Rajdhani
DATE / /

GROUP - B ASSIGNMENT - 10 (CG)

Name: Adityas Onkar Patil
Batch: G4 (SE4)
Roll No: 21449
Performance Date:
Submission Date:

TITLE: Fractals and Curves

PROBLEM STATEMENT: write c++ program to generate snowflake using concept of fractals.

LEARNING OBJECTIVES: To learn the concept of fractals and curves as a part of computer graphics.

LEARNING OUTCOMES: After completion of this assignment, students will be able to:
1. Implement fractals and curves to form figures like snow flakes, Hilbert curves etc.

S/W AND H/W REQUIREMENTS: 64-bit open source Linux or its derivative, open source c++ programming tools like GCC/G++, Qt Creator, OpenGL.

REFERENCES: 1. Programming Principles and practice using C++, Bjarne Stroustrup.
2. www.qt.io.

Rajdhani
DATE / /

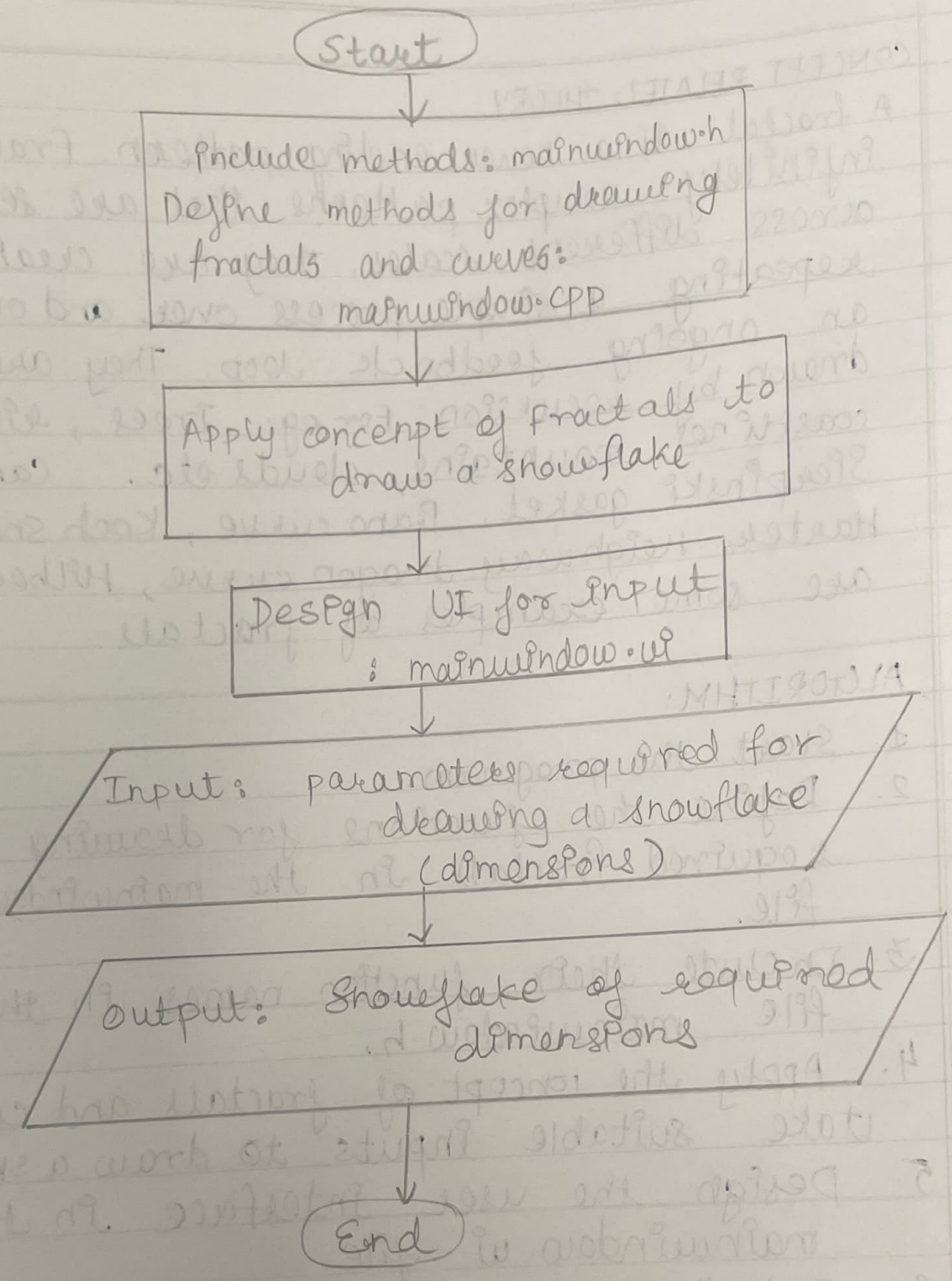
CONCEPT RELATED THEORY:

