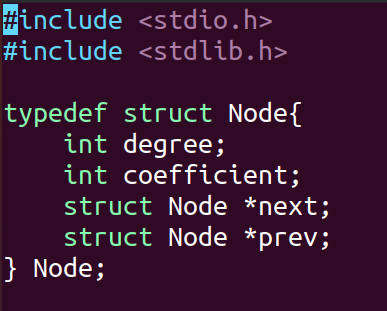
**Programming Assignment #0**

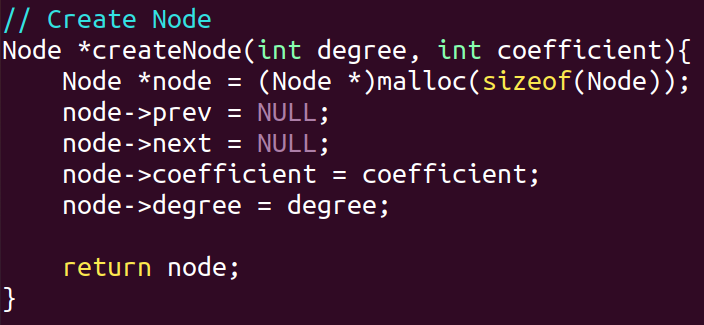
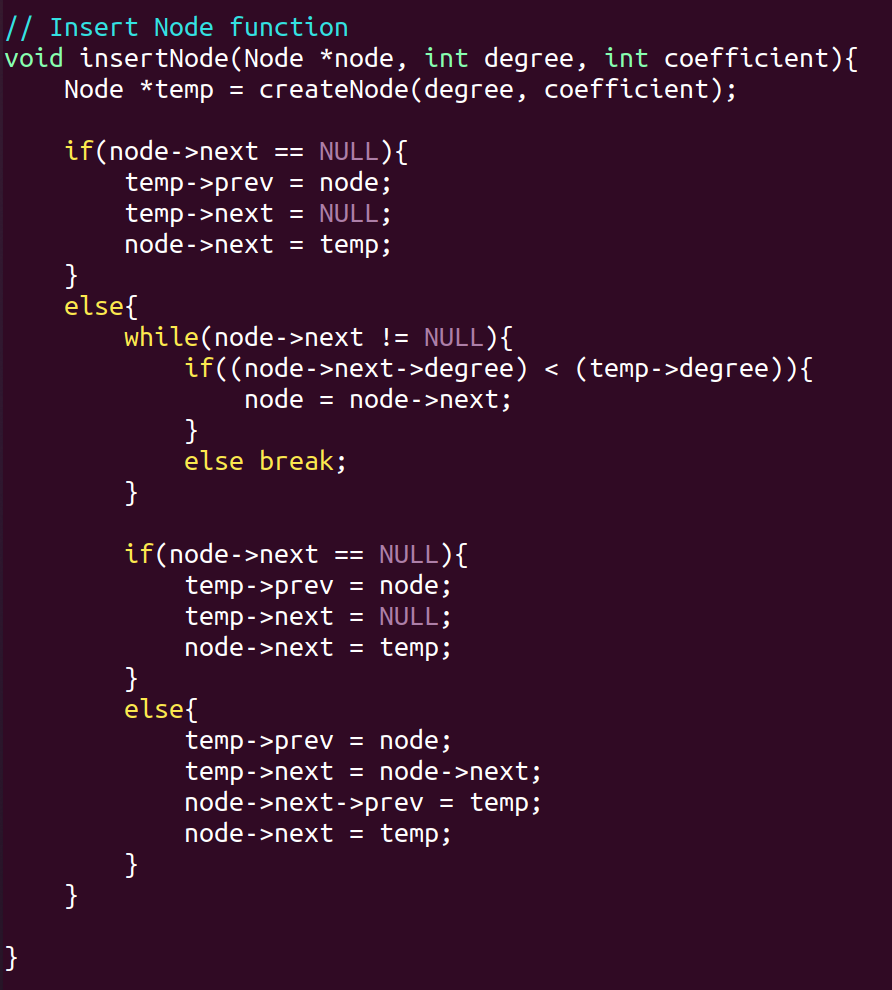
1. **Introduction**

