Multimea lui Mandelbrot

Balus Dan

Grupa 30237

1. Introducere

ML ("Meta Language") este un limbaj de programare functional general. Este cunoscut pentru utilizarea sa sistemul tip Hindley-Milner polimorfic, care atribuie automat tipurile de cele mai multe expresii fără a necesita adnotări de tip explicite și asigură siguranța tipului.

Mulțimea lui Mandelbrot este un fractal care a devenit cunoscut în afara matematicii atât pentru estetica sa, cât și pentru structura complicată, care are la bază o definiție simplă. Mulțimea lui Mandelbrot se definește ca fiind mulțimea acelor puncte c din planul complex pentru care aplicând în mod repetat polinomul complex z2 + c (pornind de la z = 0) rezultatul rămâne în interiorul unui disc de rază finită.

2. Cerinta

Generarea setului Mandelbrot.

3. Implementare/ explicatii proiect

Pentru inceput voi pune tot codul dupa care voi aduce explicatii.

Cod:

```
open Math;

fun square(x,y) = (x*x-y*y,2.0*x*y);

fun add (x,y) (u,v) = (x+u,y+v):real*real;

fun scalar (s:real) (x,y) = (s*x,s*y);

fun dot (x1,y1) (x2,y2) = (x1*x2,y1*y2):real*real;

fun sub (x,y) (u,v)  = (x-u, y-v) : real*real;

fun dist p q = let

    fun r(x,y)=sqrt(x*x+y*y)

    in r(sub p q) end;
```

```
val zero = (0.0,0.0);


fun man c z = add (square z) c;

fun twice f = f o f;
fun thrice f = f o f o f;
fun quice f = twice twice f;
fun five f = f o f o f  o f o f;
fun six f = f o f o f  o f o f o f;
fun seven f = f o f o f  o f o f o f o f;

fun t256 f = twice(twice(twice twice)) f;

fun cat p = let
      val small = 0.001;
      val a= t256 (man p) zero;
      in   if dist a (man p a)<0.001 then "1"
      else if dist a (twice (man p) a)<small then "2"
      else if dist a (thrice (man p) a)<small then "3"
      else if dist a (quice (man p) a)<small then "4"
         else if dist a (five  (man p) a)<small then "5"
         else if dist a (six  (man p) a)<small then "6"
         else if dist a (seven  (man p) a)<small then "7"
      else "*"
```

```
    end handle Overflow  => " ";

fun for (r1:real) r2 d f = if (0.0 > (r2-r1)*d) then ""
                           else (f r1)^(for (r1+d) r2 d f);
fun line x1 x2 y = (for x1 x2 ((x2-x1)/78.0) (fn i=> cat(i,y)))^"\n";
fun box((x1,y1),(x2,y2))= for y2 y1 ((y1-y2)/24.0) (fn i=> line x1 x2 i);
fun K a b = b;
val ibox=( (~2.0,~1.0),(1.0,1.0));

(* Box shifting stuff *)
fun zoom(p, q) = (add (scalar 0.75 p) (scalar 0.25 q),
          add (scalar 0.25 p) (scalar 0.75 q));
fun shift v (p,q) = let val w = dot v (sub q p) in
         (add w p, add w q) end;
val right = shift (0.5,0.0);
val left = shift (~0.5,0.0);
val up = shift (0.0,0.5);
val down = shift (0.0,~0.5);
fun doit x = K (print(box x)) x;
doit ibox;
doit (zoom ibox);
doit (zoom it);
doit (up(left it));
```

open Math;

- inclu libraria Math in proiect;

fun square(x,y) = (x*x-y*y,2.0*x*y);

- ridicarea la patrat a unui nr complex

fun add (x,y) (u,v) = (x+u,y+v):real*real;

- adunarea

fun scalar (s:real) (x,y) = (s*x,s*y);

- inmultirea cu un scalar

fun dot (x1,y1) (x2,y2) = (x1*x2,y1*y2):real*real;