A fractal is a never ending pattern. Fractals are infinitely complex patterns that are self similar across different scales. They are created by repeating a simple process over and over in an ongoing feedback loop. They are usually drawn by recursion. For eg:- Trees, ferns, coastlines, mountains, clouds etc. Cantor set, Sierpinski gasket, Peano curve, Koch snowflake, Harter Heighway dragon curve, Hilbert curve are some examples of fractals.

ALGORITHM:

1. Start the program.
2. Define the functions for drawing the required fractals in the mainwindow.cpp file.
3. Include their function names in the header file mainwindow.h.
4. Apply the concept of fractals and curves to take suitable inputs to draw a snowflake.
5. Design the user interface in the mainwindow.ui file.
6. End the program.

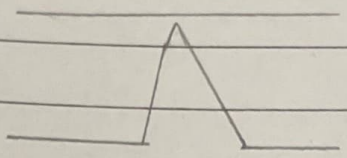
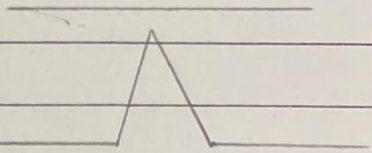
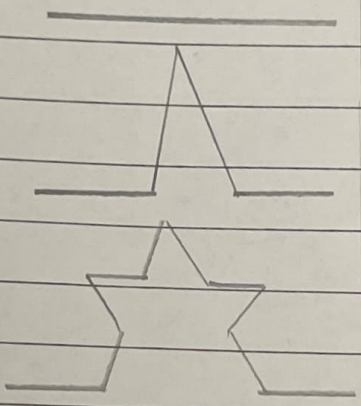
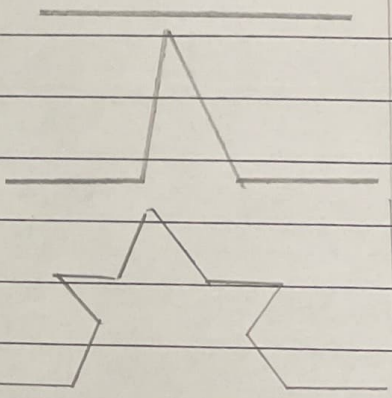
FLOWCHART:


Start
Include methods: mainwindow.h
Define methods for drawing fractals and curves: mainwindow.cpp
Apply concenpt of fractals to draw a snowflake
Design UI for input : mainwindow.ui
Input: parameters required for drawing a snowflake (dimensions)
output: Snowflake of required dimensions
End

