

```

radla;
setgraphmode (b); radla;
{

```

```

for j := 1 to 3 do
begin
for i := 1 to 50 do

```

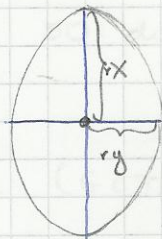
```

begin
w := random (GETMAXCOLOR) + 1; SETCOLOR(w);

```

szívek min van → rajzolónak beállító  
 a lehetőségek 1-ét

clipsisuei



meg kell adni: rx, ry, a szimmetriatengely  
 középpontját, méret

szívek (≥ 12) mintha van → mindegyiknek van egy érdje

INITGRAPH → grafikusba tétel (megnyitás)

CLOSEGRAPH → grafikusból → töröltetésbe (lezárás)

```

r := random (70) + 30; x := random (getmaxx) + 1; y := random (getmaxy) + 1;

```

CIRCLE (x, y, r) → kör rajzoló  
 'kör' középpont sugar

```
end;
```

```

radla; setgraphmode (b);

```

érték: setblkcolor  
 beállítás: set...  
 lekérdezés: get...

```
end;
```

```
radla;
```

```
{

```

```
for j := 1 to 3 do
```

```
begin
```

```
for i := 1 to 50 do
```



begin

$h := \text{random}(\overset{\text{GETMAXCOLOR}}{\text{Getmaxcolor}}) + 1; \text{setcolor}(h);$

$rx := \text{random}(70) + 30; ry := \text{random}(70) + 30;$

$x := \text{random}(\text{getmaxx}) + 1; y := \text{random}(\text{getmaxy}) + 1;$

$\text{ELIPSE}(x, y, 0, 360, rx, ry)$   
kijés dipnis

end;

:

{ - - - }

for  $j := 1$  to 3 do

begin

for  $i := 1$  to 10 do

begin

$h := \text{random}(\text{GETMAXCOLOR}) + 1; \text{SETCOLOR}(h);$

$rx := \text{random}$  ↑

$x :=$  ↑

$\text{MINTA} := \text{random}(12); \text{SETFIL\&STYLE}(\text{MINTA}, h);$

$\text{FILLELIPSE}(x, y, rx, ry)$  \*  
különb. típusú mintázat → különböző név  
 $rx = ry \rightarrow$  kör rajzol

end;

readln; setgraphmode('');

end;

{ - - - }

for  $i := 1$  to 10 do

2 ár dipnisaik u. két körrel  
rajzolás

begin

$\text{setcolor}(\overset{\text{shettér}}{\text{random}(8)}); \text{setcolor}(\text{random}(16));$

picelice: körrel  
rajzol

$x := \text{random}(\text{getmaxx} \text{ div } 2); y := \text{random}(\text{getmaxy} \text{ div } 2);$

$rx := \text{random}(100) + 30; ry := \text{random}(100) + 30;$

$c := \text{random}(270); z := \text{random}(181) + a;$   
fordított  
meghívás

if  $\text{random}(2) = 0$  then



begin schfillstyle (random (2), random (15)); pialice (x, y, c, z, rx) od

```
else
vector (x, y, c) z, rx, ry);
readln;
end;
```

```
schgraphmode (6);
{---}
```

egyes rajzolás

```
for j := 1 to 3 do
```

szere: meg kell adni a két végpontot

```
begin
```

egyes: a végpontot kell adni, és a  
színét is meg kell adni.

```
for i := 1 to 50 do
```

LINE: színek

```
begin
```

```
k := random (GETMAXCOLOR) + 1; SETCOLOR (k);
```

```
x1 := random (getmaxx) + 1; y1 := random (getmaxy) + 1;
```

```
x2 := random (getmaxx) + 1; y2 := random (getmaxy) + 1;
```

```
vastag := random (2) + 1; if vastag = 2 then vastag := vastag + 1;
```

a vonalstílus 5-féle lehet (közvetlen, pontos, keskeny, keskeny-pontos...)

```
stilus := random (5); minta := random (12);
```

```
SETLINESTYLE(STILUSstilus, MINTA, VASTAG) => kell megadni a vonal stílusát
```

```
LINE (x1, y1, x2, y2)
```

```
end;
```

```
readln; schgraphmode (6)
```

```
end;
```

```
readln;
```

```
{---}
```

```
for j := 1 to 3 do
```

```
begin
```

```
MOVEREL (1, 1); > kell megadni a graf. kursor
```

```
for i := 1 to 50 do
```



begin

$h := \text{random}(\text{GETMAXCOLOR}) + 1; \text{SETCOLOR}(h);$

$x_1 := \text{random}(\text{getmaxx}) + 1; y_1 := \text{random}(\text{getmaxy}) + 1;$

$\text{vastap} := \text{random}(2) + 1; \text{if vastap}$

⋮

{---}

for  $j := 1$  to 3 do

begin

for  $i := 1$  to 30 do

begin

$h := \text{random}$  ↑

$x_1 :=$  ↑

$x_2 :=$  ↑

$\text{MINTA} := \text{random}(12); \text{SETFILLSTYLE}(\text{MINTA}, h);$

$\text{BAR}(x_1, y_1, x_2, y_2)$  téplalapot rajzol

end;

kedln; ~~set~~ setgraphmode(6)

end;

kedln;

