

```

1 Program NagyitasKicsinyitesStb;
2 { -----
3     Nagyítás, kicsinyítés és élkiemelés. (Számítógépi grafikához)
4
5     (Adott 2-változós görbe generálása után.)
6
7     Keretprogram.
8     ----- }
9 {$f+ azért kell, hogy függvényt paraméterként adhassak át }
10 Uses
11     NewDelay,Crt,Graph;
12 Const
13     MaxN = 100; {Maximum ennyi pontból állhat a függvény}
14     Hol_a_BGI='e:\LANGS\BP\BGI'; {Itt található a BGI-driver}
15 { ----- A kirajzolandó függvények megadása: ----- }
16 Type
17     KetValtozosFv = Record
18         tolX,igX,tolY,igY:Real; {A fv. értelmezési tartománya}
19         fv: Function(x,y:Real):Real; {A fv.-t kiszámoló FUNC.}
20         min,max:Real {A fv. értékkészlete}
21     End;
22 Function x_(x,y:Real): Real;
23 Begin
24     x_:=x/100;
25 End;
26 Function x_y(x,y:Real): Real;
27 Begin
28     x_y:=x*y/100;
29 End;
30 Function xHullam_(x,y:Real): Real;
31 Begin
32     xHullam_:=Sin(x);
33 End;
34 Function Sinx_Siny(x,y:Real): Real;
35 Begin
36     Sinx_Siny:=Sin(x)*Sin(y);
37 End;
38 Function x_Cosy(x,y:Real): Real;
39 Begin
40     x_Cosy:=x*Cos(x);
41 End;
42 Function ForgEllipsz(x,y:Real): Real;
43 Begin
44     ForgEllipsz:=2*Sqr(x)+0.5*Sqr(y);
45 End;
46
47 {
48     Itt ügyelni kell, hogy a függvények maximuma és minimuma helyes legyen,
49     továbbá a összhangban álljon a MaxN és az értelmezési tartomány lépés-
50     köze!!!
51 }
52 Const
53     x_Fv : KetValtozosFv=(tolX:-4.92;igX:4.92;tolY:-4.92;igY:4.92;
54         fv:x_; min:-2.0; max:2.0);
55     x_yFv : KetValtozosFv=(tolX:-4.92;igX:4.92;tolY:-4.92;igY:4.92;
56         fv:x_y; min:-2.0; max:2.0);
57     xHullam_Fv: KetValtozosFv=(tolX:-4.92;igX:4.92;tolY:-4.92;igY:4.92;
58         fv:xHullam_; min:-1.0; max:1.0);
59     sin_sinFv : KetValtozosFv=(tolX:-4.92;igX:4.92;tolY:-4.92;igY:4.92;
60         fv:Sinx_Siny; min:-1.0; max:1.0);
61     x_CosFv : KetValtozosFv=(tolX:-pi;igX:pi;tolY:-pi;igY:pi;
62         fv:x_Cosy; min:-pi; max:pi);
63     ForgEllipszFv: KetValtozosFv=(tolX:-4.92;igX:4.92;tolY:-4.92;igY:4.92;
64         fv:ForgEllipsz; min:-0; max:75);

```

```

65 { ----- A függvényértékeket tartalmazó vektorok: ----- }
66 Type
67     FvErtek      = Array [1..MaxN,1..MaxN] of Byte;
68     FuggvenyTabla = Record
69         n,m: Integer;
70         fv : FvErtek;
71     End;
72 Var
73     ft: Array [0..3] of FuggvenyTabla;
74 { ----- A megjelenítő ablakok: ----- }
75 Type
76     Ablak = Record
77         bf,ja: Record x,y: Integer; End;
78         cimke: String;
79         szin : Integer;
80     End;
81 Var
82     a: Array [1..7] of Ablak;
83     szel,