Rajdhani

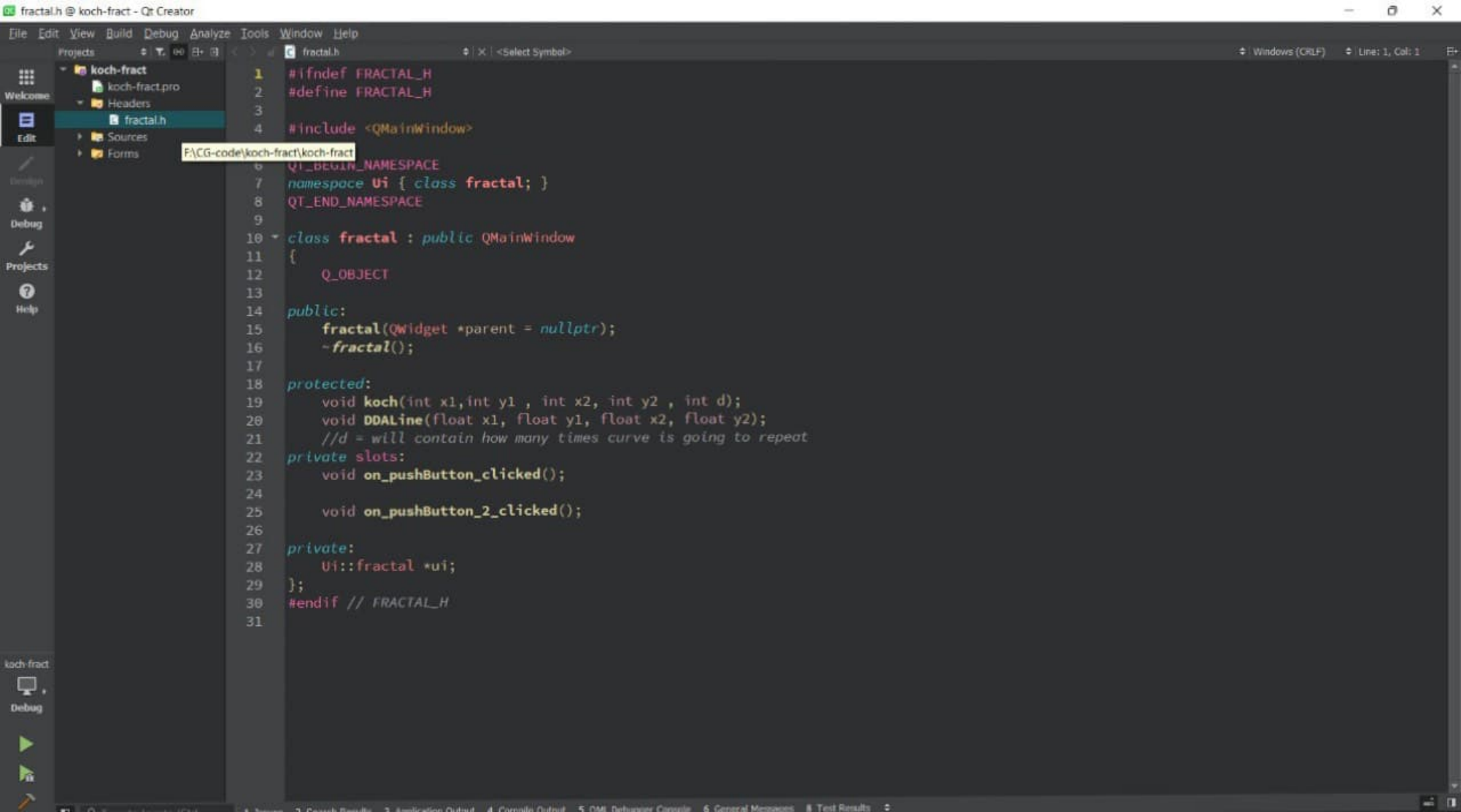
DATE / /

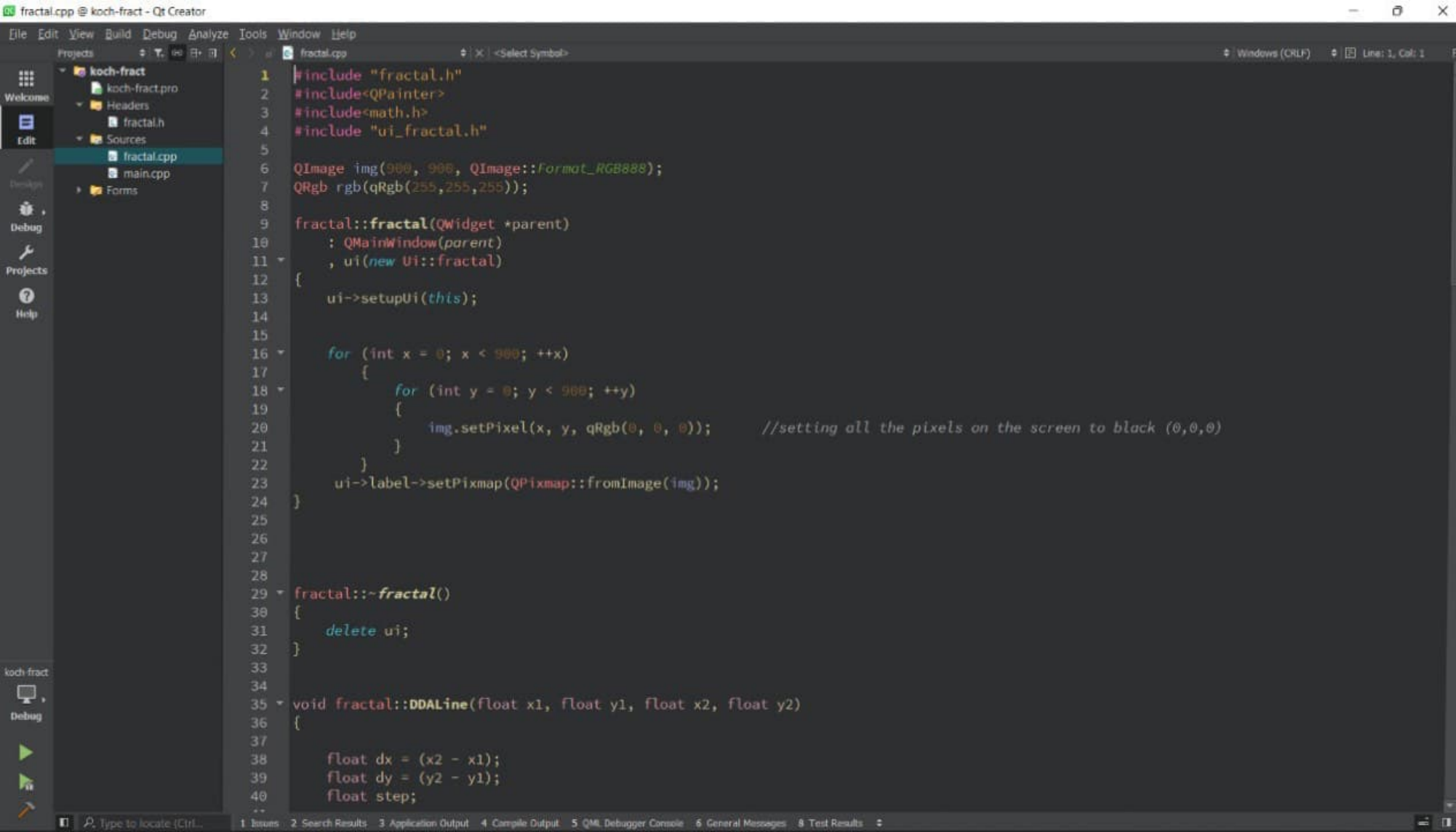
TEST CASES:

Sr.No.	Description	Expected output	Actual output	Status
1.	order: 0			Pass
2.	order: 1			Pass
3.	order: 2.			Pass.

CONCLUSION:

The knowledge and implementation of koch curves using the concept of fractals was successfully implemented.


fractal.h @ koch-fract - Qt Creator
File Edit View Build Debug Analyze Tools Window Help
Projects
koch-fract
koch-fract.pro
Headers
fractal.h
Sources
Forms
F:\CG-code\koch-fract\koch-fract
fractal.h
<Select Symbol>
Windows (CRLF)
Line: 1, Col: 1
Welcome
Edit
Design
Debug
Projects
Help
koch-fract
Debug
#ifndef FRACTAL_H
#define FRACTAL_H
#include <QMainWindow>
QT_BEGIN_NAMESPACE
namespace Ui { class fractal; }
QT_END_NAMESPACE
class fractal : public QMainWindow
{
Q_OBJECT
public:
fractal(QWidget *parent = nullptr);
~fractal();
protected:
void koch(int x1,int y1 , int x2, int y2 , int d);
void DDALine(float x1, float y1, float x2, float y2);
//d = will contain how many times curve is going to repeat
private slots:
void on_pushButton_clicked();
void on_pushButton_2_clicked();
private:
Ui::fractal *ui;
};
#endif // FRACTAL_H


fractal.cpp @ koch-fract - Qt Creator
File Edit View Build Debug Analyze Tools Window Help
Projects
koch-fract
koch-fract.pro
Headers
fractal.h
Sources
fractal.cpp
main.cpp
Forms
Welcome
Edit
Debug
Projects
Help
koch-fract
Debug
fractal.cpp
<Select Symbol>
Windows (CRLF)
Line: 1, Col: 1
#include "fractal.h"
#include<QPainter>
#include<math.h>
#include "ui_fractal.h"
QImage img(900, 900, QImage::Format_RGB888);
QRgb rgb(qRgb(255,255,255));
fractal::fractal(QWidget *parent)
: QMainWindow(parent)
, ui(new Ui::fractal)
{
ui->setupUi(this);
for (int x = 0; x < 900; ++x)
{
for (int y = 0; y < 900; ++y)
{
img.setPixel(x, y, qRgb(0, 0, 0)); //setting all the pixels on the screen to black (0,0,0)
}
}
ui->label->setPixmap(QPixmap::fromImage(img));
}
fractal::~fractal()
{
delete ui;
}
void fractal::DDALine(float x1, float y1, float x2, float y2)
{
float dx = (x2 - x1);
float dy = (y2 - y1);
float step;
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results

