ASSIGNMENT-3

GRAPHICS LAB

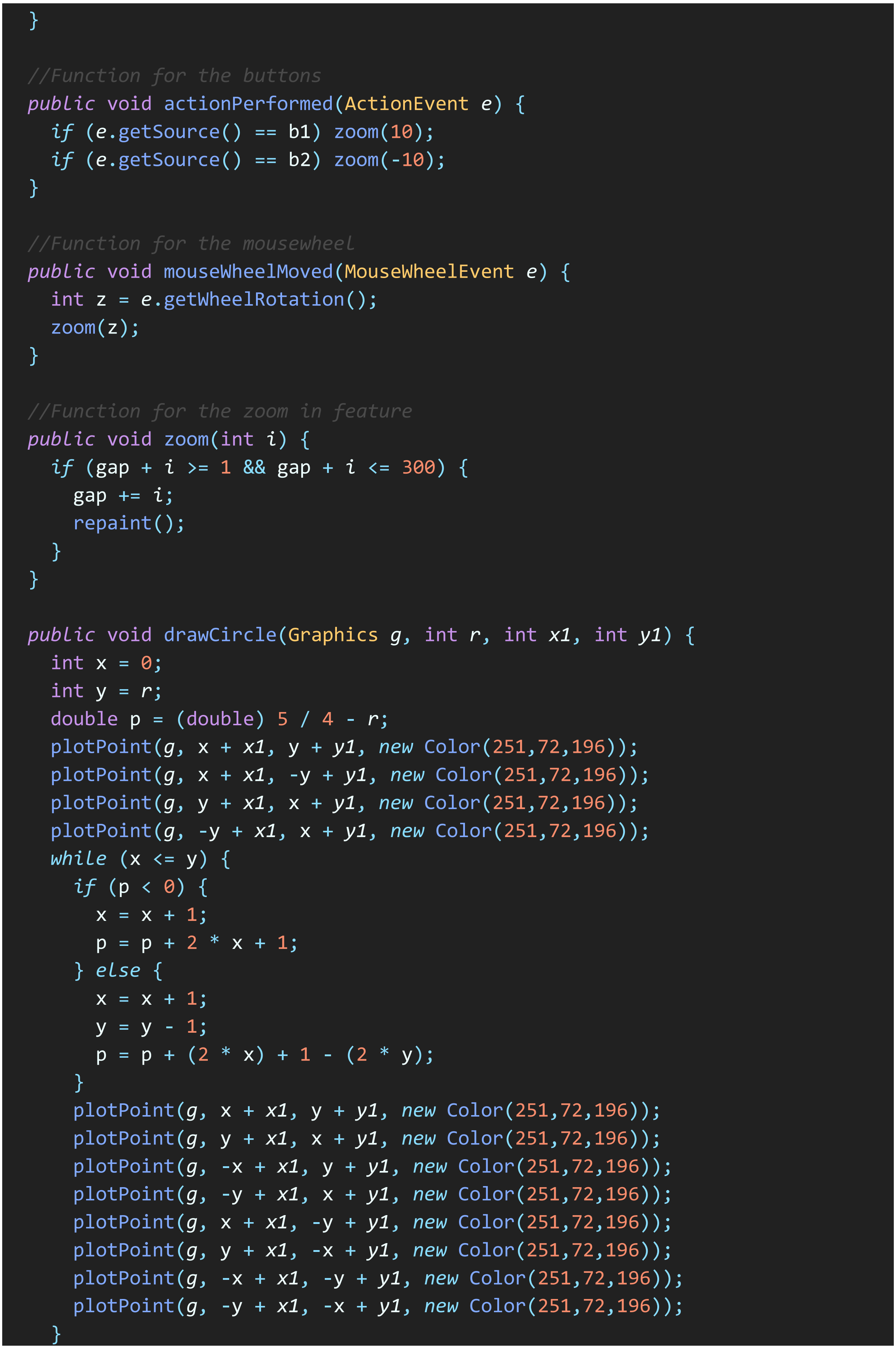
Subhajit Samanta

2020CSB046

➢ PROBLEM-1:

1. draw.java-

```
import java.applet.*;

import java.awt.*;
import java.awt.event.*;

public class draw extends Applet implements ActionListener, MouseWheelListener
{

  int originX, originY;
  int height, width;
  int gap = 20;
  Button b1 = new Button(" + ");
  Button b2 = new Button(" - ");

  public void init() {
    setBackground(Color.black);
    b1.setBackground(new Color(31, 70, 144));
    b2.setBackground(new Color(255, 229, 180));
    add(b1);
    add(b2);
    addMouseWheelListener(this);
    b1.addActionListener(this);
    b2.addActionListener(this);
  }

  public void paint(Graphics g) {
    g.setColor(Color.BLACK);
    height = getHeight();
    width = getWidth();
    originX = (getX() + width) / 2;
    originY = (getY() + height) / 2;
    // drawGrid(g);
    // drawCircle(g, 50, 0, 0);
    drawShape(g);
    }

  public void drawShape(Graphics g) {
    drawCircle(g, 100, 0, 0);
    drawCircle(g, 46, 46, 27);
    drawCircle(g, 46, -46, 27);
  drawCircle(g, 46, 0, -54);
  drawCircle(g, 8, 0, 0);