84     mag: Integer; {Aktuális szélességük és magasságuk}
85     XMax,YMax,
86     XMin,YMin: Integer; { A teljes képernyő méretei}
87 { ----- Egyéb globális változók: ----- }
88     x: Real;
89
90 Procedure Inic;
91     Var gd,gm,
92         i : Integer;
93 Begin
94     gd:=Detect;
95     InitGraph(gd,gm,Hol_a_BGI);
96     If GraphResult<>0 then
97         Begin
98             Writeln(' Legyen az aktuális alkönyvtárban a megfelelő BGI állomány!');
99             Halt(1);
100         End;
101         SetBkColor(Black); {A háttérszín legyen: fekete}
102         ClearDevice;
103         XMax:=GetMaxX; YMax:=GetMaxY;
104         XMin:=0; YMin:=0;
105         SetColor(White); {Az tintaszín legyen: fehér}
106         Rectangle(XMin,YMin, XMax,YMax);
107         szel:=(XMax Div 2)-4; mag:=(YMax Div 2)-TextHeight(' ');
108         SetTextJustify(JustifyCenter,JustifyTop); {A kiírás "origója" a szöveg}
109         OutTextXY(szel+1,2,'Nagyító program'); { felső szélének közepe }
110         With a[1] do
111             Begin
112                 bf.x:=2; bf.y:=10;
113                 ja.x:=szel-2; ja.y:=mag+8; cimke:='Eredeti'; szin:=Red
114             End;
115             With a[2] do
116                 Begin
117                     bf.x:=XMax-2-szel; bf.y:=10;
118                     ja.x:=XMax-2; ja.y:=mag+8;
119                     cimke:='Nagyítás duplázva'; szin:=Blue
120                 End;
121                 With a[3] do
122                     Begin
123                         bf.x:=2; bf.y:=YMax-2-mag;
124                         ja.x:=szel-2; ja.y:=YMax-2;
125                         cimke:='Nagyítás átlagolva'; szin:=Green
126                     End;
127                     With a[4] do
128                         Begin

```

```

129     bf.x:=XMax-2-szel; bf.y:=YMax-2-mag;
130     ja.x:=XMax-2-(szel Div 2); ja.y:=YMax-2-(mag Div 2);
131     cimke:='Elhagyó kicsinyítés'; szin:=Yellow
132 End;
133 With a[5] do
134 Begin
135     bf.x:=XMax-2-(szel Div 2); bf.y:=YMax-2-mag;
136     ja.x:=XMax-2; ja.y:=YMax-2-(mag Div 2);
137     cimke:='Atlag-kicsinyítés'; szin:=Yellow
138 End;
139 With a[6] do
140 Begin
141     bf.x:=XMax-2-szel; bf.y:=YMax-2-(mag Div 2);
142     ja.x:=XMax-2-(szel Div 2); ja.y:=YMax-2;
143     cimke:='Élkiemelés:É'; szin:=Cyan
144 End;
145 With a[7] do
146 Begin
147     bf.x:=XMax-2-(szel Div 2); bf.y:=YMax-2-(mag Div 2);
148     ja.x:=XMax-2; ja.y:=YMax-2;
149     cimke:='Élkiemelés:É-K'; szin:=Cyan
150 End;
151 For i:=1 to 7 do
152     With a[i] do
153     Begin
154         SetColor(szin);
155         SetViewport(bf.x,bf.y, ja.x,ja.y,ClipOn);
