



# Programozási nyelvek

## 3. előadás





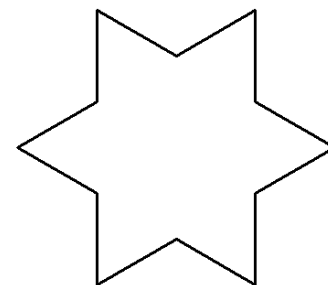
# Logo – csillagok



## Csillag egyenlő oldalú háromszögekből:

A lényeg: az ismétlődő részben a fordulatok összege legyen annyi, mint a megfelelő sokszögnél a külső szög!

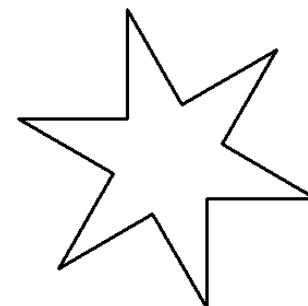
ismétlés 6 [előre 100 jobbra 120  
előre 100 balra 60]



## Csillag egyenlő szárú háromszögekből:

Ne törődjünk a sokszög oldalhosszával, a csillagot a saját oldalhosszával és a külső szögeivel adjuk meg!

ismétlés 6 [előre 100 jobbra 150  
előre 100 balra 90]





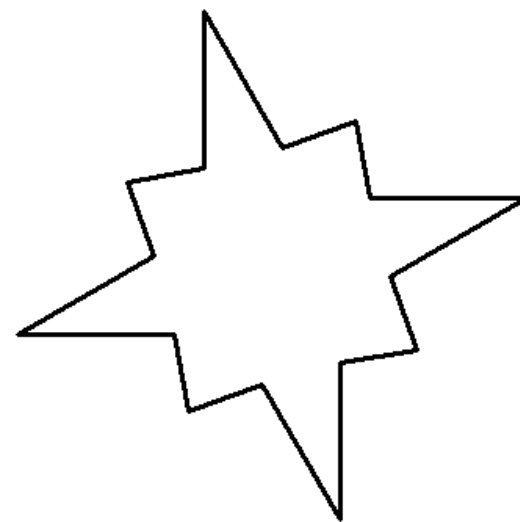
# Logo – csillagok



## Csillag többféle háromszögből:

Minden második háromszög legyen más oldalhosszú és más szögű! Egy ciklus lépésben két ágat rajzolunk, arra kell tehát figyelni, hogy a fordulatok összege a sokszög külső szögei kétszerese legyen!

```
ismétlés 4 [előre 100 jobbra 150  
előre 100 balra 80  
előre 50 jobbra 100  
előre 50 balra 80]
```





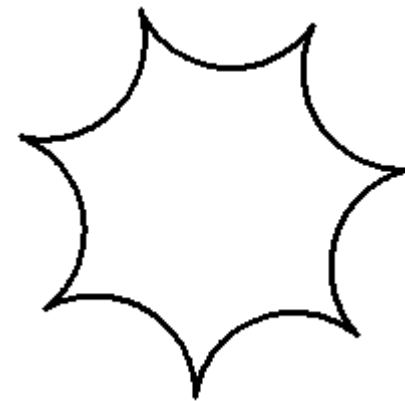
# Logo – csillagok



## Csillag körívekből:

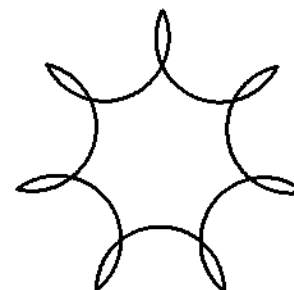
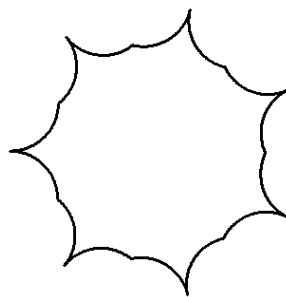
A háromszögeket persze bármilyen más alakzattal helyettesíthetjük, akár például körívekkel is!

```
ismétlés 7 [balív 60 50 1  
            balív 60 50 -1  
            jobbra 360/7+120]
```



```
eljárás balív :fok :r :l  
ismétlés :fok [előre :l*:r*3,14159/180 balra 1]  
vége
```

Ezek hogyan készülhettek:





# Logo – csillagok



## Sokszög oldalai középre illesztett sokszög:

Egy szabályos sokszög oldalai középre is tehetünk másik szabályos sokszöget, harmad akkora méretben!

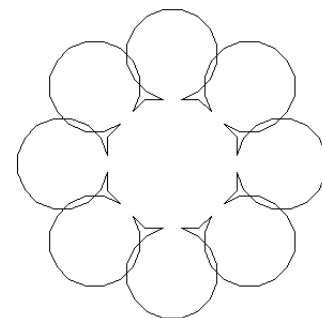
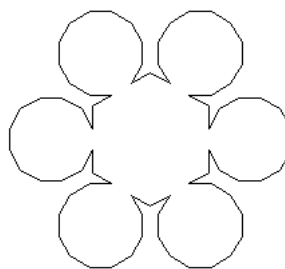
```
ismétlés :m [előre :h/3 balra 180-360/:n  
sok :n :h jobbra 180-360/:n  
előre :h/3 jobbra 360/:m]
```

```
eljárás sok :n :h
```

```
ismétlés :n-1 [előre :h jobbra 360/:n]
```

```
jobbra 360/:n
```

```
vége
```





# Logo – csillagok



## Sokszög csúcsaira illesztett sokszög:

Egy szabályos sokszög csúcsaira is tehetünk másik szabályos sokszöget (ettől az oldalak eltávolodnak egymástól)!

