```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, Grids, StdCtrls, Math;

type
  TfrmMain = class(TForm)
    sgT: TStringGrid; imgG: TImage; btnRepresenter: TButton; btnTableauImages: TButton;
    edtImMin: TEdit; Label2: TLabel; edtReMax: TEdit; Label3: TLabel; edtReMin: TEdit;
    Label4: TLabel; edtImMax: TEdit; Label5: TLabel; edtDRe: TEdit; edtDIm: TEdit;
    Label6: TLabel; Label7: TLabel; Label8: TLabel; Label9: TLabel; Label10: TLabel;
    Label11: TLabel; edtXMin: TEdit; edtYMin: TEdit; edtXMax: TEdit; edtYMax: TEdit;
    Label12: TLabel; Label1: TLabel; Label13: TLabel;
    procedure btnTableauImagesClick(Sender: TObject);
    procedure btnRepresenterClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmMain: TfrmMain;

implementation

{$R *.dfm}

type TComplex = record
                  re,im:extended
                end;

     TPixel = record
                x,y:integer;
              end;


var xMin,yMin,xMax,yMax:extended;


function inverse(z:TComplex):TComplex;
var sqrModule:extended;
begin
  sqrModule:=sqr(z.re)+sqr(z.im);
  if sqrModule=0 then begin result.re:=1E6; result.im:=1E6 end
  else begin result.re:=z.re/sqrModule; result.im:=-z.im/sqrModule end
end;


function stringToComplex(s:string):TComplex;
var p:integer;
begin
  p:=pos(' + ',s);
  result.re:=strtofloat(copy(s,1,p-1));
  result.im:=strtofloat(copy(s,p+3,length(s)-p-3));
end;


function complexToPixel(z:TComplex;imgG:TImage):TPixel;
begin
  result.x:=round((z.re-xMin)/(xMax-xMin)*(imgG.Width-1));
  result.y:=round((yMax-z.im)/(yMax-yMin)*(imgG.Height-1));
end;
```

```
procedure TfrmMain.btnTableauImagesClick(Sender: TObject);
var reMin,reMax,dRe,imMin,imMax,dIm:extended;
    i,j,imax,jmax:integer;
    z:TComplex;
begin
  reMin:=strtofloat(edtReMin.text);
  reMax:=strtofloat(edtReMax.text);
  imMin:=strtofloat(edtImMin.text);
  imMax:=strtofloat(edtImMax.text);
  dRe:=strtofloat(edtDRe.text);
  dIm:=strtofloat(edtDIm.text);
  imax:=floor((reMax-reMin)/dRe);
  jmax:=floor((imMax-imMin)/dIm);
  sgT.ColCount:=imax+2;
  sgT.RowCount:=jmax+2;
  for i:=0 to imax do sgT.Cells[i+1,0]:=floattostr(reMin+i*dRe);
  for j:=0 to jmax do sgT.Cells[0,j+1]:=floattostr(imMin+j*dIm);
  for i:=0 to imax do
    for j:=0 to jmax do
      begin
        z.re:=reMin+i*dRe;
        z.im:=imMin+j*dIm;
        sgT.Cells[i+1,j+1]:=floattostr(inverse(z).re)+' + '+floattostr(inverse(z).im)+'i'
      end
end;


procedure TfrmMain.btnRepresenterClick(Sender: TObject);
var i,j:integer;
    z:TComplex;
    p:TPixel;
begin
  imgG.Canvas.Rectangle(-1,-1,imgG.Width+1,imgG.Height+1);
  xMin:=strtofloat(edtXMin.Text);
  xMax:=strtofloat(edtXMax.Text);
  yMin:=strtofloat(edtYMin.Text);
  yMax:=strtofloat(edtYMax.Text);
  for i:=1 to sgT.ColCount-1 do
    begin
      z:=stringToComplex(sgT.Cells[i,1]);
      p:=complexToPixel(z,imgG);
      imgG.Canvas.MoveTo(p.x,p.y);
      for j:=2 to sgT.RowCount-1 do
        begin
          z:=stringToComplex(sgT.Cells[i,j]);
          p:=complexToPixel(z,imgG);
          imgG.Canvas.LineTo(p.x,p.y);
        end;
    end;
  for j:=1 to sgT.RowCount-1 do
    begin
      z:=stringToComplex(sgT.Cells[1,j]);
      p:=complexToPixel(z,imgG);
      imgG.Canvas.MoveTo(p.x,p.y);
      for i:=2 to sgT.ColCount-1 do
        begin
          z:=stringToComplex(sgT.Cells[i,j]);
          p:=complexToPixel(z,imgG);
          imgG.Canvas.LineTo(p.x,p.y);
        end;
    end;
end;

procedure TfrmMain.FormCreate(Sender: TObject);
begin
  imgG.Canvas.Pixels[0,0]:=clWhite;
end;

end.
```