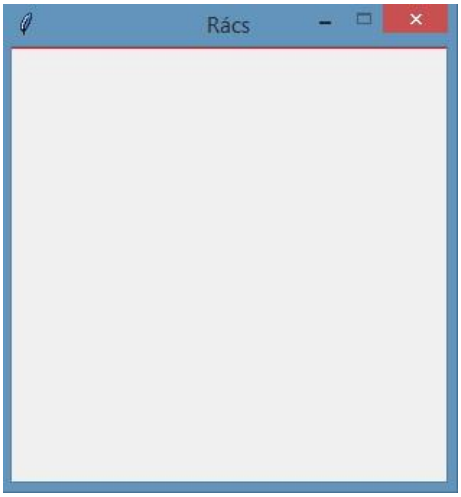
Grafikus programozás
Python nyelven 3.

Rács feladat

A feladat: rajzoljunk négyzetrácsozást, ahol a párhuzamos vonalak távolsága 10 pixel.

Haladjunk lépésenként! Először is rajzoljuk ki egy vonalat legfelülre vízszintesen! Ezt most pirosra színezem, csak hogy jobban látszódjon, később ez fekete lesz.

```
from graphics import *

w = GraphWin('Rács', 301, 301)

l = Line(Point(0, 0), Point(300, 0))
l.setOutline('red')
l.draw(w)
```



Gondolkozzunk el, hogy vajon hogyan tudjuk a többi vízszintes vonalat is kirajzolni, egymástól mondjuk 10 pixel távolságra. Ez 31 vonalat jelent – hiszen az y koordináták 0-tól 300-ig szerepelhetnek –, ami már sok ahhoz, hogy egyesével rajzoljuk meg őket, valahogy így:

```
l = Line(Point(0, 0), Point(300, 0))
l.draw(w)

l = Line(Point(0, 10), Point(300, 10))
l.draw(w)

l = Line(Point(0, 20), Point(300, 20))
l.draw(w)
...
```

Figyeljük meg, hogy két sort szeretnénk ismételni 31-szer, amik csak kevésben térnek el egymástól, ezt pirossal van jelölve. Ilyenkor érdemes használni valamilyen ciklust, ami elvégzi helyettünk az ismétlést, nekünk elég csak egyszer megírnunk ezt a két sort. Viszont ahhoz, hogy a Line objektum két paraméteréül adott pontnál változzon az y koordináta, ahhoz először is azok a helyére egy változót kell írni, utána pedig gondoskodni kell arról, hogy annak az értéke valóban meg is változzon.

Erre ad megoldást például a számlálós (for) ciklus is, hiszen annál meg kell adni egy ciklusváltozót (y), aminek a változtatási szabályát az első sorában le kell írni. Ehhez használható a range parancs. Pl. így:

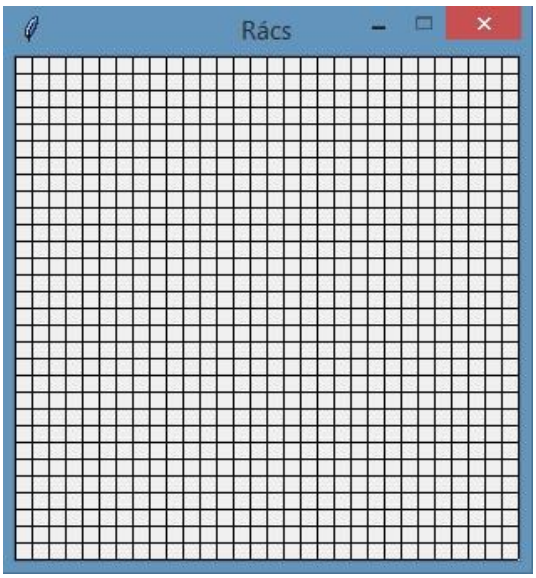
```
...
for y in range(0,301,10) :
    l = Line(Point(0, y), Point(300, y))
    l.draw(w)
```

Ebben az esetben az y kezdőértéke 0, és minden alkalommal 10-zel nő egészen addig, amíg y < 301. Tehát a mi példánkban a 300 a legnagyobb érték amit felvesz. Az eredmény:

Ugyanez a helyzet a függőleges szakaszokkal is, csak most az x koordinátákat akarjuk változtatni, hiszen a függőlegesek egymástól jobbra-balra helyezkednek el:

```
...
for x in range(0,301,10) :
    l = Line(Point(x, 0), Point(x, 300))
    l.draw(w)
```

Végül az eredmény:



Abrosz feladat

A piros-fehér kockás abrosz mintájának egy egyszerűbb változatát fogjuk megrajzolni: csak a piros és fehér színű négyzeteket fogunk kirajzolni felváltva, a pirosnak más árnyalatait nem használjuk.

