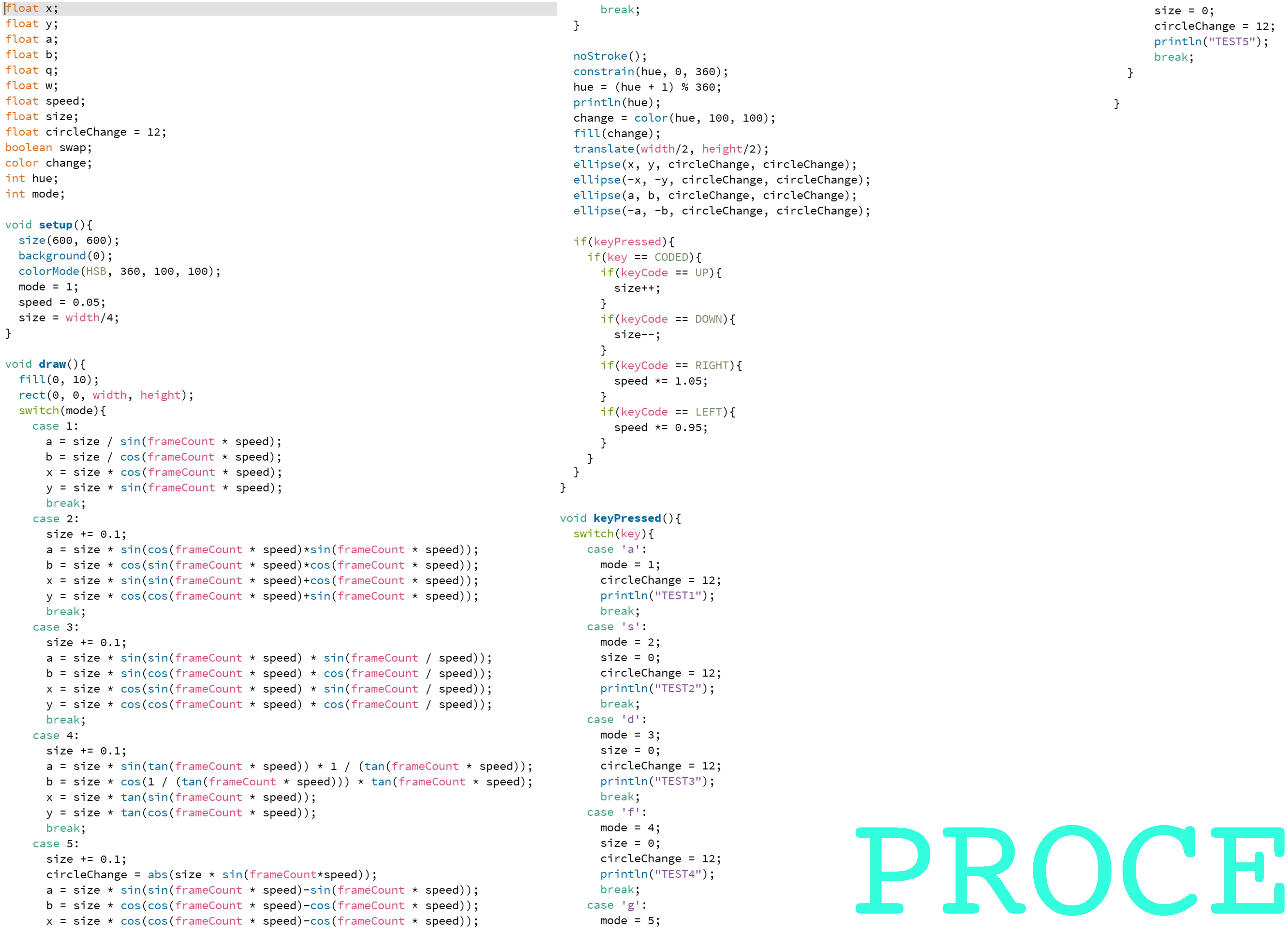
CODING

midterm

CONCEPT

sin,cos,frameCount

- came from dorkshop -



```
float x;
float y;
float a;
float b;
float q;
float w;
float speed;
float size;
float circleChange = 12;
boolean swap;
color change;
int hue;
int mode;

void setup(){
  size(600, 600);
  background(0);
  colorMode(HSB, 360, 100, 100);
  mode = 1;
  speed = 0.05;
  size = width/4;
}

void draw(){
  fill(0, 10);
  rect(0, 0, width, height);
  switch(mode){
    case 1:
      a = size / sin(frameCount * speed);
      b = size / cos(frameCount * speed);
      x = size * cos(frameCount * speed);
      y = size * sin(frameCount * speed);
      break;
    case 2:
      size += 0.1;
      a = size * sin(cos(frameCount * speed)*sin(frameCount * speed));
      b = size * cos(sin(frameCount * speed)*cos(frameCount * speed));
      x = size * sin(sin(frameCount * speed)+cos(frameCount * speed));
      y = size * cos(cos(frameCount * speed)+sin(frameCount * speed));
      break;
    case 3:
      size += 0.1;
      a = size * sin(sin(frameCount * speed) * sin(frameCount / speed));
      b = size * sin(cos(frameCount * speed) * cos(frameCount / speed));
      x = size * cos(sin(frameCount * speed) * sin(frameCount / speed));
      y = size * cos(cos(frameCount * speed) * cos(frameCount / speed));
      break;
    case 4:
      size += 0.1;
      a = size * sin(tan(frameCount * speed)) * 1 / (tan(frameCount * speed));
      b = size * cos(1 / (tan(frameCount * speed))) * tan(frameCount * speed);
      x = size * tan(sin(frameCount * speed));
      y = size * tan(cos(frameCount * speed));
      break;
    case 5:
      size += 0.1;
      circleChange = abs(size * sin(frameCount*speed));
      a = size * sin(sin(frameCount * speed)-sin(frameCount * speed));
      b = size * cos(cos(frameCount * speed)-cos(frameCount * speed));
      x = size * cos(cos(frameCount * speed)-cos(frameCount * speed));
      break;
  }

  noStroke();
  constrain(hue, 0, 360);
  hue = (hue + 1) % 360;
  println(hue);
  change = color(hue, 100, 100);
  fill(change);
  translate(width/2, height/2);
  ellipse(x, y, circleChange, circleChange);
  ellipse(-x, -y, circleChange, circleChange);
  ellipse(a, b, circleChange, circleChange);
  ellipse(-a, -b, circleChange, circleChange);

  if(keyPressed){
    if(key == CODED){
      if(keyCode == UP){
        size++;
      }
      if(keyCode == DOWN){
        size--;
      }
      if(keyCode == RIGHT){
        speed *= 1.05;
      }
      if(keyCode == LEFT){
        speed *= 0.95;
      }
    }
  }
}

void keyPressed(){
  switch(key){
    case 'a':
      mode = 1;
      circleChange = 12;
      println("TEST1");
      break;
    case 's':
      mode = 2;
      size = 0;
      circleChange = 12;
      println("TEST2");
      break;
    case 'd':
      mode = 3;
      size = 0;
      circleChange = 12;
      println("TEST3");
      break;
    case 'f':
      mode = 4;
      size = 0;
      circleChange = 12;
      println("TEST4");
      break;
    case 'g':
      mode = 5;
      size = 0;
      circleChange = 12;
      println("TEST5");
      break;
  }

}
```

show

PROCESSING