  drawCircle(g, 22, 0, 78);
  drawCircle(g, 22, 65, -38);
  drawCircle(g, 22, -65, -38);
  drawCircle(g, 10, 33, 82);
  drawCircle(g, 10, -33, 82);
  drawCircle(g, 10, -88, -14);
  drawCircle(g, 10, 88, -14);
  drawCircle(g, 10, -54, -70);
  drawCircle(g, 10, 54, -70);
  drawCircle(g, 5, 50, 80);
  drawCircle(g, 5, -50, 80);
  drawCircle(g, 6, 94, 3);
  drawCircle(g, 6, -94, 3);
  drawCircle(g, 6, 45, -83);
  drawCircle(g, 6, -45, -83);
  drawCircle(g, 4, 58, 76);
  drawCircle(g, 4, -58, 76);
  drawCircle(g, 3, 95, 14);
  drawCircle(g, 3, -95, 14);
  drawCircle(g, 3, 36, -88);
  drawCircle(g, 3, -36, -88);
  drawCircle(g, 4, 0, 50);
  drawCircle(g, 4, 42, -24);
  drawCircle(g, 4, -42, -24);
  drawCircle(g, 2, 65, 72);
  drawCircle(g, 2, -65, 72);
  drawCircle(g, 3, 95, 22);
  drawCircle(g, 3, -95, 22);
  drawCircle(g, 2, 31, -92);
  drawCircle(g, 2, -31, -92);
  drawCircle(g, 4, 23, 92);
  drawCircle(g, 4, -23, 92);
  drawCircle(g, 4, 91, -30);
  drawCircle(g, 4, -91, -30);
  drawCircle(g, 4, 69, -64);
  drawCircle(g, 4, -69, -64);
}

//Function to draw origin
public void drawOriginCircle(Graphics g) {
  g.setColor(Color.RED);
