Program Structures and Algorithms
Assignment-1

Summer-2022

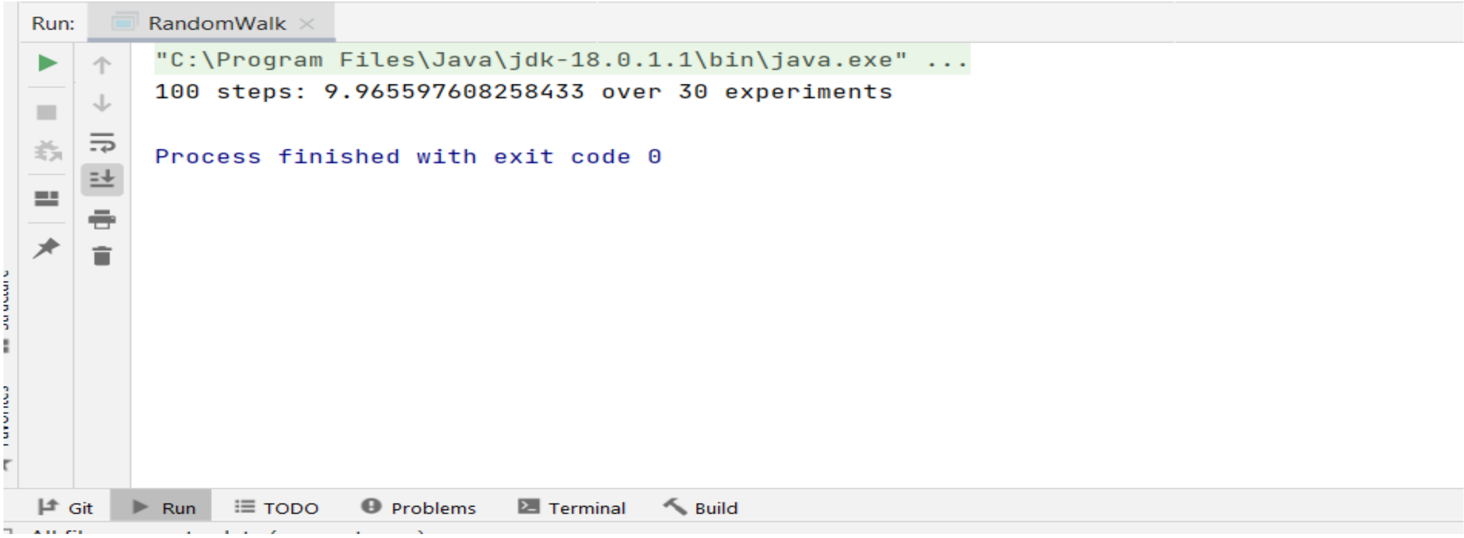
Dimpleben Kanjibhai Patel – 002965372

Problem: Random Walk

Task: To deduce the relationship between the distance of man from the lamp and the number of steps taken by the man from the original position.

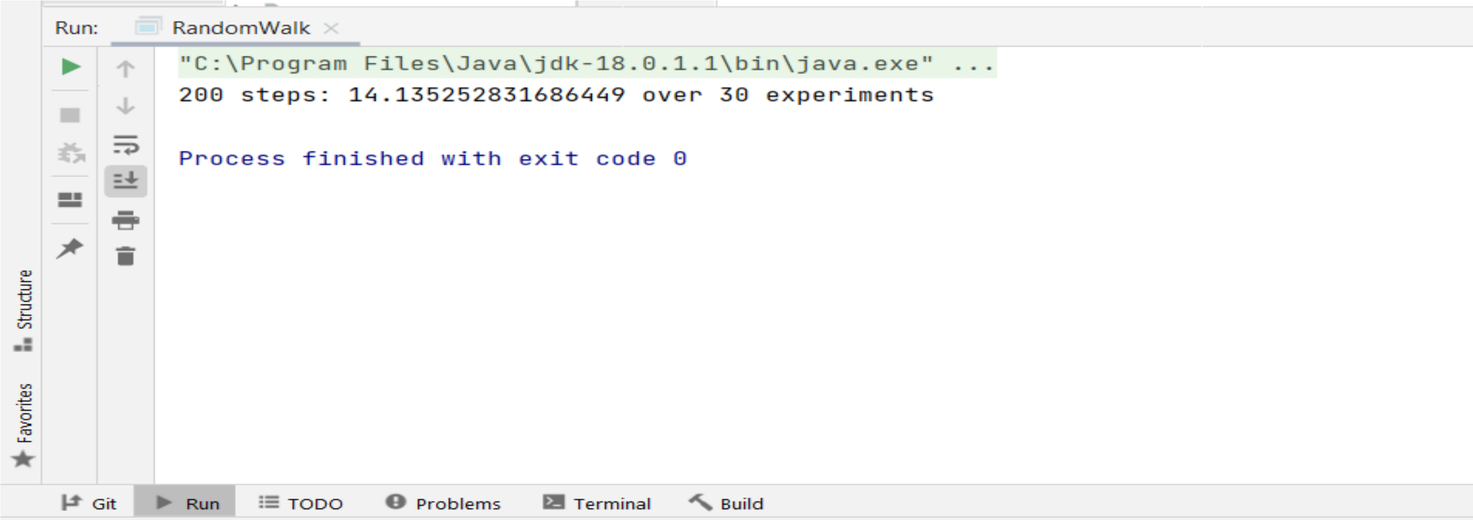
Output:

100 Steps with 30 experiments



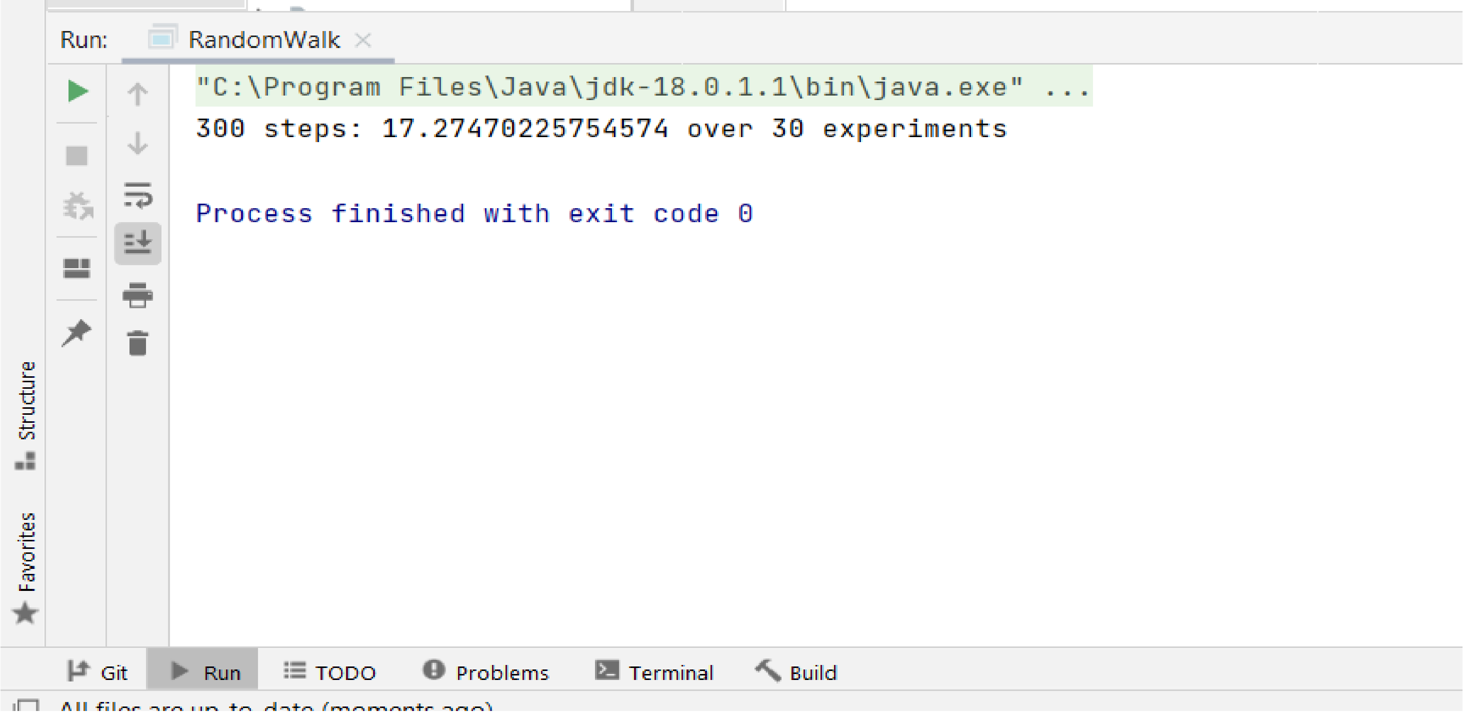
```
"C:\Program Files\Java\jdk-18.0.1.1\bin\java.exe" ...
100 steps: 9.965597608258433 over 30 experiments

Process finished with exit code 0
```

200 Steps with 30 experiments



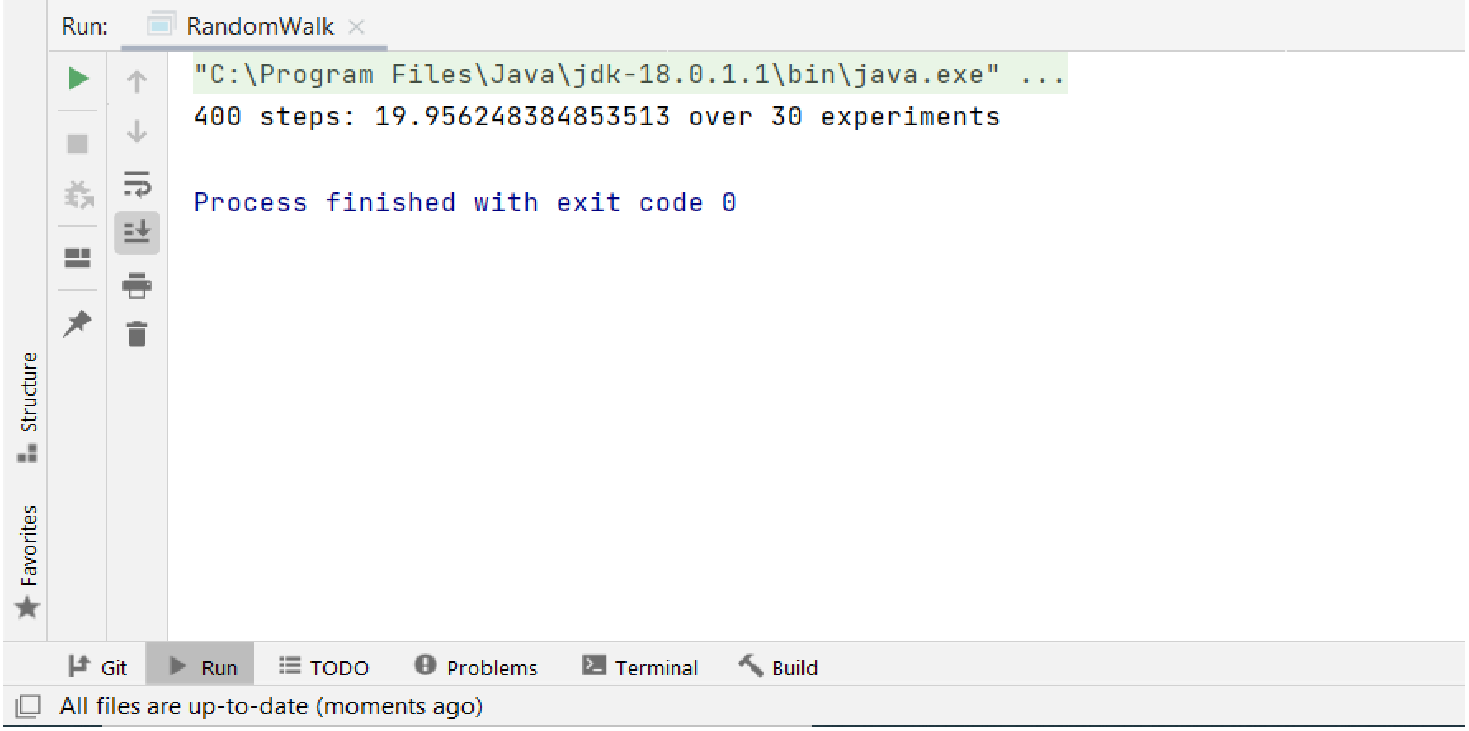
```
"C:\Program Files\Java\jdk-18.0.1.1\bin\java.exe" ...
200 steps: 14.135252831686449 over 30 experiments

Process finished with exit code 0
```

