**INFOPY** INFÓ - TANANYAG - SEGÉDLET - EXTRÁK -

2. hét: nyomkövetés, vezérlés

Nyomkövetés. Vezérlési szerkezetek: elágazások, ciklusok. Összetett vezérlési szerkezetek, összefüggő ciklusok és független változók.

BELÉPÉS

Felkészülés a laborra: A vezérlési szerkezetekről tanultak megértése.

**Tartalom** 

1. Kettő meg három 2. Sztringek különleges karakterekkel

3. A tartály

4. Szakasz hossza

5. Másodfokú egyenlet 6. Számtani sorozat

8. Adott hosszúságú szakasz 9. Adott méretű négyzet 10. Háromszög golyókból

7. A nyomkövető használata

11. Szorgalmik

## egyik = input("Írj be valamit: ")

Adott az alábbi program.

1. Kettő meg három %

masik = input("Írj be még valamit: ")

print(egyik + masik) Másold be a fejlesztőkörnyezetbe, és futtasd le! Mi történik akkor, ha azt írod be neki, hogy alma, és hogy fa? És mi történik akkor, ha 2-t és 3-at írsz be neki? (Miért nem 5?) Magyarázd meg (magyar nyelvű szövegben), hogy mi történik!

Ha ez megvolt, írd át a programot ilyenre: egyik = int(input("Írj be valamit: ")) masik = int(input("Írj be még valamit: "))

print(egyik + masik)

Figyeld meg, mi a különbség a kódban! Az első két sor változott, a harmadik nem. Próbáld most ki 2-vel és 3-mal! Mit tapasztalsz? Mi történik akkor, ha most próbálod beírni az előbbi szavakat? Magyarázd meg az eredményt!

### karaktereket \ visszaperrel kell jelölni, ha szöveg belsejében szerepelnek a forráskódban. Írd ki egyetlen egy print() utasítással az alábbi szövegeket! Ügyelj arra, hogy ahol két sorban

2. Sztringek különleges karakterekkel %

vannak a minta szerint a kimenetek, ott a programodnak is úgy kell megjelenítenie őket. A print('hello') kiírja, hogy "hello", és kezd egy új sort.

Az előadás szorzótáblás programjánál szóba jött, hogy létezik egy tabulátor nevű karakter, és hogy

azt "\t"-vel kell a forráskódban megadni. Ennek kapcsán pedig arról is, hogy a különleges

A "\t" karaktert tabulátornak nevezzük.

A következő 8-cal osztható sorszámú oszlopra lehet ugrani vele.

festéket kell vennie a lábakon álló tartály festéséhez! Ehhez használnod kell majd az első előadásokon tanultakat. Indulj ki az előadáson bemutatott kör kerülete és területe programból!

## A tartálynak ismerjük a magasságát és az átmérőjét. Mindenhol le kell

Mennyi az atmérője? 1.2

4. Szakasz hossza %

4.900885 doboz festék kell.

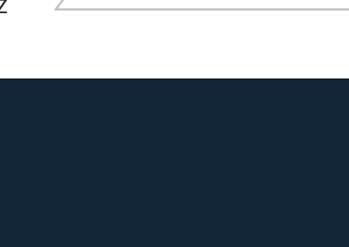
gondold át, helyes-e a képlet, amit használsz.

szakasz

3. A tartály %

festeni, az oldalát, a tetejét és az alját is. Tudjuk azt is, hogy egy doboz festék 2 m<sup>2</sup> felület lefestéséhez elegendő. Tartály festése Milyen magas? 2

Írj programot, amely segít kiszámolni a felhasználónak, hogy hány doboz



Figyelj arra, hogy a program kódjában tizedesvessző helyett tizedespontot kell használni (pl. 3.14,

persze jó a math.pi is), ugyanígy a bemenet megadásánál is. Ha nem jön ki a fenti eredmény,

Írj programot, amely a felhasználótól bekéri két síkbeli pont **x** és **y** koordinátáit, és kiírja a közéjük húzott egyenes szakasz hosszát (Pitagorasz

tételével)!