156         Rectangle(0,0, (ja.x-bf.x)-4,(ja.y-bf.y)-4);
157         OutTextXY((ja.x-bf.x) Div 2,1,cimke);
158     End;
159 End;
160
161 Procedure Feltolt(f: KetValtozosFv; lepX,lepY: Real; Var ft: FuggvenyTabla);
162     Var x,y: Real; i,j: Integer;
163 Begin
164     With f do {"f." kiemelése az "f.*" mezőhivatkozásokból}
165     Begin
166         y:=tolY; i:=0;
167         While y<=igY do
168         Begin
169             Inc(i); x:=tolX; j:=0;
170             While x<=igX do
171             Begin
172                 Inc(j);
173                 ft.fv[i,j]:=Round((fv(x,y)-min)*((MaxN-1)/(max-min)));
174                 x:=x+lepX;
175             End;{of While x}
176             y:=y+lepY;
177             ft.m:=j;
178         End;{of While y}
179         ft.n:=i;
180     End;{of With}
181 End;
182
183 Procedure FolyoGeneralas(Var ft: FuggvenyTabla);
184     Var mag,i,i0,j,j0: Integer;
185
186     Procedure Arkotas(Var i0,j0:Integer; ri1,ri2,rj1,rj2:Real);
187         Var r:Real; i,m,db:Integer;
188     Begin
189         db:=0; m:=ft.Fv[i0,j0];
190         While (i0 in [3..ft.N-2]) and (j0 in [3..ft.M-2]) do
191         Begin
192             r:=Random;

```

```

193     If
194         r<ri1      then i:=i0+1 else if
195         r<ri1+ri2  then i:=i0
196                 else i:=i0-1;
197     r:=Random;
198     If
199         r<rj1      then j:=j0+1 else if
200         r<rj1+rj2  then j:=j0
201                 else j:=j0-1;
202     ft.Fv[i,j]:=m; ft.Fv[i+1,j]:=m; ft.Fv[i,j+1]:=m; ft.Fv[i+1,j+1]:=m;
203     i0:=i; j0:=j;
204     Inc(db);
205 End{While};
206 For i:=1 to Random(db) do
207     Begin
208         If
209             ft.Fv[i0-1,j0]=m then Dec(i0) else if
210             ft.Fv[i0,j0-1]=m then Dec(j0) else if
211             ft.Fv[i0-1,j0-1]=m then Begin Dec(i0); Dec(j0) End else if
212             ft.Fv[i0+1,j0]=m then Inc(j0) else if
213             ft.Fv[j0,j0+1]=m then Inc(j0)
214         {EndIf};
215     End{For};
216 End;
217
218 Begin {FolyGeneralas}
219     ft.n:=MaxN; ft.M:=MaxN;
220     For i:=1 to ft.N do for j:=1 to ft.M do
221         Begin
222             ft.Fv[i,j]:=100;
223         End;
224         i0:=15; j0:=15;
225         ft.Fv[i0,j0]:=ft.Fv[i0,j0]-Random(9)+4;
226         mag:=ft.Fv[i0,j0];
227         Arkotas(i0,j0,0.3,0.5,0.3,0.4);
228         Arkotas(i0,j0,0.9,0.1,0.8,0.1);
229
230         i0:=45; j0:=15;
231         ft.Fv[i0,j0]:=mag;
232         Arkotas(i0,j0,0.3,0.5,0.4,0.4);
233         Arkotas(i0,j0,0.9,0.1,0.9,0.1);
234
235         i0:=30; j0:=30;
236         ft.Fv[i0,j0]:=mag;
237         Arkotas(i0,j0,0.2,0.7,0.4,0.4);