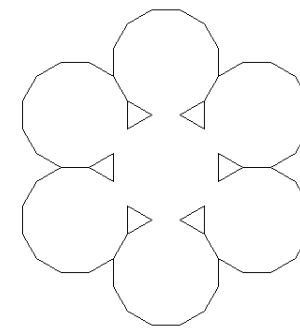
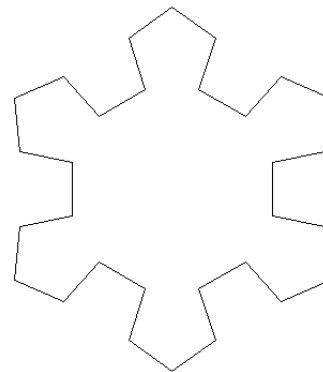
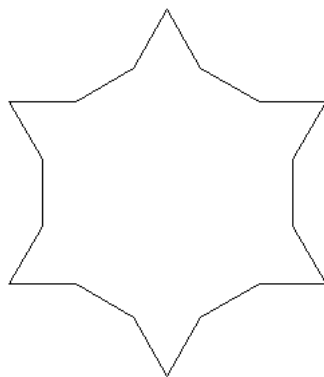
```
ismétlés :m [előre :h balra 180-360/:n-180/:m  
sok :n :h jobbra 180-360/:n+180/:m]
```

```
eljáráás sok :n :h
```

```
ismétlés :n-1 [előre :h jobbra 360/:n]
```

```
jobbra 360/:n
```

vége





# Logo – csillagok



## Sokszög csúcsaira illesztett sokszög:

Egy szabályos sokszög csúcsaira is tehetünk másik szabályos sokszöget, befelé illesztve!

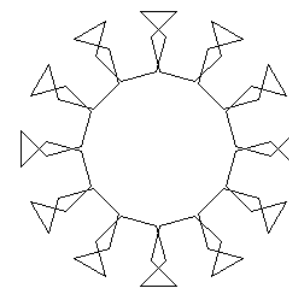
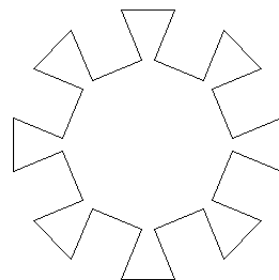
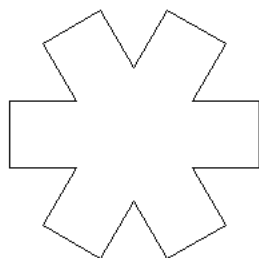
```
ismétlés :m [előre :h jobbra 180-360/:n+180/:m  
sok :n :h balra 180-360/:n-180/:m ]
```

```
eljárás sok :n :h
```

```
ismétlés :n-1 [előre :h balra 360/:n]
```

```
balra 360/:n
```

vége





# Logo – sokszög variációk



## Sokszög rekurzívan

Az  $N$  oldalú sokszögvonal 1 oldalból és egy  $N-1$  oldalú sokszögvonalból áll.

eljárás reksokszög :n :hossz :szög

előre :hossz balra :szög

ha :n>1 [reksokszög :n-1 :hossz :szög]

vége







# Logo – sokszög variációk



## Spirál

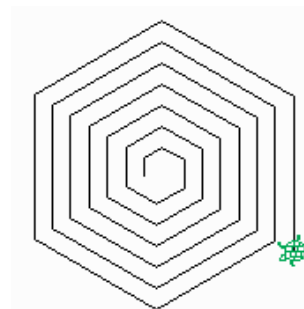
Mindig egy kicsit nagyobb távolságot tegyünk meg!

```
eljárás spirál :n :hossz :szög :táv
```

```
  előre :hossz balra :szög
```

```
  ha :n>1 [spirál :n-1 :hossz+:táv :szög :táv]
```

```
vége
```





# Logo – sokszög variációk



## Spirál

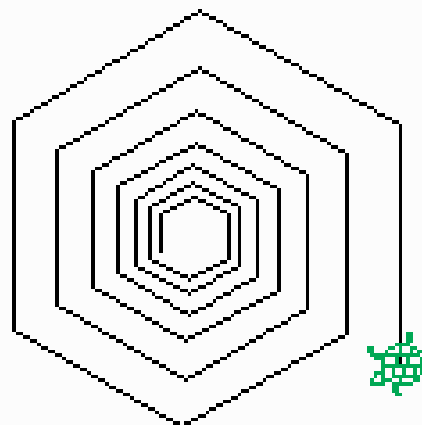
A távolságot szorozzuk egy konstanssal!

```
eljárás spirál :n :hossz :szög :szor
```

```
  előre :hossz balra :szög
```

```
  ha :n>1 [spirál :n-1 :hossz*:szor :szög :szor]
```

```
vége
```





# Logo – sokszög variációk



## Inda

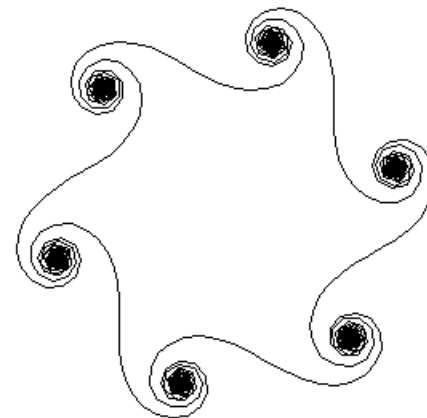
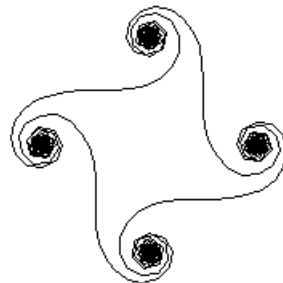
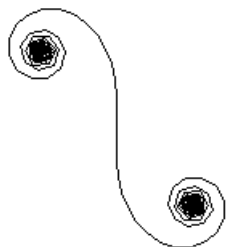
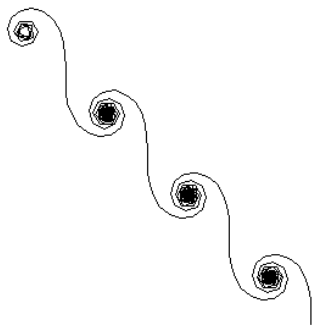
A szöget növeljük meg egy konstanssal!