  g.fillOval(originX - 5, originY - 5, 10, 10);
}

//Function for plotting points
public void plotPoint(Graphics g, int x, int y, Color c) {
  g.setColor(c);
  g.fillRect(
    originX + (x * gap) - gap / 8,
    originY - (y * gap) - gap / 8,
    gap / 2,
    gap / 2
  );
}

//Function to draw X-axis
public void drawXaxis(Graphics g) {
  g.setColor(Color.BLUE);
  g.fillRect(0, originY - 2, width, 4);
}

//Function to draw Y-axis
public void drawYaxis(Graphics g) {
  g.setColor(Color.BLUE);
  g.fillRect(originX - 2, 0, 4, height);
}

// Function to draw the Grid
public void drawGrid(Graphics g) {
  drawHorizontalLines(g);
  drawVeritcalLines(g);
}

//Function to draw the horizontal lines of the grid
public void drawHorizontalLines(Graphics g) {
  g.setColor(Color.YELLOW);
  for (int i = originX; i <= width; i += gap) {
    g.drawLine(i, 0, i, height);
  }
  for (int i = originX; i >= 0; i -= gap) {
    g.drawLine(i, 0, i, height);
  }
}

//Function to draw the vertical lines of the grid
public void drawVeritcalLines(Graphics g) {
  g.setColor(Color.YELLOW);
  for (int i = originY; i <= height; i += gap) {
    g.drawLine(0, i, width, i);
