**CODE 1: Sqrt**

**package** sqrt;

**import** java.util.Scanner;

*/\*\**

*\**

*\** ***@author*** *324921162*

*\* Muir, Thomass*

*\* ICS3U || Jeganathan*

*\* December 07, 2017*

*\* Program to calculate the square root of any number to close accuracy*

*\**

*\*/*

**public class** Sqrt {

*/\*\**

*\** ***@param*** *args the command line arguments*

*\*/*

**public static void** main(String[] args) {

*//Declare Scanner*

Scanner sc = **new** Scanner (System.in);

*//Ask user for input*

System.out.println(**"Please input a number to calculate square root:"**);

**int** intNumber = sc.nextInt();

*//Set First guess to 1.0*

**double** dblThisGuess = 1.0;

*//Counter to determine guess number*

**int** intCount = 1;

*//Declare variables for the difference, and the next guess*

**double** dblDifference;

**double** dblNextGuess;

*//Print the first guess, which is always 1.0*

System.out.println(**"Guess "** + intCount + **": "** + dblThisGuess);

*//While loop to keep program running until it reaches 10*

*//Calculate the next guess using the correct formula*

*//Print each guess to keep trach*

*//if sqrt is achieved, print the final guess*

**while**(intCount <= 10){

intCount++;

dblNextGuess = 0.5 \* (dblThisGuess + (intNumber/dblThisGuess));

dblDifference = dblNextGuess - dblThisGuess;

**if**(dblDifference <= 0.001 && dblDifference >= 0 || dblDifference >=-0.001 && dblDifference <= 0){

System.out.println(**"Guess "** + intCount + **": "** + dblNextGuess + **"\n"** + dblNextGuess + **" is our final guess for the square root of "** + intNumber);

intCount = 11;

}

*// if intCount ever reaches 10*

**else if** (intCount == 10){

System.out.println(**"-1.0"**);

}

*//if sqrt isn't achieved, calculate the next ThisGuess number*

*//Print the guess number*

**else** {

dblThisGuess = dblNextGuess;

System.out.println(**"Guess "** + intCount + **": "** + dblThisGuess);

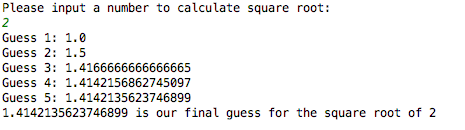
}

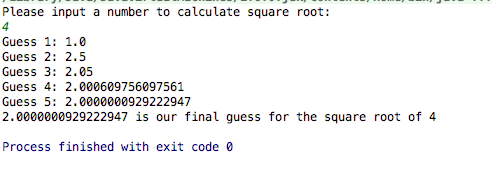
}

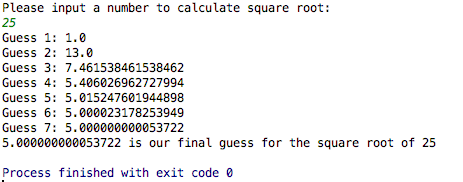
}

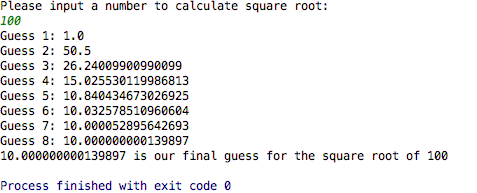
}

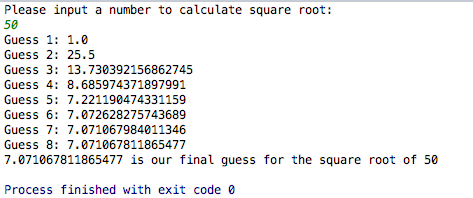
**CODE 1: Sqrt Testcases**











**CODE 2: PrimeCounter**

**package** primecounter;

*/\*\**

*\**

*\** ***@author*** *324921162*

*\* Muir, Thomass*

*\* ICS3U || Jeganathan*

*\* December 07, 2017*

*\* Read a positive integer n from the user and count the number of prime*

*\* numbers less than or equal to n*

*\**

*\**

*\*/*

**import** java.util.Scanner;

**public class** PrimeCounter {

**public static void** main(String[] args) {

*//Declare Scanner*

Scanner sc = **new** Scanner(System.***in***);

