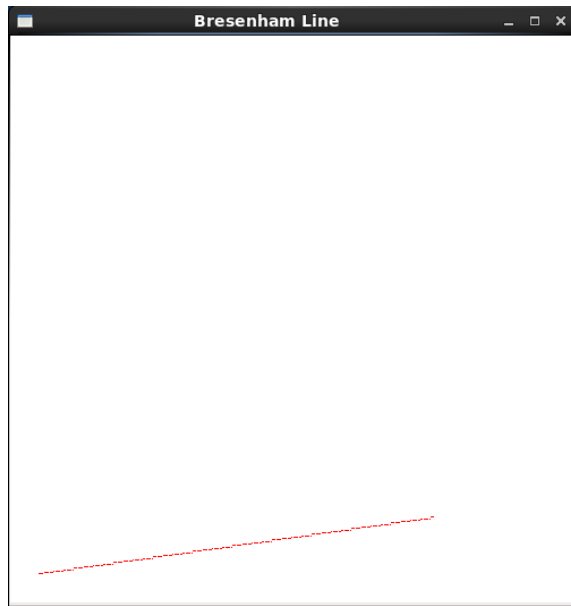


Yazhuo Liu

Homework 1

1. Output:



(20, 20) – (300, 60)



(20, 20) – (60, 300)

Partial code:

```
...
void line()
{
    int x0 = 20, y0 = 20, xn = 300, yn = 60, x, y;
    int  dx, dy,          //deltas
        pk,              //decision parameter
        k;               //looping variable

    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f( 1, 0, 0 );
    setPixel(x0, y0);    //plot first point

    // difference between starting and ending points
    dx = xn - x0;
    dy = yn - y0;
    pk = 2 * dy - dx;
    x = x0;      y = y0;
    double m = dy / dx;

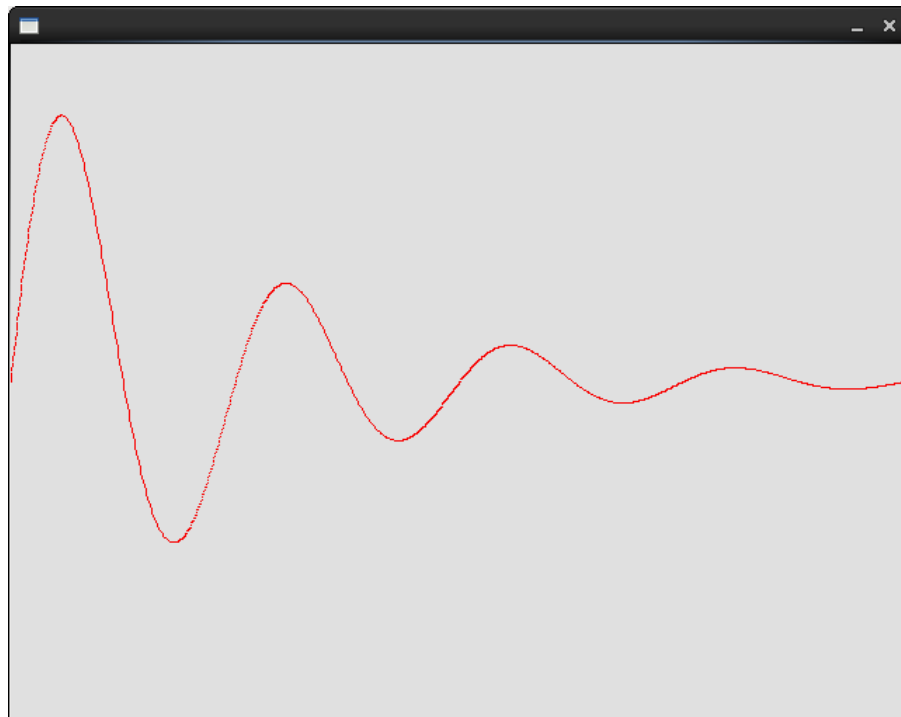
    if (m <= 1 && m >= -1) {
        for ( k = 0; k < dx-1; ++k ) {
            if ( pk < 0 ) {
                pk = pk + 2 * dy;                //calculate next pk
                //next pixel: (x+1, y )
            }
            else {
                pk = pk + 2 * dx;
                y = y + 1;
            }
            setPixel(++x, y);
        }
    }
}
```

```

    } else {
        pk = pk + 2*dy - 2*dx;    //next pixel: (x+1, y+1)
        ++y;                    //calculate next pk
    }
    ++x;
    setPixel( x, y );
}
}
else if ( m > 1 || m < -1 ) {
    for ( k = 0; k < dy-1; ++k ) {
        if ( pk < 0 ) {
            pk = pk + 2 * dx;    //calculate next pk
            //next pixel: (x+1, y )
        } else {
            //next pixel: (x+1, y+1)
            pk = pk + 2*dx - 2*dy; //calculate next pk
            ++x;
        }
        ++y;
        setPixel( x, y );
    }
}
glFlush();
}
...

