Name: 周嘉禾

Student ID: D1166506

In this program, I used recursively node to initialize the node and defined the node pointer as " polynomial". Next, I filled the function with the code:

1. initial
   Let polynomial *L=NULL.

2. getTerm
   Traverse the terms of the polynomial P and return the number of terms.

3. getCoeff
   Traverse to the assigned degree and return the term at that degree.
   Return -1 if there is any unwanted situation.

4. searchTerm
   If find the term required, return the degree of the term .
   Otherwise, return -1 as cannot find the term .

5. replaceTerm
   If the term wanted to be replaced had existed and the coefficient given wasn't zero, then traverse to the assigned degree, change the coefficient at that degree, and return degree as setting the term successfully.
   Once the coefficient given is zero, traverse to find the degree where the term is located, change the next term of the previous node, and return the degree of the term.
   Otherwise, create a term, change *P or next to the term, and return the degree of the term inserting.
   Return -1 if there is any unwanted situation.

6. clear
   Traverse the polynomial and free all of the terms.

7. is_empty
   Return 1 if the polynomial is empty.
   Otherwise, return -1.

8. printPoly
   Traverse the polynomial and print all the terms inside the polynomial.

9. polyAdd

   Traverse polynomials P1 and P2, add the term with the corresponding degree together, and restore the result into the polynomial P.

   After addition, return the polynomial P.

10. polyMinus

    Traverse polynomials P1 and P2, minus the term with the corresponding degree together, and restore the result into the polynomial P.

    After subtraction, return the polynomial P.

11. polyTime

    Traverse the polynomials P1 and P2. Let each term of P1 times each term of P2.

    If the degree wanted to insert had existed inside the polynomial P, then add the coefficient up and restore the result into the polynomial P.

    Otherwise, insert a term into polynomial P.

    After multiplication, return the polynomial P.

12. evaluate

    Calculate the result for the given number of terms, sum up the results, and return the summation.

13. Power

    Use a method called exponentiating by squaring to calculate the assigned power to the base.

At the end, in the main function, I just printed some sentences required, called the functions needed as the instruction, and proved whether the equations were satisfied or not.