DSA LAB ASSINGNMENT 1

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

ANS. #include <stdio.h>

#include <stdlib.h>

struct node {

float coeff;

int exp;

struct node \*next;

};

struct node \*createnode(float coeff, int exp) {

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

if (!temp) {

printf("Memory allocation failed\n");

return NULL;

}

temp->coeff = coeff;

temp->exp = exp;

temp->next = NULL;

return temp;

}

struct node \*createpl(int num) {

if (num <= 0) return NULL;

struct node \*head = NULL;

struct node \*temp = NULL;

for (int i = 0; i < num; i++) {

printf("Enter the exponent value: ");

int exp;

scanf("%d", &exp);

printf("Enter the coefficient value: ");

float coeff;

scanf("%f", &coeff);

struct node \*newnode = createnode(coeff, exp);

if (newnode == NULL) {

printf("Error in creating node\n");

return head;

}

if (head == NULL) {

head = newnode;

} else {

temp->next = newnode;

}

temp = newnode;

}

return head;

}

void print(struct node \*head) {

struct node \*temp = head;

while (temp != NULL) {

printf("%0.1fx^%d", temp->coeff, temp->exp);

temp = temp->next;

if (temp != NULL) printf(" + ");

}

printf("\n");

}

struct node \*add(struct node \*h1, struct node \*h2) {

struct node \*t1 = h1;

struct node \*t2 = h2;

struct node \*result = NULL;

struct node \*\*lastPtrRef = &result;

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

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

\*lastPtrRef = createnode(t1->coeff + t2->coeff, t1->exp);

t1 = t1->next;

t2 = t2->next;

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

\*lastPtrRef = createnode(t1->coeff, t1->exp);

t1 = t1->next;

} else {

\*lastPtrRef = createnode(t2->coeff, t2->exp);

t2 = t2->next;

}

lastPtrRef = &((\*lastPtrRef)->next);

}

while (t1 != NULL) {

\*lastPtrRef = createnode(t1->coeff, t1->exp);

lastPtrRef = &((\*lastPtrRef)->next);

t1 = t1->next;

}

while (t2 != NULL) {

\*lastPtrRef = createnode(t2->coeff, t2->exp);

lastPtrRef = &((\*lastPtrRef)->next);

t2 = t2->next;

}

return result;

}

int main() {

int num1, num2;

printf("Enter the number of terms in 1st polynomial: ");

scanf("%d", &num1);

struct node \*pl1 = createpl(num1);

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

scanf("%d", &num2);

struct node \*pl2 = createpl(num2);

printf("Polynomial 1: ");

print(pl1);

printf("Polynomial 2: ");

print(pl2);