238         Arkotas(i0,j0,0.9,0.1,0.9,0.1);
239
240         i0:=65; j0:=90;
241         ft.Fv[i0,j0]:=mag;
242         Arkotas(i0,j0,0.2,0.1,0.1,0.1);
243     End;
244
245 Procedure Kirajzol(a: Ablak; t: FuggvenyTabla);
246     Var i,j: Integer;
247     Begin
248         With a do
249             Begin
250                 SetViewport(bf.x+2,bf.y+1+TextHeight(' '),ja.x-2,ja.y-2,ClipOff);
251                 For i:=1 to t.n do for j:=1 to t.m do
252                     Begin
253                         PutPixel(i,j,t.fv[j,i])
254                     End;
255                 End;
256     End;

```

```

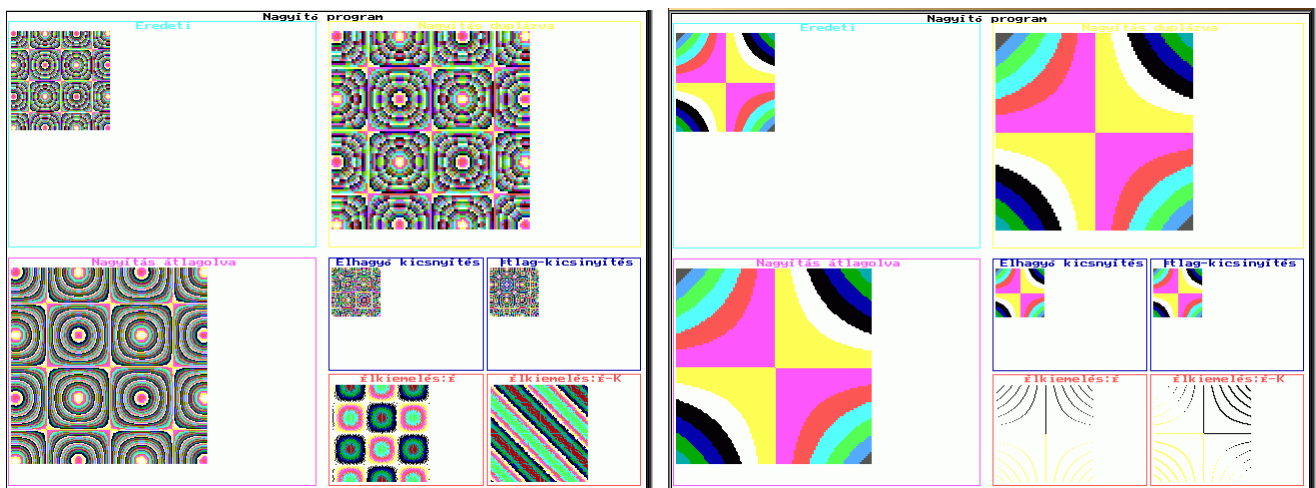
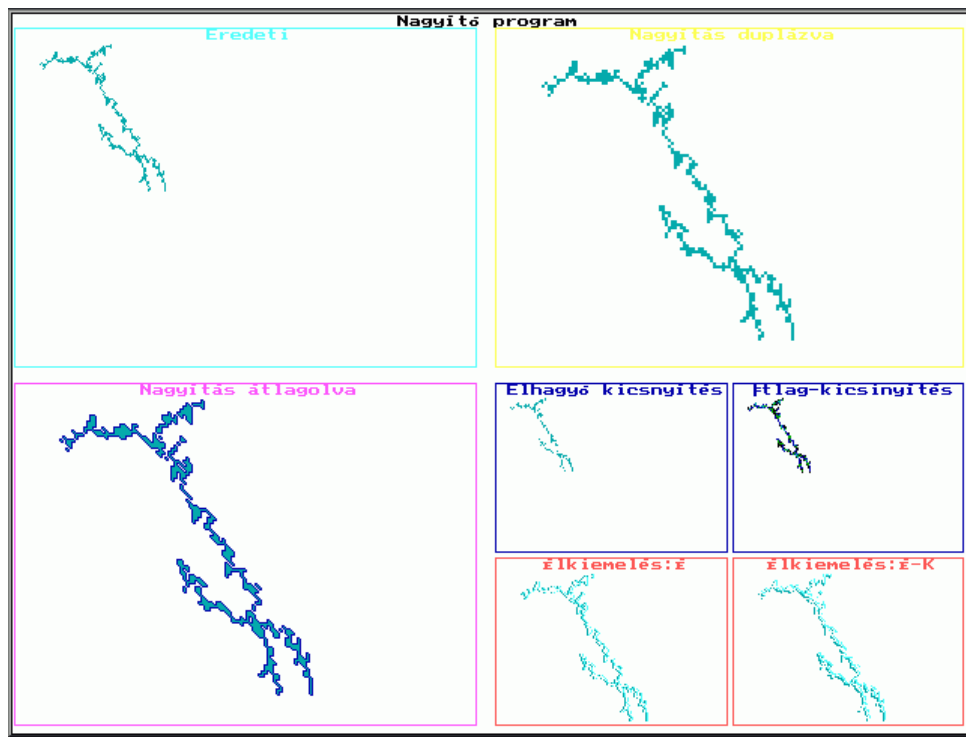
257
258 Procedure NagyitDuplazva(a: Ablak; t: FuggvenyTabla);
259     Var i,j: Integer;
260 Begin
261     { ide kell valami }
262 End;
263
264 Procedure NagyitAtlagolva(a: Ablak; t: FuggvenyTabla);
265     Var i,j: Integer; fv,fvA: Byte;
266 Begin
267     { ide kell valami }
268 End;
269
270 Procedure KicsinyitElhagyva(a: Ablak; t: FuggvenyTabla);
271     Var i,j: Integer; fv,fvA: Byte;
272 Begin
273     { ide kell valami }
274 End;
275
276 Procedure KicsinyitAtlagolva(a: Ablak; t: FuggvenyTabla);
277     Var i,j: Integer; fvA: Byte;
278 Begin
279     { ide kell valami }
280 End;
281
282 Type Irany=(Eszak, EszakKelet, Kelet, Del, Nyugat);
283
284 Procedure Elkiemeles(a: Ablak; t: FuggvenyTabla; merrol: Irany);
285     Var i,j,kx,ky: Integer; fvA: Real;
286     Const mennyire: Array [Irany,-1..1,-1..1] of Real=
287         ((0.0,-1.0,0.0),
288          (0.0,1.0,0.0),
289          (0.0,0.0,0.0)), {Eszak}
290         ((0.0,0.0,-1.0),
291          (0.0,1.0,0.0),
292          (0.0,0.0,0.0)), {EszakKelet}
293         ((0.0,0.0,0.0),
294          (0.0,1.0,-1.0),
295          (0.0,0.0,0.0)), {Kelet}
296         ((0.0,0.0,0.0),
297          (0.0,-1.0,0.0),
298          (0.0,1.0,0.0)), {Del}
299         ((0.0,0.0,0.0),
300          (0.0,-1.0,1.0),
301          (0.0,0.0,0.0)) {Nyugat});
302 Begin
303     { ide kell valami }
304 End;
305
306 Begin
307     FolyoGeneralas(ft[0]);
308     Feltolt({Sin_SinFv}ForgEllipszFv, 0.1,0.1, ft[1]);
309     Feltolt(Sin_SinFv{ForgEllipszFv}, 0.1,0.1, ft[2]);
310     Feltolt(x_yFv, 0.1,0.1, ft[3]);
311
312     Inic;
313     Kirajzol(a[1],ft[0]);
314     NagyitDuplazva(a[2],ft[0]);      NagyitAtlagolva(a[3],ft[0]);
315     KicsinyitElhagyva(a[4],ft[0]);   KicsinyitAtlagolva(a[5],ft[0]);
316     Elkiemeles(a[6],ft[0],Eszak);    Elkiemeles(a[7],ft[0],EszakKelet);
317     ReadKey;
318
319     Inic;
320     Kirajzol(a[1],ft[1]);

```

```

