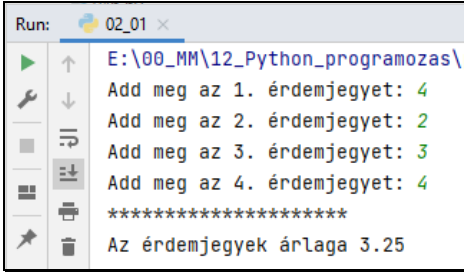
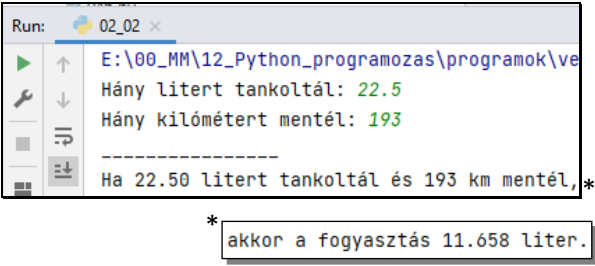
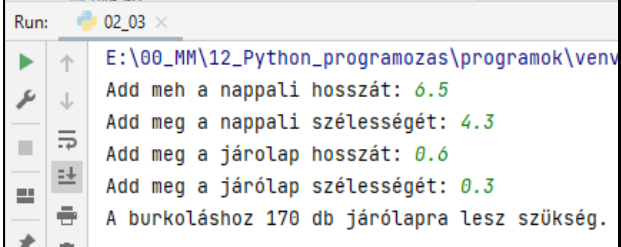
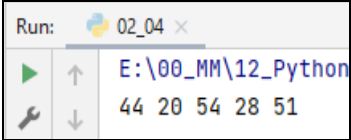
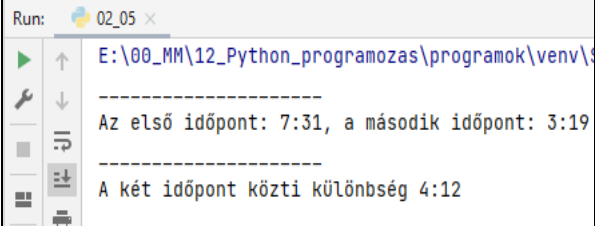
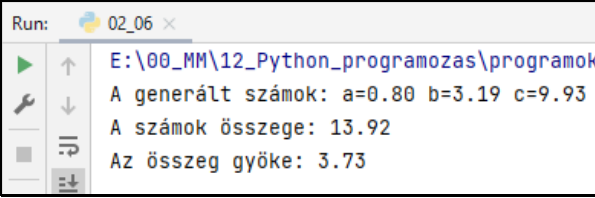
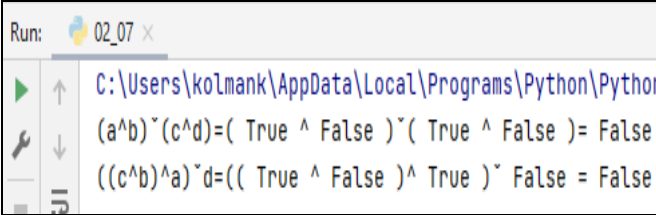
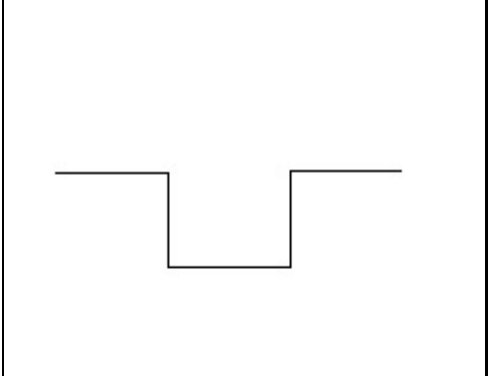
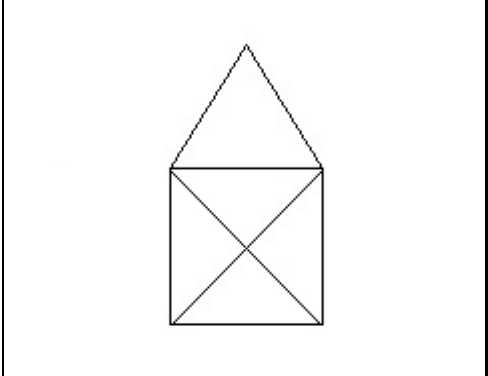
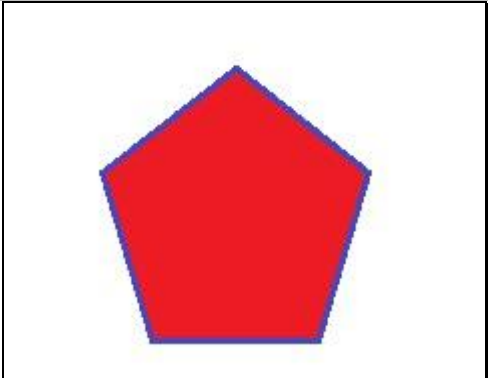


GYAKORLÓ FELADATOK (04-07. témakör)

Készítsd el a következő feladatokat utasítások alapján! A végeredmény kinézete mindig a képernyőképpel megegyező legyen! A programok neve, a feladat elején zárójelben lévő szám legyen, minden esetben!

