

## Valós számok?

A következő kis programocska meglepő eredményt ad, ha megfelelő lépésben végezzük el:  $1/3 = 1/2!$  A program a valós számok kezelésének egy sajátosságára mutat rá, ami könnyen programbeli hibához vezethet.

*51. Számoljuk ki meghatározott alkalommal az 1/3 értéket úgy, hogy szorozzuk 4-gyel, majd 1-et kivonunk belőle!*

A **meglepő** nevű kis program az **:x** változóban tárolja az egyharmad értéket, majd azt **:db** alkalommal újra számolja ugyan, de várhatóan változatlanul is hagyja, mivel az egyharmad négyszerese négyharmad, és abból levonva egyet ugyancsak egyharmadot kapunk.

```
eljárás meglepő :db
  lokálisérték "x 1/3
  ism :db [lokálisérték "x :x*4-1]
  (mutat :db "végrehajtás "után :x)
vége
```

A programot futtassuk le több paraméterrel is! Azt tapasztaljuk, hogy 22-ig semmi meglepőt nem kapunk, minden alkalommal kiírja a 0.333333 értéket, azaz az adott hat tizedesjegyes kijelzés mellett semmi meglepő nem tapasztalható. Ha viszont 23-at adjuk meg paraméternek, annál meglepőbb eredmény születik! Az eredmény 0,333334 lesz, 32 esetén pedig pont 0,5!

Mi is történt?

Sajnos a ComLOGO-ban megszokott megjelenített jegyszámot szabályozó **tizedesjegy!** parancs, illetve annak megfelelője nem létezik! A valós értékek kijelzése rögzített 6 értékes jeggyel történik meg, emiatt nem látható az ok!

Az Imagine ugyanúgy, mint minden más programozási nyelv, csak egy előre jól meghatározott pontossággal tudja kezelni a számokat. A matematikai értelemben vett valós számok emiatt csak bizonyos pontossággal ábrázolhatóak a számítógépen a lebegőpontos ábrázolás segítségével. Az egyharmad ráadásul olyan érték, amely pontosan nem ábrázolható kettes számrendszerben, mivel a kettes törtalakja végtelen kettedes jegyű, viszont egy bizonyos pontossággal, jegyszámmal igen. A **meglepő** program a pontosságot, jegyszámot leplezi le a 32 esetén, ami eszerint 64 bites, mivel a program minden lépésénél az ábrázolás hibáját 4-szeresére növeli meg, azaz két bitet emészt fel az értékes jegyekből. Ez a 64 bites ábrázolás olyan precíz, hogy igen sok lépés után okoz csak észrevehető pontatlanságot, a tízes számrendszerbeli átíráskor a kb. 20 bites pontosság is még hat tizedesnyi pontosságot biztosít.

A programozónak kell figyelnie, hogy a programjában ne okozzanak hibát a valós számok lebegőpontos ábrázolásának sajátosságai.