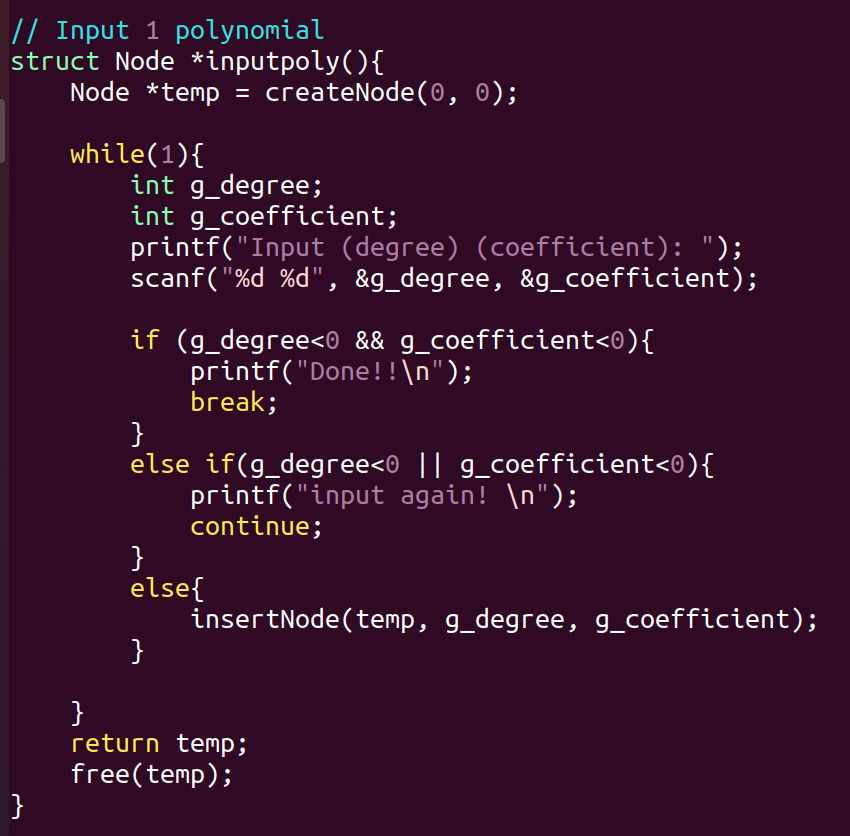
The multiplication of two polynomials is implemented through a doubly linked list.

The process of implementing the multiplication of polynomials based on the input and output of polynomials is described.

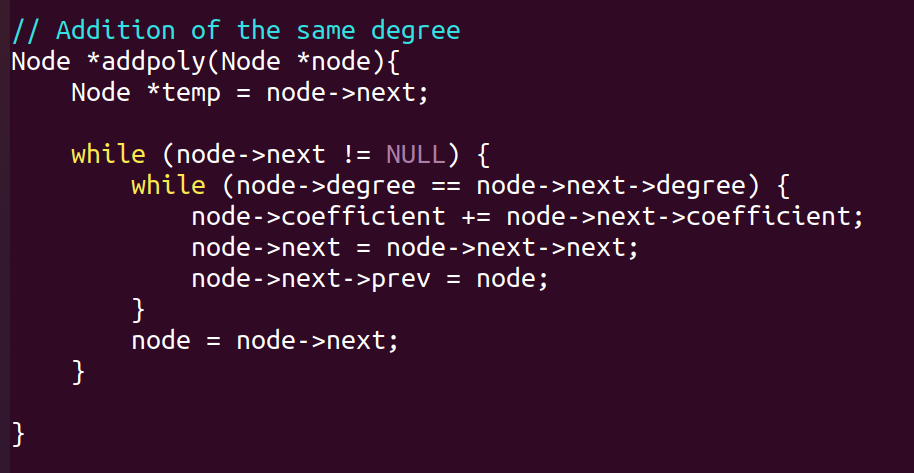
* Project
  + Your approach to the problem
    - implement a polynomial in a doubly linked list. -> 1. Inputpoly
    - output a polynomial in a doubly linked list -> 2. PrintNode
    - implement the multiplication of two polynomials -> 3. Multiply
  + What you did
    - Making Struct Node, and designate alias of Struct Node as Node



