1. feladat: Párosítás (24 pont)

A következő algoritmusokban J hatására a teknőc jobbra fordul 60 fokot, B hatására pedig balra. Az E hatására előre megy 10 egységnyit.

Melyik algoritmushoz melyik kép tartozik (van amelyik képet több algoritmus is rajzolja és van, amelyiket egyik sem)?

A. ismétlés 6 [E E B ismétlés 3 [E J] B B E J]

B. ismétlés 6 [ismétlés 3 [B E] B B E E E E]

C. ismétlés 6 [E E E E J ismétlés 3 [E B] J]

D. ismétlés 6 [E E E E B ismétlés 3 [E J] B B B]

E. ismétlés 6 [E E E E B ismétlés 3 [J E] B B B]

F. ismétlés 6 [J ismétlés 3 [B E] B B E E E E B]

|  |  |  |
| --- | --- | --- |
| 1: | 2: | 3: |
| 4: | 5. | 6: |

2. feladat: Mit rajzol (30 pont)

Mit rajzolnak az alábbi eljárások?

A. for i in range(3):   
 for j in range(2):   
 valami(10)   
 turtle.right(60)   
 for j in range(2):   
 valami(10)   
 turtle.left(120)

B. for i in range(3):   
 for j in range(2):   
 valami(10)   
 turtle.right(60)   
 turtle.forward(20)   
 for j in range(2):   
 turtle.left(120)  
 valami(10)

C. for i in range(3):   
 for j in range(2):   
 valami(10)   
 turtle.right(60)   
 for j in range(2):   
 turtle.left(120)   
 valami(10)

D. for i in range(3):   
 for j in range(2):   
 valami(10)   
 turtle.right(60)   
 for j in range(2):   
 turtle.left(120)   
 valami(10)   
 turtle.forward(20)

E. for i in range(3):   
 for j in range(2):   
 valami(10)   
 turtle.right(60)   
 turtle.left(120)   
 for j in range(2):   
 valami(10)   
 turtle.left(120)   
 turtle.forward(20)   
 jobbra 120

F. for i in range(6):   
 for j in range(2):   
 valami(10)   
 turtle.right(60)   
 turtle.forward(20)   
 turtle.left(120)   
 for j in range(2):   
 valami(10)   
 turtle.left(120)   
 turtle.forward(20)   
 turtle.right(120)

def valami(h):  
 turtle.forward(h)   
 turtle.left(60)   
 turtle.forward(h)   
 turtle.right(120)  
 turtle.forward(h)   
 turtle.left(60)   
 turtle.forward(h)

3. feladat: Függvény (24 pont)

A vala függvény egy szót kap paraméternek.

def vala(s):  
 if len(s)==0:   
 return s  
 if len(s[1:])==0:  
 return s  
 if s[0]<s[-1]:  
 return s[0]+vala(s[1:len(s)-1])  
 if s[0]>s[-1]:   
 return s[-1]+vala(s[1:len(s)-1])   
 if s[0]==s[-1]:  
 return s[0]+s[-1]+vala(s[1:len(s)-1])   
Mit ad a függvény eredményül az alábbi hívásokra?

A. vala ″abcdefg

B. vala ″abcdabcd

C. vala ″abcddcba

D. vala ″xy

E. vala ″rpg

F. vala ″aeiouklmuoiea

G. Fogalmazd meg általánosan, hogy mi a függvény feladata!

4. feladat: Hiány (32 pont)

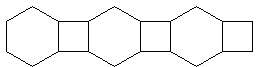
Az alábbi eljárások a háromszög, a négyzet és a hatszög eljárást használják. Kezdetben a teknőc mindegyiknél felfelé (északra) néz.

def háromszög(h):  
 for i in range(3):  
 turtle.forward(h)   
 turtle.righ(120)

def négyzet(h):  
 for i in range(4):   
 turtle.forward(h)   
 turtle.righ(90)

def hatszög(h):  
 for i in range(6):   
 turtle.forward(h)   
 turtle.righ(60)

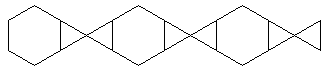
A jelölt helyekről kitöröltük a balra vagy jobbra fordulásokat. Mi lehet a helyükön?

A. 

for i in range(3):   
 hatszög(10)  
 #####   
 turtle.backward(10)  
 #####   
 turtle.forward(10)  
 #####   
 négyzet(10)   
 #####   
 turtle.forward(10)  
 #####

B. 

for i in range(3):   
 #####  
 hatszög(10)   
 for j in range(2):   
 turtle.forward(10)  
 #####  
 turtle.forward(10)  
 #####   
 négyzet(10)   
 turtle.forward(10)  
 #####  
 turtle.forward(10)  
 #####

C. 

for i in range(3):   
 hatszög(10)   
 #####   
 turtle.backward(10)   
 #####   
 turtle.forward(10)  
 #####   
 háromszög(10)   
 #####   
 turtle.forward(10)  
 háromszög(10)   
 #####   
 turtle.forward(10)   
 #####