README for MITPRIMES Code

Source Code / Project Organization

Name	Features	Comments
P1	My solution for the first computer science problem	Further details are provided below this grid
P1_test	Testing the integer factorization algorithm	Only used for internal testing, please use P1 when actually using real test cases
TestSuite	Testing P1and P2 solution with a brute force algorithm	Does not save any of the generated test data in separate files
P2	My solution for the second computer science problem	Further details are provided below this grid
README	Outlines features of all files in my submission folder + how to pass in input to programs and development computer details	This document
mathproblem-solutions	My solutions to the general math problems	
csproblem-solutions	My written responses to the computer science problem questions	
PrimeFactorization	This interface prime factorizes the resulting divisors returned by the Trial Division and Pollard Rho Algorithm and Primality Tests with Algorithms below	Some limitations of input numbers n are outlined below
ListOfPrimes	Implements an array with the first x primes where x is specified. This list is used in TrialDiv	

NthPrimeUpperBounds	Implements an algorithm that calculates the upper bound of the value of the Nth prime	NOTE: Directly taken from the Java NT Library
PrimeCountUpperBounds	Implements an algorithm that counts the number of primes within an upper bound	NOTE: Directly taken from the Java NT Library
SieveCallback	Callback function to add primes to the PrimeList	
RangedSieveofEratosthenes	Implements an efficient sieve of Eratosthenes to create a list of primes	Used in the ListOfPrimes class
BPSWPrimalityTest	Implements the Baillie-PSW Primality Test	The BPSW Test is deterministic for $N < 2^{64}$ and has no known pseudoprimes
MillerRabin	Implements the MillerRabin Test	Probabilistic Primality Test
LucasTest	Implements the Lucas Test	Probabilistic Primality Test
JacobiSymbol	Implements the algorithm to calculate the Jacobi Symbol	Jacobi Symbol is used in the Lucas Test
SqrtExact	Calculates the exact square root of an integer	NOTE: Directly taken from the Java NT Library
SqrtRange	Uses the Heron method to calculate the range of a square root where the endpoints of the range are integers	Used in Sqrt Exact if the bit method to find the sqrt does not work
PocklingtonPrimality	Implements the Pocklington Primality Test	Deterministic Test