eljárás inda :n :hossz :szög :növekedés

előre :hossz balra :szög

ha :n>1 [inda :n-1 :hossz :szög+:növekedés :szor]

vége



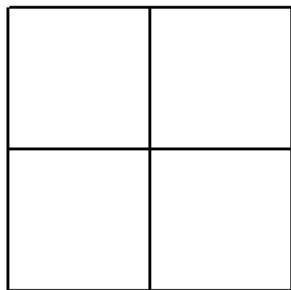


# Logo – sorminták

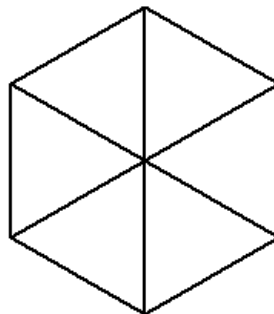


*Geometriai emlék:* a síkot háromféle szabályos sokszöggel fedhetjük le – feltétele: 360 fok osztható legyen a belső szöggel.

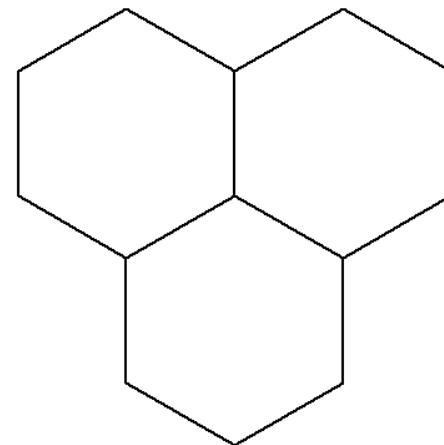
négyzettel



háromszöggel



hatszöggel



A fentiek szerint készíthetünk belőlük sormintákat, majd pedig lefedhetjük velük a síkot is.

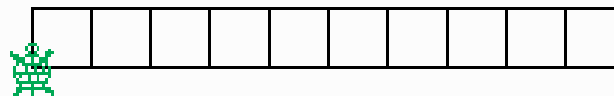




# Logo – sorminták



**Sorminta azonos elemekből** (ciklikus programok tételének alkalmazása)



Négyzet esetén `:x=:y`.

```
eljárás sorminta :n :x :y
```

```
  ismétlés :n [alapelem :x :y elmozdulás :x]
```

```
  elmozdulás -:n*:x
```

vége

```
eljárás alapelem :x :y
```

```
  ismétlés 2 [előre :y jobbra 90 előre :x jobbra 90]
```

vége

```
eljárás elmozdulás :x
```

```
  tollatfel jobbra 90 előre :x balra 90 tollatle
```

vége

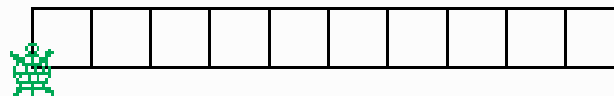




# Logo – sorminták



## Sorminta rekurzívan



Egy  $n$  elemű sorminta 1 elemből és egy  $n-1$  elemű sormintából áll:

```
eljárás sorminta :n :x :y
  ha :n>0 [alapelem :x :y elmozdulás :x
           sorminta :n-1 :x :y elmozdulás -:x]
vége
```





# Logo – sorminták



## Sorminta nem „sorban”

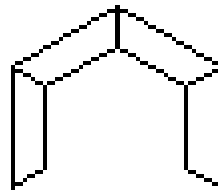
A sorminta elemeinek nem kell feltétlenül egyenes mentén elhelyezkedniük:

```
eljárás sor :n :m :o
```

```
ismétlés :n [alap :m :o előre :m balra 60]
```

```
ismétlés :n [jobbra 60 hátra :m]
```

vége





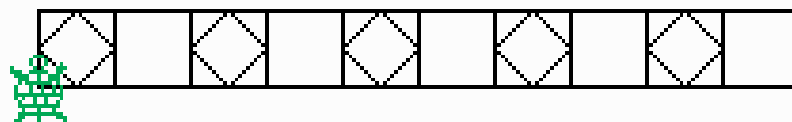
# Logo – sorminták



## Sorminta kétféle alapelemből

Alkalmazzunk közvetett rekurziót!

```
eljárás sorminta :n :x :y
  ismétlés :n [alapelem1 :x :y elmozdulás :x
              alapelem2 :x :y elmozdulás :x]
  elmozdulás - :x* :n
vége
```



Megjegyzés: A megoldás csak akkor jó, ha a kétféle alapelemből ugyanannyi van. Lehetne a végén:

Ha páratlan? :n [...]







# Logo – sorminták

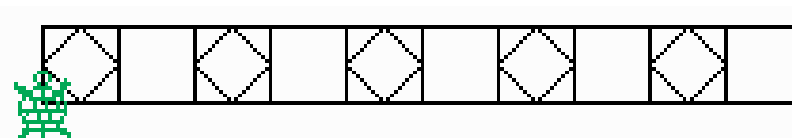


## Sorminta kétféle alapelemből

Alkalmazzunk közvetett rekurziót!

```
eljárás sorminta1 :n :x :y
  ha :n>0 [alapelem1 :x :y elmozdulás :x
           sorminta2 :n-1 :x :y elmozdulás -:x]
vége
```

```
eljárás sorminta2 :n :x :y
  ha :n>0 [alapelem2 :x :y elmozdulás :x
           sorminta1 :n-1 :x :y elmozdulás -:x]
vége
```



Megjegyzés: A kétféle alapelem lehetne különböző méretű is! Ekkor kétféle elmozdulás kellene.