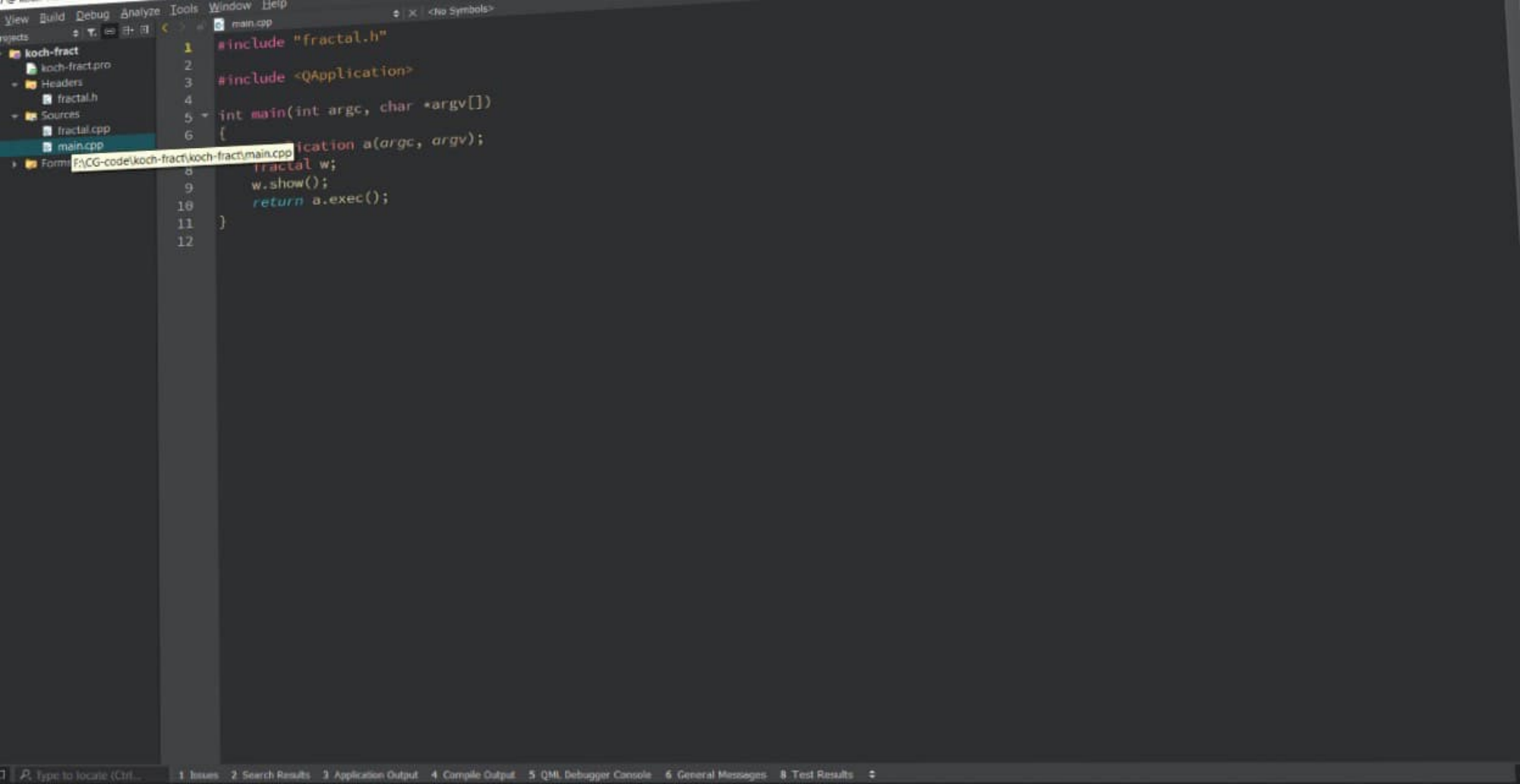
```
    }


    dx = dx / step;
    dy = dy / step;

    float x = x1, y = y1;
    int i = 1;

    while (i <= step)
    {

        x += dx;
        y += dy;
        img.setPixel(x, y, rgb);
        i++;
        ui->label->setPixmap(QPixmap::fromImage(img));

    }


}


void fractal::koch(int x1 , int y1, int x2 , int y2 , int d)
{
    double angle = 60*3.14/180;                    // Angle of equilateral triangle
    double x3 , y3 , x4, y4, x, y;
    x3 = (2*x1+x2)/3;                              // Coordinates found by using section formula
    y3 = (2*y1+y2)/3;

    x4 = (x1+2*x2)/3;
    y4 = (y1+2*y2)/3;

    x = x3 + (x4-x3)*cos(angle)+(y4-y3)*sin(angle);
    y = y3 - (x4-x3)*sin(angle)+(y4-y3)*cos(angle);

    if(d>0)
    {
        koch(x1,y1,x3,y3,d-1);                      // recursively calling the same function
        koch(x3,y3,x,y,d-1);
```