- produs scalar

fun sub (x,y) (u,v) = (x-u, y-v) : real*real;

- scaderea

fun dist p q = let fun r(x,y)=sqrt(x*x+y*y) in r(sub p q) end;

- distanta dintre 2 puncte, mai intai se face diferenta apoi se ridica la patrat

$$\mathrm{d}(\mathbf{p}, \mathbf{q}) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}.$$

fun man c z = add (square z) c;

- se foloseste pentru acel z^2 + c, functie din set (cu ajutorul careia obtinem alti fractali). Imaginea multimii lui Mandelbrot poate fi creata prin colorarea punctelor ***c***

fun twice f = f o f;

fun thrice f = f o f o f;

fun quice f = twice twice f;

fun five f = f o f o f o f o f;

fun six f = f o f o f o f o f o f;

fun seven f = f o f o f o f o f o f o f;

fun t256 f = twice(twice(twice twice)) f;

- constructorul “o” se foloseste pentru a compune o functie, rezultatul fiind tot o functie; unele numere nu pot fi in setul Mandelbrot, deoarece prin compunerea functiilor devin din ce in ce mai mari;
- functia twice, thrice, quice, five, six, seven, respectiv 256 compun f de 2, 3, 4, 5, 6, 7, respective 256 de ori;
- compunerea se foloseste pentru a creea noi fractali
- unele puncte ajung mai repede (0.2,0.2) sau oscileaza (-1.0,0.0), iar la altele le trebuiesc sute de iteratii.

fun cat p = let

val small = 0.001;

val a= t256 (man p) zero;

in if dist a (man p a)<small then "1"

else if dist a (twice (man p) a)<small then "2"

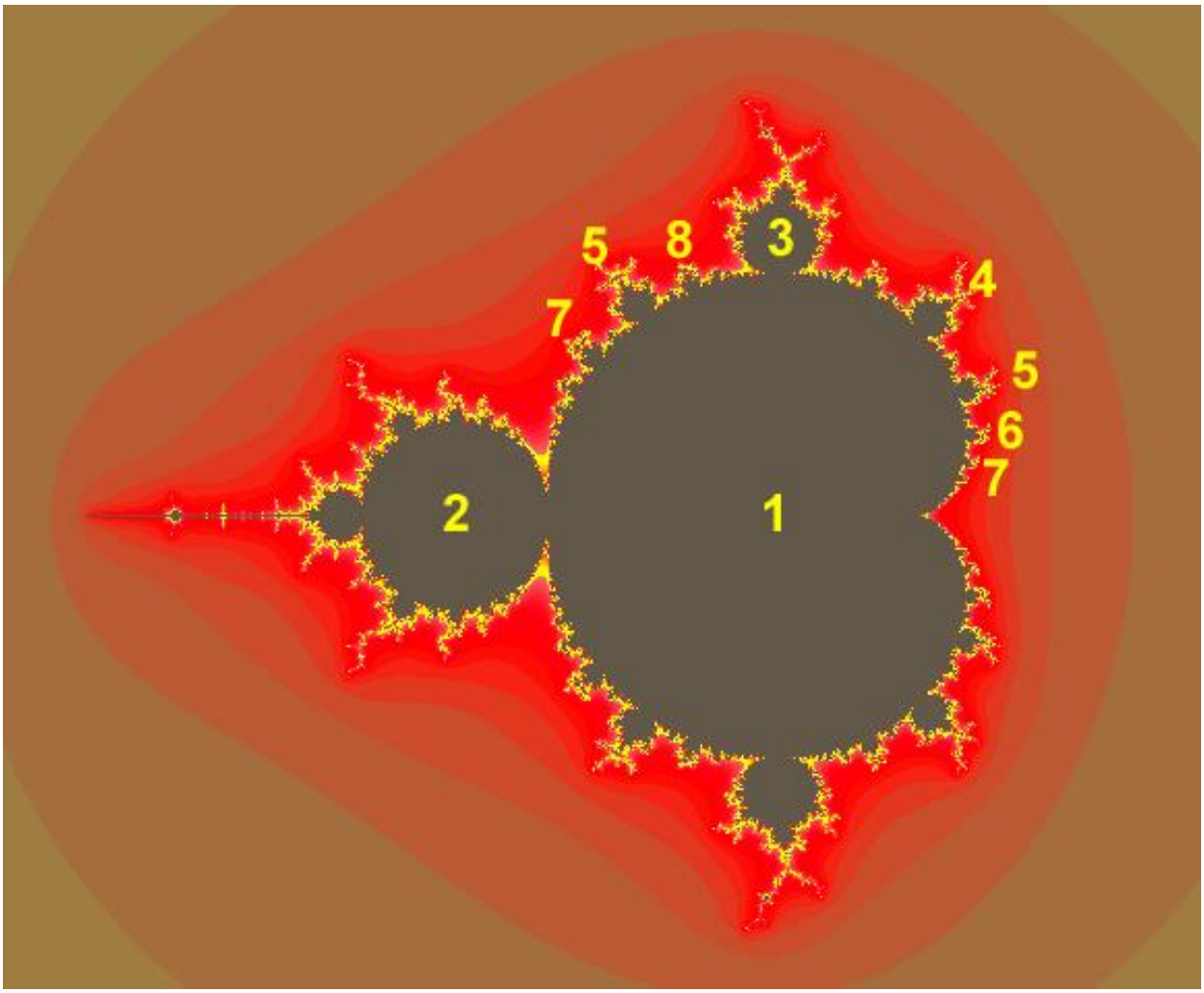
else if dist a (thrice (man p) a)<small then "3"

