)
        {
            string[] sz = sor.Split('\t');
            tipus = sz[0];
            kezdet = int.Parse(sz[1]);
            ulesek = int.Parse(sz[2]);
            henger = int.Parse(sz[3]);
            gyors = double.Parse(sz[4]);
            suly = int.Parse(sz[5]);
        }
    }
}

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

namespace opel
{
    class Program
    {
        public static List<Opelek> adatok = new List<Opelek>();
        static bool Sportauto(int ulesekszama, int cm3)
        {
            if (ulesekszama>=2 && cm3>=2000) return true;
            else return false;
        }
        static void Main(string[] args)
        {
            string[] f = File.ReadAllLines("opel.txt", Encoding.UTF8);
            foreach (var a in f) adatok.Add(new Opelek(a)); //listához adom az adatokat
            int adatokszama = adatok.Count(); //sorok száma a fájlban

            Console.WriteLine($"1. feladat:" +
                $"{"\nAz utolsó autó típusa: {adatok[adatokszama - 1].tipus}" +
                $"{"\n{adatok[adatokszama - 1].kezdet} - es, {adatok[adatokszama - 1].ulesek}"
                ülések modell.");

            int min = adatok.Min(x=>x.kezdet);
            Console.WriteLine($"2. feladat:\nAz Opel már {DateTime.Now.Year- min} éve gyárt
            autókat.");
        }
    }
}

```

```

bool van = adatok.Any(x=>x.kezdet>2010&&Sportauto(x.ulesek, x.henger));
if (van)
{
    int sorszam = adatok.FindIndex(x => x.kezdet > 2010 && Sportauto(x.ulesek,
x.henger));
    Console.WriteLine($"4. feladat:\n{adatok[sorszam].kezdet} - ben gyártott
sportautó a {adatok[sorszam].tipus}");
}
else Console.WriteLine("4. feladat:\nNincs 2010 utáni sportautó a listánkban.");

double atlag = adatok.Where(x=>x.ulesek==5&&x.gyors!=0).Average(x=>x.gyors);
Console.WriteLine($"5. feladat:\nÁtlagos gyorsulás: {Math.Round(atlag, 2)} s");

var sorba=adatok.OrderBy(x=>x.kezdet).ToList();

List<string> kiirtlista = new List<string>();
foreach (var a in sorba)
    if (a.kezdet > 2000)
    {
        string gyorsulasadat = a.gyors == 0 ? "NaN" :a.gyors.ToString();
        string sulyadat = a.suly == 0 ? "NaN" :a.suly.ToString();
        kiirtlista.Add($"{a.tipus};{a.kezdet};{a.ulesek};{a.henger};" +
            $"{gyorsulasadat};{sulyadat}");
    }
File.WriteAllLines("opel2000.csv", kiirtlista);

Console.ReadKey();

}
}
}

```