1. feladat: Drón reptetés (40 pont)

Van egy drónunk , amely épületek magasságát képes megállapítani egy lézeres érzékelővel. A drón érzékelője mindig azt az értéket adja vissza, amely a legmagasabb épület magassága az adott irányba nézve. Az alábbi felülnézeti pályán a fekete négyzetekben lévő számok azt jelentik, hogy az ott található épület hány emelet magas.

A drón helyzetét úgy tudjuk megadni, hogy megadjuk melyik cellába kerüljön (pl. B4), és melyik irányba nézzen. A drón négy irányba nézhet, felfele (F), jobbra (J), lefele (L), és balra (B). A drón nem lehet olyan helyen, ahol van épület.

Lássunk egy példát. Ha a drón a B4J parancsot kapja, akkor a B4-es cellába repül és jobbra néz. Ekkor a legnagyobb ház, amire rálát, 6 emelet magas, így a 6-os értéket fogja visszaadni. Ha a B4B parancsot kapja, akkor ugyanebben a cellában marad, de balra néz, ahol nem lát épületet, így a nulla értéket adja vissza.

Az E4J,F2L parancsokra érvényes, hogy mindegyik 6-os értéket ad vissza. (De sok más parancsot ki lehetne adni, amelyre igaz, hogy 6-os értéket ad vissza…)

Most nézzük az alábbi pályát!

|  |  |
| --- | --- |
| Adj meg olyan parancsokat, amelyek kiadásakor igazak lesznek az alábbi állítások.  Ha több olyan parancs is van, ami megfelelő lehet, add meg mindegyiket, vesszővel elválasztva.  Ha esetleg nincs megfelelő parancs, írd azt, hogy Nincs. | |
|  | |  |  |  | | --- | --- | --- | | **Kérdés** | **Állítás** | **Parancs(ok)** | | A. | A visszaadott érték 5 és lefele néz a drón |  | | B. | A visszaadott érték 1 |  | | C. | A visszaadott érték 6 de ha jobbra nézne a drón, akkor 5 lenne az érték |  | | D. | A visszaadott érték 3, de ha jobbra nézne, akkor is 3 lenne |  | | E. | A visszaadott érték 2 |  | | F. | A visszaadott érték az ezen a pályán elérhető maximális érték, és felfele néz a drón |  | |

2. feladat: Négyszögek (40 pont)

Egy négyszöghöz azonos méretű kisebb négyszögeket teszünk. Használunk egy eljárást:

def tégla(a,b):  
 for i in range(2):  
 turtle.forward(a)  
 turtle.left(120)   
 turtle.forward(b)   
 turtle.left(60)

Melyik programhoz melyik ábra tartozik?

A. def egy(a,b):  
 for i in range(2):  
 tégla (a,b)  
 turtle.forward(3\*a)   
 tégla(a,b)  
 turtle.forward(a)   
 turtle.right(60)  
 turtle.forward(3\*b)   
 turtle.right(120)

B. def kettő(a,b):   
 for i in range(2):  
 tégla(b,a)   
 turtle.forward(2\*a)   
 tégla(b,a)  
 turtle.forward(2\*a)   
 turtle.right(60)  
 turtle.forward(3\*b)   
 turtle.right(120)

C. def három(a,b):   
 for i in range(2):  
 tégla(b,a)   
 turtle.forward(4\*a)   
 turtle.right(60)  
 tégla(b,a)   
 turtle.forward(3\*b)   
 turtle.right(120)

D. def négy(a,b):   
 for i in range(2):  
 turtle.forward(a)   
 tégla(a,b)   
 turtle.forward(a)   
 tégla(a,b)   
 turtle.forward(2\*a)  
 turtle.right(60)   
 turtle.forward(3\*b)   
 turtle.right(120)

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 1 | 2 | 3 | 4 |

Mit rajzol a négy eljárás a következő hívásokra, ha a tégla eljárásban a balra helyett jobbra fordulunk?

E. egy(30,60)

F. kettő(30,60)

G. három(30,60)

H. négy(30,60)

3. feladat: Rovásírás (30 pont)

A rovásírás egyes betűi egyenes vonalakból állnak. Az alábbi eljárásokból hiányzik egy-egy utasítás. Ezek helyét egy üres dobozzal jelöltük. Add meg a hiányzó utasításokat úgy, hogy a mellékelt betűket rajzolja ki az eljárás.

 def ibetű(h):   
 turtle.forward(h)   
 turtle.backward(h/4)   
 turtle.left(105)   
 turtle.forward(h/4)   
 turtle.backward(h/2) f  
 turtle.right(105)   
 turtle.backward(3\*h/4)

 def mbetű(h):  
 turtle.forward(h)   
 for i in range(2):  
 x   
 turtle.forward(h/2)   
 turtle.right(120)   
 for i in range(2):   
 turtle.forward(h/2)   
 turtle.left(120)

 def rbetű(h):   
 turtle.forward(h)   
 turtle.backward(3\*h/5)   
 turtle.right(66)   
 turtle.forward(h/2)   
 turtle.left(66)   
 turtle.forward(2\*h/5)   
 x   
 turtle.forward(3\*h/5)   
 turtle.right(66)   
 turtle.backward(h/2)   
 turtle.left(66)   
 turtle.backward(2\*h/5)