HEADER

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = nullptr);
    void DDAline(int, int, int, int);
    void koch(int, int, int, int, int);
    ~MainWindow();

private slots:
    void on_pushButton_clicked();

    void on_pushButton_2_clicked();

private:
    Ui::MainWindow *ui;
};
#endif // MAINWINDOW_H
```

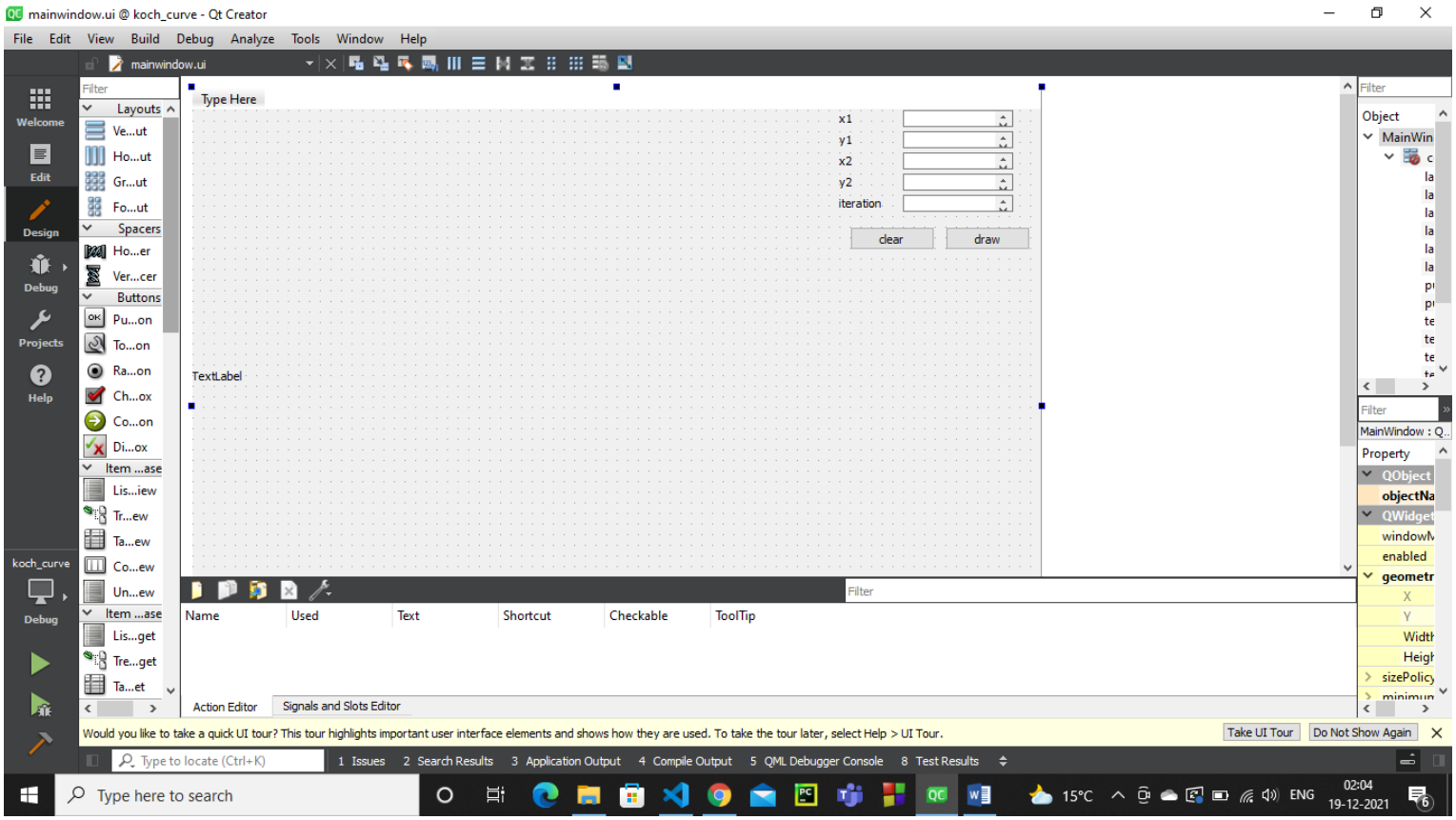
MAINWINDOW