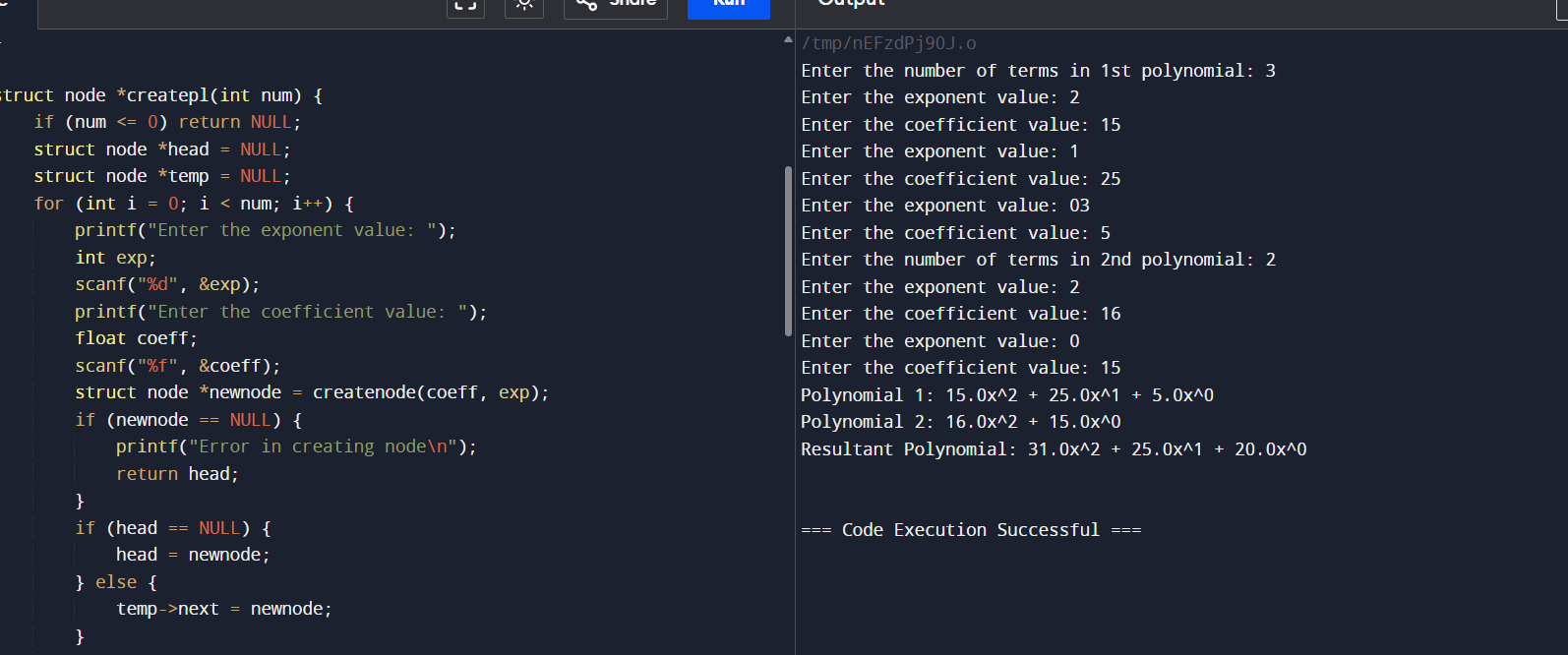
struct node \*result = add(pl1, pl2);

printf("Resultant Polynomial: ");

print(result);

return 0;

}



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.

ANS. #include <stdio.h>

#include <stdlib.h>

struct node {

float coeff;

int exp;

struct node \*next;

};

struct node \*createnode(float coeff, int exp) {

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

temp->coeff = coeff;

temp->exp = exp;

temp->next = NULL;

return temp;

}

void insert\_term(struct node \*\*head, float coeff, int exp) {

struct node \*newnode = createnode(coeff, exp);

if (\*head == NULL || (\*head)->exp < exp) {

newnode->next = \*head;

\*head = newnode;

} else {

struct node \*temp = \*head;

while (temp->next != NULL && temp->next->exp >= exp) {

temp = temp->next;

}

if (temp->exp == exp) {

temp->coeff += coeff;

free(newnode);

} else {

newnode->next = temp->next;

temp->next = newnode;

}

}

}

void print(struct node \*head) {

struct node \*temp = head;

while (temp != NULL) {

printf("%0.1fx^%d", temp->coeff, temp->exp);

temp = temp->next;

if (temp != NULL) printf(" + ");

}

printf("\n");

}

int main() {

struct node \*poly = NULL;

int num1, num2;

float coeff;

int exp;

printf("Enter the number of terms in 1st polynomial: ");

scanf("%d", &num1);

for (int i = 0; i < num1; i++) {

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

scanf("%f %d", &coeff, &exp);

insert\_term(&poly, coeff, exp);

}

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

scanf("%d", &num2);

for (int i = 0; i < num2; i++) {

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

scanf("%f %d", &coeff, &exp);

insert\_term(&poly, coeff, exp);

}

printf("Resultant Polynomial: ");

print(poly);

return 0;

}