321 NagyitDuplazva(a[2],ft[1]);      NagyitAtlagolva(a[3],ft[1]);
322 KicsinyitElhagyva(a[4],ft[1]);  KicsinyitAtlagolva(a[5],ft[1]);
323 Elkiemeles(a[6],ft[1],Eszak);    Elkiemeles(a[7],ft[1],EszakKelet);
324 ReadKey;
325
326 Inic;
327 Kirajzol(a[1],ft[2]);
328 NagyitDuplazva(a[2],ft[2]);      NagyitAtlagolva(a[3],ft[2]);
329 KicsinyitElhagyva(a[4],ft[2]);  KicsinyitAtlagolva(a[5],ft[2]);
330 Elkiemeles(a[6],ft[2],Eszak);    Elkiemeles(a[7],ft[2],EszakKelet);
331 ReadKey;
332
333 Inic;
334 Kirajzol(a[1],ft[3]);
335 NagyitDuplazva(a[2],ft[3]);      NagyitAtlagolva(a[3],ft[3]);
336 KicsinyitElhagyva(a[4],ft[3]);  KicsinyitAtlagolva(a[5],ft[3]);
337 Elkiemeles(a[6],ft[3],Eszak);    Elkiemeles(a[7],ft[3],EszakKelet);
338 ReadKey;
339
340 CloseGraph
341 End.

```



```

1 Program NagyitasKicsinyitesStb;
2 { -----
3     Nagyítás, kicsinyítés és élkiemelés. (Számítógépi grafikához)
4
5     (Adott 2-változós görbe generálása után.)
6
7     Keretprogram - Megoldás.
8     ----- }
9 {$f+ azért kell, hogy függvényt paraméterként adhassak át }
10 Uses
11     NewDelay,Crt,Graph;
12 Const
13     MaxN = 100; {Maximum ennyi pontból állhat a függvény}
14     Hol_a_BGI='e:\LANGS\BP\BGI'; {Itt található a BGI-driver}
15 { ----- A kirajzolandó függvények megadása: ----- }
16 Type
17     KetValtozosFv = Record
18         tolX,igX,tolY,igY:Real; {A fv. értelmezési tartománya}
19         fv: Function(x,y:Real):Real; {A fv.-t kiszámoló FUNC.}
20         min,max:Real {A fv. értékkészlete}
21     End;
22 Function x_(x,y:Real): Real;
23 Begin
24     x_:=x/100;
25 End;
26 Function x_y(x,y:Real): Real;
27 Begin
28     x_y:=x*y/100;
29 End;
30 Function xHullam_(x,y:Real): Real;
31 Begin
32     xHullam_:=Sin(x);
33 End;
34 Function Sinx_Siny(x,y:Real): Real;
35 Begin
36     Sinx_Siny:=Sin(x)*Sin(y);
37 End;
38 Function x_Cosy(x,y:Real): Real;
39 Begin
40     x_Cosy:=x*Cos(x);
41 End;
42 Function ForgEllipsz(x,y:Real): Real;
43 Begin
44     ForgEllipsz:=2*Sqr(x)+0.5*Sqr(y);
45 End;
46
47 {
48     Itt ügyelni kell, hogy a függvények maximuma és minimuma helyes legyen,
49     továbbá a összhangban álljon a MaxN és az értelmezési tartomány lépés-
50     köze!!!
51 }
52 Const
53     x_Fv      : KetValtozosFv=(tolX:-4.92;igX:4.92;tolY:-4.92;igY:4.92;
54                             fv:x_; min:-2.0; max:2.0);
55     x_yFv     : KetValtozosFv=(tolX:-4.92;igX:4.92;tolY:-4.92;igY:4.92;
56                             fv:x_y; min:-2.0; max:2.0);
57     xHullam_Fv: KetValtozosFv=(tolX:-4.92;igX:4.92;tolY:-4.92;igY:4.92;
58                             fv:xHullam_; min:-1.0; max:1.0);
59     sin_sinFv : KetValtozosFv=(tolX:-4.92;igX:4.92;tolY:-4.92;igY:4.92;
60                             fv:Sinx_Siny; min:-1.0; max:1.0);
61     x_CosFv   : KetValtozosFv=(tolX:-pi;igX:pi;tolY:-pi;igY:pi;
62                             fv:x_Cosy; min:-pi; max:pi);
63     ForgEllipszFv: KetValtozosFv=(tolX:-4.92;igX:4.92;tolY:-4.92;igY:4.92;
64                             fv:ForgEllipsz; min:-0; max:75);

```

```

65 { ----- A függvényértékeket tartalmazó vektorok: ----- }
66 Type
67     FvErtek      = Array [1..MaxN,1..MaxN] of Byte;
68     FuggvenyTabla = Record
69         n,m: Integer;
70         fv : FvErtek;
71     End;
72 Var
73     ft: Array [0..3] of FuggvenyTabla;
74 { ----- A megjelenítő ablakok: ----- }
75 Type
76     Ablak = Record
77         bf,ja: Record x,y: Integer; End;
78         cimke: String;
79         szin : Integer;
80     End;
81 Var
82     a: Array [1..7] of Ablak;
