1 Problem 3

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

```
//precondition: u and v are polynomials && u and v are not null
    && v != 0 && degree(u) >= degree(v)
function div(u, v)
    if degree(v) > degree(u) then
        return u / v
    q = 0
    r = u
    while degree(r) >= degree(v) do
        t = c * x^(degree(r) - degree(v))
        q = q + t
        r = r - t * v
    end while
    return q
end function
//postcondition: u = qv
```

1.2 When writing pseudocode use symbols +, -, *, and / to express rational number and polynomial arithmetic. You may also use u[i] to retrieve the coefficient at power i of polynomial u, as well as c * x^i to denote the single-term polynomial of degree i and coefficient c.

```
//precondition: u and v are polynomials && u and v are not null
    && v != 0 && degree(u) >= degree(v)
function div(u, v)
    if degree(v) > degree(u) then
        return u / v
    q = 0
    r = u
    while degree(r) >= degree(v) do
        c = r[degree(r)] / v[degree(v)]
        t = c * x^(degree(r) - degree(v))
        q = q + t
        r = r - t * v
    end while
```

```
    return q
end function
//postcondition: u = qv
```

1.3 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: $u = qv + r$ // where r is the remainder of u divided by v and q is the quotient of u divided by v.
precondition: u and v are polynomials && u and v are not null && v != 0 && degree(u) >= degree(v)
postcondition: u == qv

Proof:
Base Case:
Before the loop, $u = qv + r$ where $q = 0$ and $r = u$.
Given q = 0
Assume r != 0, then degree(r) >= degree(v)
V*0 = 0, so r >= 0

Inductive Step:
Assume $u = qv + r$ holds at itr k
$q\_new = q\_old + t$
$r\_new = r\_old - t * v$
This gives us:
$u\_new = q\_new * v + r\_new$
$u\_new = (q\_old + t)v + (r\_old - t * v\_old)$
$u\_new = q\_old * v + t * v + r\_old - t * v$
$u\_new = q\_old * v + r\_old$

This proves that the loop invariant holds at each iteration k.

Implication Post Condition:
u ==qv + r and !((degree(r) >= degree(v))
u == qv + r && r == 0, thus u == qv, implying the post condition