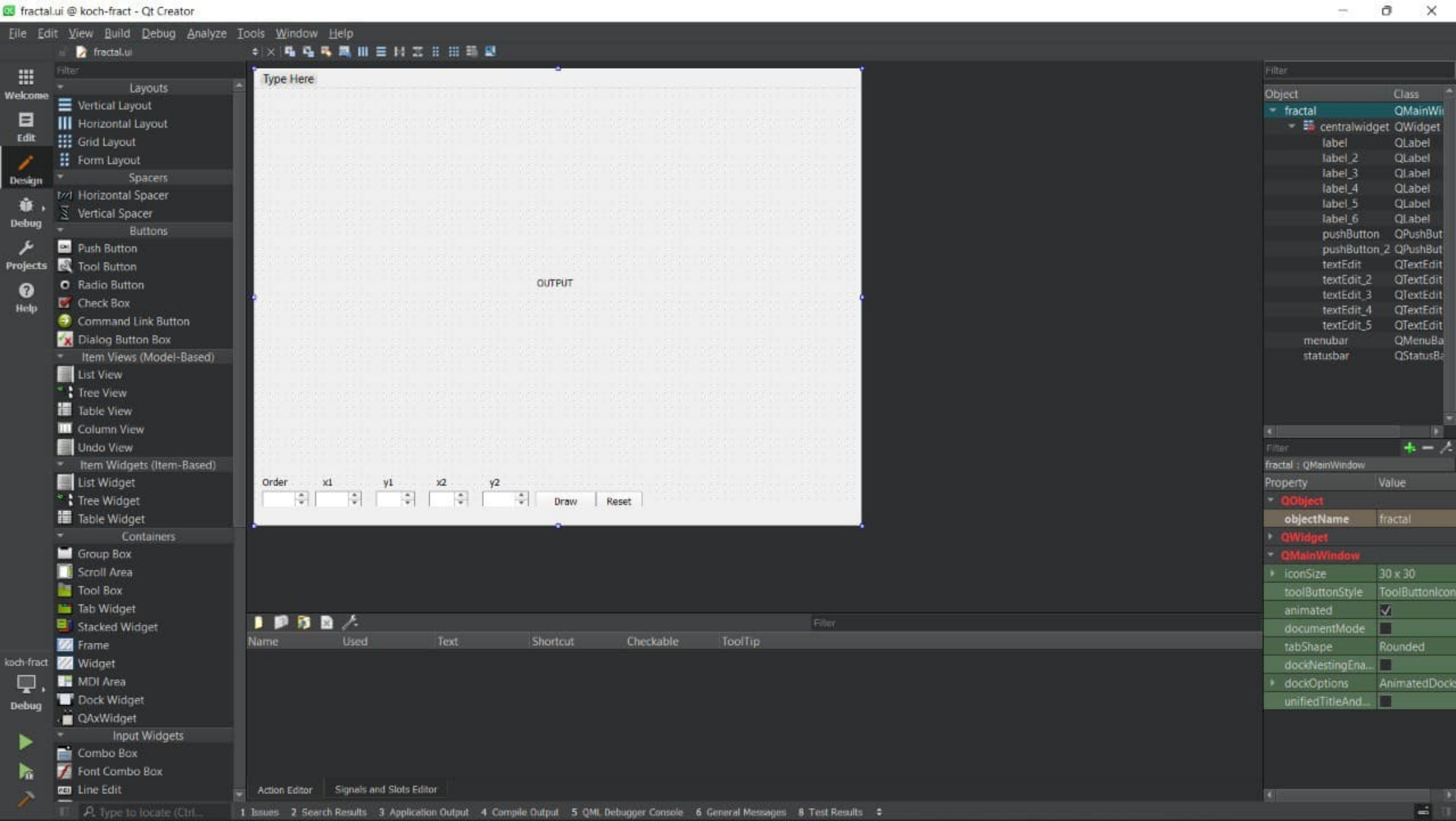
Tools Window Help

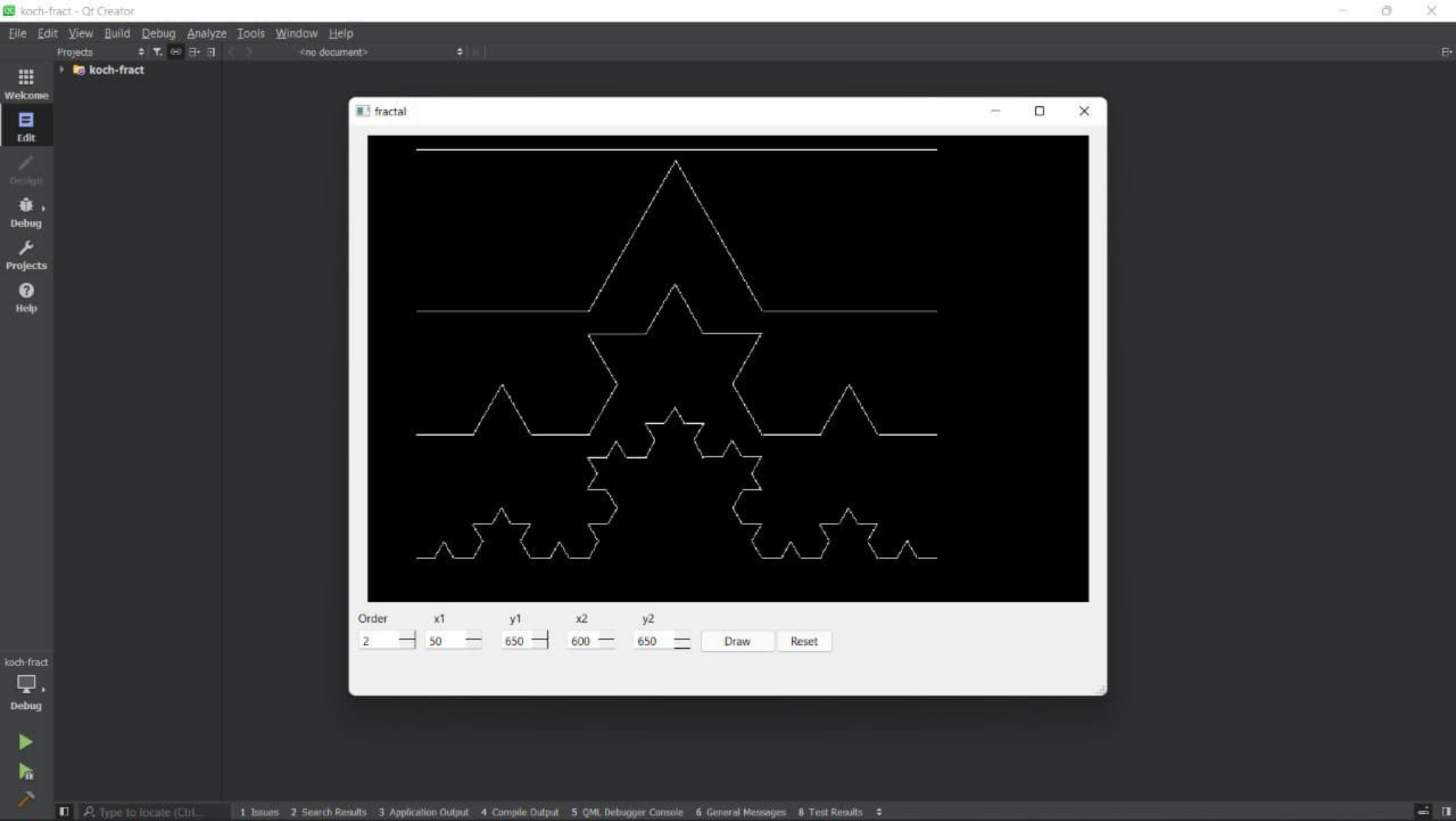
fractal.cpp

<Select Symbol>

```
97          DDALine(x,y,x4,y4);
98          DDALine(x4,y4,x2,y2);
99      }
100 }
101
102
103
104
105 void fractal::on_pushButton_clicked()
106 {
107     int order = ui->textEdit->toPlainText().toInt();
108     int a =  ui->textEdit_2->toPlainText().toInt();
109     int b =  ui->textEdit_3->toPlainText().toInt();
110     int c =  ui->textEdit_4->toPlainText().toInt();
111     int d =  ui->textEdit_5->toPlainText().toInt();
112     koch(a,b,c,d,order);
113     int count = 1;
114     for (int i = 0; i<order;order--) {
115          koch(a,b-(count*130),c,d-(count*130),order-1);
116          count++;
117     }
118    // koch(a,b-130,c,d-130,order-1);
119    // koch(a,b-260,c,d-260,order-2);
120     DDALine(a,b-430,c,d-430);
121
122 }
123
124
125 void fractal::on_pushButton_2_clicked()
126 {
127     for (int x = 0; x < 900; ++x)
128         {
129             for (int y = 0; y < 900; ++y)
130             {
131                 img.setPixel(x, y, qRgb(0, 0, 0));
132             }