83     szel,
84     mag: Integer; {Aktuális szélességük és magasságuk}
85     XMax,YMax,
86     XMin,YMin: Integer; { A teljes képernyő méretei}
87 { ----- Egyéb globális változók: ----- }
88     x: Real;
89
90 Procedure Inic;
91     Var gd,gm,
92         i : Integer;
93 Begin
94     gd:=Detect;
95     InitGraph(gd,gm,Hol_a_BGI);
96     If GraphResult<>0 then
97         Begin
98             Writeln(' Legyen az aktuális alkönyvtárban a megfelelő BGI állomány!');
99             Halt(1);
100        End;
101        SetBkColor(Black); {A háttérszín legyen: fekete}
102        ClearDevice;
103        XMax:=GetMaxX; YMax:=GetMaxY;
104        XMin:=0; YMin:=0;
105        SetColor(White); {Az tintaszín legyen: fehér}
106        Rectangle(XMin,YMin, XMax,YMax);
107        szel:=(XMax Div 2)-4; mag:=(YMax Div 2)-TextHeight(' ');
108        SetTextJustify(JustifyCenter,JustifyTop); {A kiírás "origója" a szöveg}
109        OutTextXY(szel+1,2,'Nagyító program'); { felső szélének közepe }
110        With a[1] do
111            Begin
112                bf.x:=2; bf.y:=10;
113                ja.x:=szel-2; ja.y:=mag+8; cimke:='Eredeti'; szin:=Red
114            End;
115            With a[2] do
116                Begin
117                    bf.x:=XMax-2-szel; bf.y:=10;
118                    ja.x:=XMax-2; ja.y:=mag+8;
119                    cimke:='Nagyítás duplázva'; szin:=Blue
120                End;
121                With a[3] do
122                    Begin
123                        bf.x:=2; bf.y:=YMax-2-mag;
124                        ja.x:=szel-2; ja.y:=YMax-2;
125                        cimke:='Nagyítás átlagolva'; szin:=Green
126                    End;
127                    With a[4] do
128                        Begin

```



```

129     bf.x:=XMax-2-szel; bf.y:=YMax-2-mag;
130     ja.x:=XMax-2-(szel Div 2); ja.y:=YMax-2-(mag Div 2);
131     cimke:='Elhagyó kicsinyítés'; szin:=Yellow
132 End;
133 With a[5] do
134 Begin
135     bf.x:=XMax-2-(szel Div 2); bf.y:=YMax-2-mag;
136     ja.x:=XMax-2; ja.y:=YMax-2-(mag Div 2);
137     cimke:='Atlag-kicsinyítés'; szin:=Yellow
138 End;
139 With a[6] do
140 Begin
141     bf.x:=XMax-2-szel; bf.y:=YMax-2-(mag Div 2);
142     ja.x:=XMax-2-(szel Div 2); ja.y:=YMax-2;
143     cimke:='Élkiemelés:É'; szin:=Cyan
144 End;
145 With a[7] do
146 Begin
147     bf.x:=XMax-2-(szel Div 2); bf.y:=YMax-2-(mag Div 2);
148     ja.x:=XMax-2; ja.y:=YMax-2;
149     cimke:='Élkiemelés:É-K'; szin:=Cyan
150 End;
151 For i:=1 to 7 do
152     With a[i] do
153     Begin
154         SetColor(szin);
155         SetViewport(bf.x,bf.y, ja.x,ja.y,ClipOn);
156         Rectangle(0,0, (ja.x-bf.x)-4,(ja.y-bf.y)-4);
157         OutTextXY((ja.x-bf.x) Div 2,1,cimke);
158     End;
159 End;
160
161 Procedure Feltolt(f: KetValtozosFv; lepX,lepY: Real; Var ft: FuggvenyTabla);
162     Var x,y: Real; i,j: Integer;
163 Begin
164     With f do {"f." kiemelése az "f.*" mezőhivatkozásokból}
165     Begin
166         y:=tolY; i:=0;
167         While y<=igY do
168         Begin
169             Inc(i); x:=tolX; j:=0;
170             While x<=igX do
171             Begin
172                 Inc(j);
173                 ft.fv[i,j]:=Round((fv(x,y)-min)*((MaxN-1)/(max-min)));
174                 x:=x+lepX;
175             End;{of While x}
176             y:=y+lepY;
177             ft.m:=j;
178         End;{of While y}
179         ft.n:=i;
180     End;{of With}
181 End;
182
183 Procedure FolyoGeneralas(Var ft: FuggvenyTabla);
184     Var mag,i,i0,j,j0: Integer;
