

Idő	Óra menete	Leírás	Comment
0-5	Köszönés, ismétlés	Gépek bekapcsolása. Mi volt múlt órán?	Válasz: switch case szerkezet
5-30	Lista	<p>Lista nagyon hasonlít a tömbökhöz, csak nem kell neki megadni a méretét. Folyamatosan lehet hozzá adni elemeket vagy törölni. Az elemek elérése ugyan úgy történik, mintha tömb lenne.</p> <p>Lista létrehozása:</p> <pre>List<string> napok = new List<string>();</pre> <p>Itt a napokat fogjuk eltárolni, így bele kell tennünk őket.</p> <pre>napok.Add("hétfő"); napok.Add("kedd"); napok.Add("szerda"); napok.Add("csütörtök"); napok.Add("péntek"); napok.Add("szombat"); napok.Add("vasárnap");</pre> <p>Így benne van a hét minden napja. Az elemek számát ellenőrizhetjük is:</p> <pre>Console.WriteLine(napok.Count());</pre> <p>Itt 7-et ír ki a Console-ra nekünk, tehát mind a 7 nap benne van. De rájöttünk, hogy utáljuk a hétfőt, így azt ki is tudjuk törölni:</p> <pre>napok.Remove("hétfő");</pre> <p>Így a hétfő már nem eleme és ha kiírjuk a napok lista elemeinek darabszámát megint, akkor látjuk, hogy már csak 6 eleme van. Nézzük meg most a 0. elemet:</p> <pre>Console.WriteLine(napok[0]);</pre> <p>Itt a keddet írja ki nekünk, amit szintén kitörölhetünk, de most nem a keddet adjuk meg feltételnek, hanem hogy a 0. elem nem kell nekünk:</p> <pre>napok.RemoveAt(0);</pre> <p>Csak hogy lássuk, hogy tényleg kitöröltük őket írassuk ki az elején is a lista elemeit, ott foreach ciklussal, és a végén is, de ott ciklus nélkül: Foreach (lista létrehozása után):</p> <pre>foreach(string nap in napok){ Console.WriteLine(nap); }</pre> <p>Ciklus nélkül is ugyan úgy kiíratható, mint a tömb, az elválasztó jelek nálam vesszők:</p> <pre>Console.WriteLine(String.Join(", ", napok));</pre>	
30-55	File beolvasása	<p>File beolvasásához kell először is egy változó, amiben el tudjuk tárolni a file adatait. Ahhoz, hogy ne kelljen mindig minden könyvtárban keresgélnie a programnak fordításkor, ezért külön kell meghívni őket. Eddig csak a sima System könyvtárat használtuk, most kelleni fog az Input Output is, azaz a System.IO.</p> <pre>using System.IO;</pre> <p>Most már elérjük a StreamReadert, amivel be tudjuk majd olvasni a file-okat.</p> <pre>StreamReader olvas = new StreamReader("7.input.txt");</pre> <p>Ezután a sorokat az olvas.ReadLine()-al tudjuk beolvasni, amit eltárolunk egy string változóba:</p> <pre>string sor = olvas.ReadLine();</pre>	

		<p>Emellett kell egy változó, amiben az egész file-t eltároljuk:</p> <pre>List<int> megtettkmek = new List<int>();</pre> <p>A lista azért int, mert a txt-ben is intek vannak, amik az adott nap megtett km-ek.</p> <p>A beolvasás egy while ciklussal történik, ami addig olvas, amíg a sor változó üres értéket nem kap:</p> <pre>while(sor != null){ megtettkmek.Add(int.Parse(sor)); sor = olvas.ReadLine(); }</pre> <p>Ha a file olvasás után megnézzük a listánk elemszámát, akkor meg kell egyeznie a txt hosszával.</p> <p>Ezután már csak összeadjuk a számokat és megkapjuk, hogy hány km-t tettünk meg az elmúlt napokban ezt egyesével is megtehetjük for, foreach ciklussal, de a Sum() függvény megcsinálja helyettünk:</p> <pre>Console.WriteLine(megtettkmek.Sum());</pre> <p>Ha az átlaguk kell:</p> <pre>Console.WriteLine(megtettkmek.Average());</pre>	
45-55	Fileba kiírás	<p>A file-ba kiíráshoz létre kell hozni egy StreamWriter változót:</p> <pre>StreamWriter ki = new StreamWriter("7.output.txt");</pre> <p>Majd hasonlóan, mint a Console-ra kiírásnál itt is a WriteLine/Write meghívásával ki tudunk bele írni.</p> <pre>ki.WriteLine(megtettkmek.Sum()); ki.WriteLine(megtettkmek.Average());</pre> <p>Majd be kell zárni a file-t különben nem kerül bele semmi és csak az üres file-t fogjuk látni. Ekkor kerül „mentésre” minden változtatás a 7.output.txt-ben:</p> <pre>ki.Close();</pre>	
55-60	Elköszönés	<p>Mentsék el a projektet, nyugodtan vigyék haza (GitHub).</p> <p>Gépek kikapcsolása.</p> <p>Pozitív értékelés! + Jutalom: CUKORKA</p> <p>Elköszönés</p>	<p>Cukorka, matrica csak abban az esetben jár, ha megérdemlik!</p>

Az otthoni gyakorló feladatok a [7.hazi.cs](#) file-ban elérhetőek.