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

struct Term {
    int coeff;
    int x_exponent;
    int y_exponent;
    int z_exponent;
    struct Term* next;
};

struct Term* createTerm(int coeff, int x_exp, int y_exp, int
z_exp) {
    struct Term* newTerm = (struct Term*)malloc(sizeof(struct
Term));
    newTerm->coeff = coeff;
    newTerm->x_exponent = x_exp;
    newTerm->y_exponent = y_exp;
    newTerm->z_exponent = z_exp;
    newTerm->next = NULL;
    return newTerm;
}

void insertTerm(struct Term** head, int coeff, int x_exp, int
y_exp, int z_exp) {
    struct Term* newTerm = createTerm(coeff, x_exp, y_exp,
z_exp);

    if (*head == NULL) {
        *head = newTerm;
        newTerm->next = *head;
    } else {
        struct Term* current = *head;
        while (current->next != *head) {
            current = current->next;
        }
        current->next = newTerm;
        newTerm->next = *head;
    }
}

void displayPolynomial(struct Term* head) {
    if (!head) {
        printf("Polynomial is empty.\n");
        return;
    }

    struct Term* current = head;
    do {
        printf("%dx^%dy^%dz^%d", current->coeff, current-
>x_exponent, current->y_exponent, current->z_exponent);
        current = current->next;
        if (current != head) {
            printf(" + ");
        }
    } while (current != head);
    printf("\n");
}

struct Term* addPolynomials(struct Term* poly1, struct Term*
poly2) {
    struct Term* resultPoly = NULL;
    struct Term* current1 = poly1;
    struct Term* current2 = poly2;

    do {
        int sumCoeff = current1->coeff + current2->coeff;
        insertTerm(&resultPoly, sumCoeff, current1->x_exponent,
current1->y_exponent, current1->z_exponent);

        current1 = current1->next;
        current2 = current2->next;
    } while (current1 != poly1 && current2 != poly2);

    return resultPoly;
}

void freeList(struct Term** head) {
    if (*head == NULL) {
        return;
    }

    struct Term* current = *head;
    struct Term* next;

    do {
        next = current->next;
        free(current);
        current = next;
    } while (current != *head);

    *head = NULL;
}

int main() {
    struct Term* poly1 = NULL;
    struct Term* poly2 = NULL;

    insertTerm(&poly1, 6, 2, 2, 1);
    insertTerm(&poly1, -4, 0, 1, 5);
    insertTerm(&poly1, 3, 3, 1, 1);
    insertTerm(&poly1, 2, 1, 5, 1);
    insertTerm(&poly1, -2, 1, 1, 3);

    displayPolynomial(poly1);

    insertTerm(&poly2, 2, 2, 2, 1);
    insertTerm(&poly2, 5, 0, 1, 5);
    insertTerm(&poly2, -1, 3, 1, 1);

    displayPolynomial(poly2);

    struct Term* polySum = addPolynomials(poly1, poly2);
    displayPolynomial(polySum);

    freeList(&poly1);
    freeList(&poly2);
    freeList(&polySum);

    return 0;
}
```