else if dist a (quice (man p) a)<small then "4"

else if dist a (five (man p) a)<small then "5"

else if dist a (six (man p) a)<small then "6"

```
    else if dist a (seven  (man p) a)<small then "7"
  else "*"
  end handle Overflow  => " ";
```



- se compune functia de 256 de ori in variabila a (pentru a fii sufiecient ca sa se verifici daca este in multime), dupa care se calculeaza distanta dintre a si puncte, pentru a delimita zonele;

```
fun for (r1:real) r2 d f = if (0.0 > (r2-r1)*d) then ""
                           else (f r1)^(for (r1+d) r2 d f);
fun line x1 x2 y = (for x1 x2 ((x2-x1)/78.0) (fn i=> cat(i,y)))^"\n";
fun box((x1,y1),(x2,y2))= for y2 y1 ((y1-y2)/24.0) (fn i=> line x1 x2 i);
fun K a b = b;
fun doit x = K (print(box x)) x;
doit (zoom ibox);
doit (zoom it);
doit (up(left it));
doit (up(right it));
doit (down(left it));
```

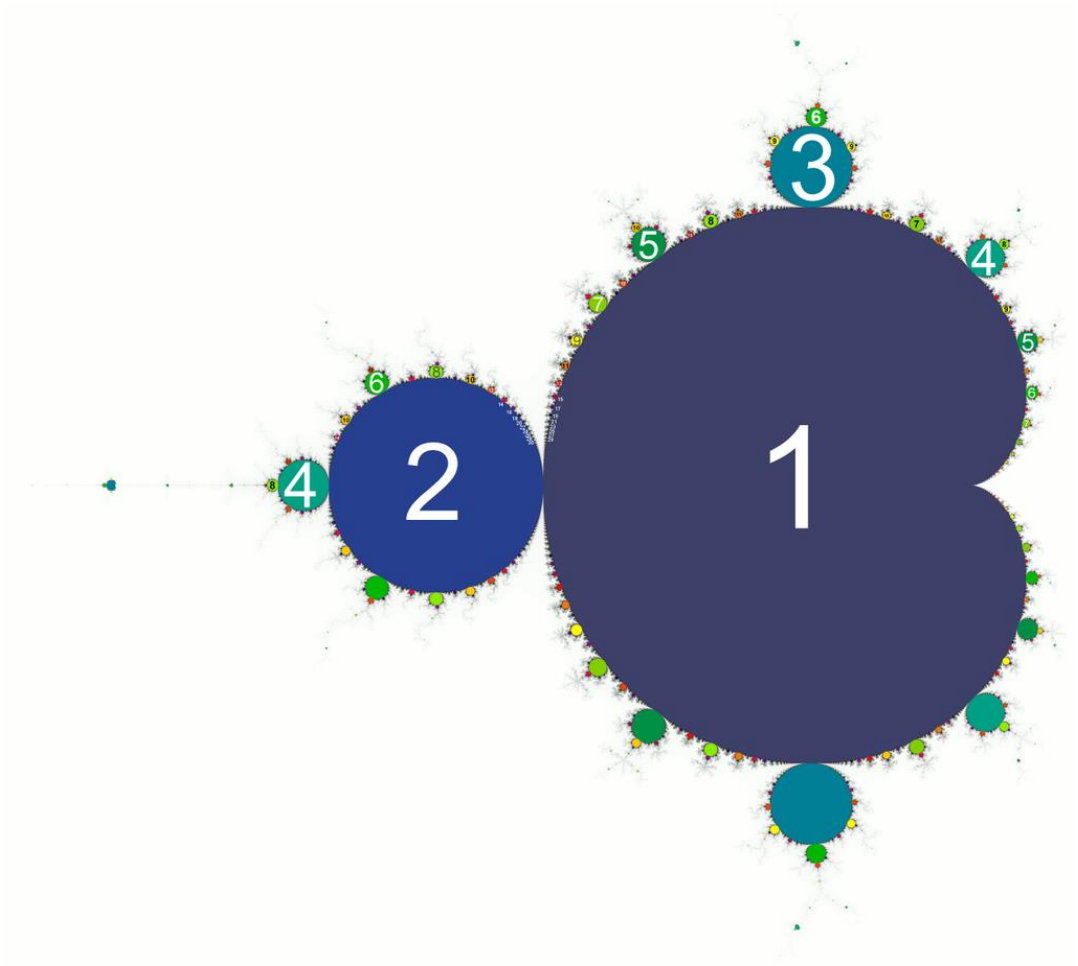
- se folosesc pentru apelarea programului si afisarea rezultatului, 78 de caractere pe linie(se trece la urmatoarea linie dup ace se numara 78, cu ajutorul caracterului “\n”), 24 de caractere pe coloana.

```
fun zoom(p, q) = (add (scalar 0.75 p) (scalar 0.25 q),
          add (scalar 0.25 p) (scalar 0.75 q));
fun shift v (p,q) = let val w = dot v (sub q p) in
         (add w p, add w q) end;
val right = shift (0.5,0.0);
val left = shift (~0.5,0.0);
val up = shift (0.0,0.5);
val down = shift (0.0,~0.5);
```

- se folosesc pentru a da zoom pe fractal.

