```
// setup servo
int servoPin = 8;
int PEN_DOWN = 20; // angle of servo when pen is down
int PEN_UP = 80;   // angle of servo when pen is up
Servo penServo;

float wheel_dia=63; //    # mm (increase = spiral out)
float wheel_base=109; //    # mm (increase = spiral in, ccw)
int steps_rev=128; //       # 512 for 64x gearbox, 128 for 16x gearbox
int delay_time=6; //        # time between steps in ms

// Stepper sequence org->pink->blue->yel
int L_stepper_pins[] = {12, 10, 9, 11};
int R_stepper_pins[] = {4, 6, 7, 5};

int fwd_mask[][4] =  {{1, 0, 1, 0},
                      {0, 1, 1, 0},
                      {0, 1, 0, 1},
                      {1, 0, 0, 1}};

int rev_mask[][4] =  {{1, 0, 0, 1},
                      {0, 1, 0, 1},
                      {0, 1, 1, 0},
                      {1, 0, 1, 0}};


void setup() {
  randomSeed(analogRead(1));
  Serial.begin(9600);
```

```
  for(int pin=0; pin<4; pin++){
    pinMode(L_stepper_pins[pin], OUTPUT);
    digitalWrite(L_stepper_pins[pin], LOW);
    pinMode(R_stepper_pins[pin], OUTPUT);
    digitalWrite(R_stepper_pins[pin], LOW);
  }
  penServo.attach(servoPin);
  Serial.println("setup");

  penup();

  delay(1000);
}


void loop(){ // draw a calibration box 4 times
ethan ();



}


// ----- HELPER FUNCTIONS -----------
int step(float distance){
  int steps = distance * steps_rev / (wheel_dia * 3.1412); //24.61
  /*
  Serial.print(distance);
  Serial.print(" ");
  Serial.print(steps_rev);
```

```
  Serial.print(" ");
  Serial.print(wheel_dia);
  Serial.print(" ");
  Serial.println(steps);
  delay(1000);*/
  return steps;
}




void ethan()
{
 pendown();
 forward(100);
 backward(100);
 left(360);
 forward(100);
 right(360);
 forward(100);
 backward(100);
 left(360);
 forward(100);
 right(360);
 forward(100);
 penup();
 forward(100);
```

```
pendown();
forward(100);
backward(50);
right(360);
forward(175);
left(360);
penup();
forward(100);
pendown();
left(360);
forward(175);
backward(87.5);
right(360);
forward(100);
left(360);
forward(87.5);
backward(175);
right(360);
penup();
forward(100);
pendown();
left(250);
forward(175);
right(450);
forward(175);
backward(87.5);
right(550);
forward(100);
backward(100);
```

```
 penup();
 backward(200);
 left(360);
 pendown();
 forward(87.5);
 backward(175);
 left(100);
 forward(175);
 left(650);
 forward(175);
 penup();



  done();      // releases stepper motor
  while(1);    // wait for reset



}


void octagon ()
{
 pendown();
  for(int x=0; x<17; x++){
    forward(100);
    left(90);
  }
```

```
  penup();
  done();      // releases stepper motor
  while(1);    // wait for reset
}

void forward(float distance){
  int steps = step(distance);
  Serial.println(steps);
  for(int step=0; step<steps; step++){
    for(int mask=0; mask<4; mask++){
      for(int pin=0; pin<4; pin++){
        digitalWrite(L_stepper_pins[pin], rev_mask[mask][pin]);
        digitalWrite(R_stepper_pins[pin], fwd_mask[mask][pin]);
      }
      delay(delay_time);
    }
  }
}



void backward(float distance){
  int steps = step(distance);
  for(int step=0; step<steps; step++){
    for(int mask=0; mask<4; mask++){
      for(int pin=0; pin<4; pin++){
        digitalWrite(L_stepper_pins[pin], fwd_mask[mask][pin]);
        digitalWrite(R_stepper_pins[pin], rev_mask[mask][pin]);
      }
      delay(delay_time);
```

```
    }
  }
}


void right(float degrees){
  float rotation = degrees / 360.0;
  float distance = wheel_base * 3.1412 * rotation;
  int steps = step(distance);
  for(int step=0; step<steps; step++){
    for(int mask=0; mask<4; mask++){
      for(int pin=0; pin<4; pin++){
        digitalWrite(R_stepper_pins[pin], rev_mask[mask][pin]);
        digitalWrite(L_stepper_pins[pin], rev_mask[mask][pin]);
      }
      delay(delay_time);
    }
  }
}


void left(float degrees){
  float rotation = degrees / 360.0;
  float distance = wheel_base * 3.1412 * rotation;
  int steps = step(distance);
  for(int step=0; step<steps; step++){
    for(int mask=0; mask<4; mask++){
      for(int pin=0; pin<4; pin++){
        digitalWrite(R_stepper_pins[pin], fwd_mask[mask][pin]);
```

```
      digitalWrite(L_stepper_pins[pin], fwd_mask[mask][pin]);
    }
    delay(delay_time);
  }
 }
}


void done(){ // unlock stepper to save battery
 for(int mask=0; mask<4; mask++){
  for(int pin=0; pin<4; pin++){
    digitalWrite(R_stepper_pins[pin], LOW);
    digitalWrite(L_stepper_pins[pin], LOW);
  }
  delay(delay_time);
 }
}


void penup(){
 delay(250);
 Serial.println("PEN_UP()");
 penServo.write(PEN_UP);
 delay(250);
}


void pendown(){
 delay(250);
```

```
  Serial.println("PEN_DOWN()");
  penServo.write(PEN_DOWN);
  delay(250);
}
```