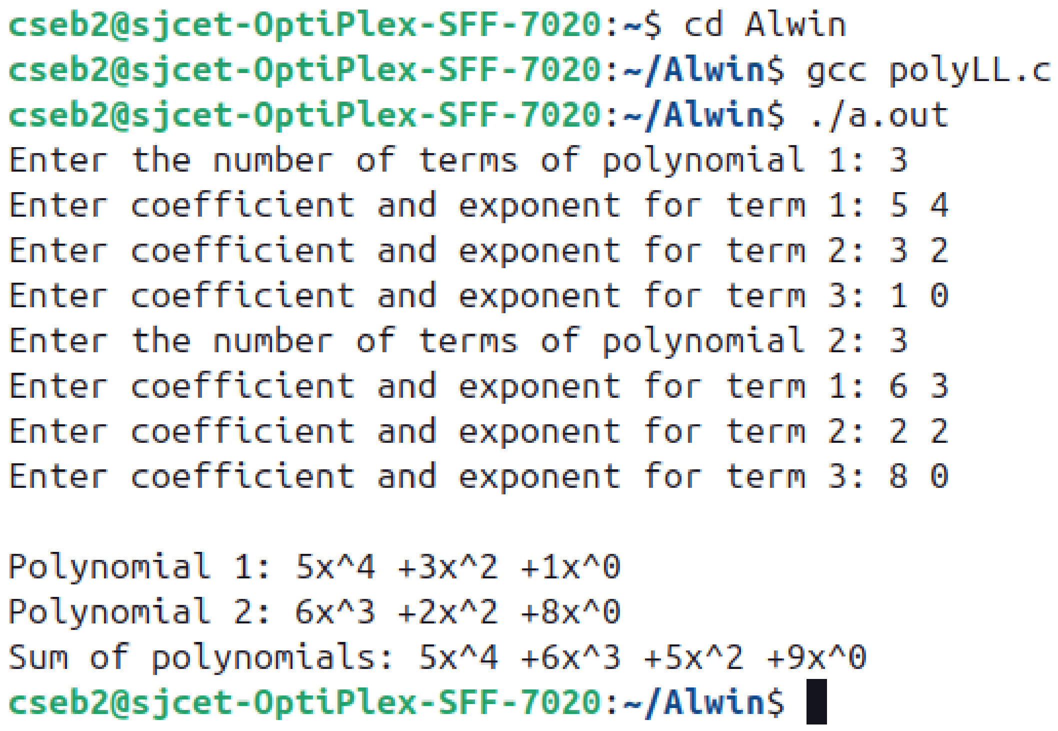
EXPERIMENT-7

Program :

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

struct Ptr {
    int coeff;
    int exp;
    struct Ptr* link;
};

struct Ptr* createNode(int coeff, int exp) {
    struct Ptr* newNode = (struct Ptr*)malloc(sizeof(struct Ptr));
    newNode->coeff = coeff;
    newNode->exp = exp;
    newNode->link = NULL;
    return newNode;
}

void insertTerm(struct Ptr** poly, int coeff, int exp) {
    struct Ptr* newNode = createNode(coeff, exp);
    if (*poly == NULL) {
        *poly = newNode;
    } else {
        struct Ptr* temp = *poly;
        struct Ptr* prev = NULL;
        while (temp != NULL && temp->exp > exp) {
            prev = temp;
            temp = temp->link;
        }
        if (temp != NULL && temp->exp == exp) {
            temp->coeff += coeff;
            free(newNode);
        } else {
            if (prev == NULL) {
```

```c
            if (prev == NULL) {
                *poly = newNode;
            } else {
                prev->link = newNode;
            }
            newNode->link = temp;
        }
    }
}

void printPolynomial(struct Ptr* poly) {
    if (poly == NULL) {
        printf("0\n");
        return;
    }

    struct Ptr* temp = poly;
    while (temp != NULL) {
        if (temp->coeff > 0 && temp != poly) {
            printf("+");
        }
        printf("%dx^%d ", temp->coeff, temp->exp);
        temp = temp->link;
    }
    printf("\n");
}

struct Ptr* addPolynomials(struct Ptr* poly1, struct Ptr* poly2) {
    struct Ptr* result = NULL;
    struct Ptr *ptr1 = poly1, *ptr2 = poly2;

    while (ptr1 != NULL) {
        insertTerm(&result, ptr1->coeff, ptr1->exp);
        ptr1 = ptr1->link;
    }

    while (ptr2 != NULL) {
        insertTerm(&result, ptr2->coeff, ptr2->exp);
        insertTerm(&result, ptr2->coeff, ptr2->exp);
        ptr2 = ptr2->link;
    }

    return result;
}

int main() {
    struct Ptr *poly1 = NULL, *poly2 = NULL, *poly3 = NULL;
    int n1, n2, coeff, exp, i;

    printf("Enter the number of terms of polynomial 1: ");
    scanf("%d", &n1);
    for (i = 0; i < n1; i++) {
        printf("Enter coefficient and exponent for term %d: ", i + 1);
        scanf("%d %d", &coeff, &exp);
        insertTerm(&poly1, coeff, exp);
    }

    printf("Enter the number of terms of polynomial 2: ");
    scanf("%d", &n2);
    for (i = 0; i < n2; i++) {
        printf("Enter coefficient and exponent for term %d: ", i + 1);
        scanf("%d %d", &coeff, &exp);
        insertTerm(&poly2, coeff, exp);
    }
    poly3 = addPolynomials(poly1, poly2);

    printf("\nPolynomial 1: ");
    printPolynomial(poly1);
    printf("Polynomial 2: ");
    printPolynomial(poly2);
    printf("Sum of polynomials: ");
    printPolynomial(poly3);

    return 0;
}
```

Output :



```
cseb2@sjcet-OptiPlex-SFF-7020:~$ cd Alwin
cseb2@sjcet-OptiPlex-SFF-7020:~/Alwin$ gcc polyLL.c
cseb2@sjcet-OptiPlex-SFF-7020:~/Alwin$ ./a.out
Enter the number of terms of polynomial 1: 3
Enter coefficient and exponent for term 1: 5 4
Enter coefficient and exponent for term 2: 3 2
Enter coefficient and exponent for term 3: 1 0
Enter the number of terms of polynomial 2: 3
Enter coefficient and exponent for term 1: 6 3
Enter coefficient and exponent for term 2: 2 2
Enter coefficient and exponent for term 3: 8 0

Polynomial 1: 5x^4 +3x^2 +1x^0
Polynomial 2: 6x^3 +2x^2 +8x^0
Sum of polynomials: 5x^4 +6x^3 +5x^2 +9x^0
cseb2@sjcet-OptiPlex-SFF-7020:~/Alwin$ 
```