#include <stdio.h>

#include <stdlib.h>

struct polynomial {

int coff;

int expo;

struct polynomial\* next;

};

typedef struct polynomial poly;

poly\* create(poly\* head, int n) {

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

int e, c;

printf("Enter the values of Exponent and Coefficient: ");

scanf("%d %d", &e, &c);

poly\* newnode = (poly\*)malloc(sizeof(poly));

newnode->expo = e;

newnode->coff = c;

newnode->next = NULL;

if (head == NULL) {

head = newnode;

} else {

poly\* prev = NULL;

poly\* curr = head;

while (curr != NULL && curr->expo > e) {

prev = curr;

curr = curr->next;

}

if (curr != NULL && curr->expo == e) {

curr->coff += c;

free(newnode);

} else if (prev == NULL) {

newnode->next = head;

head = newnode;

} else {

prev->next = newnode;

newnode->next = curr;

}

}

}

return head;

}

void show(poly\* head) {

while (head != NULL) {

printf("%dx^%d", head->coff, head->expo);

if (head->next != NULL)

printf(" + ");

head = head->next;

}

printf("\n");

}

poly\* multipoly(poly\* head1,poly\* head2){

poly\* result = NULL;

for (poly\* ptr1 = head1; ptr1 != NULL; ptr1 = ptr1->next) {

for (poly\* ptr2 = head2; ptr2 != NULL; ptr2 = ptr2->next) {

int newExpo = ptr1->expo + ptr2->expo;

int newCoff = ptr1->coff \* ptr2->coff;

// Create a new node to insert

poly\* newnode = (poly\*)malloc(sizeof(poly));

newnode->expo = newExpo;

newnode->coff = newCoff;

newnode->next = NULL;

if (result == NULL) {

result = newnode;

} else {

poly\* prev = NULL;

poly\* curr = result;

while (curr != NULL && curr->expo > newExpo) {

prev = curr;

curr = curr->next;

}

if (curr != NULL && curr->expo == newExpo) {

curr->coff += newCoff;

free(newnode);

} else if (prev == NULL) {

newnode->next = result;

result = newnode;

} else {

newnode->next = curr;

prev->next = newnode;

}

}

}

}

return result;

}

int main() {

int n;

poly\* head1 = NULL;

poly\* head2 = NULL;

for (int i = 0; i < 2; i++) {

printf("Enter number of terms in polynomial %d: ", i + 1);

scanf("%d", &n);

if (i == 0) {

head1 = create(head1, n);

} else {

head2 = create(head2, n);

}

}

poly\* result = multipoly(head1, head2);

printf("Resultant Polynomial: ");

show(result);

return 0;

}

#include <stdio.h>

#include <stdlib.h>

struct polynomial {

    int coff;

    int expo;

    struct polynomial\* next;

};

typedef struct polynomial poly;

poly\* create(poly\* head, int n) {

    for (int i = 0; i < n; i++) {

        int e, c;

        printf("Enter the values of Exponent and Coefficient: ");

        scanf("%d %d", &e, &c);

        poly\* newnode = (poly\*)malloc(sizeof(poly));

        newnode->expo = e;

        newnode->coff = c;

        newnode->next = NULL;

        if (head == NULL) {

            head = newnode;

        } else {

            poly\* prev = NULL;

            poly\* curr = head;

            while (curr != NULL && curr->expo > e) {

                prev = curr;

                curr = curr->next;

            }

            if (curr != NULL && curr->expo == e) {

                curr->coff += c;

                free(newnode);

            } else if (prev == NULL) {

                newnode->next = head;

                head = newnode;

            } else {

                prev->next = newnode;

                newnode->next = curr;

            }

        }

    }

    return head;

}

void show(poly\* head) {

    while (head != NULL) {

        printf("%dx^%d", head->coff, head->expo);

        if (head->next != NULL)

            printf(" + ");

        head = head->next;

    }

    printf("\n");

}

poly\* addingpoly(poly\* head1, poly\* head2) {

    poly\* result = NULL;

    poly\* tail = NULL;

    while (head1 != NULL && head2 != NULL) {

        poly\* temp = (poly\*)malloc(sizeof(poly));

        temp->next = NULL;

        if (head1->expo == head2->expo) {

            temp->expo = head1->expo;

            temp->coff = head1->coff + head2->coff;

            head1 = head1->next;

            head2 = head2->next;

        } else if (head1->expo > head2->expo) {

            temp->expo = head1->expo;

            temp->coff = head1->coff;

            head1 = head1->next;

        } else {

            temp->expo = head2->expo;

            temp->coff = head2->coff;

            head2 = head2->next;

        }

        if (result == NULL) {

            result = temp;

            tail = temp;

        } else {

            tail->next = temp;

            tail = temp;

        }

    }

    while (head1 != NULL) {

        poly\* temp = (poly\*)malloc(sizeof(poly));

        temp->expo = head1->expo;

        temp->coff = head1->coff;

        temp->next = NULL;

        tail->next = temp;

        tail = temp;

        head1 = head1->next;

    }

    while (head2 != NULL) {

        poly\* temp = (poly\*)malloc(sizeof(poly));

        temp->expo = head2->expo;

        temp->coff = head2->coff;

        temp->next = NULL;

        tail->next = temp;

        tail = temp;

        head2 = head2->next;

    }

    return result;

}

int main() {

    int n;

    poly\* head1 = NULL;

    poly\* head2 = NULL;

    for (int i = 0; i < 2; i++) {

        printf("Enter number of terms in polynomial %d: ", i + 1);

        scanf("%d", &n);

        if (i == 0) {

            head1 = create(head1, n);

        } else {

            head2 = create(head2, n);

        }

    }

    poly\* result = addingpoly(head1, head2);

    printf("Resultant Polynomial: ");

    show(result);

    return 0;

}