**Write a C program to implement the following operations on Singly Linked List.**

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

**Algorithm:**

**Code:**

**Polynomial addition:**

**#include <stdio.h>**

**#include <stdlib.h>**

**struct poly**

**{**

**int coeff;**

**int pow;**

**struct poly \*Next;**

**};**

**typedef struct poly Poly;**

**void Create(Poly \*List);**

**void Display(Poly \*List);**

**void Addition(Poly \*Poly1, Poly \*Poly2, Poly \*Result);**

**int main()**

**{**

**Poly \*Poly1 = malloc(sizeof(Poly));**

**Poly \*Poly2 = malloc(sizeof(Poly));**

**Poly \*Result = malloc(sizeof(Poly));**

**Poly1->Next = NULL;**

**Poly2->Next = NULL;**

**printf("Enter the values for first polynomial :\n");**

**Create(Poly1);**

**printf("The polynomial equation is : ");**

**Display(Poly1);**

**printf("\nEnter the values for second polynomial :\n");**

**Create(Poly2);**

**printf("The polynomial equation is : ");**

**Display(Poly2);**

**Addition(Poly1, Poly2, Result);**

**printf("\nThe polynomial equation addition result is : ");**

**Display(Result);**

**return 0;**

**}**

**void Create(Poly \*List)**

**{**

**int choice;**

**Poly \*Position, \*NewNode;**

**Position = List;**

**do**

**{**

**NewNode = malloc(sizeof(Poly));**

**printf("Enter the coefficient : ");**

**scanf("%d", &NewNode->coeff);**

**printf("Enter the power : ");**

**scanf("%d", &NewNode->pow);**

**NewNode->Next = NULL;**

**Position->Next = NewNode;**

**Position = NewNode;**

**printf("Enter 1 to continue : ");**

**scanf("%d", &choice);**

**} while(choice == 1);**

**}**

**void Display(Poly \*List)**

**{**

**Poly \*Position;**

**Position = List->Next;**

**while(Position != NULL)**

**{**

**printf("%dx^%d", Position->coeff, Position->pow);**

**Position = Position->Next;**

**if(Position != NULL && Position->coeff > 0)**

**{**

**printf("+");**

**}**

**}**

**}**

**void Addition(Poly \*Poly1, Poly \*Poly2, Poly \*Result)**

**{**

**Poly \*Position;**

**Poly \*NewNode;**

**Poly1 = Poly1->Next;**

**Poly2 = Poly2->Next;**

**Result->Next = NULL;**

**Position = Result;**

**while(Poly1 != NULL && Poly2 != NULL)**

**{**

**NewNode = malloc(sizeof(Poly));**

**if(Poly1->pow == Poly2->pow)**

**{**

**NewNode->coeff = Poly1->coeff + Poly2->coeff;**

**NewNode->pow = Poly1->pow;**

**Poly1 = Poly1->Next;**

**Poly2 = Poly2->Next;**

**}**

**else if(Poly1->pow > Poly2->pow)**

**{**

**NewNode->coeff = Poly1->coeff;**

**NewNode->pow = Poly1->pow;**

**Poly1 = Poly1->Next;**

**}**

**else if(Poly1->pow < Poly2->pow)**

**{**

**NewNode->coeff = Poly2->coeff;**

**NewNode->pow = Poly2->pow;**

**Poly2 = Poly2->Next;**

**}**

**NewNode->Next = NULL;**

**Position->Next = NewNode;**

**Position = NewNode;**

**}**

**while(Poly1 != NULL || Poly2 != NULL)**

**{**

**NewNode = malloc(sizeof(Poly));**

**if(Poly1 != NULL)**

**{**

**NewNode->coeff = Poly1->coeff;**

**NewNode->pow = Poly1->pow;**

**Poly1 = Poly1->Next;**

**}**

**if(Poly2 != NULL)**

**{**

**NewNode->coeff = Poly2->coeff;**

**NewNode->pow = Poly2->pow;**

**Poly2 = Poly2->Next;**

**}**

**NewNode->Next = NULL;**

**Position->Next = NewNode;**

**Position = NewNode;**

