

# Beadandók #4

- 1.) A **szamok.txt** fájl minden sorában egy-egy természetes szám található. Írj programot, mely kiírja az **osztok.txt** fájlba az adott szám osztópárjait! A kiírást kövesse az alábbi formátumot!  
12: 1 – 12, 2 – 6, 3 – 4,  
40: 1 – 40, 2 – 20, 4 – 10, 5 – 8,
- 2.) A **haromszokek.txt** fájl minden sorában három pozitív egész szám található szóközzel elválasztva. Írj programot, mely kiírja, hogy a fájl hányadik sorában található a legnagyobb **területű** háromszög adatai. (Ha több ilyen is lenne, elég csak az egyiket kiírni.) Vigyázz, nem biztos, hogy a három adat olyan, hogy lehet belőle háromszöget szerkeszteni!
- 3.) A **haromszokek.txt** fájl minden sorában három pozitív egész szám található szóközzel elválasztva. Írj programot, mely kiírja, hogy a fájl hányadik sorában található a legnagyobb **kerületű** háromszög adatai. (Ha több ilyen is lenne, elég csak az egyiket kiírni.) Vigyázz, nem biztos, hogy a három adat olyan, hogy lehet belőle háromszöget szerkeszteni!
- 4.) Írj programot, mely a **tobb\_szo.txt** állomány azon sorait átmásolja a **haromnal\_tobb\_szo.txt** állományba, melynek soraiban háromnál több szó van!
- 5.) Írj programot, mely az **krix-krax.txt** állomány azon sorait átmásolja a **tisztított.txt** állományba, melynek soraiban betű, számjegy és szóköz karaktereken kívül más karakter is szerepel! A másolás úgy történjen, hogy az átmásolandó karaktersorozatokból kerüljenek törlésre a nem betű és nem számjegy karakterek!  
például: A *jhjh#7 8\$?A eS* karaktersorozatból *jhjh7 8A eS* lesz.
- 6.) Készíts programot, mely a **nevek.txt** fájlban található nevek alapján választ ad arra a kérdésre, hogy a különböző vezetéknevekből hány darab szerepel az állományban!
- 7.) Készíts programot, mely a **nevek.txt** fájlban található nevek alapján megmondja, hogy hány olyan ember adatai vannak az állományban, ahol az adott illetőnek nincs második keresztnéve!
- 8.) Az **osztalyzatok.csv** állományban egy osztály tanulójának félévi osztályzatai szerepelnek. Írj programot, mely kiszámítja az adott tanuló átlagát két tizedesjegy pontossággal! A program írja ki az átlag mellett azt is, hogy az adott tanuló hány tantárgyból volt osztályzata! A kiírás formátuma a következő legyen:  
Szabó András átlaga:      3,28    (11 tantárgyból)  
Tóth Tibor átlaga:        4,05    (10 tantárgyból)
- 9.) A **korok.dat** állomány körök adatait tartalmazza. Az első sorban lévő szám (n) a körök számát adja meg ( $n \leq 500$ ). Ezután n sorban három-három, egymástól szóközzel elválasztott természetes szám következik. Az első és második szám a körök középpontjának x és y koordinátája ( $0 \leq x, y \leq 200$ ), a harmadik szám pedig a kör sugara ( $1 \leq r \leq 50$ ). Ezen adatok alapján a **korok\_kerulete.txt** fájlba írasd ki az egyes körök kerületét a következő formában:  

1. kör kerülete:	26,87 cm <sup>2</sup>
2. kör kerülete:	182,92 cm <sup>2</sup>
- 10.) Az **euro\_arfolyam.csv** fájl a forint-euró árfolyam alakulását tartalmazza 2016 januártól 2020 januárjáig. Az adatok mindig az adott hónap átlag árfolyamát mutatják. Minden sorban három egész következik pontosvesszővel elválasztva: év, hónap és egy euró árfolyam fillérben megadva. Tehát pl. a 31448 azt jelenti, hogy egy euró 314 Ft és 48 fillér. (A bankközi műveleteknél fillérrel is számolnak.)  
Olvasd be az adatokat, majd írd ki a képernyőre a következő formátum szerint:  
2006. január:    314 Ft 80 fillér  
2006. február:  310 Ft 13 fillér  
A program a képernyőre kiírva válaszoljon még a következő kérdésekre is!
  - A megadott időszakban mikor volt a legdrágább az euró, és mennyi volt ekkor az ára.
  - Határozd meg évenként, hogy az adott évben mennyi volt az átlagos árfolyam!

- 11.) Progországban derékszögű koordináta-rendszerben adják meg a települések helyzetét. A koordináták nem-negatív egészek, értékük legfeljebb 999. A **telepulesek.txt** állomány 500 település koordinátáit tartalmazza. Írj programot, mely választ ad a következő kérdésekre az állomány tartalma alapján!
- Melyik település (hányadik a fájlban) van legközelebb a (0;0) ponthoz?
  - Hány település van 1000 egységnél messzebb az origótól?
  - Mennyi a települések átlagos távolsága a (0;0) ponttól?
- 12.) A **penzdobas.txt** fájl 10000 pénzfeldobás eredményét tartalmazza. Írj programot, mely válaszol a következő kérdésekre az állomány tartalma alapján!
- Mennyi a fejek és az írások számának különbsége
  - A dobások hány százalékában lett fej a végeredmény?
  - Mennyi volt a leghosszabb fej sorozat?
  - Mi jött ki előbb "FIF" vagy "IFI"? (Lehet az is, hogy egyik sem!)
  - A következő játékot játsszuk: 100 forintról indulunk, fejnél nyerünk 1 forintot, írásnál veszünk 1 forintot, és ha elfogyott a pénzünk, vége a játéknak. Határozd meg, mennyi pénzünk maradt a játék végére!
- 13.) Írj programot, amely generál 500 véletlen számot 1 és 100 között, és azokat kiírja a **veletlen.txt** fájlba! Ezután ebből az állományból olvassa vissza a program a számokat és ezek alapján adjon választ a következő kérdésekre!
- Hány páros, hány páratlan szám van az állományban?
  - Hány 4-gyel osztható szám van az fájlban?
  - Mennyi a páros számok átlaga, mennyi a páratlan számok összege?
  - Hány 0 található a számokban összesen?
- 14.) Írj programot, mely az **utazas.txt** állomány tartalma alapján válaszol a következő kérdésekre!
- Hány db út van a listában?
  - Mennyi az összes utas száma?
  - Hányan utaztak Párizsba?
  - Melyik a legdrágább és melyik a legolcsóbb út?
  - Mennyi az utak átlagára?
  - Mennyi pénzt kellett az összes útra összesen befizetniük az utasoknak?
- 15.) Az **ASCII.dat** állományban a 32-es kódtól a 90-es kódig található meg az egyes karakterekhez tartozó ún. ASCII (ejtsd: eszki!) kódok. Írj programot, mely ennek az állománynak a felhasználásával visszafejti a **kodolt\_szoveg.txt** fájl tartalmát, s a végeredményt írja ki a képernyőre!
- (A kodolt\_szoveg.txt állományban az egyes karaktereknek megfelelő kódok vannak binárisan eltárolva.)