Polynomial Manipulation

Write a C program to implement the following operations on Singly Linked List.

(i) Polynomial Addition

(ii) Polynomial Subtraction

(iii) Polynomial Multiplication

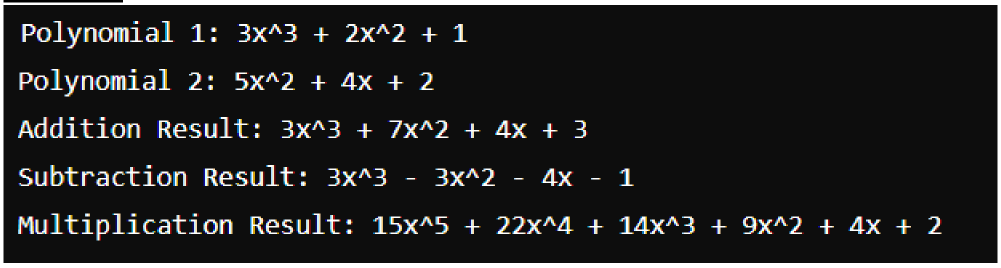
PROGRAM:

```
#include <stdio.h>
#include <stdlib.h>

typedef struct Node {
   int coeff;
   int exp;
   struct Node* next;
} Node;

// Function prototypes
Node* createNode(int coeff, int exp);
void appendNode(Node** head, int coeff, int exp);
void displayPolynomial(Node* head);
Node* addPolynomials(Node* poly1, Node* poly2);
Node* subtractPolynomials(Node* poly1, Node* poly2);
Node* multiplyPolynomials(Node* poly1, Node* poly2);

int main() {
   Node* poly1 = NULL;
   Node* poly2 = NULL;
   Node* result = NULL;

   // Example Polynomial 1: 3x^3 + 2x^2 + 1
   appendNode(&poly1, 3, 3);
   appendNode(&poly1, 2, 2);
   appendNode(&poly1, 1, 0);
```

```
    // Example Polynomial 2: 5x^2 + 4x + 2
    appendNode(&poly2, 5, 2);
    appendNode(&poly2, 4, 1);
    appendNode(&poly2, 2, 0);

    printf("Polynomial 1: ");
    displayPolynomial(poly1);

    printf("Polynomial 2: ");
    displayPolynomial(poly2);

    // Polynomial Addition
    result = addPolynomials(poly1, poly2);
    printf("Addition Result: ");
    displayPolynomial(result);

    // Polynomial Subtraction
    result = subtractPolynomials(poly1, poly2);
    printf("Subtraction Result: ");
    displayPolynomial(result);

    // Polynomial Multiplication
    result = multiplyPolynomials(poly1, poly2);
    printf("Multiplication Result: ");
    displayPolynomial(result);

    return 0;
}

Node* createNode(int coeff, int exp) {
    Node* newNode = (Node*)malloc(sizeof(Node));
    newNode->coeff = coeff;
    newNode->exp = exp;
    newNode->next = NULL;
    return newNode;
}

void appendNode(Node** head, int coeff, int exp) {
    Node* newNode = createNode(coeff, exp);
    if (*head == NULL) {
        *head = newNode;
        return;
    }
    Node* temp = *head;
    while (temp->next != NULL) {
        temp = temp->next;
    }
    temp->next = newNode;
}

void displayPolynomial(Node* head) {
    Node* temp = head;
    while (temp != NULL) {
        printf("%dx^%d", temp->coeff, temp->exp);
        if (temp->next != NULL) {
            printf(" + ");
        }
        temp = temp->next;
    }
    printf("\n");
}

Node* addPolynomials(Node* poly1, Node* poly2) {
    Node* result = NULL;
    while (poly1 != NULL && poly2 != NULL) {
        if (poly1->exp > poly2->exp) {
            appendNode(&result, poly1->coeff, poly1->exp);
            poly1 = poly1->next;
        } else if (poly1->exp < poly2->exp) {
            appendNode(&result, poly2->coeff, poly2->exp);
            poly2 = poly2->next;
        } else {
            appendNode(&result, poly1->coeff + poly2->coeff, poly1->exp);
            poly1 = poly1->next;
            poly2 = poly2->next;
        }
    }
    while (poly1 != NULL) {
        appendNode(&result, poly1->coeff, poly1->exp);
        poly1 = poly1->next;
    }
    while (poly2 != NULL) {
        appendNode(&result, poly2->coeff, poly2->exp);
        poly2 = poly2->next;
    }
    return result;
}

Node* subtractPolynomials(Node* poly1, Node* poly2) {
    Node* result = NULL;
    while (poly1 != NULL && poly2 != NULL) {
        if (poly1->exp > poly2->exp) {
            appendNode(&result, poly1->coeff, poly1->exp);
            poly1 = poly1->next;
        } else if (poly1->exp < poly2->exp) {
            appendNode(&result, -poly2->coeff, poly2->exp);
            poly2 = poly2->next;
        } else {
            appendNode(&result, poly1->coeff - poly2->coeff, poly1->exp);
            poly1 = poly1->next;
            poly2 = poly2->next;
        }
    }
    while (poly1 != NULL) {
        appendNode(&result, poly1->coeff, poly1->exp);
        poly1 = poly1->next;
    }
    while (poly2 != NULL) {
        appendNode(&result, -poly2->coeff, poly2->exp);
        poly2 = poly2->next;
    }
    return result;
}

Node* multiplyPolynomials(Node* poly1, Node* poly2) {
    Node* result = NULL;
    Node* poly2Start = poly2;
    while (poly1 != NULL) {
        poly2 = poly2Start;
        while (poly2 != NULL) {
            int coeff = poly1->coeff * poly2->coeff;
            int exp = poly1->exp * poly2->exp;
            Node* temp = result;
            Node* prev = NULL;
            while (temp != NULL && temp->exp > exp) {
                prev = temp;
                temp = temp->next;
            }
            if (temp != NULL && temp->exp == exp) {
                temp->coeff += coeff;
            } else {
                Node* newNode = createNode(coeff, exp);
                if (prev == NULL) {
                    newNode->next = result;
                    result = newNode;
                } else {
```

```
                newNode->next = prev->next;
                prev->next = newNode;
            }
        }
        poly2 = poly2->next;
    }
    poly1 = poly1->next;
  }
  return result;
}
```

OUTPUT:



```
Polynomial 1: 3x^3 + 2x^2 + 1
Polynomial 2: 5x^2 + 4x + 2
Addition Result: 3x^3 + 7x^2 + 4x + 3
Subtraction Result: 3x^3 - 3x^2 - 4x - 1
Multiplication Result: 15x^5 + 22x^4 + 14x^3 + 9x^2 + 4x + 2
```