Q1. Write a C program to add 2 polynomials which are represented using linked list and store the result in the resultant linked list.

CODE :

#include <stdio.h>

#include<stdlib.h>

struct poly

{

    float coef;

    int expo;

    struct poly \*next;

};

struct poly \*create(struct poly \*head);

struct poly \*insert(struct poly \*head, float coef, int expo);

void print(struct poly \*head);

void polyAdd(struct poly \*head1 , struct poly\*head2);

int main()

{

    struct poly \*head1 = NULL;

    struct poly \*head2 = NULL;

    struct poly \*head3 = NULL;

    printf("Enter 1st polynomial :\n");

    head1 = create(head1);

    printf("Enter 2nd polynomial :\n");

    head2 = create(head2);

    printf("\n\n1st polynomial :\n");

    print(head1);

    printf("2nd polynomial : \n");

    print(head2);

      polyAdd(head1,head2);

    return 0;

}

struct poly \*create(struct poly \*head)

{

    int n;

    float coef;

    int expo;

    printf("Enter the number of terms in the polynomial:");

    scanf("%d", &n);

    for (int i = 0; i < n; i++)

    {

        printf("Enter the coefficiant of the term %d : ", i + 1);

        scanf("%f", &coef);

        printf("Enter the exponent of the term %d : ", i + 1);

        scanf("%d", &expo);

        head = insert(head, coef, expo);

    }

    return head;

}

struct poly \*insert(struct poly \*head, float coef, int expo)

{

    struct poly \*ptr = malloc(sizeof(struct poly));

    ptr->coef = coef;

    ptr->expo = expo;

    ptr->next = NULL;

    if (head == NULL || head->expo < expo)

    {

        ptr->next = head;

        head = ptr;

    }

    else

    {

        struct poly \*temp = head;

        while (temp->next != NULL && temp->next->expo > expo)

        {

            temp = temp->next;

        }

        ptr->next = temp->next;

        temp->next = ptr;

        return head;

    }

}

void print(struct poly \* head)

{

    struct poly \* ptr = head ;

    while(ptr != NULL)

    {

        printf("%.f,%d,%p\n",ptr->coef,ptr->expo,ptr->next);

        ptr = ptr->next;

    }

}

void polyAdd(struct poly \*head1 , struct poly \*head2)

{

struct poly \*ptr1 = head1 ;

struct poly \*ptr2 = head2 ;

struct poly \*head3 = NULL ;

while(ptr1 != NULL && ptr2!= NULL)

{

    if(ptr1->expo == ptr2->expo)

    {

       head3 =  insert(head3,ptr1->coef + ptr2->coef,ptr1->expo);

        ptr1 = ptr1->next;

        ptr2 = ptr2->next;

    }

    else if(ptr1->expo> ptr2->expo)

    {

        head3 = insert(head3,ptr1->coef,ptr1->expo );

        ptr1 = ptr1->next;

    }

    else if(ptr2->expo> ptr1->expo)

    {

        head3 = insert(head3,ptr2->coef,ptr2->expo );

        ptr2 = ptr2->next;

    }

}

while(ptr1 != NULL)

{

    head3 = insert(head3,ptr1->coef,ptr1->expo);

    ptr1 = ptr1->next;

}

while(ptr2 != NULL)

{

    head3 = insert(head3,ptr2->coef,ptr2->expo);

    ptr2 = ptr2->next;

}

printf("The addition of the polynomials is : \n");

print(head3);

}

OUTPUT :

Enter 1st polynomial :

Enter the number of terms in the polynomial:3

Enter the coefficiant of the term 1 : 5

Enter the exponent of the term 1 : 4

Enter the coefficiant of the term 2 : 3

Enter the exponent of the term 2 : 2

Enter the coefficiant of the term 3 : 2

Enter the exponent of the term 3 : 1

Enter 2nd polynomial :