{---}

keret := false;

for  $j := 1$  to 8 do

begin

$\text{keret} := \text{not}(\text{keret})$

$h := \text{random}(\text{GETMAXCOLOR}) + 1; \text{SETCOLOR}(h);$

$x_1 :=$  ↑

$x_2 :=$  ↑

$\text{MINTA} :=$  ↑



BAR 3D ( $x_1, x_2, y_1, y_2$ ) ( $x_1, y_1, x_2, y_2, 50, \text{rest}$ ), <sup>100 - kérés v. más</sup>

bal felső, jobb alsó  $\rightarrow$  még pixel legyen -  
kicsinő ap  $\frac{1}{2}$

```
redln; schygraphmode (0)
end;
```

```
redln;
{ --- }
```

```
for i := 1 to 8 do
```

```
begin
```

```
  w := random(16); írási SETBKCOLOR(w); háttérszín CLEARDEVICE; 'kiválasztás' építőanyag
```

```
  repeat k := random(16) until k <> w; do
```

```
  font := random(5); írás irany := random(2); behívás merit  $\frac{1}{2}$  := random(10+1);
```

```
  SETTEXTSTYLE(font, irany, merit);
```

```
  x := random(getmaxx - 10) + 10; y := random(getmaxy - 5) + 5;
```

```
  OUTTEXTXY(x, y, 'Grafika');  $\rightarrow$  megmondja hol kezdődjék - a végén
```

```
  redln; schygraphmode(0);  $\rightarrow$  numerikus értéket van színpéldét lehet elírni
```

```
end;
```

```
redln; CLOSEGRAPH;  $\rightarrow$  lezárja a grafikus unitot
```

```
writeln('Vége');
```

```
redln;
```

```
end.
```

~~~~~

RESET CRTMODE  $\rightarrow$  át lehet menni a graf. mód lezárása  
után kar. módba

SETGRAPHMODE  $\rightarrow$  kar. mód  $\rightarrow$  graf. módba.

Graphics banner manipulation

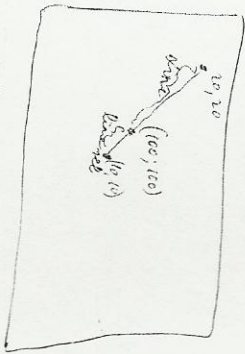
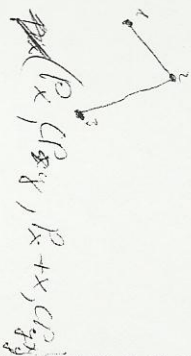
Move to (x, y : integer);

Reverse (x, y : integer);

A graphics banner edit pathfinder etc. machine printing  
line variable

Line to (x, y : integer);

Line to (x, y : integer);



Erases entire segment and all subelements  
Get Pixel (x, y : integer) : Word;

used in the  
penultimate

Allows graphics banner analysis from a screenshot

OutText (s : string);

String Release Text width (s : string) : word

String measure Text height (s : string) : word

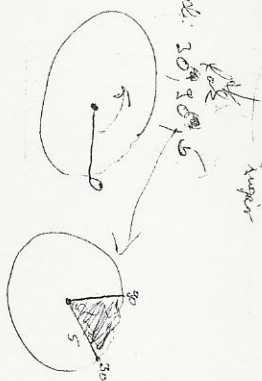
SetText Justify (r : integer)

Measure string lengths  
Left text → source = bottom bar  
center text → 1 = center text  
right text → 2 = top text

Polygon to draw with graphics  
Draw Poly (p : word; var : points)

Be is unknown a  
Fill Poly (p : word, var : points)

Reslice (x, y : integer; a, b, r : word)  
Erasing



Erasing

Sector (x, y : integer; a, b, r, x, r, y : word);

Width and height  
var



A rectangle height  
var

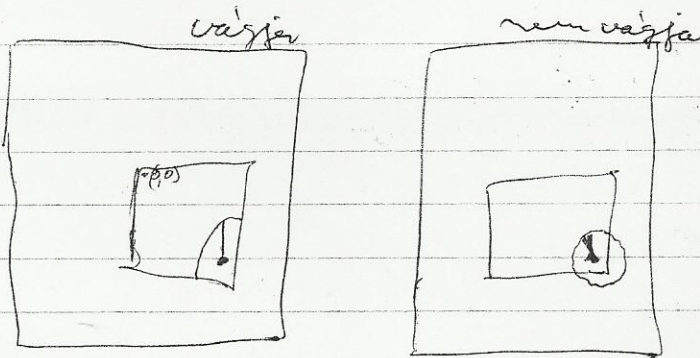
Rectangle (x1, y1, x2, y2 : integer);  
var: (x0, y0, x0, y0)



A Programmal megtekinthető a : <http://aries.eelf.hu/prog>

Kis képernyő definiál az credits képernyőben

Setviewport (  $x_1, y_1, x_2, y_2$  : integer;  $v$  : boolean )  
↓  
válasz



clearviewport;

A képernyő adatakat lehet lementeni egy adott fájlba

Getimage (  $x_1, y_1, x_2, y_2$  : integer;  $v$  : boolean );

↓  
pointer csak egy nem ártó címét tárol

ImageSize (  $x_1, y_1, x_2, y_2$  ) : word;

Var p : pointer;

Getmem ( p : pointer; Size : word );

Putimage (  $x, y$  : integer; var p,  $x$  : word );

x lehet : normal Put 0 lefedeti a képet  
XOR Put 1  
OR Put 2  
AND Put 3 Pl. fekete lesz mint  
NOT Put 4

↓  
mind 4 0100  
szé 1 0001