# Logo – sorminták

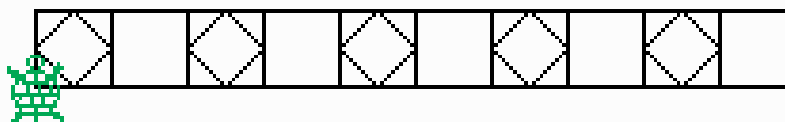


## Sorminta kétféle alapelemből

Működik közvetlen rekurzióval is, ekkor azonban tudni kell – egy logikai paraméter értékéből –, hogy melyik alapelemet kell éppen rajzolni!

```
eljárás sorminta :n :x :y :lo
  ha :n>0 [hak :lo [alapelem1 :x :y]
            [alapelem2 :x :y]
            elmozdulás :x
            sorminta :n-1 :x :y nem :lo
            elmozdulás -:x]
```

vége





# Logo – sorminták

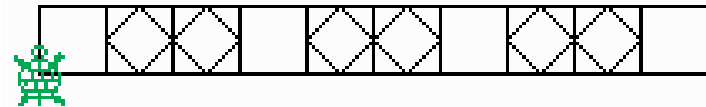


## Sorminta kétféle alapelemből

Ha nem felváltva van a kétféle elem, akkor a logikai értéket sorszámra cseréljük.

```
eljárás sorminta :n :x :y :szám
  ha :n>0 [hak :szám mod 3=0 [alapelem1 :x :y]
          [alapelem2 :x :y]
          elmozdulás :x
          sorminta :n-1 :x :y :szám+1
          elmozdulás -:x]
```

vége



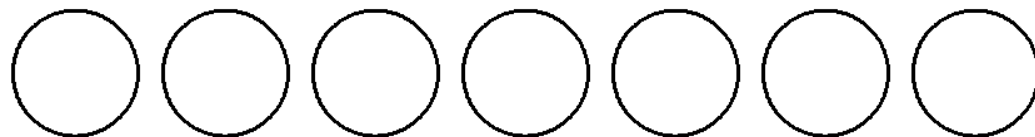
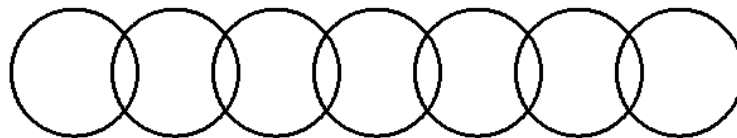
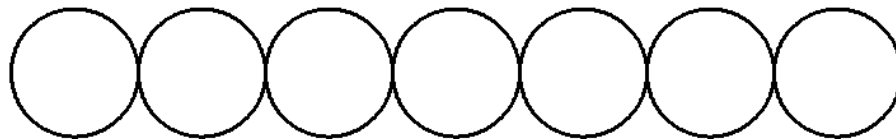


# Logo – sorminták



## Sorminta körökből

Az állapotekvivalens helyettesítés elvén a négyzeteket lecserélhetjük bármely olyan ábrára, amely ugyanazt az állapotváltozást okozza. Az alapelemek összezsúszhatnak, illetve szét is húzódnak, ha az elmozdulás mértéke nem azonos az alapelem szélességével.



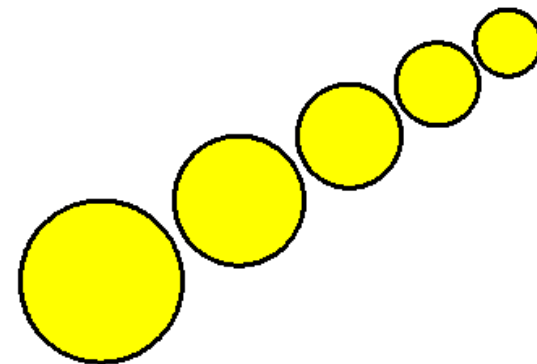


# Logo – sorminták



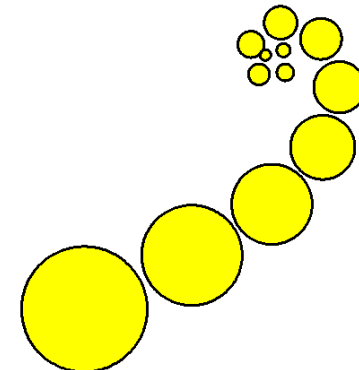
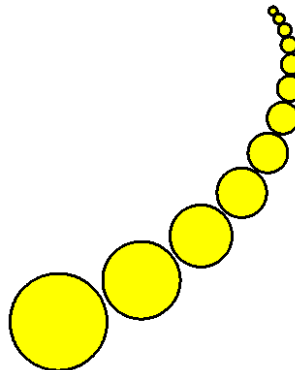
## Távolodó alakzatok

Ha az egyes elemeket nem egymás mellé tesszük, hanem pl. jobbra és felfelé is elmozdulunk, valamint az alapelem méretét adott százalékkára csökkentjük, akkor egy távolodó alakzatsort kapunk.



Mivel a méret mindig más, ezért persze csak a rekurzív megoldás jöhet szóba.

Ezek hogyan készülhettek?





# Logo – sorminták



## Sorminta más alakzatokon

A sorminta elhelyezhető más alakzatokra is.