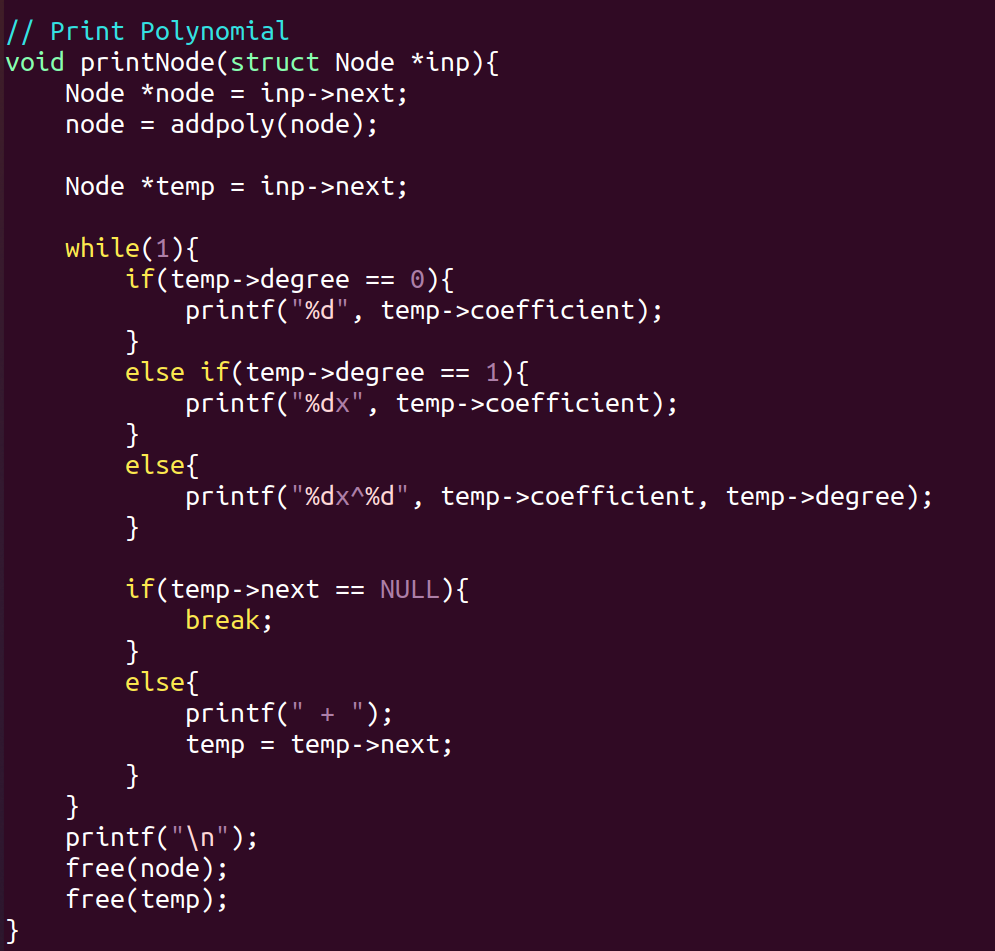
* + - Create Node
      * It is responsible for creating and initializing nodes for the first time.
    - Insert Node
      * It is responsible for adding nodes.
      * Implementation by dividing the first input case and the second input case.
    - Inputpoly
      * If all are negative, stop input.
      * Re-enter if either is negative
      * In other cases, add the number of inputs using insertnode.

