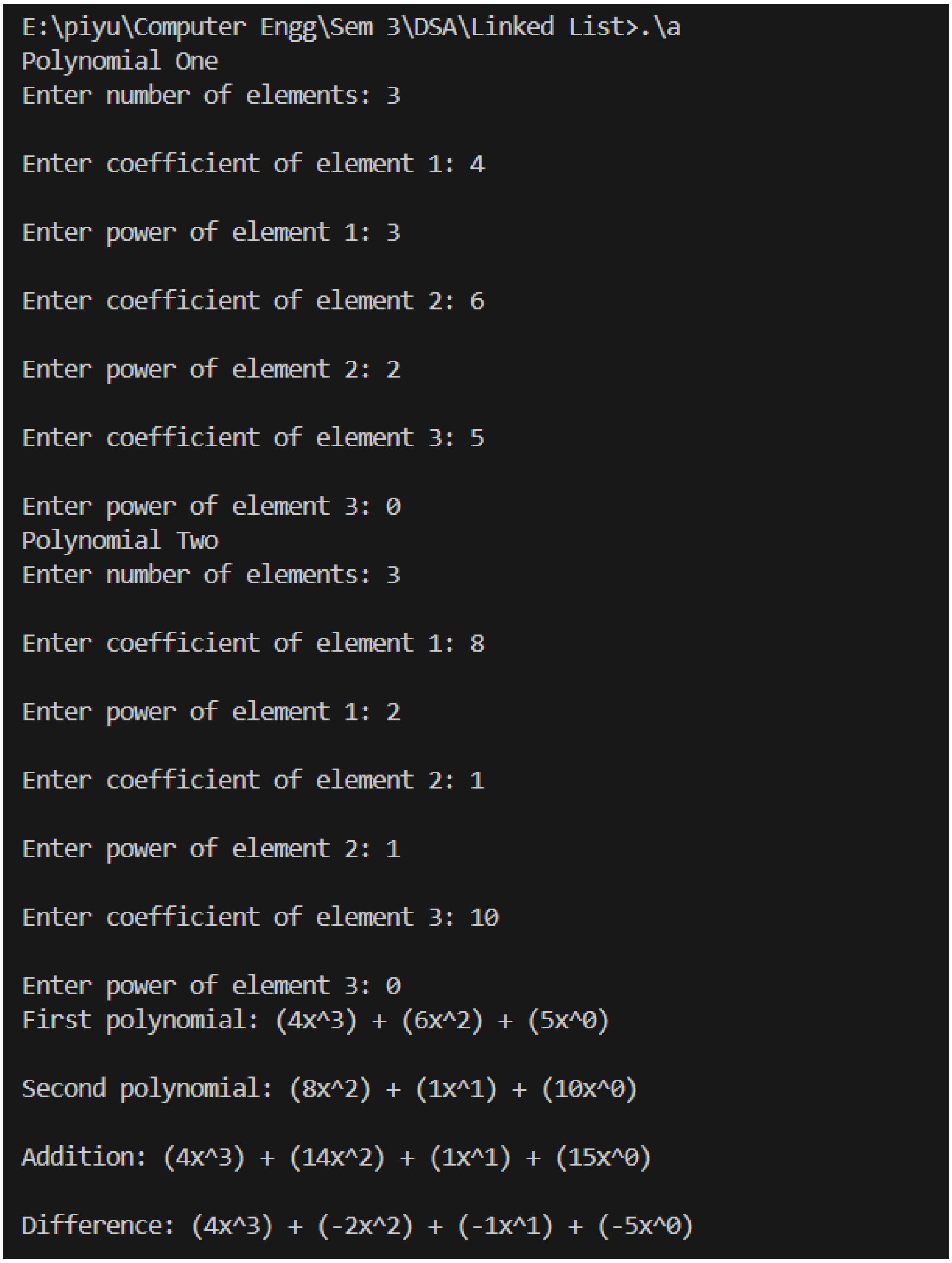
EXPERIMENT 5

Code:

```
#include <stdlib.h>
#include <stdio.h>
struct Node{
    int c,p;
    struct Node *next;
}*s1,*s2,*s3,*s4;
struct Node *create(struct Node *start){
    struct Node *nn,*temp;
    int n;
    printf("Enter number of elements: ");
    scanf("%d",&n);
    for(int i=1;i<=n;i++){
        nn=(struct Node *)malloc(sizeof(struct Node));
        printf("\nEnter coefficient of element %d: ",i);
        scanf("%d",&nn->c);
        printf("\nEnter power of element %d: ",i);
        scanf("%d",&nn->p);
        nn->next=NULL;
        if(start==NULL){
            start=nn;
            temp=start;
        }
        else{
            temp->next=nn;
            temp=temp->next;
        }
    }
    return start;
}
void result(int c, int p, struct Node **start) {
    struct Node *nn = (struct Node *)malloc(sizeof(struct Node));
    nn->c = c;
    nn->p = p;
    nn->next = NULL;

    if (*start == NULL) {
        *start = nn;
    } else {
        struct Node *temp = *start;
        while (temp->next != NULL)
            temp = temp->next;
        temp->next = nn;
    }
}
void display(struct Node *start) {
    struct Node *temp = start;
    while (temp != NULL) {
        printf("(%dx^%d)", temp->c, temp->p);
        if (temp->next != NULL)
            printf(" + ");
        temp = temp->next;
    }
    printf("\n");
}
void add(){
    struct Node *t1,*t2;
    int c,p;
    t1=s1;
    t2=s2;
    while(t1!=NULL && t2!=NULL){
        if(t1->p==t2->p){
            p=t1->p;
            c=t1->c+t2->c;
            t1=t1->next;
            t2=t2->next;
        }
        else if(t1->p>t2->p){
            p=t1->p;
            c=t1->c;
            t1=t1->next;
        }
        else{
            p=t2->p;
            c=t2->c;
            t2=t2->next;
        }
        result(c,p,&s3);
    }
    while(t1!=NULL){