*//Ask user for input*

*//Declare variables*

System.***out***.println(**"Enter a number:"**);

**int** intN = sc.nextInt();

**int** intPrimeCount = 0; *//Variable for prime count*

**int** intPrime; *//Variable to check if prime number*

**int** i, j;

**for** (i = 2; i < intN; i++) {

intPrime = 0;

**for** (j = 2; j < i; j++) {

**if** (i % j == 0)

intPrime = 1; *//Not a prime*

}

**if** (intPrime == 0) {

intPrimeCount++; *//Add to prime count*

}

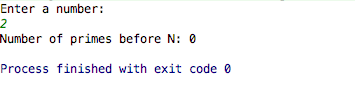
}

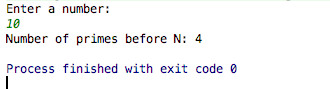
System.***out***.println(**"Number of primes before N: "** + intPrimeCount);

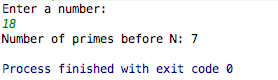
}

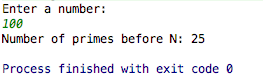
}

**CODE 2: PrimeCounter Testcases**











**CODE 3: Checksum**

*/\*\**

*\* Created by thomassmuir on 2017-12-07.*

*\*/*

**import** java.util.Scanner;

*/\*\**

*\**

*\** ***@author*** *324921162*

*\* Muir, Thomass*

*\* ICS3U || Jeganathan*

*\* December 07, 2017*

*\* A program that takes a 9-digit integer as a command line argument, computes the checksum, and prints out the 10-*

*digit ISBN number. It's ok if you don't print out any leading 0s.*

*\**

*\*/*

**public class** Checksum {

**public static void** main(String[] args) {

*//Declare Scanner*

Scanner sc = **new** Scanner (System.in);

*//Ask for user input*

System.out.print(**"Enter 9 digits of an ISBN: "**);

**int** intISBN= sc.nextInt();

*//Determine the place number of the user-inputed ISBN*

**int** intd1 = intISBN / 100000000;

**int** remainingDigits = intISBN % 100000000;

**int** intd2 = remainingDigits / 10000000;

remainingDigits %= 10000000;

**int** intd3 = remainingDigits / 1000000;

remainingDigits %= 1000000;

**int** intd4 = remainingDigits / 100000;

remainingDigits %= 100000;

**int** intd5 = remainingDigits / 10000;

remainingDigits %= 10000;

**int** intd6 = remainingDigits / 1000;

remainingDigits %= 1000;

**int** intd7 = remainingDigits / 100;

remainingDigits %= 100;

**int** intd8 = remainingDigits / 10;

remainingDigits %= 10;

**int** intd9 = remainingDigits;

*//Calculate d10*

**int** intd10 = (intd1 + intd2 \* 2 + intd3 \* 3 + intd4 \* 4 + intd5 \* 5

+ intd6 \* 6 + intd7 \* 7 + intd8 \* 8 + intd9 \* 9) % 11;

*//Print ISBN number*

**if** (intd10 == 10) {

System.out.println(**"The ISBN-10 number is "** + intd1 + intd2 + intd3 + intd4 + intd5

+ intd6 + intd7 + intd8 + intd9 + **"X"**);

}

**else** {

System.out.print(**"The ISBN-10 number is "** + intd1 + intd2 + intd3 + intd4 + intd5

+ intd6 + intd7 + intd8 + intd9 + intd10);

}

}

}

**CODE 3: Checksum Testcases**