▶ Megoldás

Hasonló feladatok

5. Másodfokú egyenlet %

 $x^2-12x+35 = 0 x_1=5, x_2=7$ 

Megoldás

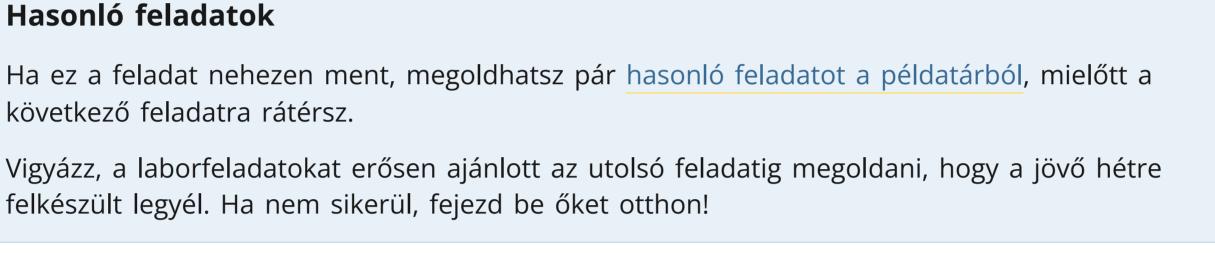
Ha ez a feladat nehezen ment, megoldhatsz pár hasonló feladatot a példatárból, mielőtt a következő feladatra rátérsz.

hossz

(0;0)-(1;1) 1.414214

(-3;2)-(5;7) 9.433981

(1;5)-(4;1) 5



aztán a jobb oldali egyenletekre is. Mit tapasztalsz? Miért? megoldás megoldás egyenlet egyenlet  $2x^2-x-6 = 0$   $x_1=2, x_2=-1.5$  $2x^2-4x+2 = 0$ ?

Próbáld ki a bal oldalt látható egyenletekre! Ezekkel ellenőrizni tudod a megoldásod. Próbáld ki

Írj programot, amely az ax²+bx+c=0 másodfokú egyenlet együtthatóit kérdezi a felhasználótól! Ezek

 $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ 

valós számok. Ezek után írja ki az egyenlet  $x_1$  és  $x_2$  megoldását! A megoldóképlet:

 $x^2+2x+10 = 0$ ?

▶ Megoldás Hasonló feladatok

Ha ez a feladat nehezen ment, megoldhatsz pár hasonló feladatot a példatárból, mielőtt a következő feladatra rátérsz. Vigyázz, a laborfeladatokat erősen ajánlott az utolsó feladatig megoldani, hogy a jövő hétre felkészült legyél. Ha nem sikerül, fejezd be őket otthon!

### 6. Számtani sorozat Olyan programot kell írnod, amelyik számtani sorozatot jelenít meg.

programot egy új fájlba:

while n <= 10:

n += 1

n = 1

n += 1

print(".")

szorzat = 1

while n > 1:

**▶** Megoldás

a = 11220

while b > 0:

temp = b

b = a % b

b = 2002

**▶** Megoldás

n -= 1

n = 8

print(".")

n = 1

számból gyököt vonni.

őket, vizsgáld meg a sorrendet, és utána kerüljenek min-be és max-ba a számok! ► Megoldás

algoritmusunk, és nem azt tettük bele, amit szerettünk volna.

print(n, end=" ") # nem kezd új sort

ablakában a szokásos Run / Run module menüpontra kattints.

jelenleg nem lényeges információ mellett) a változó értéke figyelhető meg:

File Edit Format Run Options Window Help

print(n, end = ' ')

'/tmp/ciklus.py'

\_\_package\_\_ None

loader <class ' frozen importlib.BuiltinImporter'>

Kattints még sokszor az **Over** gombra, közben figyeld a kimenetet és a Shell ablakot!

n = 1

while n <= 10:

n += 1

print('.')

felső) is! Ezek egész számok kell legyenek.

Hasonló feladatok Ebben a témakörben is találsz még feladatokat a példatárból. Ha úgy érzed, oldj meg onnan néhányat a továbblépés előtt. De vigyázz, az itteni feladatokat ajánlott végigoldani a jövő hétre, hogy felkészült legyél.

A számtani sorozat differenciáját a felhasználó adhassa meg! A sorozat két határát (alsó és

■ Hogy "reagálja le" a program azt, ha rossz sorrendben adod meg a határokat? Javítsd meg,

valamilyen semleges nevű változóba teszed először (pl. hatar1, hatar2), mert amíg nem

hogy ilyenkor is működjön! Ezt legjobb úgy csinálni, ha a két határt nem min és max, hanem

vizsgáltad meg őket, addig nem jó ötlet minimumnak és maximumnak nevezni. Tehát olvasd be

7. A nyomkövető használata % A programozási hibákat gyakran nehéz megtalálni, de sokat segít a nyomkövetés. Ennek segítségével a programot megállíthatjuk futás közben, végrehajthatjuk soronként, közben megfigyelhetjük a változók értékeit. Ez azért jó, mert:

Rögtön észre tudjuk venni, ha épp "elromlott" egy változó: rossz volt a gondolatmenetünk,

A nyomkövetés kipróbálásához, megtanulásához tedd a következőket. Először is másold be ezt a

szántunk neki. Például mert lefelejtettünk egy zárójelet, tehát kódolási hibánk van.