Először egy sornyi négyzetet rajzoljunk ki. Ehhez az előző feladathoz hasonlóan most is sokszor kéne ugyanazt leírnunk:

```
from graphics import *

w = GraphWin('Abrosz', 601, 401)

r = Rectangle(Point(0, 0), Point(10, 10)
r.setFill('red')
r.draw(w)

r = Rectangle(Point(10, 0), Point(20, 10))
r.setFill('white')
r.draw(w)

r = Rectangle(Point(20, 0), Point(30, 10))
r.setFill('red')
r.draw(w)
...
```

Ezt például úgy tudnánk egy ciklussal megoldani, ha egy piros és egy fehér négyzet kirajzolását írnánk bele, hiszen ez az, ami ismétlődik. (De mindjárt nézünk erre más megoldást!)

```
...
for x in range(0,601,20) :
    r = Rectangle(Point(x, 0), Point(x+10, 10))
    r.setFill('red')
    r.draw(w)

    r = Rectangle(Point(x+10, 0), Point(x+20, 10))
    r.setFill('white')
    r.draw(w)
```

Ha viszont azt szeretnénk, hogy a cikluson belül mindössze egy négyzet kirajzolás szerepeljen, csak a színt kelljen megválasztani attól függően, hogy hol járunk épp, akkor használjunk elágazást. Ehhez egyrészt változtatnunk kell az x ciklusváltozó lépésközét: 20 helyett 10et kell lépni, hiszen eddig két négyzetet rajzoltunk ki egyszerre, most már viszont csak egyet fogunk. Az elágazásnál az lesz a feltétel, hogy az x ciklusváltozó osztható-e 20-szal. Ha igen, akkor piros négyzetet rajzolunk, ha nem, akkor pedig fehéret. Lássuk ezt megvalósítva:

```
...
for x in range(0,601,10) :
    r = Rectangle(Point(x, 0), Point(x+10, 10))
    #szín megadása elágazással:
    if x % 20 == 0 :  #ha x-nek a 20-al vett osztási maradéka 0
        r.setFill('red')
    else :
        r.setFill('white')

    r.draw(w)          #kirajzolás
```

Ezzel nem változott a kirajzolt ábra a korábbihoz képest.

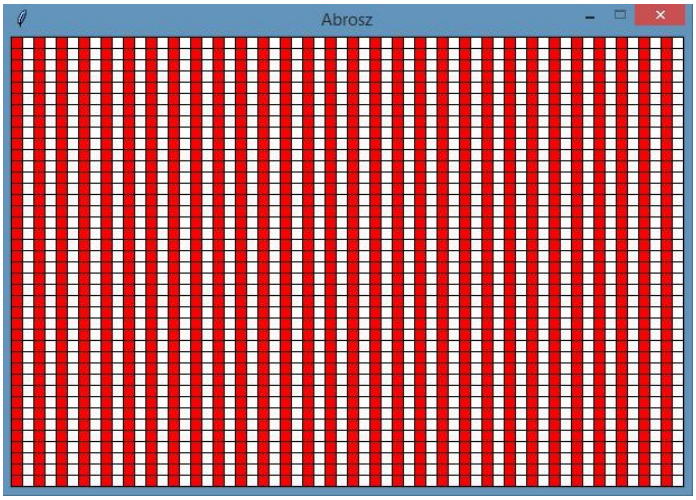
Egy sort már kirajzoltunk, jöhet a többi. Ezt is csinálhatnánk úgy, hogy a sor kirajzolását megismételjük 41-szer:

```
from graphics import *

w = GraphWin('Abrosz', 601, 401)

for x in range(0,601,10) :
    r = Rectangle(Point(x, 0), Point(x+10, 10))
    if x % 20 == 0 :
        r.setFill('red')
    else :
        r.setFill('white')
    r.draw(w)

for x in range(0,601,10) :
    r = Rectangle(Point(x, 10), Point(x+10, 20))
    if x % 20 == 0 :
        r.setFill('red')
    else :
        r.setFill('white')
    r.draw(w)
...
```