```
#include "mainwindow.h"
#include<math.h>
#include "ui_mainwindow.h"

QImage img(500, 500, QImage ::Format_RGB888 );

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    for(int i=0; i<500;i++){
        for(int j=0; j<500;j++){
```

```
            img.setPixel(i, j, qRgb(0, 0, 0));

        }
    }
    ui->label->setPixmap(QPixmap :: fromImage(img));
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::DDAline(int x1, int y1, int x2, int y2){
    int Dx = x2 - x1, Dy = y2 - y1, step, k;
    float xin, yin, x = x1, y = y1;

    if(abs(Dx) > abs(Dy)){
        step = abs(Dx);
    }
    else{
        step = abs(Dy);
    }

    xin = Dx/(float)step;
    yin = Dy/(float)step;

    img.setPixel(round(x), round(y), qRgb(255, 255, 255));
    ui->label->setPixmap(QPixmap :: fromImage(img));

    for(k = 0; k < step; k++){
        x = x + xin;
        y = y + yin;
        img.setPixel(round(x), round(y), qRgb(255, 255, 255));
        ui->label->setPixmap(QPixmap :: fromImage(img));
    }
}

void MainWindow::koch(int x1, int y1, int x2, int y2, int iter){
    //Declare required variables
    int x3, y3, x4, y4, x, y;
    const float pi = 3.1415927;
    float angle = 60*pi/180;

    //Divide the input line in three equal parts
    x3 = (2*x1 + x2)/3;
    y3 = (2*y1 + y2)/3;
    x4 = (2*x2 + x1)/3;
    y4 = (2*y2 + y1)/3;
```

```
    //Calculate the coordinate of the apex of the mid-part of line
    x = x3 + (x4 - x3)*cos(angle) + (y4-y3)*sin(angle);
    y = y3 - (x4 - x3)*sin(angle) + (y4-y3)*cos(angle);

    //If there are moe than one iterations then call koch function for each
sub-divided line
    if(iter > 1){
        koch(x1, y1, x3, y3, iter - 1);
        koch(x3, y3, x, y, iter - 1);
        koch(x, y, x4, y4, iter -1);
        koch(x4, y4, x2, y2, iter - 1);
    }
    else{
        //Display all the calculated points
        DDAline(x1, y1, x3, y3);
        DDAline(x3, y3, x, y);
        DDAline(x, y, x4, y4);
        DDAline(x4, y4, x2, y2);
    }
}

void MainWindow::on_pushButton_clicked()
{
    //Clear
    for(int i=0; i<500;i++){
        for(int j=0; j<500;j++){
            img.setPixel(i, j, qRgb(0, 0, 0));

        }
    }
    ui->label->setPixmap(QPixmap :: fromImage(img));
}


void MainWindow::on_pushButton_2_clicked()
{
    //Draw kotch curve
    int x1, y1, x2, y2, iter;
    x1 = ui->textEdit->toPlainText().toInt();
    y1 = ui->textEdit_2->toPlainText().toInt();
    x2 = ui->textEdit_3->toPlainText().toInt();
    y2 = ui->textEdit_4->toPlainText().toInt();
    iter = ui->textEdit_5->toPlainText().toInt();

    koch(x1, y1, x2, y2, iter);


}
```

UI:


mainwindow.ui @ koch_curve - Qt Creator
File Edit View Build Debug Analyze Tools Window Help
mainwindow.ui
Type Here
x1
y1
x2
y2
iteration
clear
draw
TextLabel
Name
Used
Text
Shortcut
Checkable
ToolTip
Action Editor
Signals and Slots Editor
Would you like to take a quick UI tour? This tour highlights important user interface elements and shows how they are used. To take the tour later, select Help > UI Tour.
Take UI Tour
Do Not Show Again
Type here to search
02:04
19-12-2021