# Assertion 2.

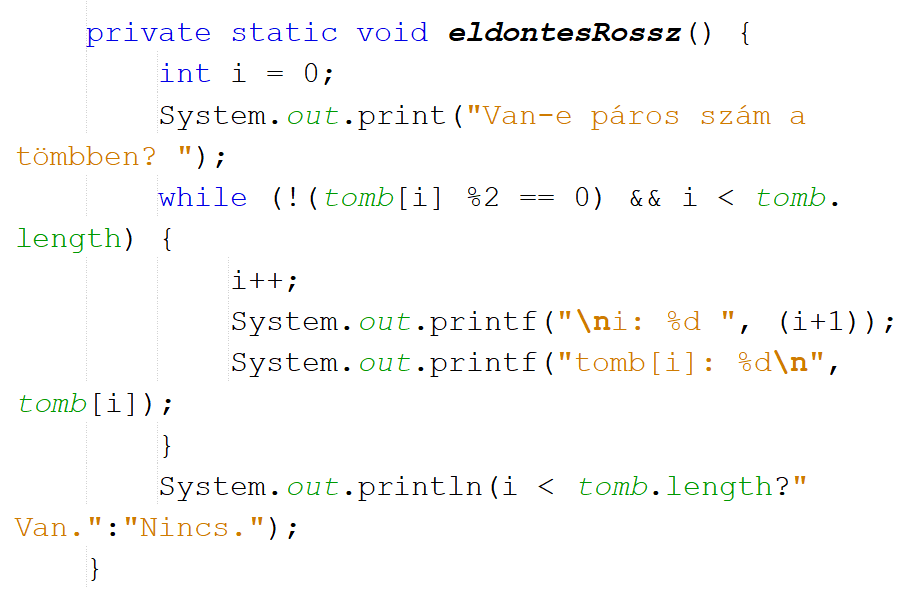
Talán úgy a legegyszerűbb megfogalmazni, hogy miről is szól az assert, hogy olyan komment, ami “válaszol” a feltevéseinkre, állításainkra.

Az assert után tehát megfogalmazzuk azt, hogy mit várunk el a programnak azon a pontján, a “:” után pedig azt az üzenetet íratjuk ki, ami az állítás helytelensége/tagadása esetén hasznos lehet számunkra.

A következő feladatokat oldjátok meg páros munkában!

Lehetőség szerint dolgozzatok azzal, akivel múlt órán is.

## Feladatok

1. Egy egész számokból álló 10 elemű tömböt (lehet megadva) rendezzünk növekvő sorrendbe! (az alapkód mellékelve: *rendezes.docx*).
   1. Előfeltétel ellenőrzése: a tömb (x) nem null értékű (assert x != null : "null passed to x";)
   2. Utófeltétel ellenőrzése: rendezett a tömb (assert isSorted(x): "array not sorted";) (A rendezés kódját megkapjátok (mellékelve), csak az ellenőrzéseket és az *isSorted(x)* metódust (függvényt, logikai típusú visszatérési értékkel) kell *nektek megírni*, illetve elhelyezni.)
   3. A múlt órán tanult debuggolás segítségével próbáljátok megérteni a rendezési algoritmust, és beszéljétek meg, hogy a kérdésként megírt kommentek helyére mit tennétek válaszként.
   4. A kommentek helyett használhattok assert-eket is (a rendezési algoritmuson belül is tehát), hogy ellenőrizzétek azt, hogy helyesen gondolkodtok-e.
2. Írjátok meg jól az eldöntés tételét, kiindulva az alábbi rossz változatból! (használjatok segítségül asserteket és/vagy debuggolást is!)  
   
   1. mit jelenthet az utolsó sor? (nézzétek meg [itt](https://www.baeldung.com/java-ternary-operator)!)
3. Bizonyítsátok be assertek és az alábbi specifikáció segítségével, hogy az   
   r == a ∗ b kifejezés teljesen megfelel az r = a + a + a + … + a kifejezésnek, ahol az a változó b-szer szerepel az összegzésben, ha b > 0! [[1]](#footnote-0)  
   *Specifikáció:***Bemenet**: a, b (egész)  
   **Kimenet**: r  
   **Előfeltétel**: b > 0 (ellenőrzés)  
   **Utófeltétel**: r = a ∗ b (ellenőrzés)  
   *Pszeudo kód:*Szorzas():  
    r := 0  
    c := b   
    Ciklus amíg (c > 0)   
    r := r + a  
    c := c − 1  
    Ciklus vége  
   Eljárás vége
4. Készítsetek 10 véletlen számból álló sorozatot (lista vagy tömb segítségével), mely lehetséges kockadobások sorozatát tárolja! Kommentek helyett használj asserteket!

1. <https://www.rose-hulman.edu/class/se/OldFiles/csse373/Spring2009/Resources/assert.pdf> [↑](#footnote-ref-0)