185
186     Procedure Arkotas(Var i0,j0:Integer; ri1,ri2,rj1,rj2:Real);
187         Var r:Real; i,m,db:Integer;
188     Begin
189         db:=0; m:=ft.Fv[i0,j0];
190         While (i0 in [3..ft.N-2]) and (j0 in [3..ft.M-2]) do
191         Begin
192             r:=Random;

```

```

193     If
194         r<ri1      then i:=i0+1 else if
195         r<ri1+ri2  then i:=i0
196                 else i:=i0-1;
197     r:=Random;
198     If
199         r<rj1      then j:=j0+1 else if
200         r<rj1+rj2  then j:=j0
201                 else j:=j0-1;
202     ft.Fv[i,j]:=m; ft.Fv[i+1,j]:=m; ft.Fv[i,j+1]:=m; ft.Fv[i+1,j+1]:=m;
203     i0:=i; j0:=j;
204     Inc(db);
205 End{While};
206 For i:=1 to Random(db) do
207 Begin
208     If
209         ft.Fv[i0-1,j0]=m then Dec(i0) else if
210         ft.Fv[i0,j0-1]=m then Dec(j0) else if
211         ft.Fv[i0-1,j0-1]=m then Begin Dec(i0); Dec(j0) End else if
212         ft.Fv[i0+1,j0]=m then Inc(j0) else if
213         ft.Fv[j0,j0+1]=m then Inc(j0)
214     {EndIf};
215 End{For};
216 End;
217
218 Begin {FolyGeneralas}
219     ft.n:=MaxN; ft.M:=MaxN;
220     For i:=1 to ft.N do for j:=1 to ft.M do
221 Begin
222     ft.Fv[i,j]:=100;
223 End;
224     i0:=15; j0:=15;
225     ft.Fv[i0,j0]:=ft.Fv[i0,j0]-Random(9)+4;
226     mag:=ft.Fv[i0,j0];
227     Arkotas(i0,j0,0.3,0.5,0.3,0.4);
228     Arkotas(i0,j0,0.9,0.1,0.8,0.1);
229
230     i0:=45; j0:=15;
231     ft.Fv[i0,j0]:=mag;
232     Arkotas(i0,j0,0.3,0.5,0.4,0.4);
233     Arkotas(i0,j0,0.9,0.1,0.9,0.1);
234
235     i0:=30; j0:=30;
236     ft.Fv[i0,j0]:=mag;
237     Arkotas(i0,j0,0.2,0.7,0.4,0.4);
238     Arkotas(i0,j0,0.9,0.1,0.9,0.1);
239
240     i0:=65; j0:=90;
241     ft.Fv[i0,j0]:=mag;
242     Arkotas(i0,j0,0.2,0.1,0.1,0.1);
243 End;
244
245 Procedure Kirajzol(a: Ablak; t: FuggvenyTabla);
246     Var i,j: Integer;
247 Begin
248     With a do
249 Begin
250     SetViewport(bf.x+2,bf.y+1+TextHeight(' '),ja.x-2,ja.y-2,ClipOff);
251     For i:=1 to t.n do for j:=1 to t.m do
252 Begin
253     PutPixel(i,j,t.fv[j,i])
254 End;
255 End;
256 End;

```

```

257
258 Procedure NagyitDuplazva(a: Ablak; t: FuggvenyTabla);
259   Var i,j: Integer;
260 Begin
261   { ide kell valami }
262   With a do
263     Begin
264       SetViewPort(bf.x+2,bf.y+1+TextHeight(' '),ja.x-2,ja.y-2,ClipOff);
265       For i:=1 to t.n do for j:=1 to t.m do
266         Begin
267           PutPixel(2*i-1,2*j-1,t.fv[j,i]); PutPixel(2*i,2*j-1,t.fv[j,i]);
268           PutPixel(2*i-1,2*j,t.fv[j,i]); PutPixel(2*i,2*j,t.fv[j,i])
269         End;
270     End;
271 End;
272
273 Procedure NagyitAtlagolva(a: Ablak; t: FuggvenyTabla);
274   Var i,j: Integer; fv,fvA: Byte;
275 Begin
276   { ide kell valami }
277   With a do
278     Begin
279       SetViewPort(bf.x+2,bf.y+1+TextHeight(' '),ja.x-2,ja.y-2,ClipOff);
280       For i:=1 to t.n-1 do for j:=1 to t.m-1 do
281         Begin
282           fv:=t.fv[j,i]; PutPixel(2*i-1,2*j-1,fv);
283           fvA:=(fv+t.fv[j+1,i]) Div 2; PutPixel(2*i-1,2*j,fvA);
284           fvA:=(fv+t.fv[j,i+1]) Div 2; PutPixel(2*i,2*j-1,fvA);
285           fvA:=(fv+t.fv[j+1,i+1]) Div 2; PutPixel(2*i,2*j,fvA);
286         End;
287     End;
288 End;
289
290 Procedure KicsinyitElhagyva(a: Ablak; t: FuggvenyTabla);
291   Var i,j: Integer; fv,fvA: Byte;
292 Begin
293   { ide kell valami }
294   With a do
295     Begin
296       SetViewPort(bf.x+2,bf.y+1+TextHeight(' '),ja.x-2,ja.y-2,ClipOff);
297       For i:=1 to t.n Div 2 do for j:=1 to t.m Div 2 do
298         Begin