Azt is látjuk, ha kódolási hibánk van; ha nem azt a műveletet végzi el a programsor, mint amit

A program ilyenkor elindult, de még egyelőre meg van akasztva a legelső sor előtt. A Debug Control ablak over gombjának nyomkodásával lehet végrehajtani a következő sort. A befejezett ciklustörzsek után a kiemeneten elkezdenek megjelenni a számok. A forráskódban látszik, melyik az aktuális sor, a szürke sáv mutatja, hol tart épp a végrehajtás. A Debug Control ablakban pedig (sok

ciklus.py - /tmp/ciklus.py (3.5.2)

Debug Contro

Debug Control nevű ablak. Ebben pipáld be a [ ] Source jelölőnégyzetet! Utána a program

Mentés után a Shell ablak menüjében kattints a Debug / Debugger menüpontra. Erre megjelenik egy

Go Step Over Out Quit Stack Source ciklus.py:3: <module>() bdb'.run(), line 431: exec(cmd, globals, locals) > '\_main\_ '.<module>(), line 3: print(n, end = ' \_builtins\_\_ <module 'builtins' (built-in)>

```
Miután kitapasztaltad, válaszolj a nyomkövető használatával az alábbi kérdésekre!
 ■ Mennyi az alábbi programban az n változó értéke, amikor a legalsó sorhoz ér a végrehajtás, a
    print(".")-hoz?
     while n <= 10:
         print(n, end=" ")  # nem kezd új sort
 Alább egy program, amely 8 faktoriálisát számítja ki, és 40320-at ír a kimenetre.
         szorzat *= n
     print(szorzat)
    Értsd meg a nyomkövető segítségével a program működését! A nyomkövető használatával –
    újabb print() beírása nélkül – mondd meg, hogy mennyi a szorzat változó értéke, amikor az n
   változó éppen 3-ra változott!
```

a = tempprint("lnko =", a) Megint új print() beírása nélkül: mennyi a b változó értéke, amikor az a épp 374-re változott?

■ A következő program Euklidész algoritmusa 🗹 segítségével határozza meg két szám

legnagyobb közös osztóját, jelen esetben az 11220-ét és a 2002-ét, ami 22:

# Mekkora legyen a szakasz? +---+

legyen:

8. Adott hosszúságú szakasz %

Aztán próbáld ki, mi történik a print("-" \* 4) sor hatására. Hogy lehetne ezt felhasználni? Írd meg így is a programot! ► Megoldás

Írd meg a programot előbb úgy, hogy ciklust használsz a – jelek kiírásához!

Írj egy programot, amely kér a felhasználótól egy számot, és kirajzol egy + és – jelekből álló

szakaszt. Pl. ha a szám 4, akkor a képernyőn a lenti ábra jelenjen meg, vagyis a belsejében 4 db –

```
9. Adott méretű négyzet %
Az előző feladat rajzolást végző programrészéből készíts egy olyan másolatot, amely + és -
karakterek helyett | és . karaktereket használ, mint jobb oldalon! Végül pedig, ezt ciklusba téve, írj
egy olyan változatot, amelyben egy adott méretű négyzet jelenik meg.
 |....|
```

• • • • +---+ A forráskód szerkesztését kényelmesebb billentyűzeten végezni, mint egérrel. Szinte minden funkció elérhető gyorsbillentyűkön keresztül. Néhány fontosabb:

■ Ctrl + ], Ctrl + []: a kijelölt részlet indentálása (különösen hasznos ennél a feladatnál: amikor

- egy meglévő kódrészletet szeretnél ciklusba betenni, vagy ciklusból kivenni!), ■ Ctrl + C, Ctrl + V: a szokásos másolás és beillesztés, ■ Ctrl + Z: "undo", szerkesztés visszavonása. ► Megoldás
- 10. Háromszög golyókból % Írjunk programot, amely kér a felhasználótól egy számot (n), és utána egy akkora háromszöget

■ [Home], [End]: sor elejére, sor végére ugrás,

■ Shift + ↑, ↓, ←, →: forráskódrészlet kijelölése,

# rajzol a képernyőre "o" betűkből, hogy annak éppen a megadott számú sora van! Például n=3 esetén az itt látható ábra keletkezzen.

Hasonló feladatok

000 00000 ► Megoldás

Itt mindig az a lényeg, hogy a rajz alapján rájöjj, milyen vezérlési szerkezettel állítható az elő. Ebben a témakörben is találsz még feladatokat a példatárból.

11. Szorgalmik %

Ha mindennel elkészültél, dolgozhatsz a szorgalmi feladatokon is. Ezek plusz pontért küldhetők be.

```
InfoPy – BProf ProgAlap
főoldal · admin portál · elérhetőség · licenc · 3 rss
BME EET, 2009-2023.
```