FELADAT LEÍRÁSA	KÉPERNYŐKÉP
<p>1. (02_01.py) Készíts programot, amely bekér 4 érdemjegyet, majd írasd ki az átlagukat, század pontossággal!</p>	 <pre> Run: 02_01 x E:\00_MM\12_Python_programozas\ Add meg az 1. érdemjegyet: 4 Add meg az 2. érdemjegyet: 2 Add meg az 3. érdemjegyet: 3 Add meg az 4. érdemjegyet: 4 ***** Az érdemjegyek átlaga 3.25 </pre>
<p>2. (02_02.py) Készíts programot, mellyel kiszámolod, hogy egy autó mennyit fogyaszt 100 kilométeren! Kérd be, hogy hány litert tankoltál, és hogy hány kilométert mentél! (liter ->float, km->int) Képlet: $Fogyasztás = \text{tankolt_mennyiség} * 100 / \text{megtett_km}$ Az eredményt ezred pontossággal írasd ki!</p>	 <pre> Run: 02_02 x E:\00_MM\12_Python_programozas\programok\ve Hány litert tankoltál: 22.5 Hány kilométert mentél: 193 ----- Ha 22.50 litert tankoltál és 193 km mentél,* * akkor a fogyasztás 11.658 liter. </pre>
<p>3. (02_03.py) Burkolni szeretnéd járólappal a nappalidat. Számold ki, hogy darab járólagra lesz szükséged! Készíts programot mely bekéri a nappal egyik, másik oldalát méterben. Nem egész számokat is használhassunk a programban! Majd kérd be egy járólap méretét! Hosszát és a szélességét! (A járólapok mérete kisebb, mint 1 méter. pl.: 0,3m*0,3m, 0,3m*0,6m, 0,3m*0,2m) Arra figyelj, hogy mindig 10%-al túl kell tervezni a szükséges mennyiséget! A végső darabszám viszont egész legyen!</p>	 <pre> Run: 02_03 x E:\00_MM\12_Python_programozas\programok\venv\ Add meh a nappali hosszát: 6.5 Add meg a nappali szélességét: 4.3 Add meg a járólap hosszát: 0.6 Add meg a járólap szélességét: 0.3 A burkoláshoz 170 db járólagra lesz szükség. </pre>
<p>4. (02_04.py) Készíts olyan programot, mellyel ötöslottó sorsolást modellezel! (Az ötöslottóban 1-90-ig vannak számok.)</p>	 <pre> Run: 02_04 x E:\00_MM\12_Python 44 20 54 28 51 </pre>
<p>5. (02_05.py) Készíts egy programot, melyben generálsz két véletlenszerű időpontot órában és percben meghatározva, majd jelenítsd meg a két időpont közti különbséget! Használd az abs() függvényt!</p>	 <pre> Run: 02_05 x E:\00_MM\12_Python_programozas\programok\venv\ ----- Az első időpont: 7:31, a második időpont: 3:19 ----- A két időpont közti különbség 4:12 </pre>
<p>6. (02_06.py) Generálgj három darab véletlen valós számot 1 és 10 között, majd szorozd össze őket, és végül vonjad a gyököt a szorzatból! Készíts programot, hogy a képernyőkép úgy nézzen ki, ahogy a mintán!</p>	 <pre> Run: 02_06 x E:\00_MM\12_Python_programozas\programok A generált számok: a=0.80 b=3.19 c=9.93 A számok összege: 13.92 Az összeg gyöke: 3.73 </pre>

<p>7. (02_07.py) Készíts programot, melyben az elején megadsz négy változóban (a,b,c,d) True és False értékeket, majd a következő képleteket old meg, a képernyőkép alapján! (\sim \rightarrow vagy, \wedge \rightarrow és) $(a^b)^{\sim}(c^d)$ $((c^b)^a)^{\sim}d$</p>	
<p>8. (02_08.py) Rajzold meg a jobb oldalon lévő ábrát! A szakaszok egyforma hosszúságúak, 100 kp! A program lefutás után várjon, amíg a felhasználó bezárja az ablakot!</p>	
<p>9. (02_09.py) Rajzold meg a jobb oldalon lévő „ház” ábrát úgy, hogy egyetlen vonalat használj! (magyarul nem emelheted fel a „tollat”! A „ház” oldalai 100 képpontosak, viszont van olyan vonal, aminek a méretét Pithagorasz tétellel ki kel számolnod! A program lefutás után várjon, amíg a felhasználó bezárja az ablakot!</p>	
<p>10. (02_10.py) Rajzolj kék színnel egy piros kitöltésű ötszöget! Az oldalak hossza legyen 100 képpont! A program lefutás után várjon, amíg a felhasználó bezárja az ablakot, vagy használjuk a time.sleep() parancsot és várjon az ablak/program bezárásával 2 másodpercet!</p>	



Hasznos parancs a grafikus felület használatakor a time.sleep():

Ha a rajzolt alakzatnál azt szeretnénk, hogy ne azonnal ugorjon ki a programból a lefutás végén, akkor a kódunk végére írjuk be a **time.sleep(2.5)** utasítást! Ami azt eredményezi, amikor megrajzolta az eredeti utasítások alapján az alakzatot, akkor 2,5 másodpercet vár. A zárójelben mindig float számot kell megadni, másodpercben!

A kód elején meg kell adni az „**import time**” függvény könyvtárt, hogy tudjuk használni a time.sleep() parancsot! (Olyan mint a turtle és a math.)

A PyCharm lehet, hogy összevonja a program elején az import utasításokat.

```
1 import time
2 from turtle import *
```

```
6 time.sleep(2.0)
```

```
1 import ...
1 import time
2 from turtle import *
```