```

2. Output:



$$f(x) = e^{-|x|} \sin (2\pi x)$$

Partial code:

```
...
double f ( double x )
{
    double y = exp ( -fabs ( x ) ) * sin ( 2 * 3.1415926 * x );

    return y;
}

int main()
{
    const int VWIDTH = 640;
    const int VHEIGHT = 480;

    const Point center ( VWIDTH/2, VHEIGHT/2 ); //center of screen
    Surface surf( VWIDTH, VHEIGHT, (char *) "draw_line" );

    surf.clearScreen();                //clear screen
    surf.updateSurface();
    // SDL_Delay ( 1000 );              //dealy one second, just for demo

    surf.setBackgroundColor ( 0xe0, 0xe0, 0xe0 ); //set background to grey

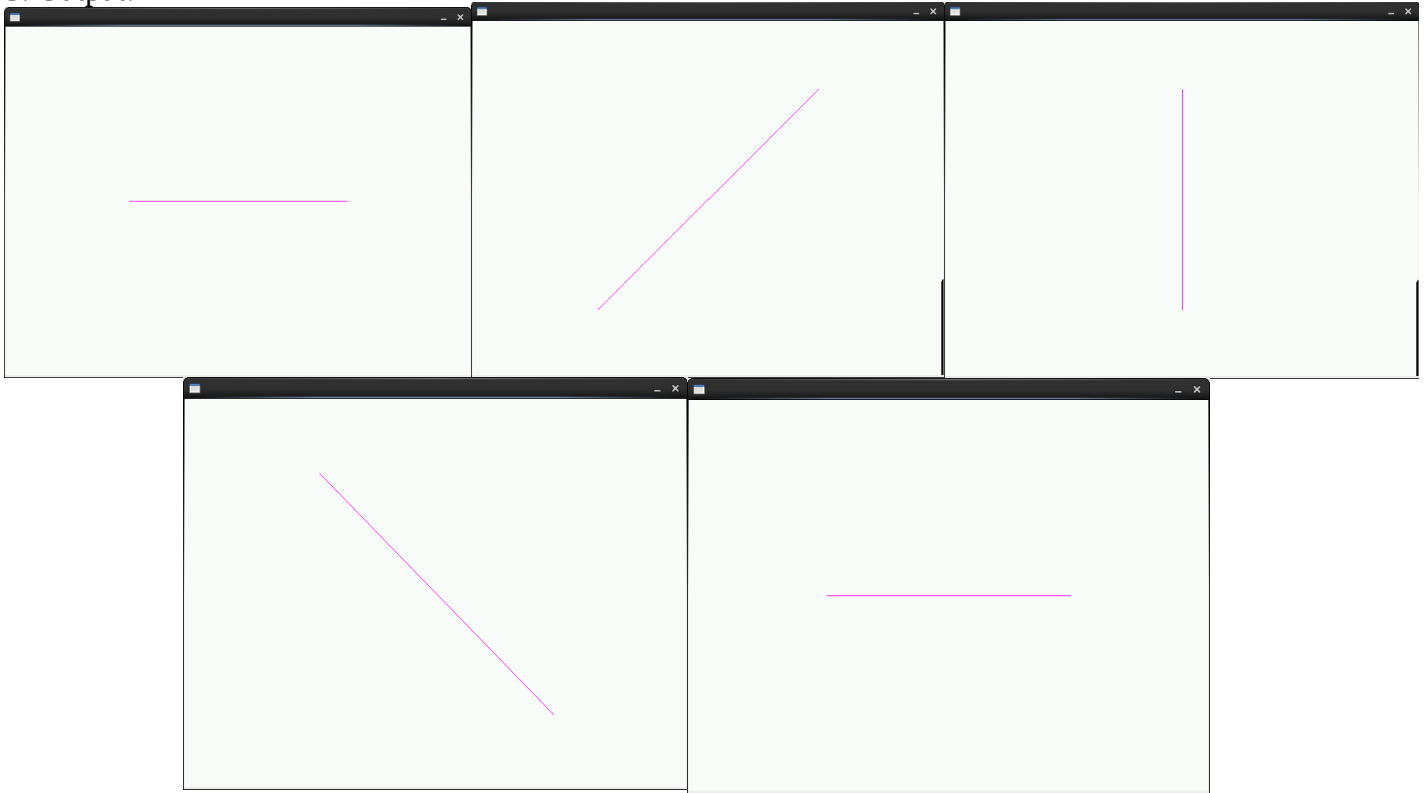
    //draw a line
    surf.setColor ( 0xff, 0, 0 );      //using red color

    int sx, sy;
    double x, y, a, b, c, d;
    a = VWIDTH / 4.0;
    c = 0;
    b = -VHEIGHT / 2.0;
    d = VHEIGHT / 2.0;

    x = 0.0;                          //initial position
    y = f ( 0.0 );
    sx = (int) ( a * x + c );
    sy = (int) ( b * y + d );
    surf.moveTo ( sx, sy );

    for ( x = 0; x < 4.0; x += 0.005 ) {
        y = f ( x );
        sx = (int) ( a * x + c );
        sy = (int) ( b * y + d );
        surf.lineTo ( sx, sy );
    }
    ...
```