133         }
134     ui->label->setPixmap(QPixmap::fromImage(img));
135
136 }
```


View Build Debug Analyze Tools Window Help
koch-fract
koch-fract.pro
Headers
fractal.h
Sources
fractal.cpp
main.cpp
Forms
F:\CG-code\koch-fract\koch-fract\main.cpp
main.cpp
<No Symbols>
#include "fractal.h"
#include <QApplication>
int main(int argc, char *argv[])
{
ication a(argc, argv);
fractal w;
w.show();
return a.exec();
}
Type to locate (Ctrl...
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 6 General Messages 8 Test Results


fractal.ui @ koch-fract - Qt Creator
File Edit View Build Debug Analyze Tools Window Help
Welcome
Edit
Design
Debug
Projects
Help
koch-fract
Debug
Filter
Layouts
Vertical Layout
Horizontal Layout
Grid Layout
Form Layout
Spacers
Horizontal Spacer
Vertical Spacer
Buttons
Push Button
Tool Button
Radio Button
Check Box
Command Link Button
Dialog Button Box
Item Views (Model-Based)
List View
Tree View
Table View
Column View
Undo View
Item Widgets (Item-Based)
List Widget
Tree Widget
Table Widget
Containers
Group Box
Scroll Area
Tool Box
Tab Widget
Stacked Widget
Frame
Widget
MDI Area
Dock Widget
QAxWidget