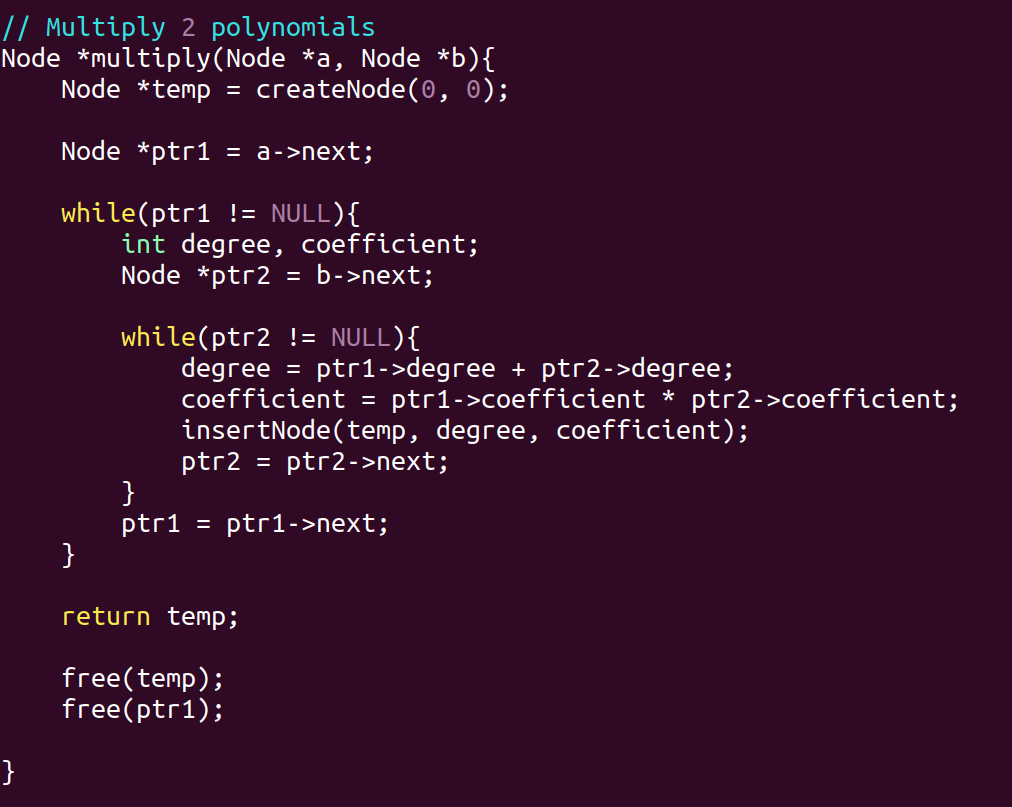


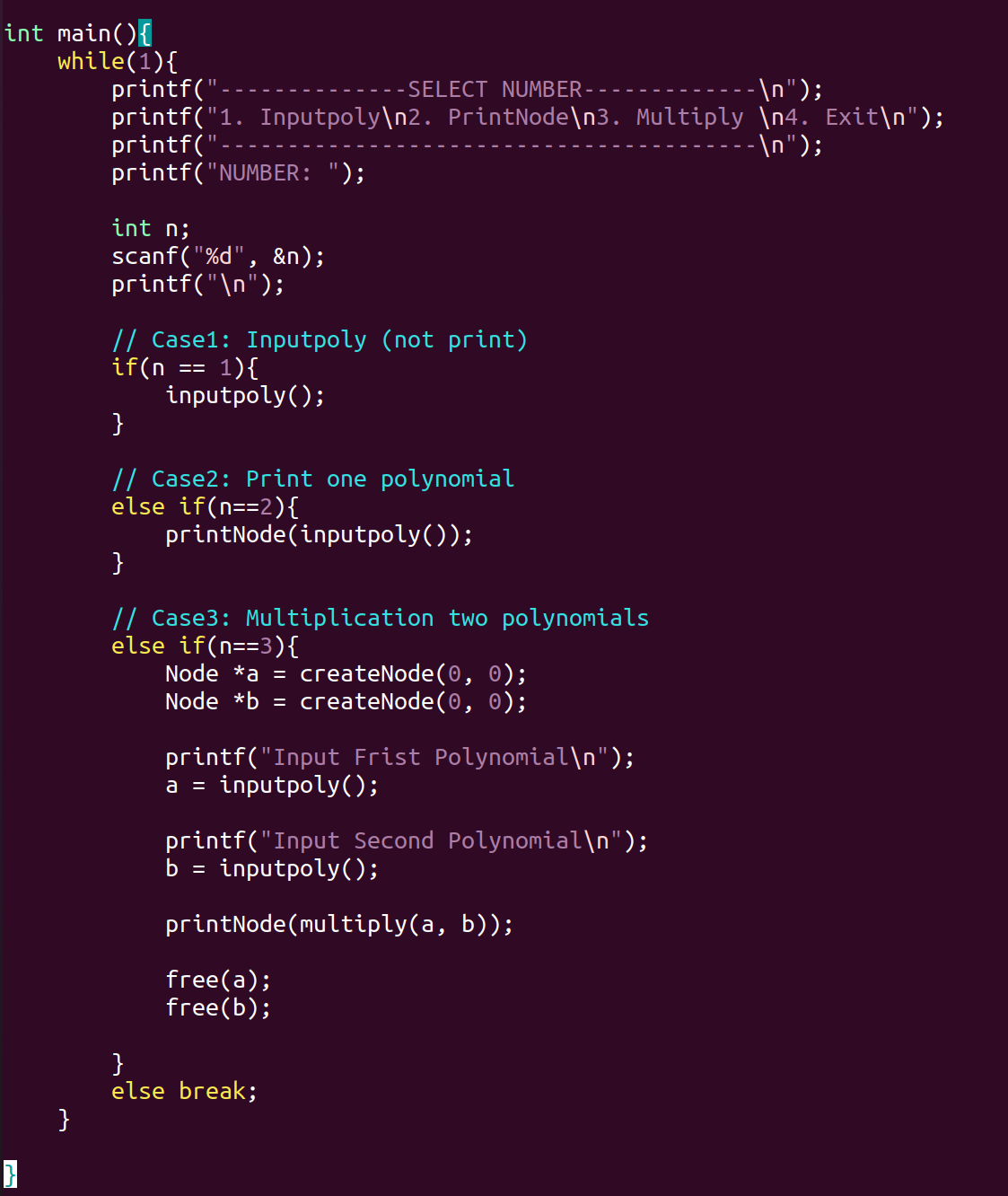
* + - addpoly
      * if the same degree, add it.