3. Output:



Code:

...

```
surf.setColor (0xff, 0, 0xff);  
surf.moveTo (center.x - 150, center.y);  
surf.forward (300, 1);  
  
surf.updateSurface();  
SDL_Delay ( 2000 );  
surf.clearScreen();  
  
surf.moveTo (center.x - 150, center.y + 150);  
surf.lineTo (center.x + 150, center.y - 150);  
  
surf.updateSurface();  
SDL_Delay ( 2000 );  
surf.clearScreen();  
  
surf.moveTo (center.x, center.y + 150);  
surf.lineTo (center.x, center.y - 150);  
  
surf.updateSurface();  
SDL_Delay ( 2000 );  
surf.clearScreen();  
  
surf.moveTo (center.x + 150, center.y + 150);
```

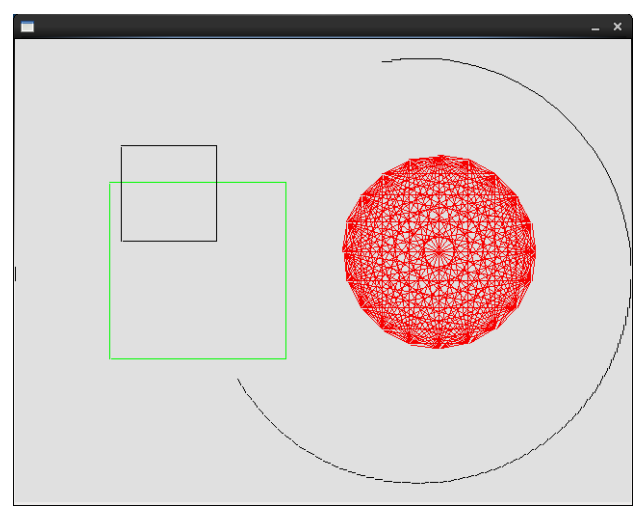
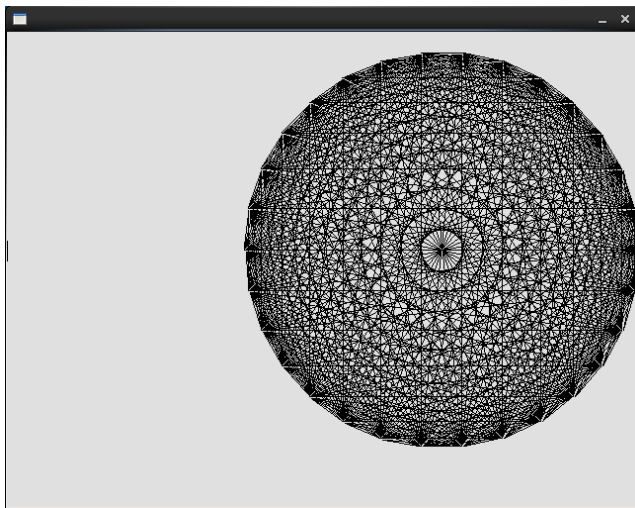
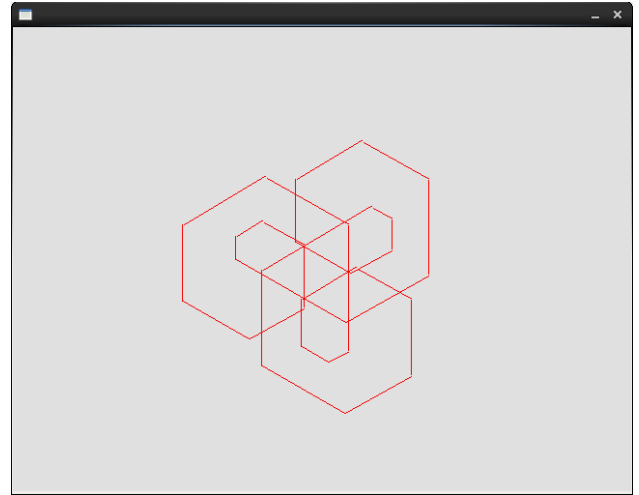
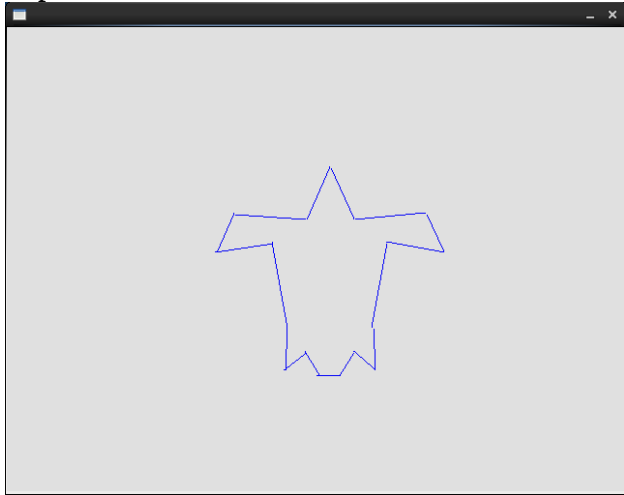
```
surf.lineTo (center.x - 150, center.y - 150);
```

```
surf.updateSurface();  
SDL_Delay ( 2000 );  
surf.clearScreen();
```

```
surf.moveTo (center.x - 150, center.y);  
surf.forward (300, 1);
```

...