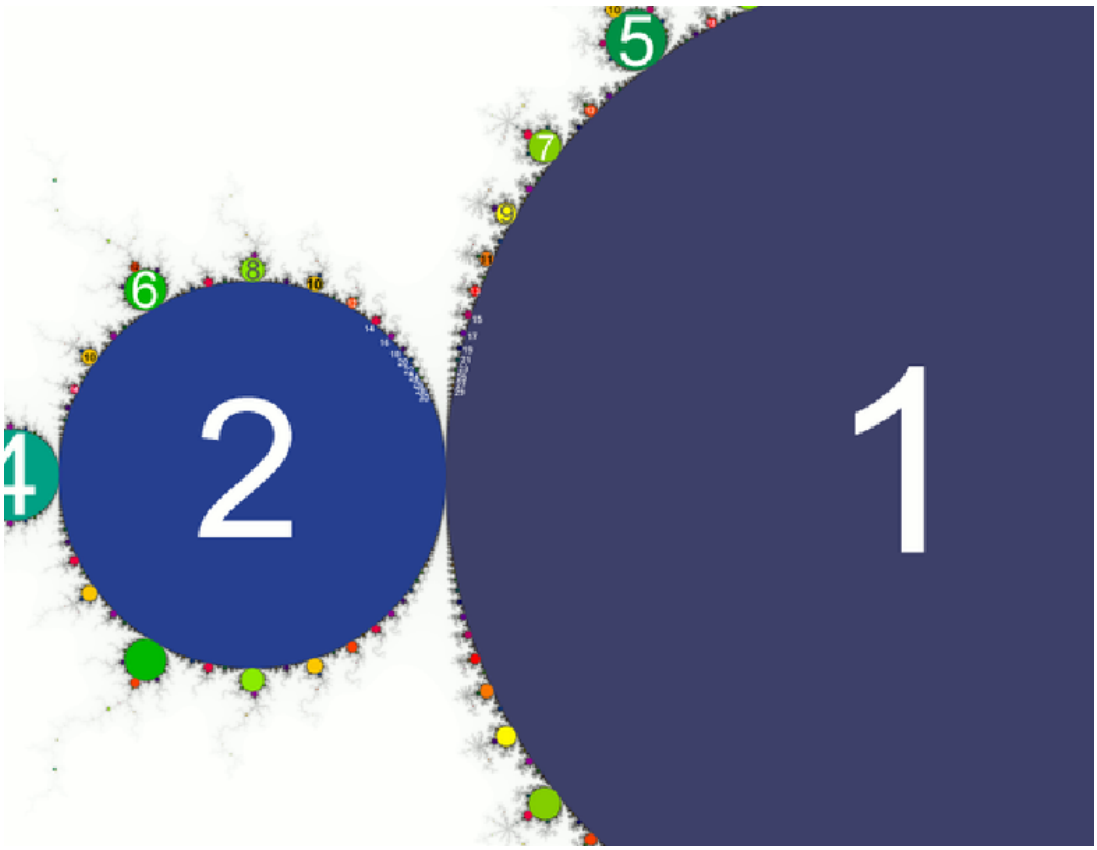
doit ibox;

```
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111*311111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111*333331111111111111111111111111111111
1111111111111111111111111111111111111111111111111111133*111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111511*211111111111711111111111111111111111111
1111111111111111111111111111111111111111111111111*111111111111111111444111111111111111111111
1111111111111111111111111111111111111111111111771111111111111111111111111*111111111111111111
1111111111111111111111111111111111111111111*111111111111111111111111111551111111111111111111
1111111111111111111111116*1**1*11111*111111111111111111111111111111111111111111111111111111111
1111111111111111111111222222222*1*111111111111111111111111111111111111111111111111111111111111
1111111111111111111*22222222222**111111111111111111111111111111111111111111111111111111111111
1111116111111111**4442222222222222211111111111111111111111111111111111111111111111111111111111
1111111111111111111*22222222222**111111111111111111111111111111111111111111111111111111111111
1111111111111111111222222222*1*111111111111111111111111111111111111111111111111111111111111111
1111111111111111111116*1**1*11111*111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111*111111111111111111111111111551111111111111111111111111111
1111111111111111111111111111111111771111111111111111111111*111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111133*111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111*333331111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111*311111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
```



