

Idő	Óra menete	Leírás	Comment
0-5	Köszönés, ismétlés	Gépek bekapcsolása. Mi volt múlt órán?	Válasz: Lista, file-ból olvasás, file-ba írás
5-15	Programozási tételek	Ahhoz, hogy hatékonyabb legyen a programunk és időt takarítsunk meg programozási tételeket hoztak létre, amiknek a helyessége bizonyítható. Ide tartozik pl. a sorozatszámítás, összesítés, megszámlálás, rendezés, kiválogatás.	
15-55	Sorozatszámítás	<p>Feladat: Egy futár vezeti, hogy melyik nap hány km-t tett meg 1 címért. Egyik listában tároljuk el a napok számát, másokban az aznapi fuvarok sorszámát, harmadikban pedig a megtett távokat.</p> <p>Hozzunk létre egy listát, amiben eltároljuk a kézbesítésekért kapott pénzt, majd írassuk ki, mennyit keresett velük összesen. És egy txtbe a napot, sorszámot, út hosszát és az érte kapott pénzt.</p> <p>Ahhoz, hogy átláthatóbb legyen a kódunk megismerkedünk a függvényekkel is mellette, így egy függvény fogja visszaadni a pénzüsszeget, amit elmentünk majd a listába.</p> <p>Hozzuk létre a listákat és töltsük fel adatokkal:</p> <pre>List<int> nap = new List<int>(); List<int> fuvar = new List<int>(); List<int> hossz = new List<int>(); List<int> fizetes = new List<int>();</pre> <p>Az adatokat hozzá a 8. input.txt-ben megtalálod, 3-as tagolásokban. Első a nap utána a fuvar majd a hossz tömb összetartozó eleme.</p> <p>Létrehozuk a két változót, ami segíti a beolvasásunkat:</p> <pre>StreamReader olvas = new StreamReader("8. input.txt"); string sor = olvas.ReadLine();</pre> <p>Majd jön a do while ciklus, amibe egymás után olvasunk be 3 sort, így megtartva a 3-as tagolást.</p> <pre>nap.Add(int.Parse(sor)); fuvar.Add(int.Parse(olvas.ReadLine())); hossz.Add(int.Parse(olvas.ReadLine()));</pre> <p>Ezek után, ha ezt beolvastuk, akkor ki tudjuk számolni az érte kapott fizetést:</p> <pre>fizetes.Add(fizet(hossz[adatok]));</pre> <p>itt hívjuk meg a fizet függvényünket, amiben if-ekkel kiszámoljuk, hogy mennyit fog keresni fuvaronként (1-2: 500Ft; 3-5: 700 Ft; 6-: 900Ft):</p> <p>Függvényt statikusát hozunk létre, mert a vissza adott értéket nem változtatjuk meg, illetve visszatérési értéknek számot kérünk így int lesz a típusa:</p> <pre>static int fizet(int tav)</pre> <p>A távolságot várja majd, mint értéket, amivel számolni fog.</p> <p>kell egy fizetés változó, amibe bele rakjuk majd az összeget, amennyit kap a fuvarért és azzal fog vissza térni:</p> <pre>int fizetes;</pre>	

		<p>Ezek után már csak az if szerkezetet kell megírni:</p> <pre> if (tav < 3) fizetes = 500; else { if (tav < 6) fizetes = 700; else { fizetes = 900; } } </pre> <p>Ha ez megvan, akkor a függvény vissza térésének megadjuk a fizetés változót és kész is van a függvényünk:</p> <pre> return fizetes; </pre> <p>Ezek után, hogy ki tudjuk írni kell a StreamWriter a do while ciklus elé és akkor nem kell még egy ciklust írunk hozzá:</p> <pre> StreamWriter ki = new StreamWriter("8. output.txt"); </pre> <p>Majd a már megírt ciklusunk végére a txt-be írás:</p> <pre> ki.WriteLine(nap.Last() + ".nap\t" + fuvar.Last() + ".fuvar\t" + hossz.Last() + " km\t" + fizetes.Last() + "Ft"); </pre> <p>Ezekon kívül kell még egy beolvasás, amit feltételként fogunk vizsgálni, hogy van-e még a txt-ben sor, mert ha nincs, akkor a do while ciklusunk kilép és a program futása a txt bezárásával leáll:</p> <pre> while(sor != null); ki.Close(); </pre>	
55-60	Elköszönés	<p>Mentsék el a projektet, nyugodtan vigyék haza (GitHub). Gépek kikapcsolása. Pozitív értékelés! + Jutalom: CUKORKA Elköszönés</p>	<p>Cukorka, matrica csak abban az esetben jár, ha megérdemlik!</p>

Az otthoni gyakorló feladatok a [8.hazi.cs](#) file-ban elérhetőek.