#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 multiplication(Poly\* Poly1, Poly\* Poly2, Poly\* result);

void insertPoly(Poly\*\* result, int coeff, int pow);

int main() {

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

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

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

Poly1->Next = NULL;

Poly2->Next = NULL;

result->Next = NULL;

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

create(Poly1);

printf("Equation is: ");

display(Poly1);

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

create(Poly2);

printf("Equation is: ");

display(Poly2);

multiplication(Poly1, Poly2, result);

printf("Product of polynomials: ");

display(result);

return 0;

}

void create(Poly\* List) {

int choice;

Poly\* position, \* NewNode;

position = List;

do {

NewNode = malloc(sizeof(Poly));

printf("Enter the coeff: ");

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

printf("Enter the pow: ");

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

NewNode->Next = NULL;

position->Next = NewNode;

position = NewNode;

printf("Enter 1 to continue, 0 to stop: ");

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(" + ");

}

}

printf("\n");

}

void insertPoly(Poly\*\* result, int coeff, int pow) {

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

NewNode->coeff = coeff;

NewNode->pow = pow;

NewNode->Next = NULL;

Poly\* temp = \*result;

while (temp->Next != NULL && temp->Next->pow > pow) {

temp = temp->Next;

}

if (temp->Next != NULL && temp->Next->pow == pow) {

temp->Next->coeff += coeff;

} else {

NewNode->Next = temp->Next;

temp->Next = NewNode;

}

}

void multiplication(Poly\* Poly1, Poly\* Poly2, Poly\* result) {

Poly\* temp1 = Poly1->Next;

Poly\* temp2;

while (temp1 != NULL) {

temp2 = Poly2->Next;

while (temp2 != NULL) {

int coeff = temp1->coeff \* temp2->coeff;

int pow = temp1->pow + temp2->pow;

insertPoly(&result, coeff, pow);

temp2 = temp2->Next;

}

temp1 = temp1->Next;

}

}