Input Widgets
Combo Box
Font Combo Box
Line Edit
Type Here
OUTPUT
Order
x1
y1
x2
y2
Draw
Reset
Object
Class
fractal
QMainWi
centralwidget
QWidget
label
QLabel
label_2
QLabel
label_3
QLabel
label_4
QLabel
label_5
QLabel
label_6
QLabel
pushButton
QPushBut
pushButton_2
QPushBut
textEdit
QTextEdit
textEdit_2
QTextEdit
textEdit_3
QTextEdit
textEdit_4
QTextEdit
textEdit_5
QTextEdit
menubar
QMenuBa
statusbar
QStatusBa
fractal : QMainWindow
Property
Value
QObject
objectName
fractal
QWidget
QMainWindow
iconSize
30 x 30
toolButtonStyle
ToolButtonIcon
animated
documentMode
tabShape
Rounded
dockNestingEna...
dockOptions
AnimatedDock
unifiedTitleAnd...
Name
Used
Text
Shortcut
Checkable
ToolTip
Action Editor
Signals and Slots Editor
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results


koch-fract - Qt Creator
File Edit View Build Debug Analyze Tools Window Help
Projects
<no document>
koch-fract
Welcome
Edit
Design
Debug
Projects
Help
koch-fract
Debug
fractal
Order
x1
y1
x2
y2
2
50
650
600
650
Draw
Reset
Type to locate (Ctrl...
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
6 General Messages
8 Test Results