**}**

**}**

**Polynomial Subtraction:**

**#include <stdio.h>**

**#include <stdlib.h>**

**struct poly**

**{**

**int coeff;**

**int pow;**

**struct poly \*Next;**

**};**

**typedef struct poly Poly;**

**void Create(Poly \*List);**

**void Display(Poly \*List);**

**void Subtraction(Poly \*Poly1, Poly \*Poly2, Poly \*Result);**

**int main()**

**{**

**Poly \*Poly1 = malloc(sizeof(Poly));**

**Poly \*Poly2 = malloc(sizeof(Poly));**

**Poly \*Result = malloc(sizeof(Poly));**

**Poly1->Next = NULL;**

**Poly2->Next = NULL;**

**printf("Enter the values for first polynomial :\n");**

**Create(Poly1);**

**printf("The polynomial equation is : ");**

**Display(Poly1);**

**printf("\nEnter the values for second polynomial :\n");**

**Create(Poly2);**

**printf("The polynomial equation is : ");**

**Display(Poly2);**

**Subtraction(Poly1, Poly2, Result);**

**printf("\nThe polynomial equation subtraction result is : ");**

**Display(Result);**

**return 0;**

**}**

**void Create(Poly \*List)**

**{**

**int choice;**

**Poly \*Position, \*NewNode;**

**Position = List;**

**do**

**{**

**NewNode = malloc(sizeof(Poly));**

**printf("Enter the coefficient : ");**

**scanf("%d", &NewNode->coeff);**

**printf("Enter the power : ");**

**scanf("%d", &NewNode->pow);**

**NewNode->Next = NULL;**

**Position->Next = NewNode;**

**Position = NewNode;**

**printf("Enter 1 to continue : ");**

**scanf("%d", &choice);**

**} while(choice == 1);**

**}**

**void Display(Poly \*List)**

**{**

**Poly \*Position;**

**Position = List->Next;**

**while(Position != NULL)**

**{**

**printf("%dx^%d", Position->coeff, Position->pow);**

**Position = Position->Next;**

**if(Position != NULL && Position->coeff > 0)**

**{**

**printf("+");**

**}**

**}**

**}**

**void Subtraction(Poly \*Poly1, Poly \*Poly2, Poly \*Result)**

**{**

**Poly \*Position;**

**Poly \*NewNode;**

**Poly1 = Poly1->Next;**

**Poly2 = Poly2->Next;**

**Result->Next = NULL;**

**Position = Result;**

**while(Poly1 != NULL && Poly2 != NULL)**

**{**

**NewNode = malloc(sizeof(Poly));**

**if(Poly1->pow == Poly2->pow)**

**{**

**NewNode->coeff = Poly1->coeff - Poly2->coeff;**

**NewNode->pow = Poly1->pow;**

**Poly1 = Poly1->Next;**

**Poly2 = Poly2->Next;**

**}**

**else if(Poly1->pow > Poly2->pow)**

**{**

**NewNode->coeff = Poly1->coeff;**

**NewNode->pow = Poly1->pow;**

**Poly1 = Poly1->Next;**

**}**

**else if(Poly1->pow < Poly2->pow)**

**{**

**NewNode->coeff = -(Poly2->coeff);**

**NewNode->pow = Poly2->pow;**

**Poly2 = Poly2->Next;**

**}**

**NewNode->Next = NULL;**

**Position->Next = NewNode;**

**Position = NewNode;**

**}**

**while(Poly1 != NULL || Poly2 != NULL)**

**{**

**NewNode = malloc(sizeof(Poly));**

**if(Poly1 != NULL)**

**{**

**NewNode->coeff = Poly1->coeff;**

**NewNode->pow = Poly1->pow;**

**Poly1 = Poly1->Next;**

**}**

**if(Poly2 != NULL)**

**{**

**NewNode->coeff = -(Poly2->coeff);**

**NewNode->pow = Poly2->pow;**

**Poly2 = Poly2->Next;**

**}**

**NewNode->Next = NULL;**

**Position->Next = NewNode;**

**Position = NewNode;**

**}**

**}**

**Polynomial Multiplication:**

**#include <stdio.h>**

**#include <stdlib.h>**

**typedef struct Node {**

**int data;**

**int power;**

**struct Node \* next;**

**} Node;**

**Node \* getNode(int data, int power) {**

**Node \* ref = (Node \* ) malloc(sizeof(Node));**

**if (ref == NULL) {**

**return NULL;**

**}**

**ref->data = data;**

**ref->power = power;**

**ref->next = NULL;**

**return ref;**

**}**

**void updateRecord(Node \* ref, int data, int power) {**

**ref->data = data;**

**ref->power = power;**

**}**

**typedef struct MultiplyPolynomial {**

**struct Node \* head;**

**} MultiplyPolynomial;**

**MultiplyPolynomial \* getMultiplyPolynomial() {**

**MultiplyPolynomial \* ref = (MultiplyPolynomial \* ) malloc(sizeof(MultiplyPolynomial));**

**if (ref == NULL) {**

**return NULL;**

**}**

**ref->head = NULL;**

**return ref;**

**}**

**void insert(MultiplyPolynomial \* ref, int data, int power) {**

**if (ref->head == NULL) {**

**ref->head = getNode(data, power);**

**} else {**

**Node \* node = NULL;**

**Node \* temp = ref->head;**

**Node \* location = NULL;**

**while (temp != NULL && temp->power >= power) {**

**location = temp;**

**temp = temp->next;**

**}**

**if (location != NULL && location->power == power) {**

**location->data = location->data + data;**

**} else {**

**node = getNode(data, power);**

**if (location == NULL) {**

**node->next = ref->head;**

**ref->head = node;**

**} else {**

**node->next = location->next;**

**location->next = node;**

**}**

**}**

**}**

**}**

**MultiplyPolynomial \* multiplyPolynomials(MultiplyPolynomial \* ref, MultiplyPolynomial \* other) {**

**MultiplyPolynomial \* result = getMultiplyPolynomial();**

**Node \* poly1 = ref->head;**

**Node \* temp = other->head;**

**int power\_value = 0;**

**int coefficient = 0;**

**while (poly1 != NULL) {**

**temp = other->head;**

**while (temp != NULL) {**

**power\_value = poly1->power + temp->power;**

**coefficient = poly1->data \* temp->data;**

**insert(result, coefficient, power\_value);**

**temp = temp->next;**

**}**

**poly1 = poly1->next;**

**}**

**return result;**

**}**

**void display(MultiplyPolynomial \* ref) {**

**if (ref->head == NULL) {**

**printf("Empty Polynomial ");**

**}**

**printf(" ");**

**Node \* temp = ref->head;**

**while (temp != NULL) {**

**if (temp != ref->head) {**

**printf(" + %d", temp->data);**

**} else {**

**printf("%d", temp->data);**

**}**

**if (temp->power != 0) {**

**printf("x^%d", temp->power);**

**}**

**temp = temp->next;**

**}**

**printf("\n");**

**}**

**int main() {**

**MultiplyPolynomial \* a = getMultiplyPolynomial();**

**MultiplyPolynomial \* b = getMultiplyPolynomial();**

**insert(a, 9, 3);**

**insert(a, 4, 2);**

**insert(a, 3, 0);**

**insert(a, 7, 1);**

**insert(a, 3, 4);**

**insert(b, 7, 3);**

**insert(b, 4, 0);**

**insert(b, 6, 1);**

**insert(b, 1, 2);**

**printf("\n Polynomial A\n");**

**display(a);**

**printf(" Polynomial B\n");**

**display(b);**

**MultiplyPolynomial \* result = multiplyPolynomials(a, b);**

**printf(" Result\n");**

**display(result);**

**}**

**OUTPUT**

**Polynomial A**

**3x^4 + 9x^3 + 4x^2 + 7x^1 + 3**

**Polynomial B**

**7x^3 + 1x^2 + 6x^1 + 4**

**Result**

**21x^7 + 39x^6 + 67x^5 + 91x^4 + 55x^3 + 52x^2 + 46x^1 + 12**