* + - printNode
      * output polynomials for each
      * constant
      * primary case
      * other cases (degree>=2)

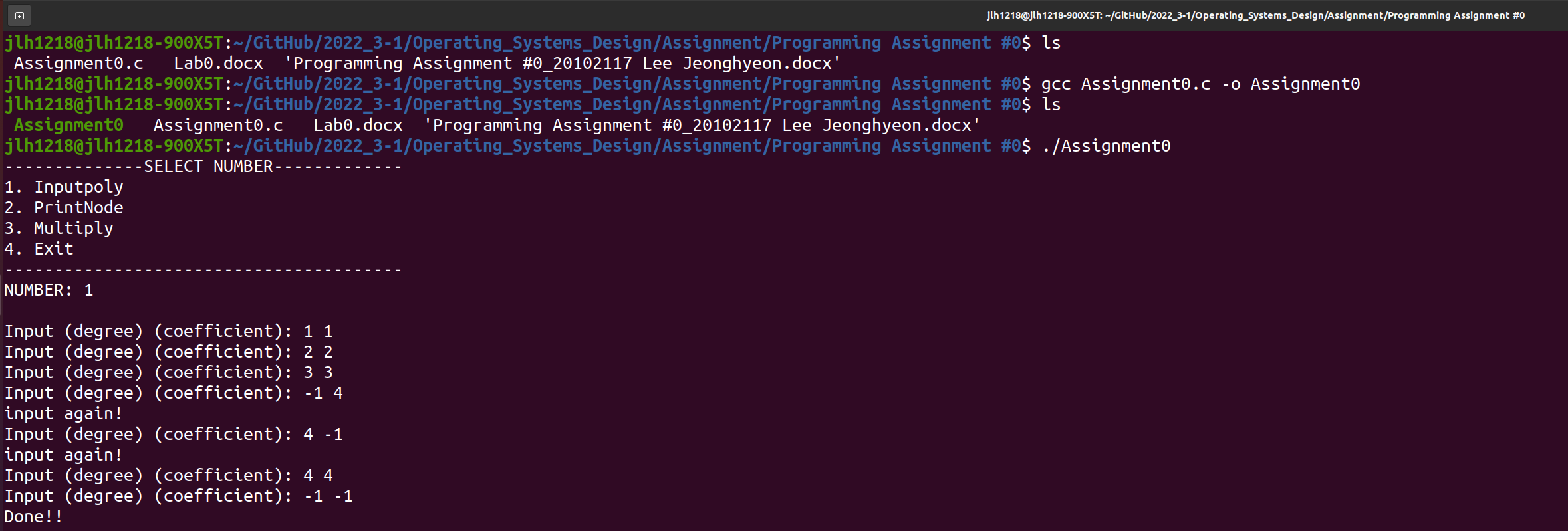
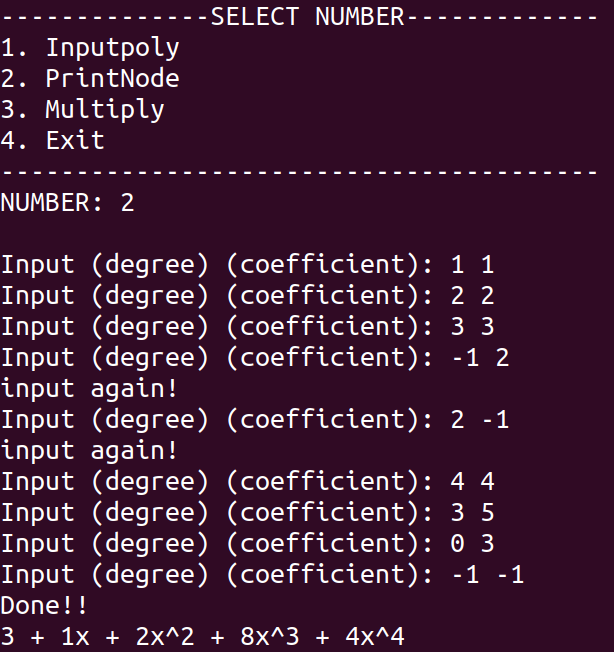
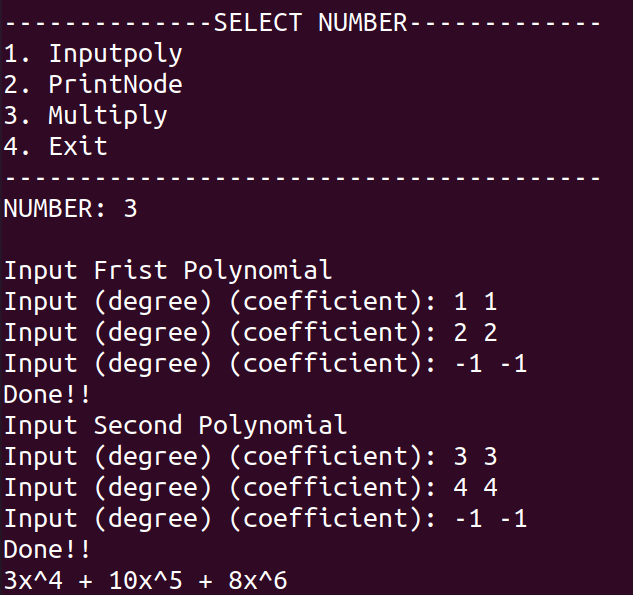
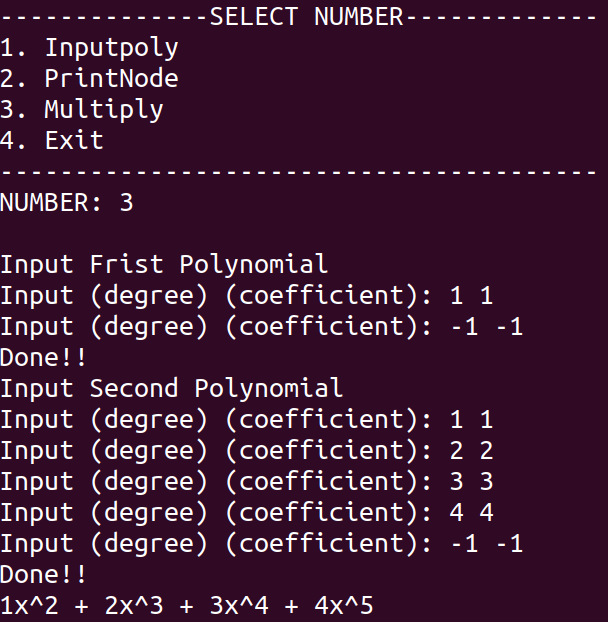


