Or Brener
Student ID#: 1140102

Assignment 2
Polynomial Arithmetic in Ada
Reflection Report

Program design:

- The main menu prompts the user for their desired arithmetic;
- Addition: first the user must input 2 valid polynomials, then they are added by like terms.
- Subtraction: first the user must input 2 valid polynomials, then they are subtracted by reversing the signs of the second polynomial (multiplying itself by -1) and then adding them together.
    - A simple mathematical proof for this: $A - B = A + (-B)$
- Multiplication: first the user must input 2 valid polynomials, then each term of the first polynomial is multiplied for the whole second polynomial (stored in a temporary polynomial). Then the temporary polynomial is summed with the result (all temps are summed together to get the product).
- Evaluation: first the user inputs a valid polynomial and a valid integer value for x. That value is then substituted into the polynomial and evaluated using simple mathematical rules.
- At any point the user can exit the program by entering menu option 5.

Notes:

- I have a linked list that stores a polynomial
    - Each node has an integer coefficient, integer degree, and pointer to the next node
    - The linked list is ordered from largest degree to smallest degree
    - Terms with a coefficient are not stored
    - An empty polynomial is just null
- In readpoly(), the user inputs the highest exponent in the polynomial and then inputs the coefficients for each of the terms (degree = highest exponent to zero)
- Note that for all instances where user input is prompted, accurate input validation is executed (checks that it is an integer)
- Note that all arithmetic works with polynomials of different sizes and degrees. Once the arithmetic is computed, the result is outputted to the user.

Programming Process/Reflection:

Ada is definitely a headache to study and program in (it is a legacy language after all). The hardest challenge that I found was that it was quite difficult to find helpful resources online when running into bugs (although the professor's website was definitely a great resource!). With that being said, Ada's strict type restrictions and exception throwing make it a very good fit for a program such as this where only integers are valid input.