

10. hét: számrendszerek, számábrázolás

Czirkos Zoltán · 2023.11.07.

A számrendszerekről szóló előadáshoz kapcsolódó feladatok. Néhány számábrázolási probléma megjelenése a programokban.

Felkészülés a laborra:

- Az [operátorokról tanultak](#) átismétlése.
- A [számábrázolásról, számrendszerekről tanultak](#) megértése.

Tartalom

- [Harmadik kis ZH a műszaki menedzsereknek](#)
- [Pót ZH](#)
- [Operátorok és kiértékelés](#)
- [Lebegőpontos](#)
- [Hol a hiba?](#)
- [Kiírás adott számrendszerben I.](#)
- [Kiírás adott számrendszerben II.](#)
- [Beolvasás adott számrendszerben](#)
- [Konverziós függvények](#)
- [További feladatok](#)

1. Harmadik kis ZH a műszaki menedzsereknek

A hétfői alkalmakon lesz a harmadik kis ZH a műszaki menedzsereknek

2. Pót ZH

[Jelentkeztél pót ZH-ra](#), ha írnod kell? Ha még nem, tedd meg most, már nem sok idő van hátra!

3. Operátorok és kiértékelés

Adjuk meg az alábbi kifejezésekhez tartozó kifejezésfát, figyelembe véve az operátorok precedenciáját!

- $6 + 2 * 3$
- $2 * 6 - 5 / 3$
- $a = b + c$
- $t[i + 2] * 3$
- $5 * - 6$
- $5 - * 6$

A legegyszerűbb ascii karakterekkel megrajzolni és úgy feltölteni. Ebben a feladatban nem kell kódolni, csak rajzolni.

Kétszer kettő:

```
*
/ \
2  2
```

► [Megoldás](#)

4. Lebegőpontos

Mit írnak ki az alábbi program egyes sorai? Próbáld meg kitalálni! Futtasd le a programot, és magyarázd meg az eredményt!

```
print("1.", 1e200 / 1e-200)
print("2.", "igaz" if 1e3 + 1 == 1e3 else "hamis")
print("3.", "igaz" if 1e30 + 1 == 1e30 else "hamis")
```

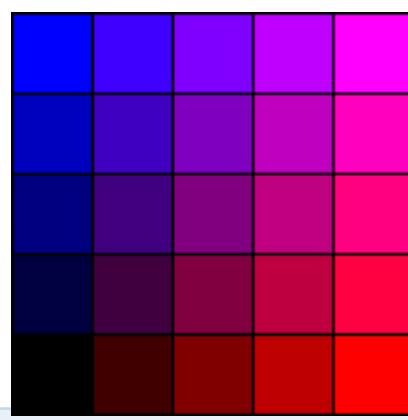
► [Megoldás](#)

5. Hol a hiba?

Az alábbi program egy színátmenetes rajtot hivatott elkészíteni. A színek balról jobbra egyre pirosabbak, és alulról felfelé egyre kékebbek. A színek komponensek értéke mindig 0-tól 1-ig változik. Az osztások számát a program a felhasználótól kéri, az egyes kis négyzetek oldalhossza 30 képpont.

A program némely osztásszámmra kifogástalanul működik, például **db = 5**-re a képen látható ábrát készíti. Más osztások esetén elromlik, **db = 10** esetén szétcsúszik az ábra.

Mi a probléma? Magyarázd meg a jelenséget, és mutasd meg a nyomkövetőben is! Javítsd meg a programot, hogy minden osztásszámmra helyesen működjön!



```
import turtle

def negyzet(a):
    turtle.begin_fill()
    for i in range(0, 4):
        turtle.forward(a)
        turtle.left(90)
    turtle.end_fill()

def main():
    a = 30 # oldalhossz
    db = int(input("Hány darabból? "))

    turtle.speed(0)
    b = 0.0
    while b <= 1.0:
        r = 0.0
        while r <= 1.0:
            turtle.fillcolor(r, 0, b)
            negyzet(a)
            turtle.forward(a)
            r += 1/(db-1)
            turtle.backward(db*a)
            turtle.left(90)
            turtle.forward(a)
            turtle.right(90)
            b += 1/(db-1)

        turtle.done()

main()
```

► [Megoldás](#)

6. Kiírás adott számrendszerben I.

Írj programot, amelyik a megadott számot a megadott számrendszerbe alakítja át, és beteszi úgy egy sztringbe! Pl. a 9-es szám 2-es számrendszerben "**1001**". Az algoritmusod tetszőleges lehet, de a végén egy **print(szam)** utasítással ki kell tudni írni a számot, ahol a **szam** annak a változónak a neve, amiben az eredményt előállítottad.

Elég, ha előbb csak tízes számrendszerig működik a program! Ha működik jól a programod, akkor utána alakítsd át úgy, hogy nagyobb alap esetén is működjön! A 10-et, és annál nagyobb számjegyeket ilyenkor betűkkel szokás jelölni. Pl. 16-osban a 0...15 számjegyek: 012...89ABCDE.

► [Tipp](#)

► [Megoldás](#)

7. Kiírás adott számrendszerben II.

Működik az előző feladatban megírt programod 0-ra is? Az algoritmustól függően előfordulhat, hogy 0-ra üres sztringet ad. Ha ilyen megoldást adtál, akkor egészítsd ki!

Oldd meg továbbá azt, hogy negatív számot is kaphasson! A pozitív számokat nem kell plusszal jelölni, a negatívak elé viszont kerüljön mínusz karakter!

8. Beolvasás adott számrendszerben

Írj programot, amely kap egy sztringet és egy számrendszernek a számát, majd megadja azt az egész számot, amit az adott sztringbeli számjegysorozat reprezentál! Például ha a felhasználó beírja, hogy 2-es számrendszer, és 1101, akkor ki kell írni, hogy 13. Először elég, ha tízes számrendszerig működik a program, csak utána írd át úgy, hogy működjön nagyobb alap esetén is!

```
Hányas számrendszerben fogsz írni?
16
Írd be a számot!
fce2
A beolvasott szám 10-es számrendszerben: 64738
```

Ha elkészültél a saját algoritmussal, ellenőrizd különféle számokra a beépített konverzióval! Pl. **int("fce2", 16)** a fenti példa alapján **64738**-at ad.

► [Tipp](#)

► [Megoldás](#)

9. Konverziós függvények

Dolgozd át úgy az előző feladatok „kiírás adott számrendszerben” és „beolvasás adott számrendszerben” programkódjait, hogy azok függvények legyenek! Például:

- szamrendszerbol("fce2", 16) → 64738.**
- szamrendszerbe(64738, 16) → "fce2".**

Készítsd fel a függvényeket hibák kezelésére! Pl. ha adott számrendszerben nem létezik egy számjegy, akkor dobjon kivételt az első függvény. Ha negatív számrendszerbe átalakítást kérünk, akkor dobjon kivételt a második, és így tovább. Milyen hibalehetőségek lehetnek még?

10. További feladatok

Ha végeztél, dolgozz a [nagy házi feladaton](#)!