    // add coordinate text
  }
  for (int i = originY; i >= 0; i -= gap) {
    g.drawLine(0, i, width, i);
  }
```

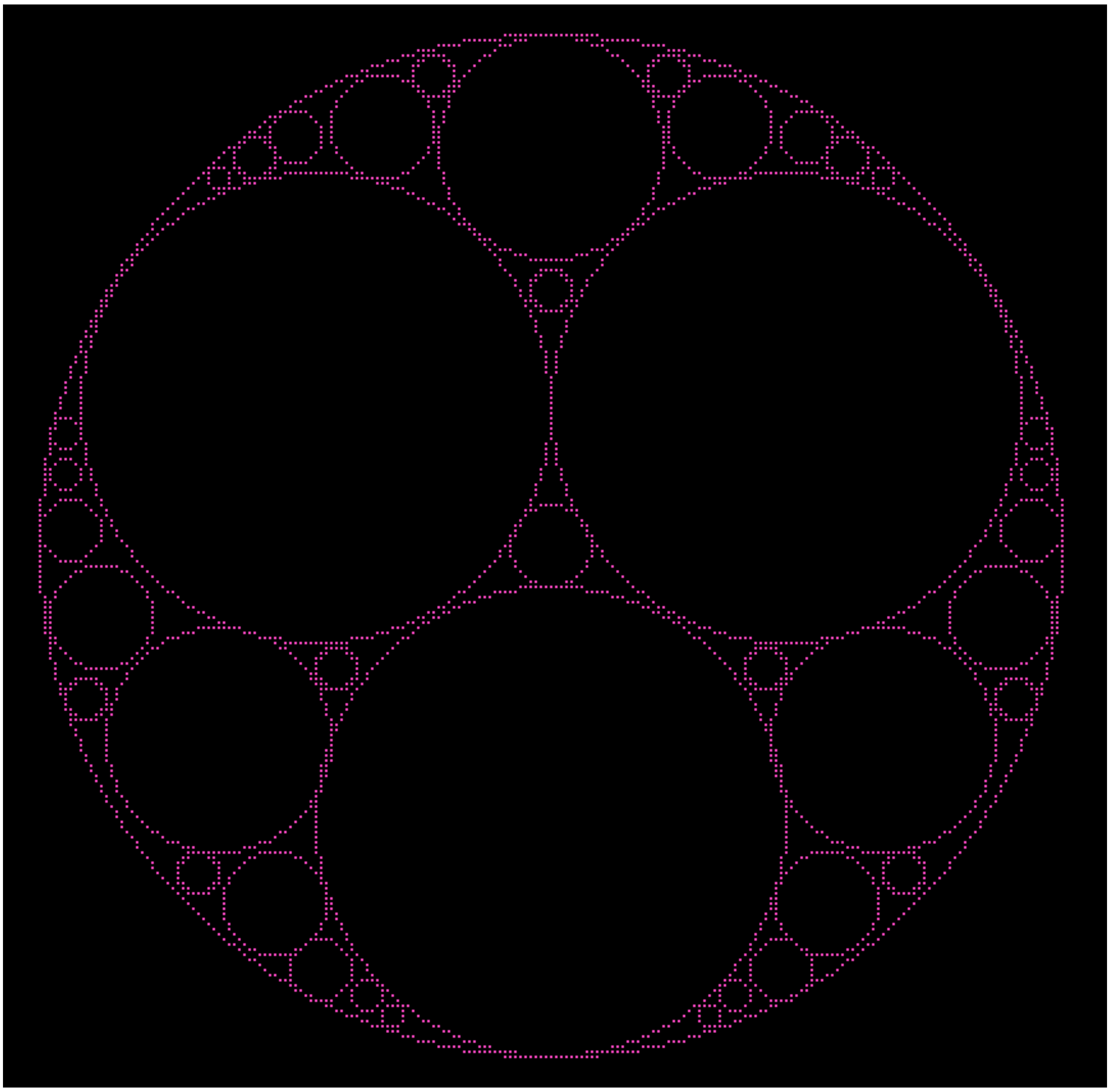


```
}

//Function for the buttons
public void actionPerformed(ActionEvent e) {
  if (e.getSource() == b1) zoom(10);
  if (e.getSource() == b2) zoom(-10);
}

//Function for the mousewheel
public void mouseWheelMoved(MouseWheelEvent e) {
  int z = e.getWheelRotation();
  zoom(z);
}

//Function for the zoom in feature
public void zoom(int i) {
  if (gap + i >= 1 && gap + i <= 300) {
    gap += i;
    repaint();
  }
}

public void drawCircle(Graphics g, int r, int x1, int y1) {
  int x = 0;
  int y = r;
  double p = (double) 5 / 4 - r;
  plotPoint(g, x + x1, y + y1, new Color(251,72,196));
  plotPoint(g, x + x1, -y + y1, new Color(251,72,196));
  plotPoint(g, y + x1, x + y1, new Color(251,72,196));
  plotPoint(g, -y + x1, x + y1, new Color(251,72,196));
  while (x <= y) {
    if (p < 0) {
      x = x + 1;
      p = p + 2 * x + 1;
    } else {
      x = x + 1;
      y = y - 1;
      p = p + (2 * x) + 1 - (2 * y);
    }
    plotPoint(g, x + x1, y + y1, new Color(251,72,196));