4. Output:



Code:

```
...  
void draw_no_3 (Surface &surf, int L)  
{  
    surf.turn( 330 );  
    surf.forward( L/3*4, 1 );  
    surf.turn( 60 );  
    surf.forward( L, 1 );
```

```

surf.turn( 60 );
surf.forward( L, 1 );
surf.turn( 60 );
surf.forward( L/6*5, 1 );
surf.turn( 60 );
surf.forward( L/6*5, 1 );
surf.turn( 60 );
surf.forward( L/3*2, 1 );
surf.turn( 60 );
surf.forward( L/3*2, 1 );
surf.turn( 60 );
surf.forward( L/2, 1 );
surf.turn( 60 );
surf.forward( L/3, 1 );
surf.turn( 60 );
surf.forward( L/4, 1 );
surf.turn( 60 );
}

```

```

void draw_3(Surface & surf, int L)
{
    draw_no_3( surf, L );
    surf.turn( 30 );
    draw_no_3( surf, L );
    surf.turn( 30 );
    draw_no_3( surf, L );
}

```

```

//draw turtle 1
void draw_turtle1( Surface & surf, int L)
{
    surf.forward( L * 0.8, 1 );
    surf.turn( 60 );
    surf.forward( L, 1 );
    surf.turn( 260 );
    surf.forward( L, 1 );
    surf.turn( 135 );
    surf.forward( L * 1.5, 1 );
    surf.turn( 345 );
    surf.forward( L * 3, 1 );
    surf.turn( 270 );
    surf.forward( L * 2, 1 );
    surf.turn( 125 );
    surf.forward( L * 1.5, 1 );
    surf.turn( 70 );
    surf.forward( L * 2.5, 1 );
    surf.turn( 290 );
    surf.forward( L * 2, 1 );
    surf.turn( 130 );
}

```

```

surf.forward( L * 2, 1 );
surf.turn( 290 );
surf.forward( L * 2.5, 1 );
surf.turn( 70 );
surf.forward( L * 1.5, 1 );
surf.turn( 125 );
surf.forward( L * 2, 1 );
surf.turn( 270 );
surf.forward( L * 3, 1 );
surf.turn( 345 );
surf.forward( L * 1.5, 1 );
surf.turn( 135 );
surf.forward( L, 1 );
surf.turn( 260 );
surf.forward( L, 1 );
surf.turn( 60 );
}

//draw rosette with N-sided polygon
void rosette (Surface &surf, int N, int radius )
{
    if ( N < 3 ) return;
    Point pt[N+1];

    int cx = surf.getCP().x;
    int cy = surf.getCP().y;
    double angle = 0; //initial angle
    double angleInc = 2 * 3.14159265 / N; //angle increment
    pt[0] = Point ( ( int) (radius * cos( angle ) + cx),
                    ( int) ( radius * sin ( angle ) + cy ) );
    for ( int k = 1; k < N; k++ ) { //repeat n times
        angle += angleInc;
        pt[k] = Point ( ( int) (radius * cos( angle ) + cx),
                        ( int) ( radius * sin ( angle ) + cy ) );
    }
    for ( int i = 0; i < N - 1; i++ ) {
        for ( int j = i + 1; j < N; j++ ) {
            surf.moveTo ( pt[i] ); //connect all vertices
            surf.lineTo ( pt[j] );
        }
    }
} //rosette

...

//draw an arc
surf.setColor (0, 0, 0);
surf.moveTo (center.x + 100, center.y);
draw_arc ( surf, 100, 220, 260, 250 );

```

```
//draw a big square
surf.setColor (0, 0xff, 0);
surf.moveTo (center.x - 130, center.y);
draw_polygon (surf, 4, 130, 45);

//draw a small square
surf.setColor (0, 0, 0);
surf.moveTo (center.x - 160, center.y - 80);
draw_polygon (surf, 4, 70, 45);

//draw an 8-sided rosette
surf.setColor ( 0xff, 0, 0 );           //using black color
surf.moveTo ( center.x + 120, center.y - 20 );
rosette ( surf, 20, 100 );
...
```

Report:

The homework problems are pretty hard for me to do. I don't even think I can complete it. I did my best but I couldn't really understand the materials taught in class. Part one of the homework I got a and b, but have troubles doing c and d. Part two I did it but I don't think I really understood what the question is asking. Part three I feel I did well, I successfully animated the rotation of the line. Part four is the most difficult one, I could only do four images. I feel like the homework is so much harder than the labs, especially when doing it without any help.