

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);
    ~MainWindow();

private slots:
    void on_pushButton_clicked();

    void on_pushButton_2_clicked();

    void on_pushButton_3_clicked();

    void on_pushButton_4_clicked();

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

MAINWINDOW

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
QImage img(400,400,QImage::Format_RGB888);

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
}

MainWindow::~MainWindow()
```

```

{
    delete ui;
}

class Transformation{
    friend MainWindow;
public:
    int a[3][3];int t[3][3];
    Transformation(){
        a[0][2]=a[1][2]=a[2][2]=1;
        a[0][0]=200;a[0][1]=200;
        a[1][0]=213;a[1][1]=248;
        a[2][0]=248;a[2][1]=213;

    }

    void operator *(float b[3][3]){
        for(int i=0;i<3;i++){
            for(int j=0;j<3;j++){
                t[i][j]=0;
                for(int k=0;k<3;k++){
                    t[i][j]=t[i][j]+(a[i][k]*b[k][j]);
                }
            }
        }
    }

    void DDA(int x1,int y1,int x2,int y2){
        float x,y,dx,dy,step;
        int i;
        dx=x2-x1;
        dy=y2-y1;
        if(abs(dx)>abs(dy)){
            step=abs(dx);
        }
        else{
            step=abs(dy);
        }

        dx=dx/step;
        dy=dy/step;
        x=x1;y=y1;
        i=1;
        while(i<=step){
            x=x+dx;
            y=y+dy;
            display(x,y);
            i++;
        }
    }
}

```

```

    }
}
void display(float x,float y){
    img.setPixel(x,y,qRgb(255,255,255));
}
};

Transformation t1;

void MainWindow::on_pushButton_clicked()
{
    //for grid lines
    t1.DDA(0,200,400,200);
    t1.DDA(200,0,200,400);

    //coordinates for triangle
    t1.DDA(200,200,213,248);
    t1.DDA(213,248,248,213);
    t1.DDA(248,213,200,200);

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

void MainWindow::on_pushButton_2_clicked()
{
    int tx,ty;
    float b[3][3];
    tx=ui->textEdit->toPlainText().toInt();
    ty=ui->textEdit_2->toPlainText().toInt();
    b[0][0]=b[1][1]=b[2][2]=1;
    b[0][1]=b[0][2]=b[1][0]=b[1][2]=0;
    b[2][0]=tx;b[2][1]=ty;
    t1*b;
    t1.DDA(t1.t[0][0],t1.t[0][1],t1.t[1][0],t1.t[1][1]);
    t1.DDA(t1.t[1][0],t1.t[1][1],t1.t[2][0],t1.t[2][1]);
    t1.DDA(t1.t[2][0],t1.t[2][1],t1.t[0][0],t1.t[0][1]);
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            t1.a[i][j]=t1.t[i][j];
        }
    }
    ui->label->setPixmap(QPixmap::fromImage(img));
}

```

```

void MainWindow::on_pushButton_3_clicked()
{
    float x;
    float cost,sint,b[3][3];
    x=ui->textEdit_3->toPlainText().toFloat();
    x=x*3.14159/180;
    cost=cos(x);sint=sin(x);
    b[2][2]=1;
    b[0][2]=b[1][2]=0;
    b[2][0]=(-200*cost)+(200*sint)+200;
    b[2][1]=(-200*sint)-(200*cost)+200;
    b[0][0]=cost;b[0][1]=sint;
    b[1][0]=-sint;b[1][1]=cost;
    t1*b;
    t1.DDA(t1.t[0][0],t1.t[0][1],t1.t[1][0],t1.t[1][1]);
    t1.DDA(t1.t[1][0],t1.t[1][1],t1.t[2][0],t1.t[2][1]);
    t1.DDA(t1.t[2][0],t1.t[2][1],t1.t[0][0],t1.t[0][1]);
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            t1.a[i][j]=t1.t[i][j];
        }
    }
    ui->label->setPixmap(QPixmap::fromImage(img));
}

```

```

void MainWindow::on_pushButton_4_clicked()
{
    float sx,sy,b[3][3];
    sx=ui->textEdit_4->toPlainText().toFloat();
    sy=ui->textEdit_5->toPlainText().toFloat();
    b[0][0]=sx;b[1][1]=sy;
    b[2][2]=1;
    b[0][1]=b[0][2]=b[1][0]=b[1][2]=0;
    b[2][0]=-(sx*200)+200;
    b[2][1]=-(sy*200)+200;
    t1*b;
    t1.DDA(t1.t[0][0],t1.t[0][1],t1.t[1][0],t1.t[1][1]);
    t1.DDA(t1.t[1][0],t1.t[1][1],t1.t[2][0],t1.t[2][1]);
    t1.DDA(t1.t[2][0],t1.t[2][1],t1.t[0][0],t1.t[0][1]);
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            t1.a[i][j]=t1.t[i][j];
        }
    }
    ui->label->setPixmap(QPixmap::fromImage(img));
}

```

UI