300 Steps with 30 experiments



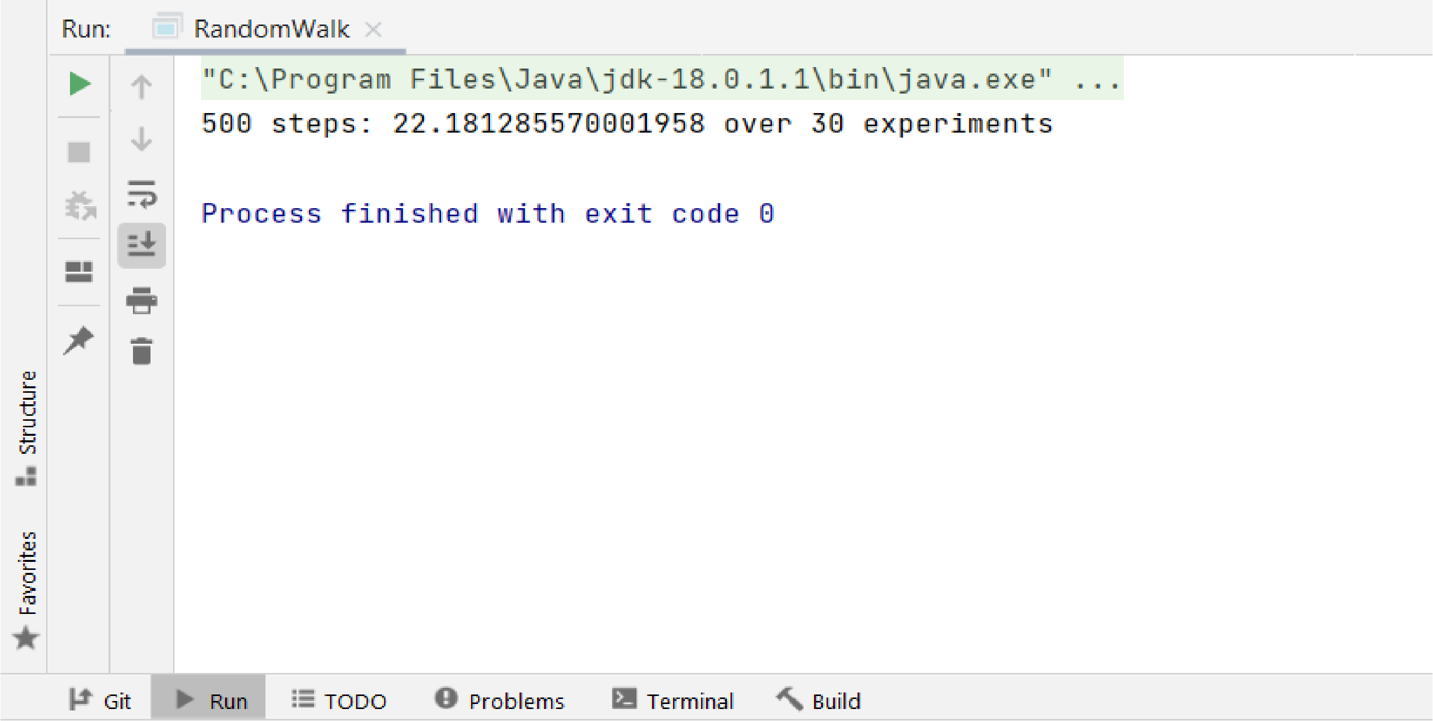
```
"C:\Program Files\Java\jdk-18.0.1.1\bin\java.exe" ...
300 steps: 17.27470225754574 over 30 experiments

Process finished with exit code 0
```

