```
//CG Lab 03 Scan Fill



#include "mainwindow.h"
#include "ui_mainwindow.h"
#include"QColorDialog"
#include<QMouseEvent>
#include<QtDebug>
#include<QTime>
#include<iostream>

using namespace std;


QImage img(500,500,QImage::Format_RGB888);
QColor color;
int ver=0,temp,i,j,a[20],b[20];
float slope[20],dx,dy,x[20];

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    ver=0;
    start=true;
}

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


void MainWindow::on_pushButton_clicked()
{
    color=QColorDialog::getColor();
}

void MainWindow::dda(int x1,int y1, int x2, int y2)
{
   int dx,dy,p;
   int x,y;
   int i=0;
   x=x1;
   y=y1;
    dx=abs(x2-x1);
    dy=abs(y2-y1);
    if(dx>=dy)
    {
      p=2*dy-dx;
      while(i<=dx)
```

```
      {img.setPixel(x,y,color.rgb());
          if(p<0)
          {
              p=p+2*dy;
          }
          else{
              p=p+2*(dy-dx);
              y=y+sign(y2-y1);
          }
          x=x+sign(x2-x1);
          i++;
      }
        }
    else{

        p=2*dx-dy;
        while(i<=dy)
        {img.setPixel(x,y,color.rgb());
            if(p<0)
            {
                p=p+2*dx;
            }
            else{
                p=p+2*(dx-dy);
                x=x+sign(x2-x1);
            }
            y=y+sign(y2-y1);
            i++;
        }
    }
    ui->label->setPixmap(QPixmap::fromImage(img));
}

int MainWindow::sign(float x)
{
    if(x<0)
        return -1;
    else
        return 1;
}

void MainWindow::mousePressEvent(QMouseEvent *ev)
{if(start)
    {
        int p=ev->pos().x();
        int q=ev->pos().y();
        a[ver]=p;
        b[ver]=q;
        if(ev->button()==Qt::RightButton)
        {
          dda(a[0],b[0],a[ver-1],b[ver-1]);
          start=false;
        }
        else{
            if(ver>0)
            {
               dda(a[ver],b[ver],a[ver-1],b[ver-1]);
```

```
            }
        }
        ver++;
    }

}

void MainWindow::on_pushButton_2_clicked()
{   int ymax=0,ymin=100000000;
            a[ver]=a[0];
            b[ver]=b[0];
            for( int i=0;i<ver-1;i++)
            {
               if(b[i]>ymax)
               ymax=b[i];
               if(b[i]<ymin)
               ymin=b[i];
                }
            for(int i=0;i<ver;i++){
                dx=a[i+1]-a[i];
                dy=b[i+1]-b[i];
                if(dx==0.0){slope[i]=1.0;}
                if(dy==0.0){slope[i]=0.0;}
                if(dx!=0.0 and dy!=0.0){
                    slope[i]=float(dx/dy);
                }
            }


            for(int y=0;y<500;y++){
                int index=0;
                for(int i=0;i<ver;i++){
                    if((y>=b[i] and y<b[i+1])or(y>=b[i+1] and y<b[i])){
                        x[index]=a[i]+slope[i]*(y-b[i]);
                        index++;
                    }
                }

                for(int i=0;i<index-1;i++){
                    for(int k=0;k<index-1-i;k++){
                        if(x[k]>x[k+1]){
                            temp=x[k];
                            x[k]=x[k+1];
                            x[k+1]=temp;
                        }
                    }
                }
                //calling line drawing algorithm
                for(int j=0;j<index;j+=2){
                    dda(x[j],y,x[j+1],y);
                }
            }

}
```