Enter the number of terms in the polynomial:3

Enter the coefficiant of the term 1 : 4

Enter the exponent of the term 1 : 4

Enter the coefficiant of the term 2 : 2

Enter the exponent of the term 2 : 2

Enter the coefficiant of the term 3 : 1

Enter the exponent of the term 3 : 0

1st polynomial :

5,4,00981848

3,2,00981860

2,1,00000000

2nd polynomial :

4,4,009815F0

2,2,00981608

1,0,00000000

The addition of the polynomials is :

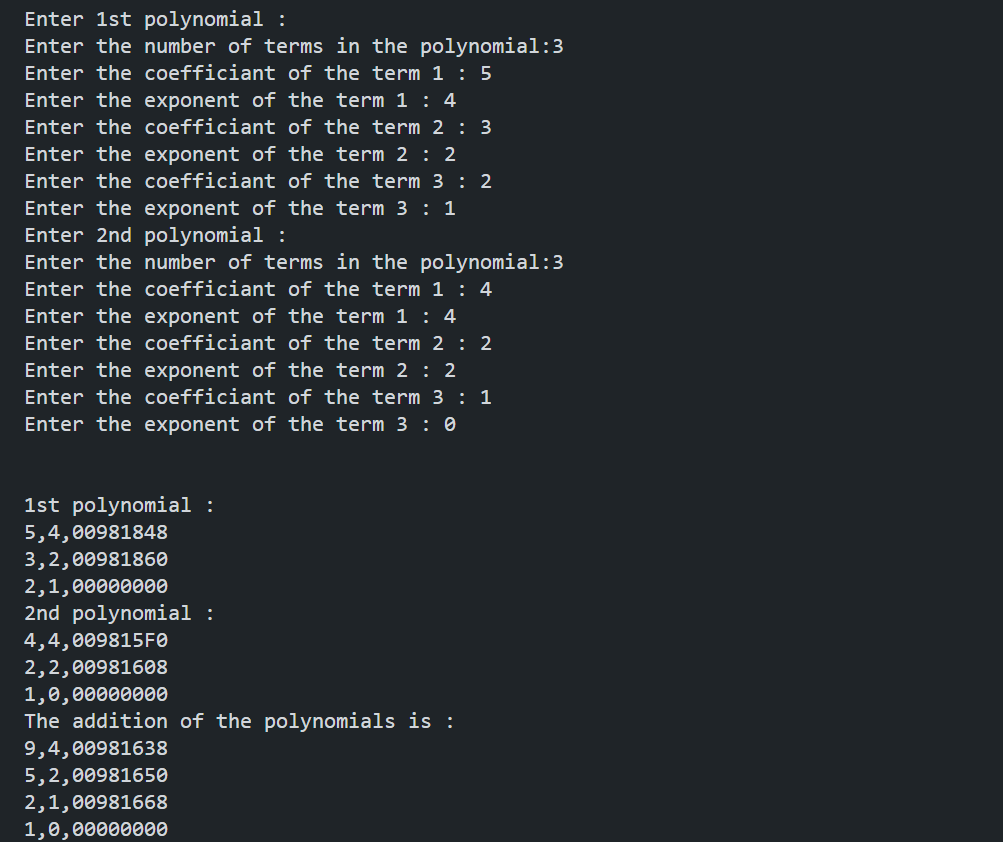
9,4,00981638

5,2,00981650

2,1,00981668

1,0,00000000

SCREENSHOT :



Q2. Write a C program to add two polynomials where for the first polynomial is required to be represented using a linked list. For the 2nd polynomial instead of creating a new linked list, add coefficients as soon as you get it as an input from the user. In this entire process you will create only one linked list for two polynomials and for result also.

