Exericses: Day 1

Exercise 1: Here is an example of a Java while loop:

```java
int n = 10;
while(n > 0) {
    System.out.println(n);
    n--;
}
```
The block of code is executed as long as the boolean condition is satisfied. This fragment of code would print the integers from 10 down to 1.

(a) Remember that the Taylor series

\[
\arctan x = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{2k + 1}
\]

is valid for \(|x| < 1\). Moreover, for a given value of \(x\), this gives an alternating series. Using a while loop to add terms until the error is sufficiently small, create a method

```java
public static double arctan(double x, double tolerance)
```

that uses the Taylor series to compute \(\arctan x\) with an error less than \(\text{tolerance}\). Try to generate the individual terms in the series efficiently; that is, try to use one term in the series to generate the next.

(b) Remembering that

\[
\frac{\pi}{4} = \arctan \frac{1}{2} + \arctan \frac{1}{3},
\]

create a program using your method to compute \(\pi\) with an error less than \(10^{-n}\) where \(n\) is input into the program. If \(n = 20\), does your program give the correct result? Explain why or why not.

Exercise 2: There is a Division Algorithm for the Gaussian integers: that is, given Gaussian integers \(a\) and \(b\) where \(b \neq 0\), there are (not necessarily unique) Gaussian integers \(q\) and \(r\) such that

\[a = bq + r, \quad |r| < |b|\]

In fact, to find \(q\) you may view \(b\) as a complex number and find the closest Gaussian integer to the complex number \(a/b\). This shows that \(r\) may be chosen to satisfy

\[|r| \leq \frac{|b|}{\sqrt{2}}.\]

Beginning with the \texttt{GaussianInteger} class we constructed in the lecture (and given in the notes), add an instance method

```java
public static GaussianInteger[] dividedBy(GaussianInteger b)
```
that returns an array with two GaussianIntegers, \( q \) and \( r \), that result from dividing the Gaussian integer represented by the instance by \( b \). You will want to use the expression `(int) Math.round(double x)` to find the closest integer to a real number \( x \). You may wish to add in a few auxiliary methods to help out.

Also construct a program that reads two GaussianIntegers from the command line and prints the quotient and remainder:

```java
$ java DivisionAlgorithm 10 5 1 1
q = 8 + -2 i, r = 0 + -1 i
```

**Optional Exercise 3:** Implement the Euclidean Algorithm to find the greatest common divisor of two Gaussian integers by creating a class method

```java
public static GaussianInteger gcd(GaussianInteger a, GaussianInteger b)
```

Here is a helpful fact: given two Gaussian integers \( a \) and \( b \), and Gaussian integers \( q \) and \( r \) such that

\[
a = bq + r,
\]

the greatest common divisor of \( a \) and \( b \) is the same as the greatest common divisor of \( b \) and \( r \). This fact leads to the Euclidean Algorithm, presented for the integers in Chapter 1 of the notes.

**Optional Exercise 4:** Study the class `java.lang.String` and write a program that reads a String from the command line and writes it backward.

```java
$ java Backward Hello
olleH
```