231501058 CS23231 – D a t a S t r u c t u r e s

**Ex. No.: 3**

**Polynomial Manipulation**

**Date: 8/3/24**

**Write a C prog ram to imple men t the following operations on Singly Linked List .**

1. **Polynomial Addition**
2. **Polynomial Subtraction**
3. **Polynomial Multiplication**

**Algorithm:**

#include <stdio.h>

#include <malloc.h>

struct node {

int coeff;

int exp;

struct node\* next;

struct node\* prev;

};

struct node\* head1 = NULL;

struct node\* head2 = NULL;

struct node\* headResult = NULL;

void insertTerm(struct node\*\* head, int coeff, int exp) {

struct node\* newnode = (struct node\*)malloc(sizeof(struct node)); if (newnode != NULL) {

newnode->coeff = coeff;

newnode->exp = exp;

newnode->next = NULL;

newnode->prev = NULL;

if (\*head == NULL) {

\*head = newnode;

} else {

struct node\* t = \*head;

while (t->next != NULL) {

1. = t->next;

}

t->next = newnode;

newnode->prev = t;

}

}

}

void display(struct node\* head) {

struct node\* t = head;

while (t != NULL) {

if (t->coeff > 0 && t != head)



**Dept of Artificial Intelligence and Machine Learning** | **Rajalakshmi Engineering College**

. 17

231501058

CS23231 – D a t a S t r u c t u r e s

printf("+ ");

printf("%dx^%d ", t->coeff, t->exp);

1. = t->next;

}

printf("\n");

}

struct node\* addPolynomials(struct node\* head1, struct node\* head2) { struct node\* result = NULL;

struct node\* t1 = head1;

struct node\* t2 = head2;

while (t1 != NULL && t2 != NULL) {

if (t1->exp == t2->exp) {

insertTerm(&result, t1->coeff + t2->coeff, t1->exp);

t1 = t1->next;

t2 = t2->next;

} else if (t1->exp > t2->exp) {

insertTerm(&result, t1->coeff, t1->exp);

t1 = t1->next;

} else {

insertTerm(&result, t2->coeff, t2->exp);

t2 = t2->next;

}

}

while (t1 != NULL) {

insertTerm(&result, t1->coeff, t1->exp);

t1 = t1->next;

}

while (t2 != NULL) {

insertTerm(&result, t2->coeff, t2->exp);

t2 = t2->next;

}

return result;

}

struct node\* multiplyPolynomials(struct node\* head1, struct node\* head2) { struct node\* result = NULL;

struct node\* t1 = head1;

struct node\* t2 = head2;

|  |  |
| --- | --- |
| while (t1 != NULL) { |  |
| t2 = head2; |  |
| while (t2 != NULL) { |  |
| insertTerm(&result, t1->coeff \* t2->coeff, t1->exp + t2->exp); |  |
| t2 = t2->next; |  |
| } |  |
| **Dept of Artificial Intelligence and Machine Learning** | **Rajalakshmi Engineering College** | . 18 |



231501058 CS23231 – D a t a S t r u c t u r e s

t1 = t1->next;

}

struct node\* t = result;

struct node\* tPrev = NULL;

while (t != NULL && t->next != NULL) {

tPrev = t;

struct node\* tNext = t->next;

while (tNext != NULL) {

if (t->exp == tNext->exp) {

t->coeff += tNext->coeff;

tPrev->next = tNext->next;

if (tNext->next != NULL) {

tNext->next->prev = tPrev;

}

free(tNext);

tNext = tPrev->next;

} else {

tPrev = tNext;

tNext = tNext->next;

}

}

1. = t->next;

}

return result;

}

int main() {

insertTerm(&head1, 5, 2);

insertTerm(&head1, 4, 1);

insertTerm(&head1, 2, 0);

insertTerm(&head2, 5, 1);

insertTerm(&head2, 5, 0);

printf("Polynomial 1: ");

display(head1);

printf("Polynomial 2: ");

display(head2);

headResult = addPolynomials(head1, head2);

printf("\nAddition Result: ");

display(headResult);

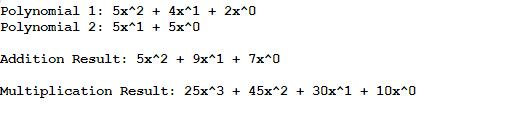
headResult = multiplyPolynomials(head1, head2); printf("\nMultiplication Result: "); display(headResult);

|  |  |
| --- | --- |
| return 0;} |  |
| **Dept of Artificial Intelligence and Machine Learning** | **Rajalakshmi Engineering College** | . 19 |



231501058 CS23231 – D a t a S t r u c t u r e s

**OUTPUT**



**Dept of Artificial Intelligence and Machine Learning** | **Rajalakshmi Engineering College**

. 20