**Name:**

**Advanced Programming in Java**

**Lab Exercise 11.21.2024**

1. Given a number, a, and you want to find its square root. One way to do that is to start with a rough guess about the answer, x0, and then improve the guess using this formula:



For example, if we want to find the square root of 9, and we start with x0 = 6, then



which is closer. We can repeat the procedure, using x1 to calculate x2, and so on. In this case, x2 = 3.075 and x3 = 3.00091. So it converges quickly on the correct answer.

Write a method called squareRoot that takes a double and returns an approximation of the square root of the parameter, using this technique. You should not use Math.sqrt. As your initial guess, you should use a=2. Your method should iterate until it gets two consecutive estimates that differ by less than 0.0001. You can use Math.abs to calculate the absolute value of the difference.

1. A word is said to be “abecedarian” if the letters in the word appear in alphabetical order. For example, the following are all six-letter English abecedarian words:

abdest, acknow, acorsy, adempt, adipsy, agnosy, be\_st, behint, beknow, bijoux, biopsy, cestuy, chintz, deux, dehors, dehort, deinos, diluvy, dimpsy

Write a method called isAbecedarian that takes a String and returns a boolean indicating whether the word is abecedarian.

1. A word is said to be a “doubloon” if every letter that appears in the word appears exactly twice. Here are some examples doubloons found in the dictionary:

Abba, Anna, appall, appearer, appeases, arraigning, beriberi, bilabial, Caucasus, coco, Dada, deed, Emmett, Hannah, horseshoer, intestines, Isis, mama, Mimi, murmur, noon, Otto, papa, peep, reappear, redder, sees, Shanghaiings, Toto

Write a method called isDoubloon that takes a string and checks whether it is a doubloon. To ignore case, invoke the toLowerCase method before checking.

1. **Number reversal**

For a given input number, return the number in reverse. So, an input of 3956 should return 6593.