De látszik, hogy ez már tényleg nem túl praktikus. Főleg, hogy mindenütt a piros számokat át kéne írni. Éppen ezért használjunk ismét ciklust, csak most a téglalapot meghatározó pontok y koordinátáinak kell különböző értéket adnunk, úgyhogy ehhez is az új ciklusváltozót lehet használni. A korábban megírt 7 soros programrészletet kell az új cikluson belülre másolni, hiszen az jelenti egy sor kirajzolását, és éppen ez az, amit ismételni szeretnénk:

```
...
for y in range(0,401,10) :
    for x in range(0,601,10) :
        r = Rectangle(Point(x, y), Point(x+10, y+10))
        if x % 20 == 0 :
            r.setFill('red')
        else :
            r.setFill('white')
        r.draw(w)
```

Ezzel úgynevezett egymásba ágyazott ciklusokat kaptunk, hiszen egy cikluson belülre raktunk egy másikat. Viszont ennek az eredménye még nem tökéletes nekünk a feladat szempontjából:



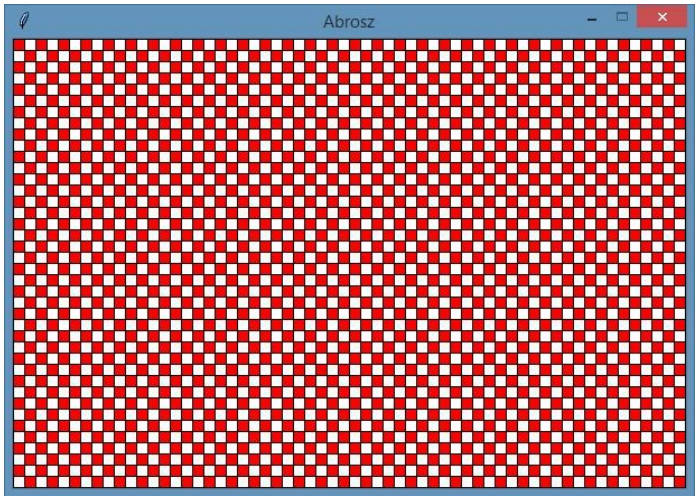
Pontosan ugyanazt a sort másoltuk le többször, mi viszont azt szeretnénk, ha ezek el lennének csúsztatva egymáshoz képest: piros négyzet alatt fehér legyen és fordítva. Ehhez meg kell vizsgálnunk, hogy mitől függ, hogy milyen színt használunk az adott négyzetnél, és ezen kell változtatni. Ez az elágazásunk feltételénél dől el, tehát ezt fogjuk módosítani.

Ha éppen „páratlanodik" sorban vagyunk, akkor minden páratlan „sorszámú" négyzetet színezzünk pirosra, de a „párosadik" sorokban pedig épp fordítva. Nézzük, hogy mik a páratlan sorszámú sorok! Az ilyeneknél az y ciklusváltozó értéke például 0, 20, 40, 60, …, hiszen minden lépésben 10-zel változik ennek az értéke, de nekünk csak minden második sor kell.

Tehát vagy akkor kell piros négyzetet rajzolnunk, ha „páratlanodik" sornak a „páratlanodik" négyzetét rajzoljuk ki, vagy ha „párosadik" sornak a „párosadikát". Az 1. esetben az x és y ciklusváltozó is osztható hússzal (tehát 0 maradékot adnak), a 2. esetben pedig 10 maradékot ad mindkettő egyaránt. Ezt fogjuk kihasználni, eszerint írjuk át a feltételt:

```
...
for y in range(0,401,10) :
    for x in range(0,601,10) :
        r = Rectangle(Point(x, y), Point(x+10, y+10))
        if x % 20 == y % 20 :
            r.setFill('red')
        else :
            r.setFill('white')
        r.draw(w)
```

Ezzel pedig sikerült is elérni a kívánt hatást:



Megjegyzések

A legtöbb esetben ugyanúgy használható mindkét fajta ciklus: a számlálós (for) és az elöltesztelős (while) is. Ez utóbbira valószínűleg következő órán látunk példát.

A rács feladat megoldásában pár számot átírva tudunk rajzolni tic-tac-toe pályát, az abrosz adatot módosítva pedig akár sakktáblát is lehet csinálni.