```
9…include <stdlib.h>
#include <math.h>

typedef struct node {
    int coeff, x, y, z;
    struct node *next;
} NODE;

/* Create header node */
NODE* createHeader() {
    NODE *h = (NODE*)malloc(sizeof(NODE));
    h->next = h;
    return h;
}

/* Insert at end (simple) */
void insertTerm(NODE *h, int c, int x, int y, int z) {
    NODE *temp = (NODE*)malloc(sizeof(NODE));
    temp->coeff = c;
    temp->x = x;
    temp->y = y;
    temp->z = z;

    NODE *p = h;
    while (p->next != h)
        p = p->next;

    temp->next = h;
    p->next = temp;
}

/* Display the polynomial */
void display(NODE *h) {
    NODE *p = h->next;
    if (p == h) {
        printf("0\n");
        return;
    }
    while (p != h) {
        printf("%+d x^%d y^%d z^%d ", p->coeff, p->x, p->y, p->z);
        p = p->next;
    }
    printf("\n");
}
```

```
/* Evaluate polynomial */
double evaluate(NODE *h, double x, double y, double z) {
    double sum = 0;
    NODE *p = h->next;
    while (p != h) {
        sum += p->coeff * pow(x, p->x) * pow(y, p->y) * pow(z, p->z);
        p = p->next;
    }
    return sum;
}

/* Add two polynomials simply by copying all terms */
NODE* addPoly(NODE *A, NODE *B) {
    NODE *C = createHeader();

    NODE *p = A->next;
    while (p != A) {
        insertTerm(C, p->coeff, p->x, p->y, p->z);
        p = p->next;
    }

    p = B->next;
    while (p != B) {
        insertTerm(C, p->coeff, p->x, p->y, p->z);
        p = p->next;
    }

    return C;
}

int main() {

    /* Polynomial P(x,y,z) */
    NODE *P = createHeader();

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

    printf("Polynomial P(x,y,z):\n");
    display(P);
```

```
    double x, y, z;
    printf("Enter values of x y z: ");
    scanf("%lf %lf %lf", &x, &y, &z);

    printf("Value = %lf\n", evaluate(P, x, y, z));

    /* Adding two polynomials */
    NODE *POLY1 = createHeader();
    NODE *POLY2 = createHeader();

    insertTerm(POLY1, 3, 2, 1, 0);
    insertTerm(POLY1, 4, 1, 2, 1);

    insertTerm(POLY2, -3, 2, 1, 0);
    insertTerm(POLY2, 5, 1, 2, 1);

    NODE *SUM = addPoly(POLY1, POLY2);

    printf("\nPOLY1:\n");
    display(POLY1);

    printf("POLY2:\n");
    display(POLY2);

    printf("SUM (simple append of terms):\n");
    display(SUM);

    return 0;
}
```