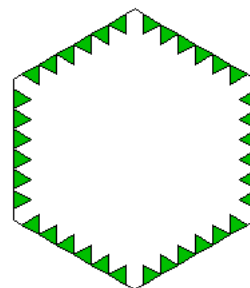
eljárás sok :n :db :h

ismétlés :n [előre :h/2

ismétlés :db [háromszög :h előre :h]

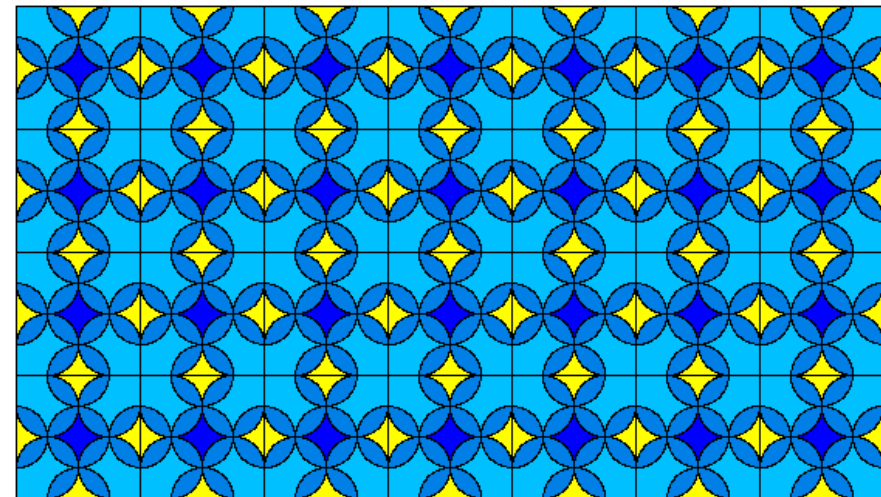
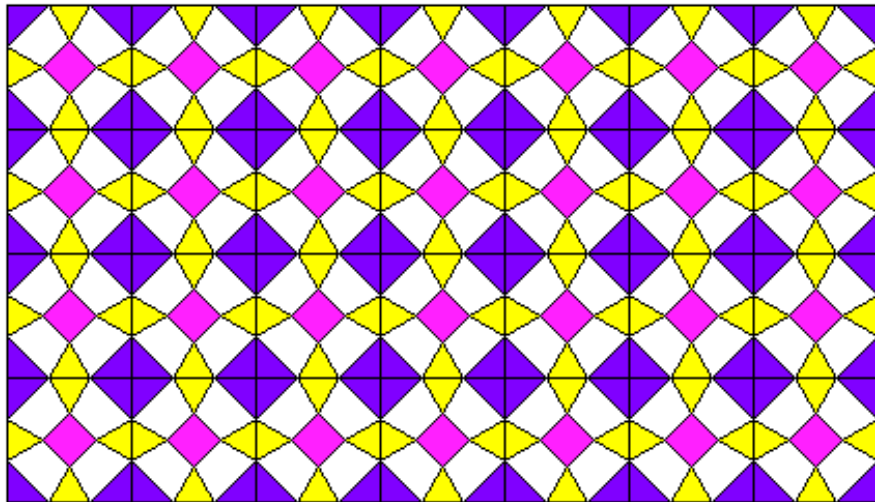
előre :h/2 jobbra 360/:n]

vége





# Logo – Mozaik





# Logo – Mozaik



## Mozaik

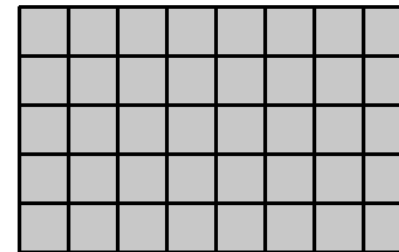
A legegyszerűbb mozaik a négyzetrács, adott számú sorból és oszlopból áll. Megírható ciklussal is és rekurzívan is.

```
eljárás mozaik :m :n :x
  ismétlés :m [sorminta :n :x felfelé :x]
  lefelé :m* :x
```

vége

```
eljárás mozaik :m :n :x
  ha :m>0 [sorminta :n :x felfelé :x
          mozaik :m-1 :n :x lefelé :x]
```

vége







# Logo – Mozaik



## Mozaik látszólag kétféle elemből

Az alábbi mozaik látszólag kétféle elemből áll (nyolcszögekből, illetve négyzetekből), pedig csak nyolcszögeket rajzolunk.

```
eljárás mozaik :m :n :x
```

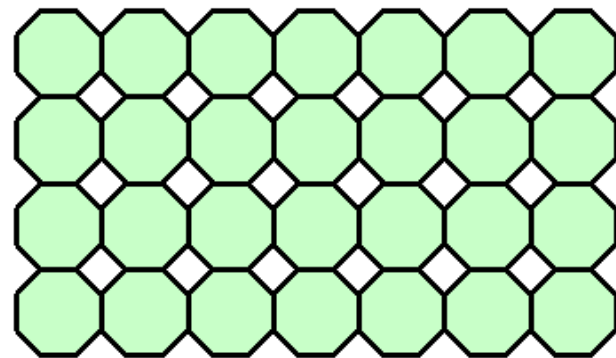
```
  ha :m>0 [sorminta :n :x felfelé :x
```

```
          mozaik :m-1 :n :x lefelé :x]
```

vége

Miért a rekurzív változatot használjuk?

A felfelé és a lefelé eljárás most a nyolcszög oldalain megy végig. Ha  $:m^* :x$  lépést tenne meg, akkor mást kapnánk.