400 Steps with 30 experiments



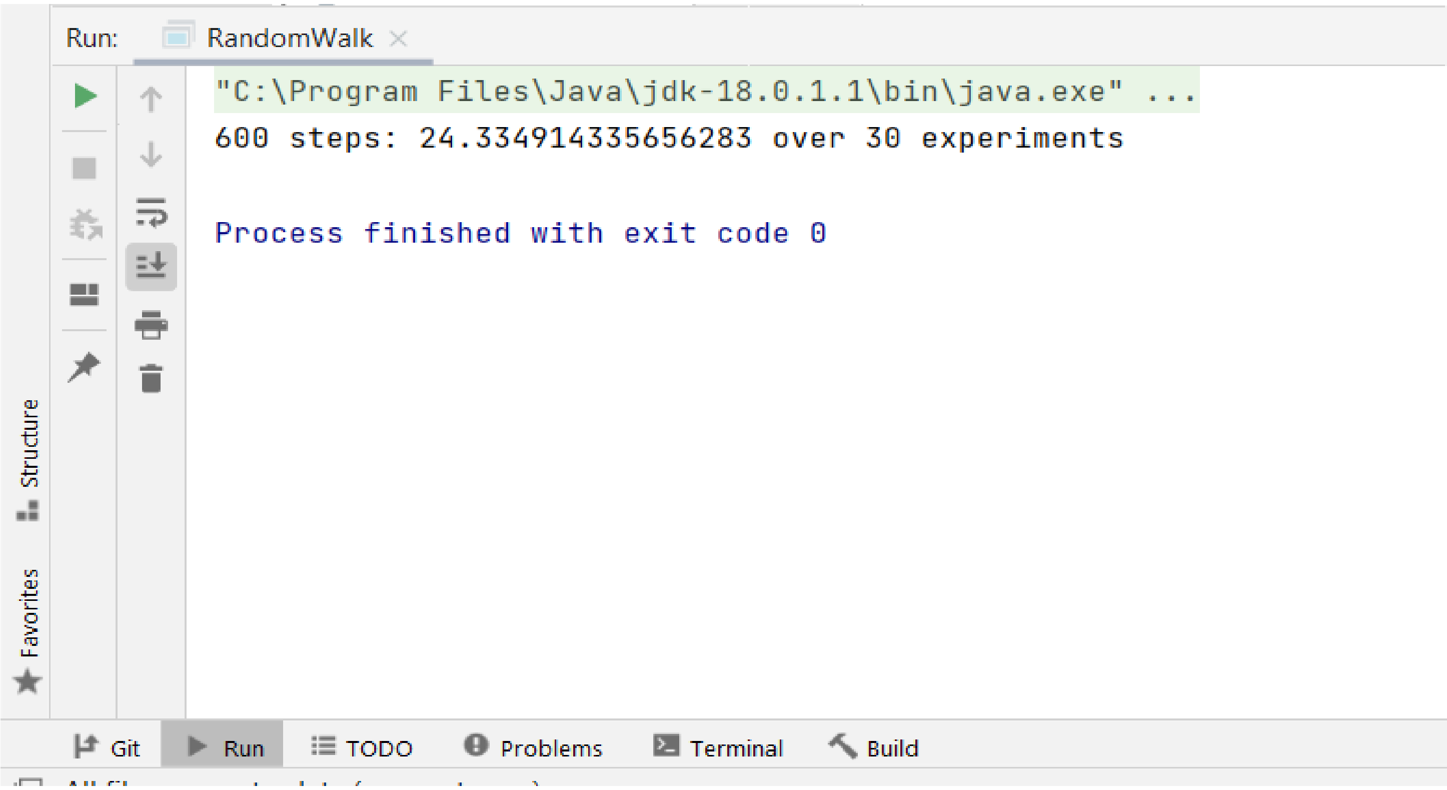
```
"C:\Program Files\Java\jdk-18.0.1.1\bin\java.exe" ...
400 steps: 19.956248384853513 over 30 experiments

Process finished with exit code 0
```

500 Steps with 30 experiments



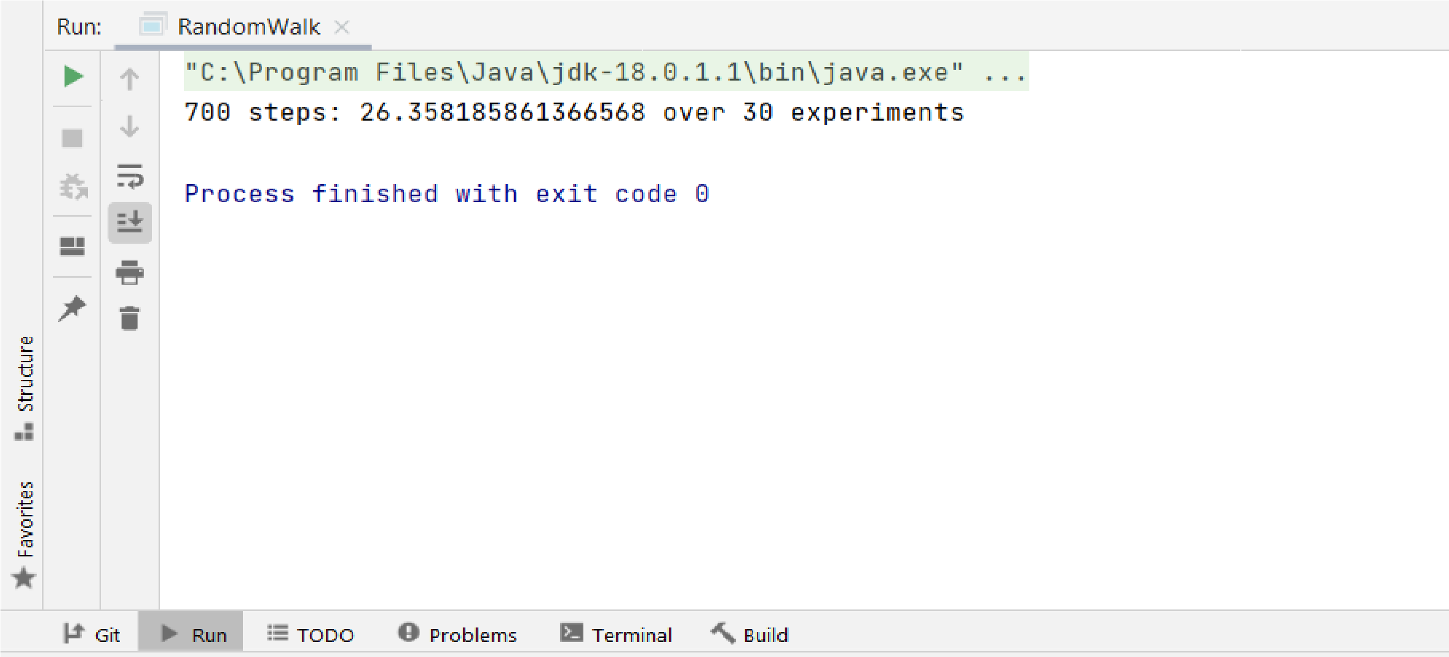
```
"C:\Program Files\Java\jdk-18.0.1.1\bin\java.exe" ...
500 steps: 22.181285570001958 over 30 experiments

Process finished with exit code 0
```