    plotPoint(g, y + x1, x + y1, new Color(251,72,196));
    plotPoint(g, -x + x1, y + y1, new Color(251,72,196));
    plotPoint(g, -y + x1, x + y1, new Color(251,72,196));
    plotPoint(g, x + x1, -y + y1, new Color(251,72,196));
    plotPoint(g, y + x1, -x + y1, new Color(251,72,196));
    plotPoint(g, -x + x1, -y + y1, new Color(251,72,196));
    plotPoint(g, -y + x1, -x + y1, new Color(251,72,196));
  }
  }
}
```

2. index.html-

```
<html>

  <head> </head>
  <body>
    <applet code="draw.class" width="800" height="800"></applet>
  </body>
</html>
```

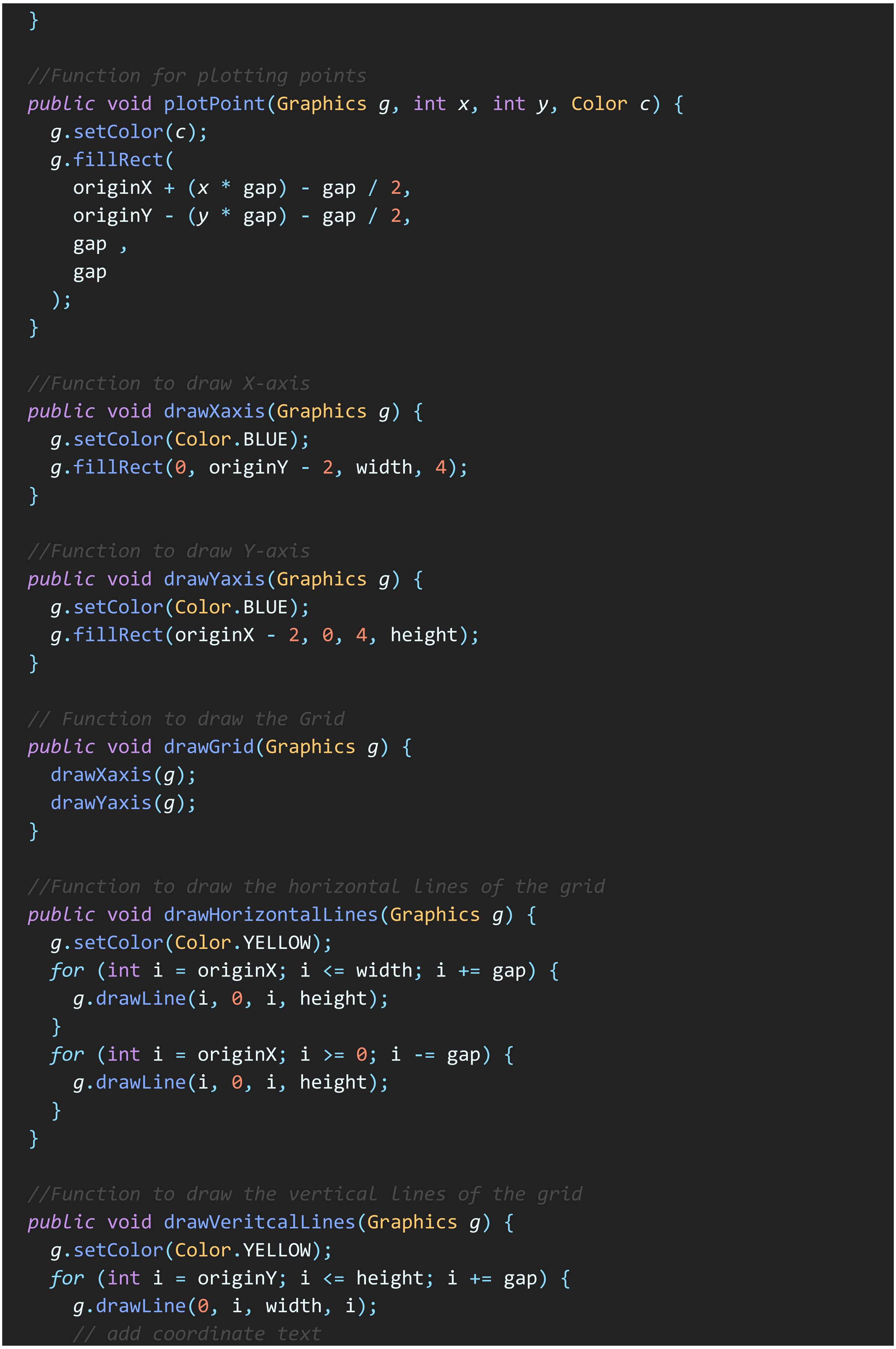
3. Output-



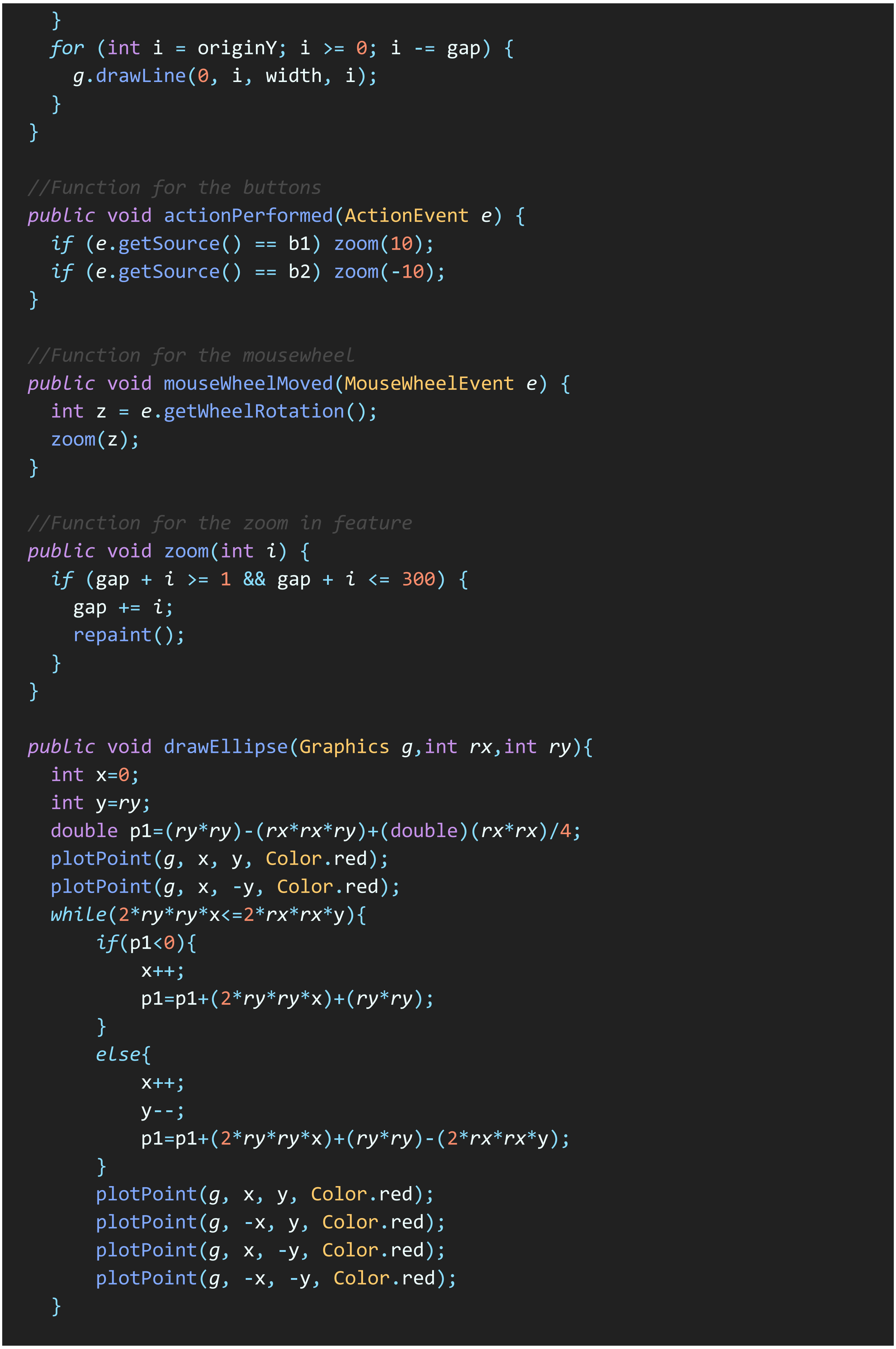
➢ PROBLEM-2:

1. draw.java-

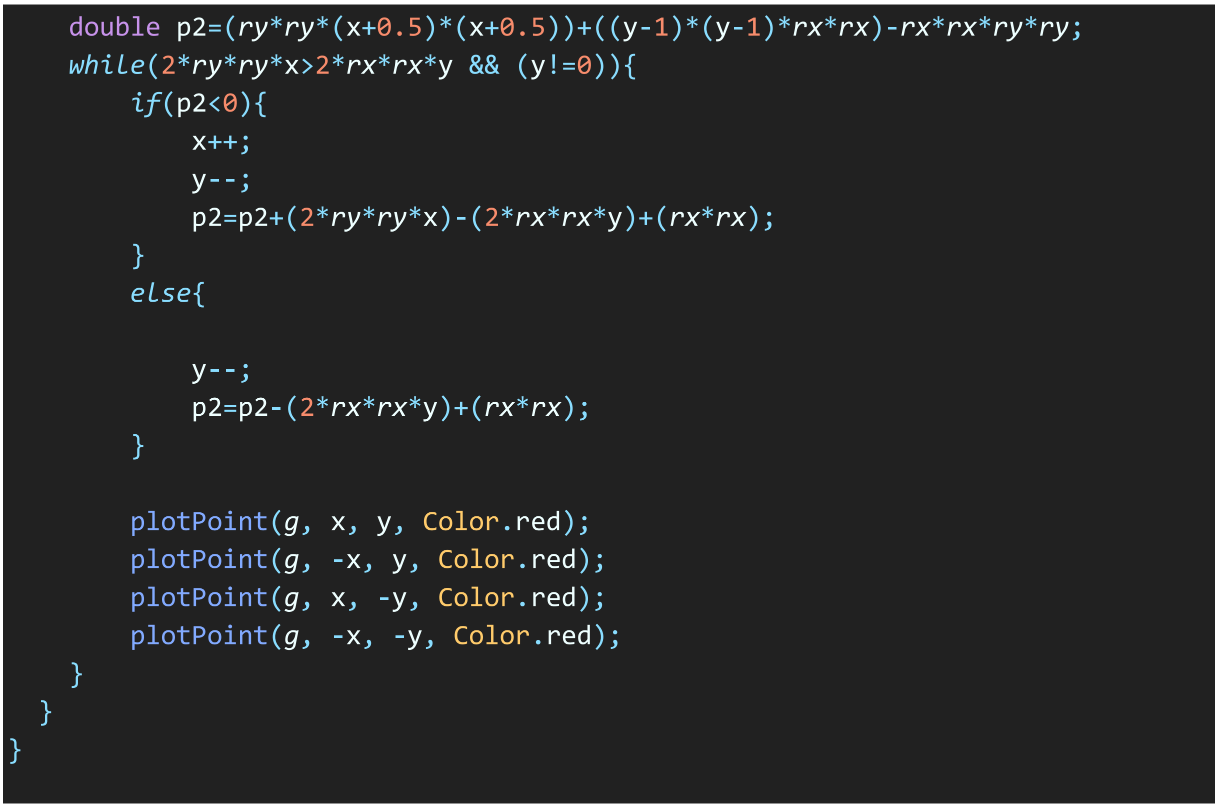
```
import java.applet.*;

import java.awt.*;
import java.awt.event.*;

public class draw extends Applet implements ActionListener, MouseWheelListener
{

  int originX, originY;
  int height, width;
  int gap = 20;
  Button b1 = new Button(" + ");
  Button b2 = new Button(" - ");

  public void init() {
    setBackground(Color.black);
    b1.setBackground(new Color(31, 70, 144));
    b2.setBackground(new Color(255, 229, 180));