# Logo – Mozaik



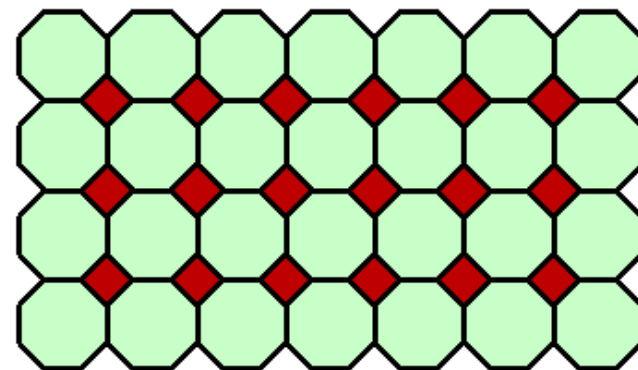
## Mozaik látszólag kétféle elemből

Ha csak nyolcszögeket rajzolunk, akkor hogyan készülhetett ez az ábra?

A piros részeket a mozaik elkészülte után színeztük be, mintha eggyel keve-

sebb sor- és oszlopszámú másik mozaikot rajzoltunk volna, az eredetihez képest megfelelően eltolva, amiben az alapelem a *semmi* pirosra színezve.

Mit kellene tenni, ha a széleken levő háromszögeket is pirosra szeretnénk festeni? Nagy piros téglalapra rajzoljuk rá a mozaikot!



töltőmód! 1 – a tollszínnel határolt területet festi





# Logo – Mozaik

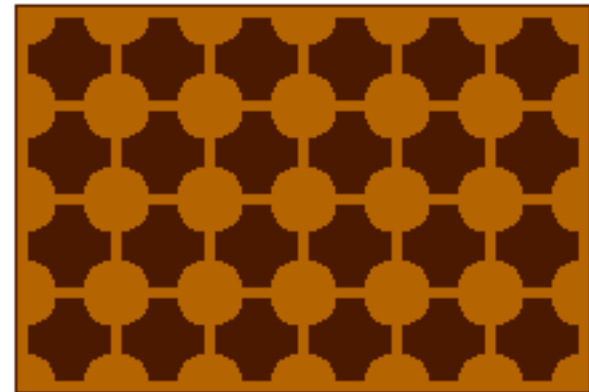


## Mozaik egyféle elemből, alaplagra

A mozaikot ráhelyezhetjük tetszőleges alakzatra, pl. téglalagra:

```
eljárás mozaik :n :m :h  
ismétlés :n [sor :m :h tollatfel előre 3,5*:h tollatle]  
tollatfel hátra :n*3,5*:h+1,5*:h balra 90  
előre :h/2 jobbra 90 tollatle  
téglalap :n*3,5*:h+:h/2 :m*3,5*:h+:h/2  
vége
```

Lehetne előbb a téglalap, amire  
rárajzolnánk a mozaikot.





# Logo – Mozaik



## Mozaik kétféle elemből

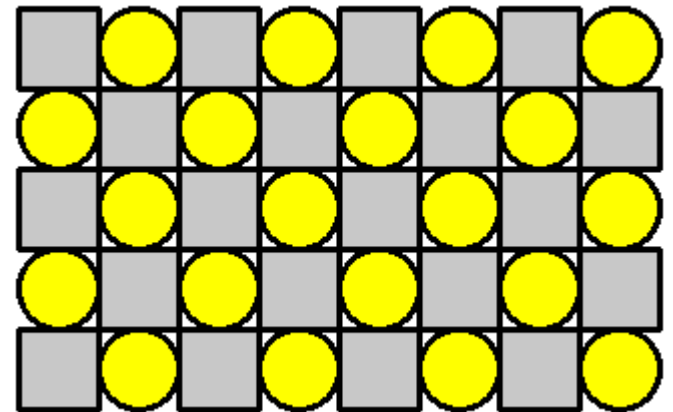
Ugyanazt az elvet követhetjük, mint a sormintánál, azaz csak a rekurzív megoldás működik.

```
eljárás mozaik :m :n :x :lo
```

```
  ha :m>0 [sorminta :n :x :lo felfelé :x
```

```
          mozaik :m-1 :n :x nem :lo lefelé :x]
```

```
vége
```





# Logo – Mozaik



## Mozaik nem téglalap alapon, növekvő méretben

Ha a sorminta elemei nem egyenes vonal mentén helyezkednek el, akkor mozaik is különlegesebb alakzatot fedhet le:

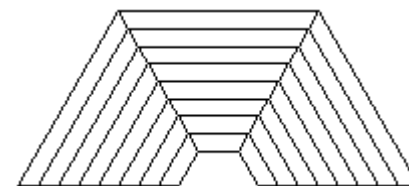
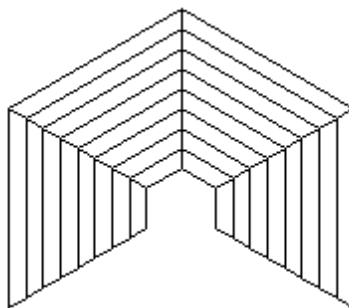
eljárás mozaik :m :n :méret :oldal

  sor :n :méret :oldal

  ha :m>1 [jobbra 120 előre :oldal balra 120

    mozaik :m-1 :n :oldal+:méret :oldal]

vége





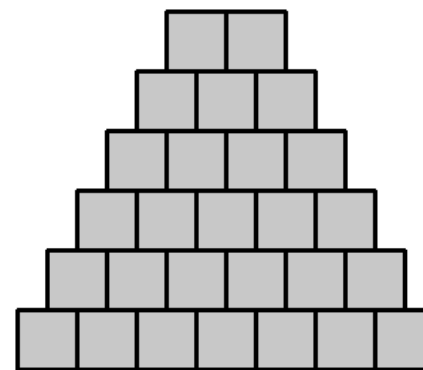
# Logo – Mozaik



## Piramis

Ugyancsak rekurzív megoldást kell írunk, ha a sorok hosszát szeretnénk változtatni.

```
eljárás piramis :m :n :x  
  ha :m>0 [sorminta :n :x felfelé :x  
          piramis :m-1 :n-1 :x lefelé :x]  
vége
```





# Logo – Mozaik



## Rombusz-mozaik

Mozaikot nemcsak négyzetekből (kis változtatással téglalapokból), hanem pl. rombuszokból is készíthetünk.

```
eljárás rmozaik :m :n :x :sz  
  ha :m>0 [rsor :n :x felfelé :x  
    rmozaik :m-1 :n :x :sz lefelé :x]
```

vége

A felfelé és a lefelé eljárás, valamint az alapelem nyilván változik. A fő eljárás ugyanaz, a sor sem változik,

ha a rombusz rajzolásra beépítjük a megfelelő kezdő fordulást.





