Homework 3

Problem 3:

Write a pseudocode algorithm for polynomial division. Write your answer in the file answers/problem3.pdf.

```
Method div(RatPoly P){

    Return NaN if P or X are NaN

    Q = new empty Poly()
    R = Poly (Coefficients of this, represented as X)

    Return 0 if P.degree  > R.degree

    While ( !( P.degree  > R.degree) and !R.equals(ZERO) ){
        C = Lead coefficient of R / Lead coefficient of P
        Q = Q + C                          //adding division to polynomial
        R = r – C * P                      //changing R's value
    }

    Return Q
}
```

State the loop invariant for the main loop and prove partial correctness. Write your answer in the file answers/problem3.pdf. For the proof question, you do not need to handle division by zero; however, you will need to do so in the Java program. Important: write your pseudocode, invariants, and proofs first, then write the Java code. Going backwards will be harder.

Loop invariant:
$X = R + (P*Q)$

Proof:
The Base Case:
X = some value x
Q = 0
P = p
R = x
So, the result would be $X = 0* p + x$ ➔ $X = x$, which is <u>true</u>

Assume $X = R + (P*Q)$:

Next iteration:
$X = (R-C*P)+(P*(Q+C))$
$X = R - C*P + Q*P - C*P$
$X = R + (Q*P)$, <u>True</u>