600 Steps with 30 experiments



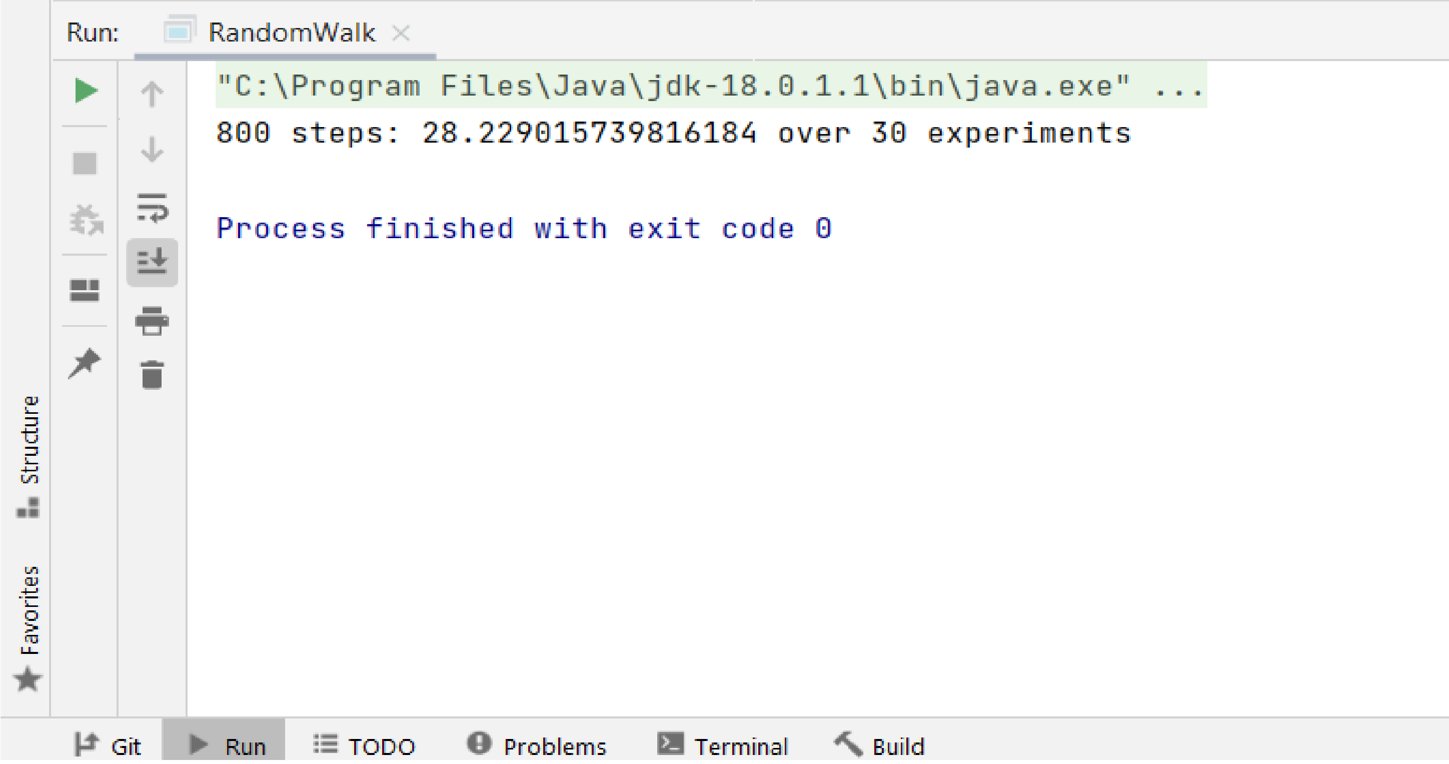
```
"C:\Program Files\Java\jdk-18.0.1.1\bin\java.exe" ...
600 steps: 24.334914335656283 over 30 experiments

Process finished with exit code 0
```