# Logo – Mozaik



## Rombusz-mozaik

eljárás rsor :n :x :sz

  ha :n>0 [rombusz :x :sz elmozdulás :x

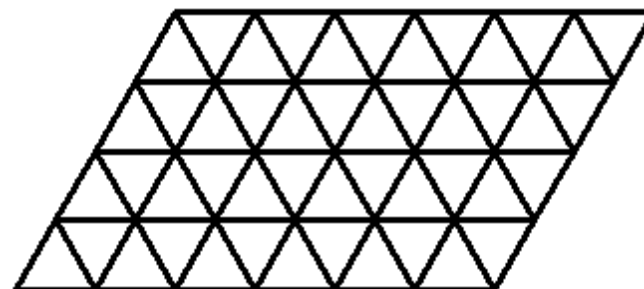
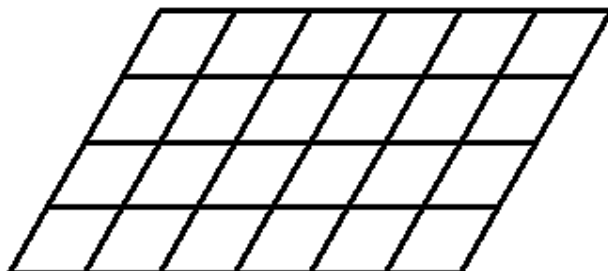
    rsor :n-1 :x :sz elmozdulás -:x]

vége

## Háromszög-mozaik

Ha 60 fokos szöget választunk a rombuszhoz, akkor érdekes lehetőség adódik a rombusz egyik átlójának behúzásával.

Megkapjuk a háromszög-mozaikot:







# Logo – Mozaik

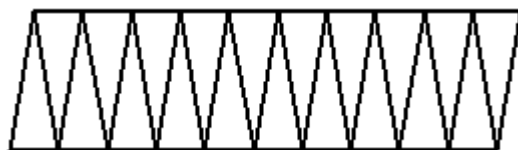


## Háromszög-mozaik

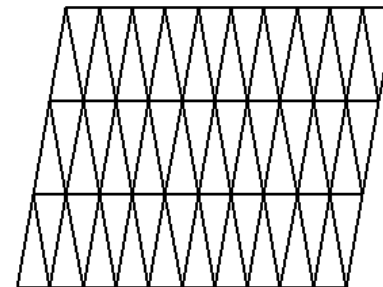
Ha nem egyenlő oldalú háromszögekből építkezünk, akkor máshogy kell eljárunk. Készítsünk kétféle háromszögsort!



A bal szélüket könnyű egymáshoz illeszteni:



Belőlük pedig már csak egy lépés a mozaik:





# Logo – Mozaik



## Háromszög-mozaik

A háromszög eljárás – most először – ne legyen állapotátlat-  
szó, sőt csak a háromszög két oldalát rajzoljuk meg!



eljárás háromsz :h :sz

jobbra :sz/2 előre :h jobbra 180-:sz

előre :h balra 180-:sz/2

vége

Ha a bal szélén kezdjük a rajzolást és a jobb szélén fejezzük be,  
akkor nagyon könnyű ilyeneket egymás mellé tenni:





# Logo – Mozaik



## Háromszög-mozaik

Most már csak az alsó oldalakat kell megrajzolni. Ha nem tudjuk kiszámolni a hosszát, akkor alkalmazzuk újra az xypoz! utasítást!

eljárás hsor :n :h :sz :x :y :i

ismétlés :n [háromsz :h :sz]

xypoz! :x :y irány! :i

vége



A felső részéhez:

jobbra :sz/2 előre :h jobbra 180-:sz

hsor :n :h :sz xpoz ypoz irány



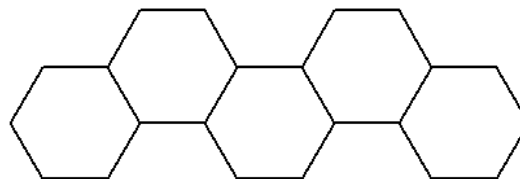
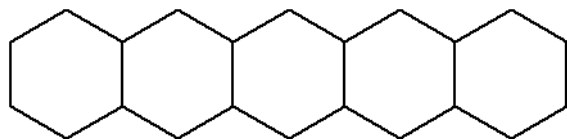


# Logo – Mozaik

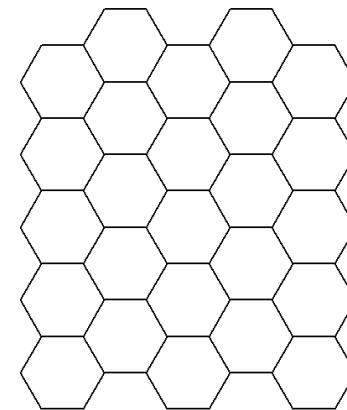
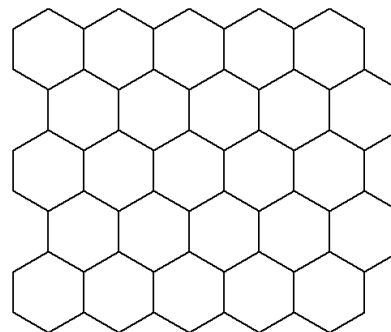


