1. **A text document containing an explanation of your designed data type, your specification and your pseudo-code for all operations.**

**Explanation of data type:**

typedef struct

{

polyNode\* head;

polyNode\* current;

} polynomial;

In our data type, named “polynomial”, there are two nodes named “head” and ”current” these are of type “polyNode”

typedef struct n {

term d; // STORE A TERM IN THE NODE

struct n\* successor;

} polyNode;

The data type “polyNode” contains two things, the first is of data type “term” named “d”.

The second is creating a pointer to an empty node.

The data type “term” contains a double and an int, named “coefficient” and “exponent”

**Your specification:**

**Pseudo-code for all operations**

**Pseudo-code for mainProgram.c:**

Create two polynomial pointers named “poly1” and “poly2” and set both values to null.

Print to screen that the program is creating new polynomials and to input the values for the first and second polynomial.

Set “poly1” and “poly2” equal to the return of the method createPoly after each time it’s called.

Display “poly1” and “poly2” that were just created on the screen using the displayPoly function and passing in the “poly1” and “poly2” variable names that was just created.

Print to the screen that the program is Adding the polynomials.

Call the addPoly function and pass in both “poly1” and “poly2”

Print to the screen that the program is Subtracting the polynomials.

Call the subPoly function and pass in both “poly1” and “poly2”

Print to the screen that the program is Multiplying the polynomial.

Call the multiplyPoly function and pass in “poly1”

Print to the screen that the program is Dividing the polynomial.

Call the dividePoly function and pass in “poly1”

Print to the screen that the program is normalizing the polynomial.

//NOT SURE YET

Print to the screen that the program is returning the order of polynomial

Call the orderPoly function and pass in the “poly1” variable.

Print to the screen that the program is deleting polynomial.

Call the deletePoly function and pass in the “poly1” variable.

Print to screen that the program is displaying the polynomial

Call the displayPoly function and pass in the “poly1” variable.

Print to the screen that we are exiting the program

Return exit success to indicate successful program execution status.