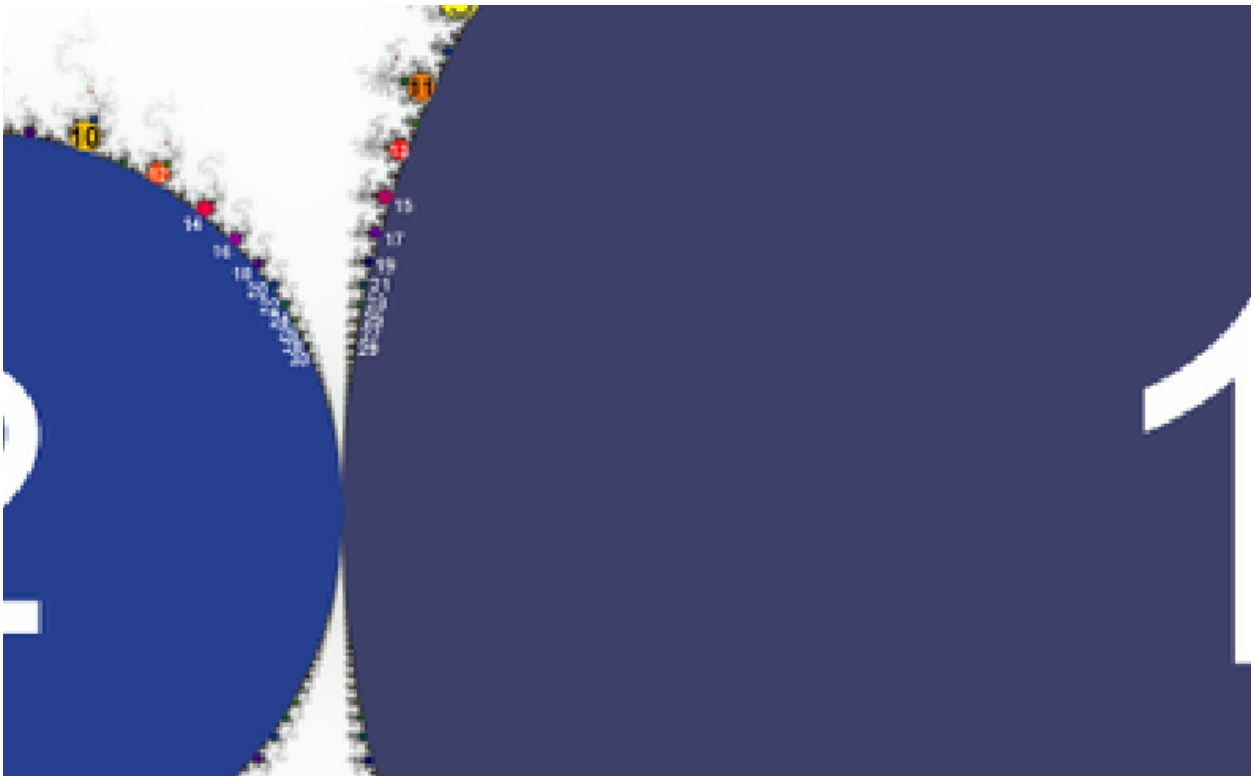
doit (zoom ibox);

```
val it = ((~2.0, ~1.0), (1.0, 1.0)): (real * real) * (real * real)
11111111111111111111111111111111111111**11111111111111111111111111111111111144
1111111111111111111111111111111111111*511111111111111111111111111111111111111
11111111111111111111111111111111177*7111111111111111111111111111111111111111
11111111111111111111111111111111**11111111111111111111111111111111111111111
11111111111111111111111111111111**21111111111111111111111111111111111111111
11111111111111111111111111111111*111111111111111111111111111111111111111111
1111166*11*****11*1111111111*1111111111111111111111111111111111111111111111
111111*2222222222222*111111**211111111111111111111111111111111111111111111
11116222222222222222222*111**111111111111111111111111111111111111111111111
11*22222222222222222222*11*1111111111111111111111111111111111111111111111
1*222222222222222222222*1*11111111111111111111111111111111111111111111111
*2222222222222222222222221211111111111111111111111111111111111111111111111
42222222222222222222222222211111111111111111111111111111111111111111111111
*2222222222222222222222221211111111111111111111111111111111111111111111111
1*22222222222222222222*1*1111111111111111111111111111111111111111111111111
11*2222222222222222222*11*1111111111111111111111111111111111111111111111111
11116222222222222222222*111**111111111111111111111111111111111111111111111
111111*2222222222222*111111**211111111111111111111111111111111111111111111
1111166*11*****11*1111111111*1111111111111111111111111111111111111111111111
11111111111111111111111111111111*111111111111111111111111111111111111111111
11111111111111111111111111111111**21111111111111111111111111111111111111111
11111111111111111111111111111111**11111111111111111111111111111111111111111
11111111111111111111111111111111177*7111111111111111111111111111111111111111
11111111111111111111111111111111111*511111111111111111111111111111111111111
```



