Project 0

This project extends Problem 2-3 from the textbook. The student must show that Horner's Rule correctly evaluates a polynomial in less running time than a straightforward term-by-term calculation.

The following code fragment implements Horner's Rule for numerical solution of a polynomial:

```
1    y = 0;
2    i = n;
3    while (i >= 0) {
4         y = a[i] + x * y;
5         i = i - 1;
6    }
```

a. Determine the asymptotic running time of this code fragment analytically. Confirm your analysis with at least five different data sets in a computer program. One of these must be the instructor-provided `10_coefficients.txt`.

b. Write pseudo code to compute each term of a polynomial from scratch. Determine the asymptotic running time of your code analytically. Confirm your analysis with the same data sets used for Part a. Discuss how the naive algorithm compares with Horner's Rule.

c. Prove analytically that the following is a loop invariant for lines 3-6 of Part a:

$$y = \sum_{k=0}^{n-(i+1)} a_{k+i+1} x^k$$

Confirm with a computer program that the loop invariant holds while evaluating a polynomial of degree $n = 10$, by using the instructor-provided `10_coefficients.txt` data file. The printed output must provide values for initialization, maintenance, and termination. The maintenance values must show both the actual and loop invariant values for each loop.

d. Argue that the given code fragment for Horner's Rule correctly evaluates a polynomial.

In addition to the source files and input files, project deliverables will be outputs of the following three items, as separate pages:

- Time for Horner's Rule as a function of *n*
- Time for Naive Algorithm as a function of *n*
- Loop Invariant Output

The project due date is September 9, 2009.