700 Steps with 30 experiments



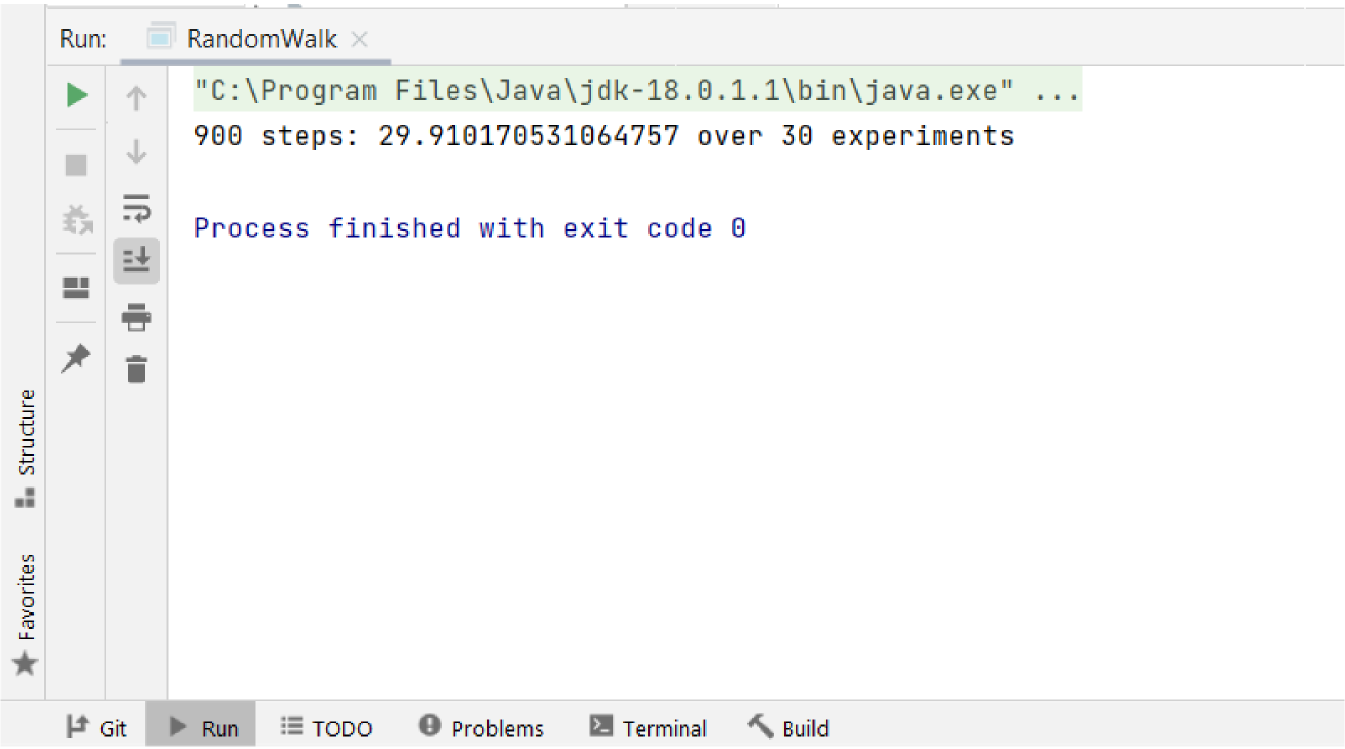
```
"C:\Program Files\Java\jdk-18.0.1.1\bin\java.exe" ...
700 steps: 26.358185861366568 over 30 experiments

Process finished with exit code 0
```

800 Steps with 30 experiments



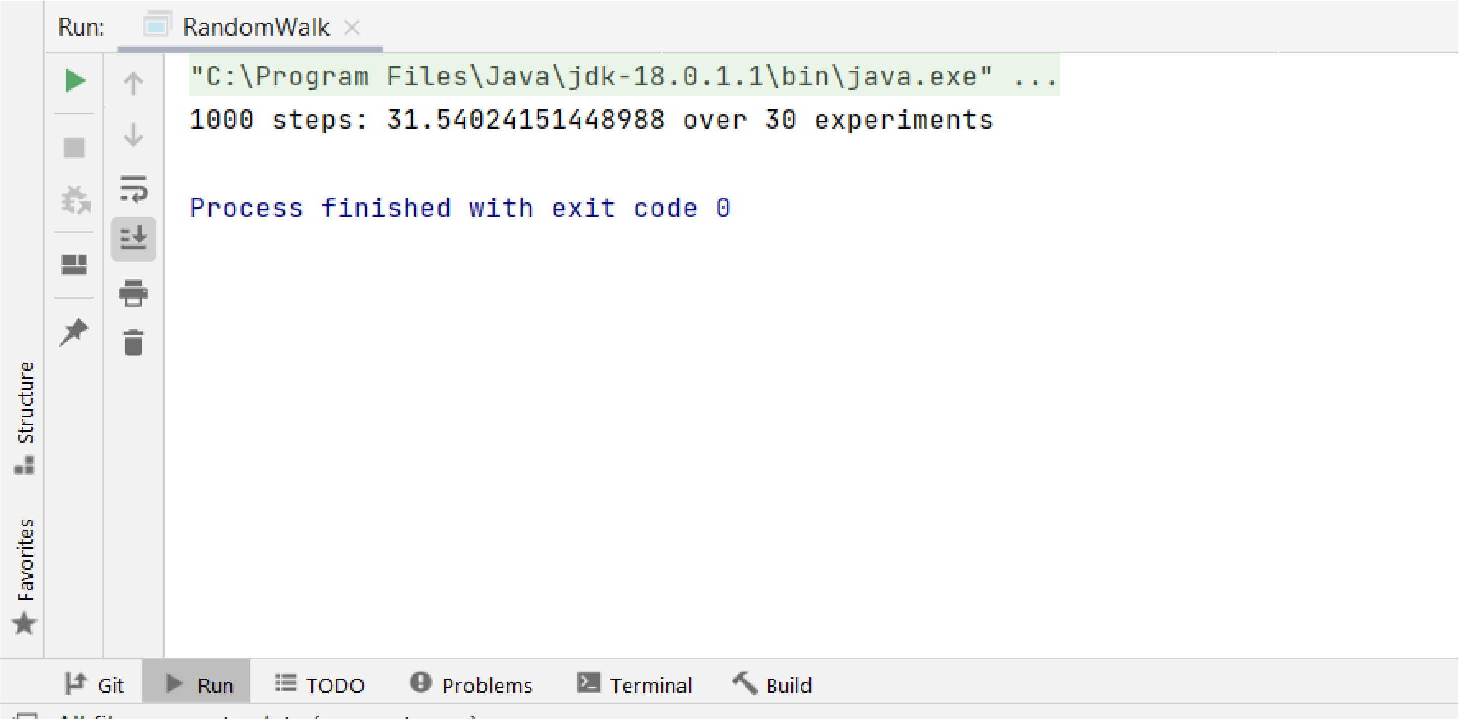
```
"C:\Program Files\Java\jdk-18.0.1.1\bin\java.exe" ...
800 steps: 28.229015739816184 over 30 experiments

Process finished with exit code 0
```