doit (zoom it);

```
val it = ((~1.25, ~0.5), (0.25, 0.5)): (real * real) * (real * real)
1111111111111111***1111111111111111111111111111111111111111111111111111111111111111
**111111111111111***1111111111111111111111111111111111111111111111111111111111111111
***11111111111****2111111111111111111111111111111111111111111111111111111111111111
222**1111111111**211111111111111111111111111111111111111111111111111111111111111111
222222**1111111***1111111111111111111111111111111111111111111111111111111111111111
22222222*111111**2111111111111111111111111111111111111111111111111111111111111111
222222222*1111**2111111111111111111111111111111111111111111111111111111111111111
2222222222**11**2111111111111111111111111111111111111111111111111111111111111111
22222222222*11**1111111111111111111111111111111111111111111111111111111111111111
222222222222*1*2111111111111111111111111111111111111111111111111111111111111111
22222222222221*2111111111111111111111111111111111111111111111111111111111111111
222222222222212211111111111111111111111111111111111111111111111111111111111111
222222222222222211111111111111111111111111111111111111111111111111111111111111
222222222222212211111111111111111111111111111111111111111111111111111111111111
22222222222221*2111111111111111111111111111111111111111111111111111111111111111
222222222222*1*2111111111111111111111111111111111111111111111111111111111111111
22222222222*11**1111111111111111111111111111111111111111111111111111111111111111
2222222222**11**2111111111111111111111111111111111111111111111111111111111111111
222222222*1111**2111111111111111111111111111111111111111111111111111111111111111
22222222*111111**2111111111111111111111111111111111111111111111111111111111111111
222222**1111111***1111111111111111111111111111111111111111111111111111111111111111
222**1111111111**211111111111111111111111111111111111111111111111111111111111111111
***11111111111****2111111111111111111111111111111111111111111111111111111111111111
**111111111111111***1111111111111111111111111111111111111111111111111111111111111111
```



doit (up(left it));

```
**1111111111111111***1111111111111111111111111111111111111111111111111111111111111111111111111111
val it = ((~0.875, ~0.25), (~0.125, 0.25)): (real * real) * (real * real)
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111****
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111*1***511
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111***51111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111*777*1***5111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111*7777**711111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111**71111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111***211111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111*11**711111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111*****2111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111***1111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111***1111111111111
111111111*111111111111111***1111111111111111111111111111111111111111111111***1111111111111111
111111111*6666*11111*1*1******1*11**1111111111111111111111111111111111111111***1111111111111111
111111111*6666*****22222222222222****1**1111111111111111111111111111111111111***1111111111111111
11111111111**62222222222222222222222222***11111111111111111111111111111111****2111111111111111
111111111**22222222222222222222222222222**1111111111111111111111111111111111**211111111111111111
1111111*62222222222222222222222222222222**1111111111111111111111111111111111***11111111111111111
1111***22222222222222222222222222222222222*111111111111111111111111111111111**211111111111111111
111**22222222222222222222222222222222222222*1111111111111111111111111111111111**211111111111111111
11**22222222222222222222222222222222222222222**11111111111111111111111111111111**211111111111111111
11*22222222222222222222222222222222222222222222*11111111111111111111111111111111**111111111111111111
1*22222222222222222222222222222222222222222222222*1111111111111111111111111111111*1*211111111111111111
**22222222222222222222222222222222222222222222222221111111111111111111111111111111*211111111111111111
*42222222222222222222222222222222222222222222222222221111111111111111111111111111112211111111111111111
42222222222222222222222222222222222222222222222222222111111111111111111111111111111111111111111111111
```

