1a.

```
public static int hailstoneLength (int n) {
        int length = 0;
        int length = 1;
        while (n != 1) {
            if (
        for (int length = 1; n != 1; length ++) {
                if (n % 2 == 0) {
                        n = n/2;
                }else if (n % 2 == 1) {
                        n = (3*n) + 1;
                }
        }
        return length;
}
```

1b.

```
public static boolean isLongSeq(int n) {
        int length = Hailstone.hailstoneLength(n);
        if (length > n) {
                return true;
        }
        return false;
}
```

1c.

```
public static double propLong (int n) { int total=0; int count   int total=0; int count=0;
        for (int i=1; i<=n; i++) {
                if (Hailstone.isLongSeq) {
                        total++;
                }
        }
        return (double) total)/((double) n);
```

```
public class GameSpinner {
    private int numSectors;
    private int runLength;
    private int prev;
        public GameSpinner (int numSectors) {
            this.numSectors = numSectors;
            this.runLength = 0;
        }

        public int spin() {
            int val = (int) (Math.random()*numSectors);
            if (val == prev) {
                runLength++;
            }
            else {
                runLength = 1;
            }
            prev = val;
            return val;
        }

        public int currentRun() {
            return this.runLength;
        }
```