299           PutPixel(i,j,t.fv[2*j,2*i]);
300         End;
301     End;
302 End;
303
304 Procedure KicsinyitAtlagolva(a: Ablak; t: FuggvenyTabla);
305   Var i,j: Integer; fvA: Byte;
306 Begin
307   { ide kell valami }
308   With a do
309     Begin
310       SetViewPort(bf.x+2,bf.y+1+TextHeight(' '),ja.x-2,ja.y-2,ClipOff);
311       For i:=1 to t.n Div 2 do for j:=1 to t.m Div 2 do
312         Begin
313           fvA:=(t.fv[2*j-1,2*i-1]+t.fv[2*j,2*i-1]+t.fv[2*j-1,2*i]+t.fv[2*j,2*i]) Div 4;
314           PutPixel(i,j,fvA);
315         End;
316     End;
317 End;
318
319 Type Irany=(Eszak, EszakKelet, Kelet, Del, Nyugat);
320

```

```

321 Procedure Elkiemeles(a: Ablak; t: FuggvenyTabla; merrol: Irany);
322 Var i,j,kx,ky: Integer; fvA: Real;
323 Const mennyire: Array [Irany,-1..1,-1..1] of Real=
324         ((0.0,-1.0,0.0),
325          (0.0,1.0,0.0),
326          (0.0,0.0,0.0)), {Eszak}
327         ((0.0,0.0,-1.0),
328          (0.0,1.0,0.0),
329          (0.0,0.0,0.0)), {EszakKelet}
330         ((0.0,0.0,0.0),
331          (0.0,1.0,-1.0),
332          (0.0,0.0,0.0)), {Kelet}
333         ((0.0,0.0,0.0),
334          (0.0,-1.0,0.0),
335          (0.0,1.0,0.0)), {Del}
336         ((0.0,0.0,0.0),
337          (0.0,-1.0,1.0),
338          (0.0,0.0,0.0)) {Nyugat});
339 Begin
340     { ide kell valami }
341     With a do
342     Begin
343         SetViewPort(bf.x+2,bf.y+1+TextHeight(' '),ja.x-2,ja.y-2,ClipOff);
344         For i:=2 to t.n-1 do for j:=2 to t.m-1 do
345             Begin
346                 fvA:=128;
347                 For kx:=-1 to 1 do for ky:=-1 to 1 do
348                     fvA:=fvA+mennyire[merrol,kx,ky]*t.fv[j+kx,i+ky];
349                     PutPixel(i,j,Round(fvA));
350             End;
351         End;
352     End;
353
354 Begin
355     FolyoGeneralas(ft[0]);
356     Feltolt({Sin_SinFv}ForgEllipszFv, 0.1,0.1, ft[1]);
357     Feltolt(Sin_SinFv{ForgEllipszFv}, 0.1,0.1, ft[2]);
358     Feltolt(x_yFv, 0.1,0.1, ft[3]);
359
360     Inic;
361     Kirajzol(a[1],ft[0]);
362     NagyitDuplazva(a[2],ft[0]);           NagyitAtlagolva(a[3],ft[0]);
363     KicsinyitElhagyva(a[4],ft[0]);       KicsinyitAtlagolva(a[5],ft[0]);
364     Elkiemeles(a[6],ft[0],Eszak);       Elkiemeles(a[7],ft[0],EszakKelet);
365     ReadKey;
366
367     Inic;
368     Kirajzol(a[1],ft[1]);
369     NagyitDuplazva(a[2],ft[1]);           NagyitAtlagolva(a[3],ft[1]);
370     KicsinyitElhagyva(a[4],ft[1]);       KicsinyitAtlagolva(a[5],ft[1]);
371     Elkiemeles(a[6],ft[1],Eszak);       Elkiemeles(a[7],ft[1],EszakKelet);
372     ReadKey;
373
374     Inic;
375     Kirajzol(a[1],ft[2]);
376     NagyitDuplazva(a[2],ft[2]);           NagyitAtlagolva(a[3],ft[2]);
377     KicsinyitElhagyva(a[4],ft[2]);       KicsinyitAtlagolva(a[5],ft[2]);
378     Elkiemeles(a[6],ft[2],Eszak);       Elkiemeles(a[7],ft[2],EszakKelet);
379     ReadKey;
380
381     Inic;
382     Kirajzol(a[1],ft[3]);
383     NagyitDuplazva(a[2],ft[3]);           NagyitAtlagolva(a[3],ft[3]);
384     KicsinyitElhagyva(a[4],ft[3]);       KicsinyitAtlagolva(a[5],ft[3]);

```

```

385     Elkiemeles(a[6],ft[3],Eszak);      Elkiemeles(a[7],ft[3],EszakKelet);
386     ReadKey;
387
388     CloseGraph
389 End.
390
391 212-re alternativakent beszurhato:
392  (*)
393         fv:=(t.fv[i,j]+t.fv[i+1,j]+t.fv[i,j+1]+t.fv[i+1,j+1]) Div 4;
394         PutPixel(2*i-1,2*j-1,fv);
395         PutPixel(2*i,2*j-1,fvA);
396         PutPixel(2*i,2*j-1,fvA);
397         PutPixel(2*i,2*j-1,fvA);
398  *)

```