    add(b1);
    add(b2);
    addMouseWheelListener(this);
    b1.addActionListener(this);
    b2.addActionListener(this);
  }

  public void paint(Graphics g) {
    g.setColor(Color.BLACK);
    height = getHeight();
    width = getWidth();
    originX = (getX() + width) / 2;
    originY = (getY() + height) / 2;
    drawGrid(g);
    // drawCircle(g, 50, 0, 0);
    drawEllipse(g, 20, 10);
  }

  //Function to draw origin
  public void drawOriginCircle(Graphics g) {
    g.setColor(Color.RED);
    g.fillOval(originX - 5, originY - 5, 10, 10);
```



```
}

//Function for plotting points
public void plotPoint(Graphics g, int x, int y, Color c) {
  g.setColor(c);
  g.fillRect(
    originX + (x * gap) - gap / 2,
    originY - (y * gap) - gap / 2,
    gap ,
    gap
  );
}

//Function to draw X-axis
public void drawXaxis(Graphics g) {
  g.setColor(Color.BLUE);
  g.fillRect(0, originY - 2, width, 4);
}

//Function to draw Y-axis
public void drawYaxis(Graphics g) {
  g.setColor(Color.BLUE);
  g.fillRect(originX - 2, 0, 4, height);
}

// Function to draw the Grid
public void drawGrid(Graphics g) {
  drawXaxis(g);
  drawYaxis(g);