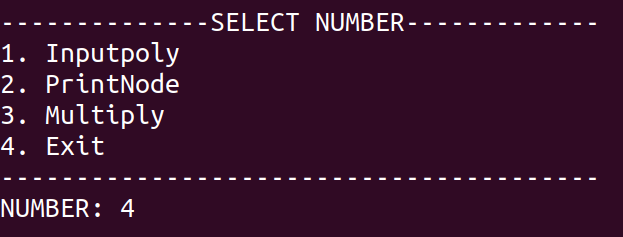
* + - multiply
      * implementation of the multiplication of two polynomials
    - main
      * Describe the implementation process in 4 views
      * 1. Input polynomial.
      * 2. Output of polynomial.
      * 3. Multiplication of two polynomials.
      * 4. Exit.



* + Design
    - What you already had (and where it came from)
      * Reference

<https://dojang.io/mod/page/view.php?id=645>

* + - What you added/changed
    - For parts, include close-up drawings
      * Compile Assignment.c file
      * Case1: Input polynomial
        + If degree or coefficient is negative, reenter it.
      * Case2: Print polynomial
        + Additional output when same degree is entered
      * Case3: Multiplication two polynomials
        + can be calculated even with different expression lengths
      * Case4: Exit -> Break;



1. **Conclusions**

The structure of the doubly linked list could be directly implemented and understood.

While experiencing errors in the middle, various attempts such as changing pointers were made to derive results.

It was difficult to understand the results obtained by chance in the implementation process, but it was an opportunity to think about why these results came out, look for them, and learn.

Obviously, it may not be a perfect understanding, but I was able to better understand the concept of a pointer and the concept of a memory while implementing it myself.