

Write a Java Program to solve each one of the problem. Please name your project as **mention in each question**. Save all your project into one folder called **firtsname_lastname_C5_HW** and compress he folder drop into the Google Drive

- 1) Write a java program called **Sqrt.java** to Calculate the square root using *Newton's method*. This is an iterative method: it starts with a guess at the answer and improves the guess over and over until it is "close enough". If we are looking for the square root of x , and the current guess is *this_guess*, then the next guess is given by the following expression:
$$\text{next_guess} = 0.5 \times (\text{this_guess} + x/\text{this_guess})$$
This becomes the new value for *this_guess* and the whole process is repeated. Your function should start with a first guess of 1.0, and continue until the difference between two successive guesses is less than 0.001 (That's the definition of "accuracy".) If that takes more than 10 iterations, your program should stop and return -1.0. There is a library function `math.sqrt`. **Do not use it.** (Exponentiation operator). The idea is to calculate the square root using only simpler arithmetic operations that are present in every programming language, **not** to try to find some roundabout way of computing the answer! Be aware that all the values in this part of the exercise are floats. If you accidentally use integers, you may run into serious mysterious troubles, so please be careful to provide decimal parts for numeric constants. (You should use integers where appropriate, of course; for example, count the iterations with an integer.)
- 2) **Counting primes.** Write a program **PrimeCounter.java** that takes one command-line argument N and prints out the number of primes less than N .
- 3) **Checksums.** The International Standard Book Number (**ISBN**) is a 10 digit code that uniquely specifies a book. The rightmost digit is a *checksum* digit which can be uniquely determined from the other 9 digits from the condition that $d_1 + 2d_2 + 3d_3 + \dots + 10d_{10}$ must be a multiple of 11 (here d_i denotes the i th digit from the right). The checksum digit d_{10} can be any value from 0 to 10: the ISBN convention is to use the value X to denote 10. *Example:* the checksum digit corresponding to 020131452 is 5 since is the only value of d_{10} between 0 and 10 for which $d_1 + 2*2 + 3*5 + 4*4 + 5*1 + 6*3 + 7*1 + 8*0 + 9*2 + 10*d_{10}$ is a multiple of 11. Write a program `ISBN.java` 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 0's.