Transformations/Recursion Lab

1) Rewrite the following function:

```
void drawSpokes(int centerx, int centery, int radius) {
  int steps = 30;
  float angle = 2*PI/steps;

  stroke(random(255), random(255), random(255), 127);
  for (int i=0; i<steps; i++) {
    float x = cos(angle*i)*radius + centerx;
    float y = sin(angle*i)*radius + centery;

    line(centerx, centery, x, y);
  }
}
```

a) Define a corresponding Spokes class. The class contain the following data fields, x, y, radius, steps, angle and c (for color), of which x, y, radius and steps should be set via the constructor

```
class Spoke {
  float x;
  float y;
  int steps;
  float radius;
  color c;

  Spoke(float x, float y, float radius, int steps) {
    this.x = x;
    this.y = y;
    this.radius = radius;
    this.steps = steps;
    c = color(random(255),random(255),random(255),127);
  }
}
```

b) Write its display() method with transformations instead of coordinate calculation with trigs.

```
  void display() {
    float angle = 2*PI/steps;
    stroke(c);
    pushMatrix();
    translate(x, y);
    for (int i=0; i<steps; i++) {
      pushMatrix();
      rotate(angle*i);
      line(0, 0, radius, 0);
      popMatrix();
    }
    popMatrix();
  }
```

Add a setup() to complete a program which will create 50 spokes at random locations in the sketch window, with size varying from 20 to 50 and number of spokes varying from 10 to 30. Store them in an array and draw.

```
Spoke[] spokes = new Spoke[50];
void setup() {
  size(500, 500);
  background(255);

  for (int i=0; i<50; i++) {
    spokes[i] = new Spoke(random(width), random(height),
                random(20, 50), int(random(10, 30)));
  }

  for (int i=0; i<50; i++) {
    spokes[i].display();
  }
}
```

2) Trace the following code. (Draw a table with the appropriate variables and fields.)

```
int mystery (int x) {
  if (x/10 < 10) {
    return x;
  }
  else {
    return x%10 + mystery(x/10);
  }
}

println(mystery(12345));
```

3) Trace the following code. (Draw a table with the appropriate variables and fields.)

```
int mystery (int num) {
  if (num==1) {
    return num;
  }
  else {
    return (num/2) * mystery(num-1);
  }
}

println(mystery(4));
```

4) Write a recursive function `int power(int n, int i)` that takes two integer parameters `n` and `i`. The function should return the i^{th} power of `n`.

```
int power(int n, int i) {
  if (i == 0) {
    return 1;
  }
  else {
    return n*power(n, i-1);
  }
}
```

5) Write a recursive function `void rev(int[] a, int start, int end)` that takes an integer array `a`, a starting index `start` and an ending index `end` and reverses the portion of the array between the indices. For example, if array `a` is declared and initialized as follows:

```
int[] a = {1, 2, 3, 4, 5};
```

then the call `rev(a, 0, a.length-1)` will result in the contents of `a` completely reversed.

```
void rev(int[] a, int s, int e) {
  int tmp;

  if (s >= e) {
    return;
  }
  else {
    tmp = a[s];
    a[s] = a[e];
    a[e] = tmp;
    rev(a, s+1, e-1);
  }
}
```