## Hatszög-mozaik

A hatszög-mozaiknál meg kellene gondolni, hogy nézzen ki egy sor! Két alapvető lehetőségünk van:



Ha azonban mozaikot rajzolunk belőlük, azonnal látszik, hogy egymás 90 fokkal elforgatottjai:



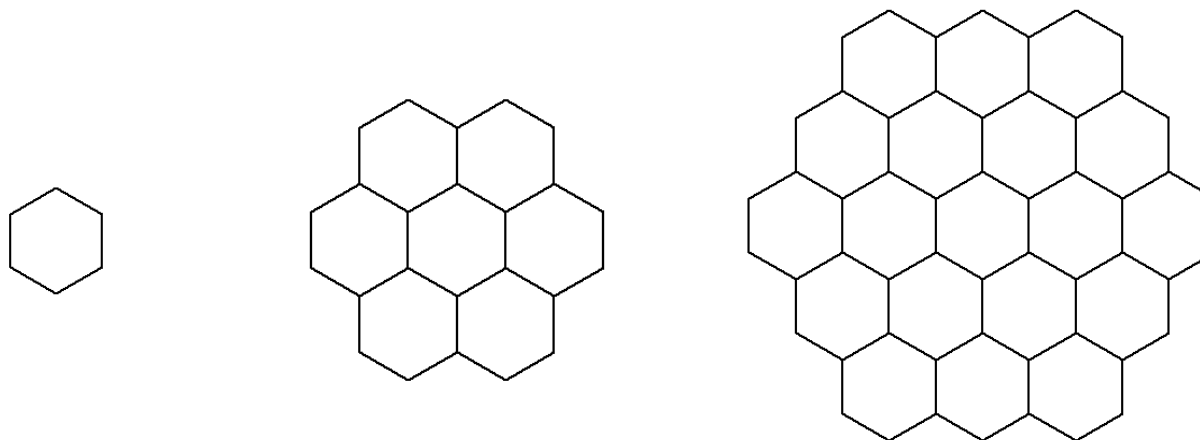


# Logo – Mozaik



## Hatszög-mozaik

A hatszög-mozaikot azonban másképp is elképzelhetjük: hatszögek valahány körben helyezkednek el egymás körül:



A következő körben soronként 4-5-6-7-6-5-4 hatszög lenne. Tehát a megoldás két piramis rajzolása hatszögsorokból.



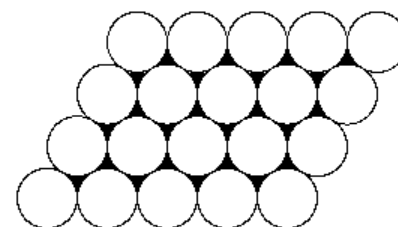
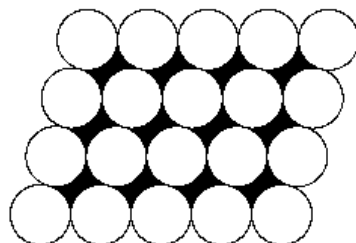
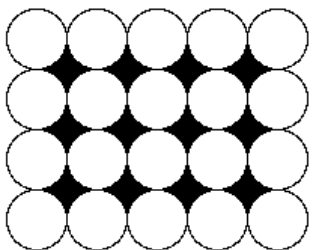


# Logo – Mozaik



## Körmozaik

Körmozaiknál a kérdéses a sík körökkel és köztük levő közökkel kitöltése. Variációk:



Itt is két menetben rajzolunk. Először a körökből készül mozaik, utána pedig ehhez képest eltolva, üres alapelemekből, festéssel töltjük ki a közöket.

Másik megoldás: nagy fekete téglalapra rajzolunk körmozaikot, fehérrel töltve a köröket.



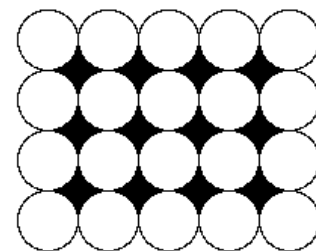


# Logo – Mozaik



## Körmozaik

Megoldási ötlet: a második körsor rajzolásánál lehet festeni a körközöket:



eljárás körmozaik :m :n :r :sz

tollatfel körsor :n :r jobbra :sz előre  $2^* :r$

balra :sz

ismétlés :m-1 [körsor :n :r sorokközött :n-1 :r

jobbra :sz előre  $2^* :r$  balra :sz]

jobbra :sz hátra  $2^* :m^* :r$  balra :sz tollatle

vége





# Logo – Mozaik



## Körmozaik

eljárás körsor :n :r

ismétlés :n [kör :r jobbra 90 előre 2\*:r balra 90]

jobbra 90 hátra 2\*:n\*:r balra 90

vége

eljárás sorokközött :n :r

ismétlés :n [körív :r 200 balra 90 előre 1 tölt

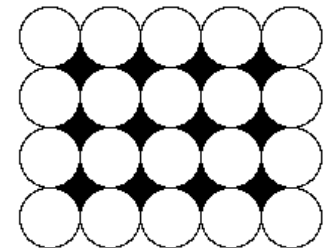
hátra 1 jobbra 90 körív :r 60 balra 90 előre 1

tölt hátra 1 jobbra 90 körív :r 100

jobbra 90 előre 2\*:r balra 90]

jobbra 90 hátra 2\*:n\*:r balra 90

vége







# Logo