900 Steps with 30 experiments



```
"C:\Program Files\Java\jdk-18.0.1.1\bin\java.exe" ...
900 steps: 29.910170531064757 over 30 experiments

Process finished with exit code 0
```

1000 Steps with 30 experiments



```
"C:\Program Files\Java\jdk-18.0.1.1\bin\java.exe" ...
1000 steps: 31.54024151448988 over 30 experiments

Process finished with exit code 0
```

Conclusion:

The simulation is performed by keeping the number of experiments constant i.e., equal to 30 and varying the number of steps from 100 to 1000. The result of the simulation is shown in the screenshot. It is observed that the distance(D) is approximately equal to the square root of the steps taken(m).

$$\mathbf{D} = \sqrt{m}$$

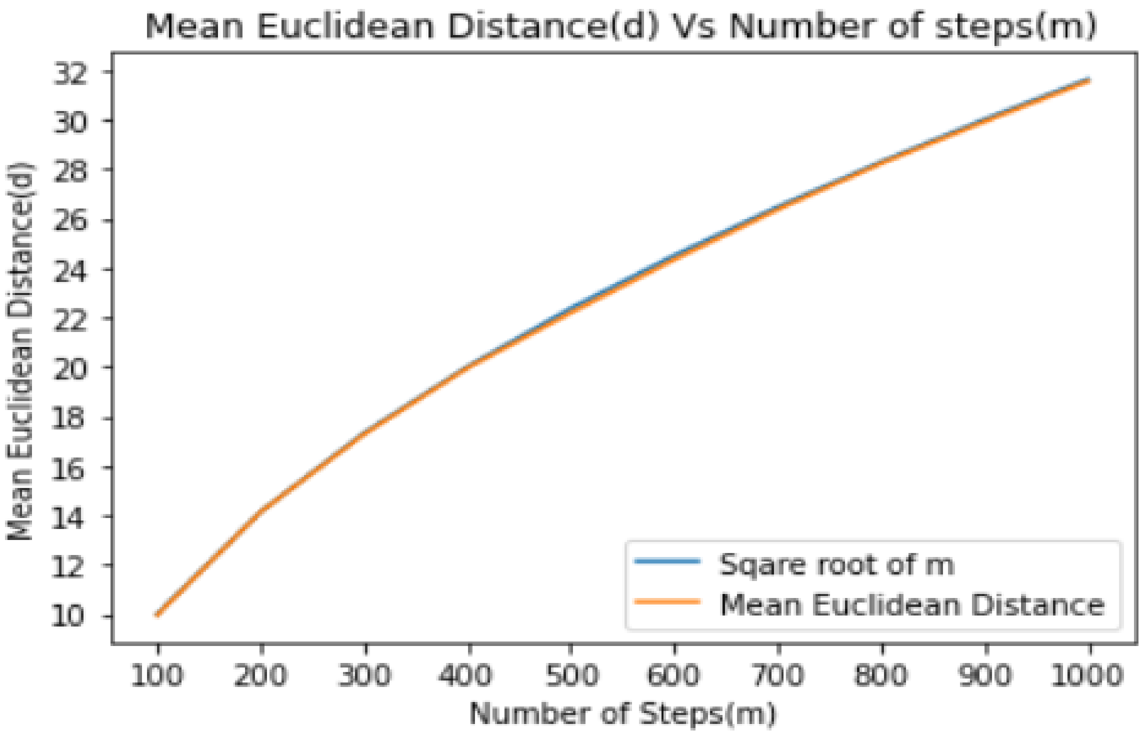
D => Mean Euclidean Distance of man from the lamp post.
m => number of steps taken by man

Evidence that supports the conclusion:

Number of Steps(m)	Number of Experiments (n)	Mean Euclidean Distance (D)	Actual Value of Square root of m ($\sqrt{m}$)
100	30	9.96	10
200	30	14.13	14.14
300	30	17.27	17.32
400	30	19.95	20
500	30	22.18	22.36
600	30	24.33	24.49
700	30	26.35	26.45
800	30	28.22	28.28
900	30	29.91	30
1000	30	31.54	31.62

Graphical Representation:

The graphical representation of the above table is as shown below. The x-axis represents the number of steps taken by man and y-axis represent the Mean Euclidean distance of man from original position. It is the comparison graph of actual value of square root of steps(m) and the Mean Euclidean Distance(D) that we calculated. It can be observed that the above relationship is approximately satisfied.