CODE :

#include <stdio.h>

#include <stdlib.h>

struct node

{

    float coef;

    int expo;

    struct node \*next;

};

struct node \*create(struct node \*head);

struct node \*insert(struct node \*head, float coef, int expo);

struct node \*Second(struct node \*head);

struct node \*Add(struct node \*head, float coef, int expo);

void display(struct node \*head);

int main()

{

    struct node \*head = NULL;

    head = create(head);

    head = Second(head);

    display(head);

    return 0;

}

struct node \*create(struct node \*head)

{

    int n;

    float coef;

    int expo;

    printf("Enter the Number of terms in the 1st polynomial :\n");

    scanf("%d", &n);

    for (int i = 0; i < n; i++)

    {

        printf("Enter the coefficiant of the term %d :\n ", i + 1);

        scanf("%f", &coef);

        printf("Enter the exponent of the term %d :\n ", i + 1);

        scanf("%d", &expo);

        head = insert(head, coef, expo);

    }

    return head;

}

struct node \*insert(struct node \*head, float coef, int expo)

{

    struct node \*ptr = (struct node \*)malloc(sizeof(struct node));

    struct node \*temp = head;

    ptr->coef = coef;

    ptr->expo = expo;

    ptr->next = NULL;

    if (head == NULL || head->expo < expo)

    {

        ptr->next = head;

        head = ptr;

    }

    else

    {

        while (temp->next != NULL && temp->next->expo > expo)

        {

            temp = temp->next;

        }

        ptr->next = temp->next;

        temp->next = ptr;

    }

    return head;

}

struct node \*Second(struct node \*head)

{

    int n;

    float coef;

    int expo;

    printf("Enter the Second polynomial to be added :\n");

    printf("Enter the number of terms in the 2nd :\n ");

    scanf("%d", &n);

    for (int i = 0; i < n; i++)

    {

        printf("Enter the coefficiant of the  term %d :\n", i + 1);

        scanf("%f", &coef);

        printf("Enter the exponent  of the term %d :\n", i + 1);

        scanf("%d", &expo);

        head = Add(head, coef, expo);

    }

    return head;

}

struct node \*Add(struct node \*head, float coef, int expo)

{

    struct node \*temp = head;

    if (head->expo < expo)

    {

        struct node \*new = (struct node \*)malloc(sizeof(struct node));

        new->coef = coef;

        new->expo = expo;

        new->next = head;

        head = new;

        return head;

    }

    else

    {

        while (temp != NULL)

        {

            if (temp->expo == expo)

            {

                temp->coef = temp->coef + coef;

                return head;

            }

            temp = temp->next;

        }

        temp = head;

        struct node \*new = (struct node \*)malloc(sizeof(struct node));

        new->coef = coef;

        new->expo = expo;

        new->next = NULL;

        while (temp->next != NULL && temp->next->expo > expo)

        {

            temp = temp->next;

        }

        new->next = temp->next;

        temp->next = new;

    }

    return head;

}

void display(struct node \*head)

{

    struct node \*temp = head;

    printf("The Final polynomial after Addition is : \n");

    while (temp != NULL)

    {

        printf(" %.1fx^%d", temp->coef, temp->expo);

        temp = temp->next;

        if (temp != NULL)

        {

            printf(" + ");

        }

    }

}

OUTPUT :

Enter the Number of terms in the 1st polynomial :

3

Enter the coefficiant of the term 1 :

5

Enter the exponent of the term 1 :

3

Enter the coefficiant of the term 2 :

4

Enter the exponent of the term 2 :

2

Enter the coefficiant of the term 3 :

2

Enter the exponent of the term 3 :

1

Enter the Second polynomial to be added :

Enter the number of terms in the 2nd :

3

Enter the coefficiant of the term 1 :

5

Enter the exponent of the term 1 :

3

Enter the coefficiant of the term 2 :

9

Enter the exponent of the term 2 :

9

Enter the coefficiant of the term 3 :

2

Enter the exponent of the term 3 :

0

The Final polynomial after Addition is :

9.0x^9 + 10.0x^3 + 4.0x^2 + 2.0x^1 + 2.0x^0

SCREENSHOT :

