Q. Add two polynomials

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

struct poly {

int coeff;

int deg;

struct poly \*next;

};

int main(){

int n=0,c=0,d=0;

struct poly \*polynomial1 = (struct poly\*)malloc(sizeof(struct poly));

polynomial1->coeff = 0;

polynomial1->deg = 0;

polynomial1->next=NULL;

struct poly \*temp=polynomial1;

printf("Enter the first polynomial, specifying the coefficient and degree respectively.\n");

printf("Number of terms\n");

scanf("%d", &n);

int p=0;

while(p<n){

printf("\nCoefficient:");

scanf("%d", &c);

printf("Degree:");

scanf("%d", &d);

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

temp->next=newterm;

newterm->coeff = c;

newterm->deg = d;

newterm->next=NULL;

temp=newterm;

p++;

}

n=0;

temp=polynomial1->next;

struct poly \* polynomial2 = (struct poly\*)malloc(sizeof(struct poly));

polynomial2->coeff = 0;

polynomial2->deg = 0;

polynomial2->next=NULL;

struct poly \*temp2 = polynomial2;

printf("\n\nEnter the second polynomial\n");

printf("Number of terms\n");

scanf("%d", &n);

int q=0;

while(q<n){

printf("\nCoefficient:");

scanf("%d", &c);

printf("Degree:");

scanf("%d", &d);

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

temp2->next=newterm;

newterm->coeff = c;

newterm->deg = d;

newterm->next=NULL;

temp2=newterm;

q++;

}

temp2=polynomial2->next;

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

result->coeff = 0;

result->deg=0;

result->next=NULL;

struct poly \* temp3 = result;

while(temp!=NULL && temp2!=NULL){

if(temp->deg == temp2->deg){

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

temp3->next = resultterm;

resultterm->coeff = (temp->coeff) + (temp2->coeff);

resultterm->deg = temp->deg;

resultterm->next=NULL;

temp3=resultterm;

temp=temp->next;

temp2=temp2->next;

}

else{

if(temp->deg > temp2->deg){

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

temp3->next = resultterm;

resultterm->coeff = (temp->coeff);

resultterm->deg = (temp->deg);

temp3=resultterm;

temp=temp->next;

}

if(temp->deg < temp2->deg){

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

temp3->next = resultterm;

resultterm->coeff = temp2->coeff;

resultterm->deg = temp2 ->deg;

temp3=resultterm;

temp2=temp2->next;

}

}

}

temp3 = result->next;

int i=0;

printf("Resultant polynomial:\n");

while(temp3!=NULL){

i++;

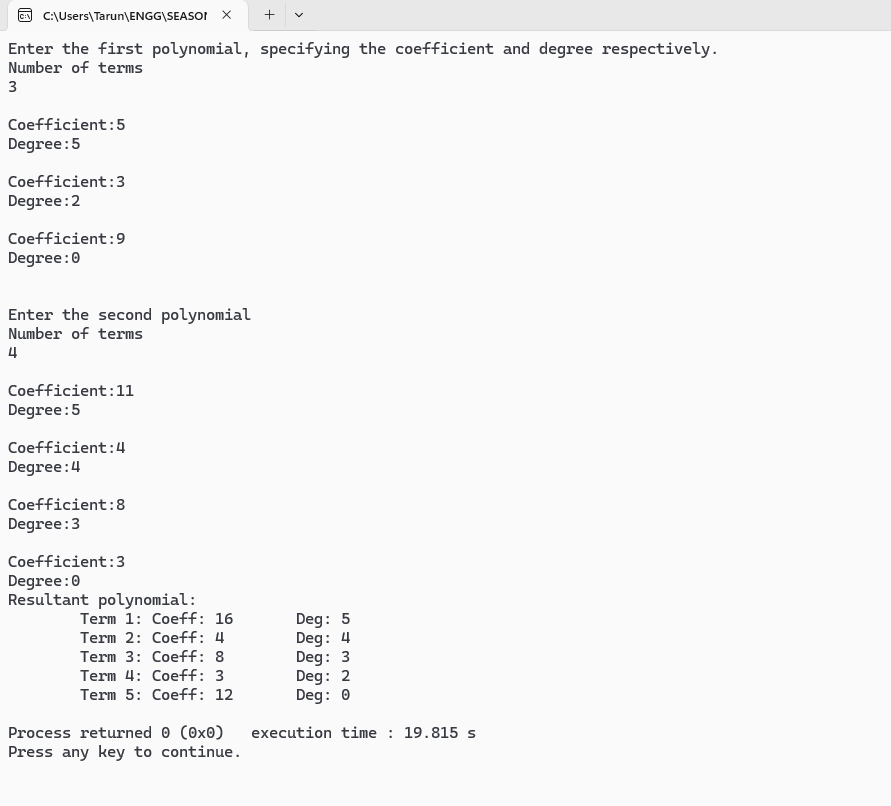
printf("\tTerm %d:\tCoeff: %d\tDeg: %d\n",i, temp3->coeff, temp3->deg);

temp3=temp3->next;

}

return 0;

}



Q. Polynomial addition using single linked list

#include <stdio.h>

#include <stdlib.h>

struct poly {

int coeff;

int deg;

struct poly \*next;

};

void addTerm(struct poly \*\*head, int coeff, int deg) {

struct poly \*temp = \*head;

struct poly \*prev = NULL;

while (temp != NULL && temp->deg > deg) {

prev = temp;

temp = temp->next;

}

if (temp != NULL && temp->deg == deg) {

temp->coeff += coeff;

} else {

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

newterm->coeff = coeff;

newterm->deg = deg;

newterm->next = temp;

if (prev == NULL) {

\*head = newterm;

} else {

prev->next = newterm;

}

}

}

void printPoly(struct poly \*head) {

struct poly \*temp = head;

int i = 1;

while (temp != NULL) {

printf("Term %d: Coeff: %d Deg: %d\n", i++, temp->coeff, temp->deg);

temp = temp->next;

}

}

int main() {

int n = 0, c = 0, d = 0;

struct poly \*polynomial1 = NULL;

printf("Enter the first polynomial, specifying the coefficient and degree respectively.\n");

printf("Number of terms: ");

scanf("%d", &n);

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

printf("\nCoefficient: ");

scanf("%d", &c);

printf("Degree: ");

scanf("%d", &d);

addTerm(&polynomial1, c, d);

}

printf("\nEnter the second polynomial\n");

printf("Number of terms: ");

scanf("%d", &n);

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

printf("\nCoefficient: ");

scanf("%d", &c);

printf("Degree: ");

scanf("%d", &d);

addTerm(&polynomial1, c, d);

}

printf("\nResultant polynomial:\n");

printPoly(polynomial1);

return 0;}