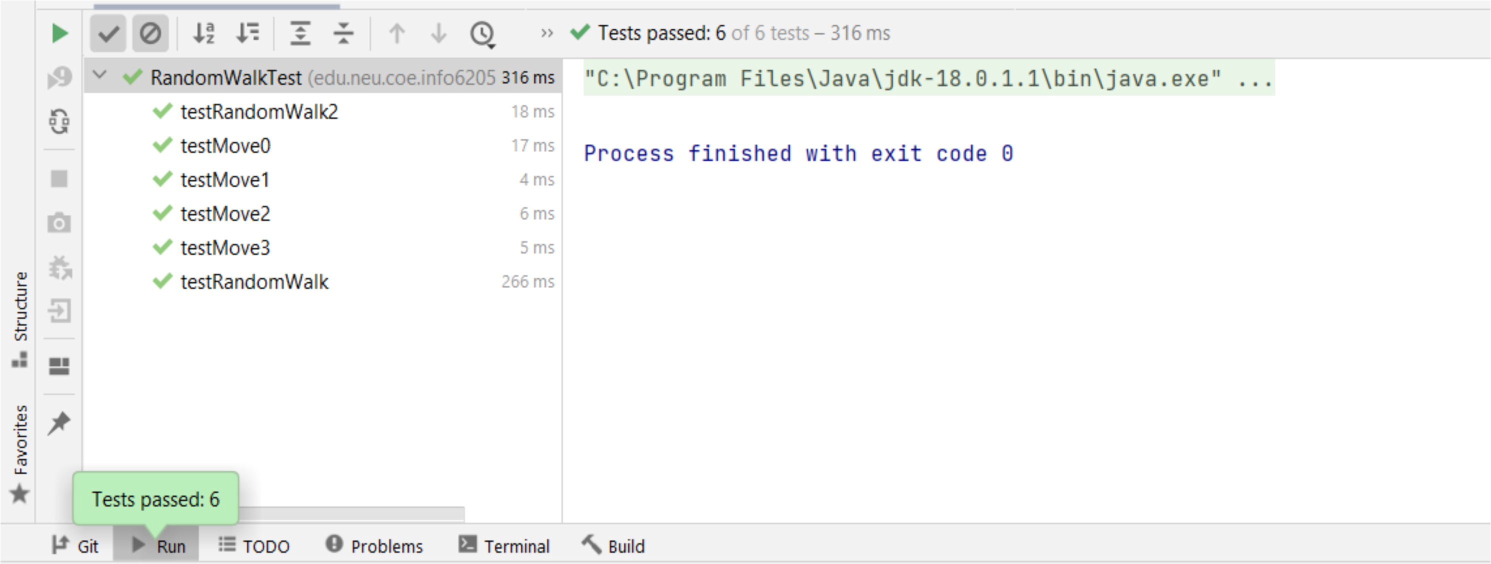


Code:

RandomWalk.java

```
/*
 * Copyright (c) 2017. Phasmid Software
 */

package edu.neu.coe.info6205.randomwalk;

import java.util.Random;

public class RandomWalk {

    private int x = 0;
    private int y = 0;

    private final Random random = new Random();

    /**
     * Private method to move the current position, that's to say the drunkard
moves
     *
     * @param dx the distance he moves in the x direction
     * @param dy the distance he moves in the y direction
     */
    private void move(int dx, int dy) {
        x += dx;
        y += dy;
    }
    /**
     * Perform a random walk of m steps
     *
     * @param m the number of steps the drunkard takes
     */
    private void randomWalk(int m) {
        for(int i=1; i<=m; i++){
            randomMove();
        }
    }

    /**
     * Private method to generate a random move according to the rules of the
situation.
     * That's to say, moves can be (+-1, 0) or (0, +-1).
     */
    private void randomMove() {
        boolean ns = random.nextBoolean();
        int step = random.nextBoolean() ? 1 : -1;
        move(ns ? step : 0, ns ? 0 : step);
    }

    /**
     * Method to compute the distance from the origin (the lamp-post where the
drunkard starts) to his current position.
     *
     * @return the (Euclidean) distance from the origin to the current position.
     */
    public double distance() {
        double distance= Math.pow(x,2)+Math.pow(y,2);
        return Math.sqrt(distance);
    }

    /**
     * Perform multiple random walk experiments, returning the mean distance.
     *
     * @param m the number of steps for each experiment
     * @param n the number of experiments to run
     * @return the mean distance
     */
    public static double randomWalkMulti(int m, int n) {
        double totalDistance = 0;
        for (int i = 0; i < n; i++) {
            RandomWalk walk = new RandomWalk();
            walk.randomWalk(m);
            totalDistance = totalDistance + walk.distance();
        }
        return totalDistance / n;
    }

    public static void main(String[] args) {
        if (args.length == 0)
            throw new RuntimeException("Syntax: RandomWalk steps [experiments]");
        int m = Integer.parseInt(args[0]);
        int n = 30;
        if (args.length > 1) n = Integer.parseInt(args[1]);
        double meanDistance = randomWalkMulti(m, n);
        System.out.println(m + " steps: " + meanDistance + " over " + n + "
experiments");
    }

}
```

Unit test Result:



```
Tests passed: 6 of 6 tests – 316 ms

RandomWalkTest (edu.neu.coe.info6205 316 ms
    testRandomWalk2                   18 ms
    testMove0                         17 ms
    testMove1                          4 ms
    testMove2                          6 ms
    testMove3                          5 ms
    testRandomWalk                   266 ms
```

```
"C:\Program Files\Java\jdk-18.0.1.1\bin\java.exe" ...

Process finished with exit code 0
```