}

//Function to draw the horizontal lines of the grid
public void drawHorizontalLines(Graphics g) {
  g.setColor(Color.YELLOW);
  for (int i = originX; i <= width; i += gap) {
    g.drawLine(i, 0, i, height);
  }
  for (int i = originX; i >= 0; i -= gap) {
    g.drawLine(i, 0, i, height);
  }
}

//Function to draw the vertical lines of the grid
public void drawVeritcalLines(Graphics g) {
  g.setColor(Color.YELLOW);
  for (int i = originY; i <= height; i += gap) {
    g.drawLine(0, i, width, i);
    // add coordinate text
```

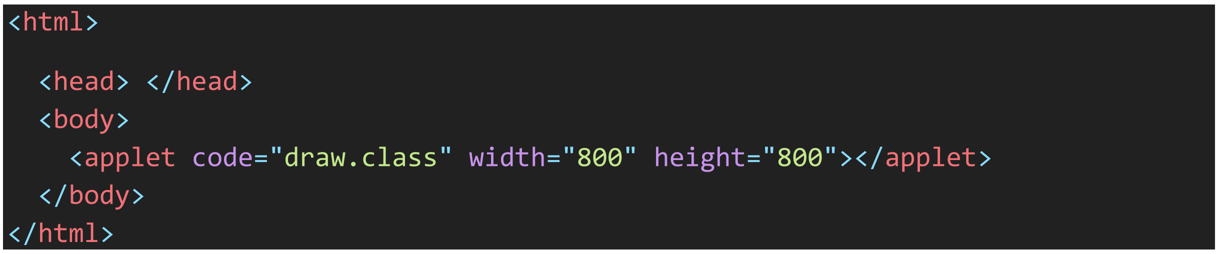


```
  }
  for (int i = originY; i >= 0; i -= gap) {
    g.drawLine(0, i, width, i);
  }
}

//Function for the buttons
public void actionPerformed(ActionEvent e) {
  if (e.getSource() == b1) zoom(10);
  if (e.getSource() == b2) zoom(-10);
}

//Function for the mousewheel
public void mouseWheelMoved(MouseWheelEvent e) {
  int z = e.getWheelRotation();
  zoom(z);
}

//Function for the zoom in feature
public void zoom(int i) {
  if (gap + i >= 1 && gap + i <= 300) {
    gap += i;
    repaint();
  }
}

public void drawEllipse(Graphics g,int rx,int ry){
  int x=0;