        result(t1->c,t1->p,&s3);
        t1=t1->next;
    }
    while(t2!=NULL){
        result(t2->c,t2->p,&s3);
        t2=t2->next;
    }
}
void subtract(){
    struct Node *t1,*t2;
    int c,p;
    t1=s1;
    t2=s2;
    while(t1!=NULL && t2!=NULL){
        if(t1->p==t2->p){
            p=t1->p;
            c=t1->c-t2->c;
            t1=t1->next;
            t2=t2->next;
        }
        else if(t1->p>t2->p){
            p=t1->p;
            c=t1->c;
            t1=t1->next;
        }
        else{
            p=t2->p;
            c=-(t2->c);
            t2=t2->next;
        }
        result(c,p,&s4);
    }
    while(t1!=NULL){
        result(t1->c,t1->p,&s4);
        t1=t1->next;
    }
    while(t2!=NULL){
        result(-(t2->c),t2->p,&s4);
        t2=t2->next;
    }
}

void main(){
    printf("Polynomial One\n");
    s1=create(s1);
    printf("Polynomial Two\n");
    s2=create(s2);
    printf("First polynomial: ");
    display(s1);
    printf("\nSecond polynomial: ");
    display(s2);
    add();

    printf("\nAddition: ");

    display(s3);

    subtract();

    printf("\nDifference: ");

    display(s4);

}
```

Output:



```
E:\piyu\Computer Engg\Sem 3\DSA\Linked List>.\a
Polynomial One
Enter number of elements: 3

Enter coefficient of element 1: 4

Enter power of element 1: 3

Enter coefficient of element 2: 6

Enter power of element 2: 2

Enter coefficient of element 3: 5

Enter power of element 3: 0
Polynomial Two
Enter number of elements: 3

Enter coefficient of element 1: 8

Enter power of element 1: 2

Enter coefficient of element 2: 1

Enter power of element 2: 1

Enter coefficient of element 3: 10

Enter power of element 3: 0
First polynomial: (4x^3) + (6x^2) + (5x^0)

Second polynomial: (8x^2) + (1x^1) + (10x^0)

Addition: (4x^3) + (14x^2) + (1x^1) + (15x^0)

Difference: (4x^3) + (-2x^2) + (-1x^1) + (-5x^0)
```