**Funding Circle Coding Challenge**

In this coding exercise, program calculates the next Nth prime number by assuming a number NUM to be the Nth prime. Then the number NUM is checked for factoring by previous N-1 prime numbers. Moreover, if the value of n-1th prime number is less than the value of root of nth prime number i.e. root of NUM the p also checks the factors between n-1th prime number and root of value of nth prime number i.e. Root of NUM if no factors found for the number the nth prime number. i.e. NUM then NUM is the Nth prime number.

If the factor is found, then the next number NUM+1 is checked for the prime number.

**Complexity Analysis:**

Worst Case Complexity:

O(Pn \* Max(n-1, n-1))

Where n is the nth prime number

Pn is value of the nth prime number

Pn-1 is the value of (n-1)th prime number

For Example: if n = 10;

Pn = 29;

Pn-1 =23

**Installation and Usage Instructions**

**Requirements:**

* Java version: 1.8
* Junit libraries:
  + junit-4.13-beta-3
  + hamcrest-core-1.3

**How to run the application on windows:**

* Unzip the folder
* Compile the source code using following command:
  + - javac -d classes Prime.java
* Run the application using following command
  + - java -cp classes Prime

**How to run the Junit test cases:**

* Compile the source code using following command
  + - javac -d classes Prime.java
    - javac -cp "lib/\*;classes/;." TestPrime.java
* Run the application using following command
  + - java -cp "lib/\*;classes/;." org.junit.runner.JUnitCore TestPrime

**Screenshots of the successfully run:**