doit (up(right it));

```
val it = ((~1.25, 0.0), (~0.5, 0.5)): (real * real) * (real * real)
11111111111111111111111111111111111111111111111111111111111111111111111111**3333333333
11111111111111111111111111111111111111111111111111111111111111111111111111113333333333
11111111111111111111111111111111111111111111111111111111111111111111111111*333333333
11111111111111111111111111111111111111111111111111111111111111111111111111*33333333
11111111111111111111111111111111111111111111111111111111111111111111111111**33333
11111111111111111111111111111111111111111111111111111111111111111111111111*111111*******3
11111111111111111111111111111111111111111111111111111111111111111111111111***11111*******33331111111
11111111111111111111111111111111111111111111111111111111111111111111111111**111*11111111111*******331111111111111111111
11111111111111111111111111111111111111111111111555555*111111******211111111111111111111111111
11111111111111111111111111111111111111111*55555555111****5111111111111111111111111111111111
11111111111111111111111111111111111111111555555****51111111111111111111111111111111111111
11111111111111111111111111111111111111111**551111111111111111111111111111111111111111111
11111111111111111111111111111111111111111****51111111111111111111111111111111111111111111
11111111111111111111111111111111111111*1***51111111111111111111111111111111111111111111111
11111111111111111111111111111111111111***51111111111111111111111111111111111111111111111111
11111111111111111111111111111111*777*1***51111111111111111111111111111111111111111111111111
11111111111111111111111111111111*7777**711111111111111111111111111111111111111111111111111
11111111111111111111111111111111**711111111111111111111111111111111111111111111111111111
11111111111111111111111111111111***211111111111111111111111111111111111111111111111111111
11111111111111111111111111111111*11**711111111111111111111111111111111111111111111111111
11111111111111111111111111111111*****211111111111111111111111111111111111111111111111111
11111111111111111111111111111111***111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111***111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111***111111111111111111111111111111111111111111111111111111
```

doit (down(left it))

```
val it = ((~0.875, 0.25), (~0.125, 0.75)): (real * real) * (real * real)
1111111111111111111111111111111111111111111111111111111111111111111111111111****
11111111111111111111111111111111111111111111111111111111111111111111111*1***511
11111111111111111111111111111111111111111111111111111111111111111111111***51111
111111111111111111111111111111111111111111111111111111111111111*777*1***5111111
111111111111111111111111111111111111111111111111111111111111111*7777**711111111
111111111111111111111111111111111111111111111111111111111111111111**71111111111
1111111111111111111111111111111111111111111111111111111111111111***211111111111
111111111111111111111111111111111111111111111111111111111111*11**71111111111111
11111111111111111111111111111111111111111111111111111111111*****211111111111111
111111111111111111111111111111111111111111111111111111111111***1111111111111111
11111111111111111111111111111111111111111111111111111111111***11111111111111111
1111111111*11111111111111***111111111111111111111111111111***111111111111111111
111111111*6666*11111*1*1******1*11**111111111111111111111***1111111111111111111
111111111*6666*****222222222222222****1**111111111111111***11111111111111111111
11111111111**62222222222222222222222222***111111111111****211111111111111111111
111111111**2222222222222222222222222222222**11111111111**2111111111111111111111
1111111*6222222222222222222222222222222222222**1111111***1111111111111111111111
1111***2222222222222222222222222222222222222222*111111**21111111111111111111111
111**2222222222222222222222222222222222222222222*1111**211111111111111111111111
11**222222222222222222222222222222222222222222222**11**211111111111111111111111
11*22222222222222222222222222222222222222222222222*11**111111111111111111111111
1*2222222222222222222222222222222222222222222222222*1*2111111111111111111111111
**222222222222222222222222222222222222222222222222221*2111111111111111111111111
*4222222222222222222222222222222222222222222222222212211111111111111111111111111
4222222222222222222222222222222222222222222222222222222111111111111111111111111
```