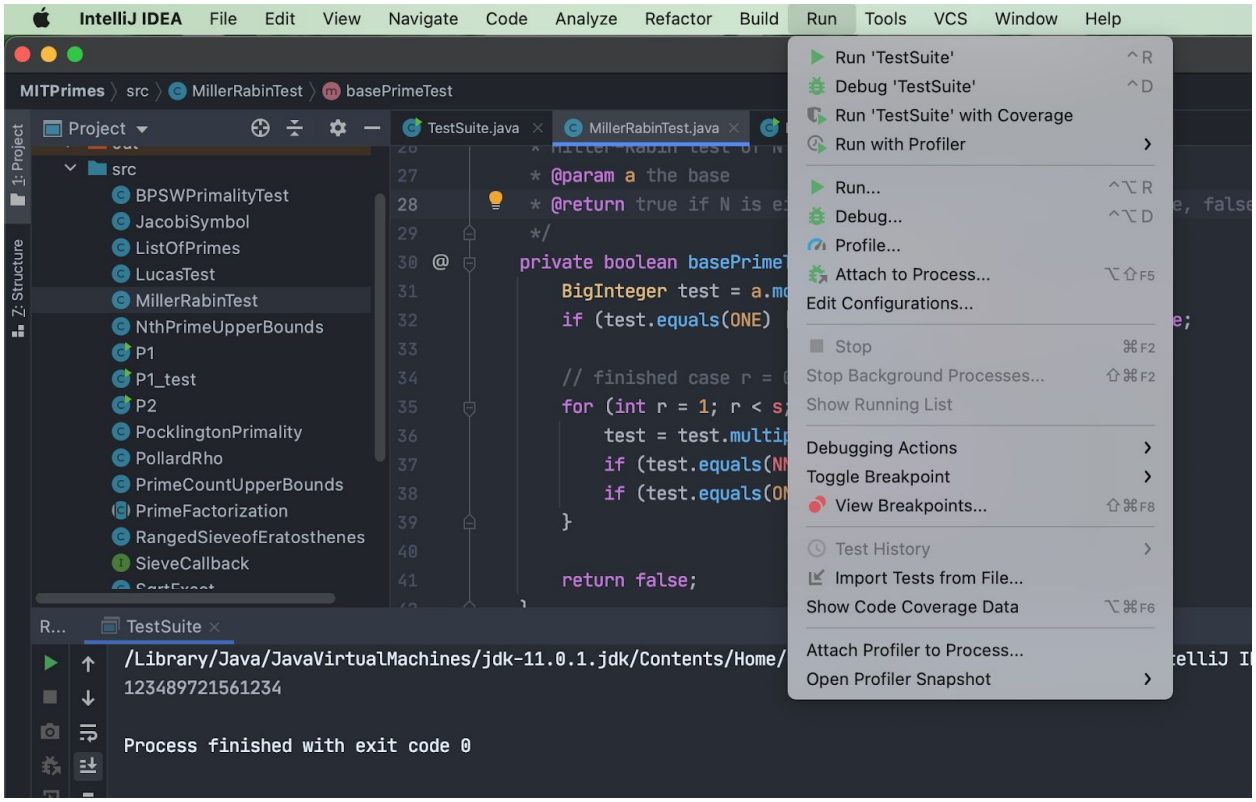
Compiling and Building

Programming Language: Java 11.0.1
Development Framework: IntelliJ IDEA.
Platform: Macbook Pro Big Sur on Intel Core i7

The code uses only standard Java libraries.

Running the code in IntelliJ

To compile code in IntelliJ, simply press the open button and select the MITPrimes project folder. To run each of the problems run P1 or P2 respectively by pressing Run (On the top menu bar) > Run > and then select the file to run.



Input is passed in through the interactive run console. Paste the input and then press enter to run the program.

Program: P1 Input
Paste the string in the first line of the program and press enter
Program: P2 Input
Paste both the string and the number with a space between them in the same line.

To run the test suite, uncomment the method that you are interested in using.

Other Details

- Both programs P1 and P2 do not accept strings with 0 -- throws an intentional error

- Both programs P1 and P2 do not accept strings with characters other than zeroes -- throws an intentional error
- Both programs P1 and P2 do not accept empty strings
- If the product given in P2 does not satisfy requirements -- throws an intentional error

References

- All primality proving algorithms are preexisting popularly used methods:
    - Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms, Third Edition. The MIT Press, 3rd edition, 2009.
    - The Pollard Rho Algorithm - Introduction To Computer Science by Robert Sedgewick and Kevin Wayne
    - BPSW Algorithm
        - Baillie–PSW primality test
        - Baillie-PSW Primality Test -- from Wolfram MathWorld
    - MillerRabinTest
        - Strong pseudoprime
    - Lucas Test
        - https://en.wikipedia.org/wiki/Lucas_pseudoprime#Strong_Lucas_pseudoprimes
    - Jacobi Symbol
        - Jacobi symbol
    - Square Root of Number (Heron's or Babylonian Method)
        - https://en.wikipedia.org/wiki/Methods_of_computing_square_roots#Babylonian_method
    - AKS Algorithm - AKS primality test
    - Pocklington Primality Test
        - https://en.wikipedia.org/wiki/Pocklington_primality_test#math_7
        - Looking for a fast deterministic primality test for numbers above 64 bits
    - Sieve of Eratosthenes - https://github.com/kimwalisch/primesieve
    - Double to Integer Converter
        - Double-precision floating-point format
        - https://de.wikipedia.org/wiki/IEEE_754#Allgemeines
        - BitMasks - Java Math for Engineers - JAVA Programming for Engineers - page 275
        - Bias value - IEEE floating point
        - 52-bit significant Significand
    - Java Number Theory Library - TilmanNeumann/java-math-library
    - Wolfram Alpha to verify my prime factorization method
- Brute Force Testing Suite

    - Print all ways to break a string in bracket form
    - Efficient program to print all prime factors of a given number
- Conditional Probability - Conditional probability
- Continuous Probability - https://en.wikipedia.org/wiki/Probability_distribution#Continuous_probability_distribution
- Continuous Distributions - 1.8-1.9: Continuous Random Variables [10pt] 1.10.1: Uniform Distribution (Continuous) [10pt] 1.10.4
- Almost Surely definition - Almost surely