  int y=ry;
  double p1=(ry*ry)-(rx*rx*ry)+(double)(rx*rx)/4;
  plotPoint(g, x, y, Color.red);
  plotPoint(g, x, -y, Color.red);
  while(2*ry*ry*x<=2*rx*rx*y){
      if(p1<0){
          x++;
          p1=p1+(2*ry*ry*x)+(ry*ry);
      }
      else{
          x++;
          y--;
          p1=p1+(2*ry*ry*x)+(ry*ry)-(2*rx*rx*y);
      }
      plotPoint(g, x, y, Color.red);
      plotPoint(g, -x, y, Color.red);
      plotPoint(g, x, -y, Color.red);
      plotPoint(g, -x, -y, Color.red);
  }
```
]



```
    double p2=(ry*ry*(x+0.5)*(x+0.5))+((y-1)*(y-1)*rx*rx)-rx*rx*ry*ry;
    while(2*ry*ry*x>2*rx*rx*y && (y!=0)){
        if(p2<0){
            x++;
            y--;
            p2=p2+(2*ry*ry*x)-(2*rx*rx*y)+(rx*rx);
        }
        else{

            y--;
            p2=p2-(2*rx*rx*y)+(rx*rx);
        }

        plotPoint(g, x, y, Color.red);
        plotPoint(g, -x, y, Color.red);
        plotPoint(g, x, -y, Color.red);
        plotPoint(g, -x, -y, Color.red);
    }
  }
}
```

2. index.html-



```
<html>

  <head> </head>
  <body>
    <applet code="draw.class" width="800" height="800"></applet>
  </body>
</html>
```

3. Output-