#include <stdio.h>

#include <stdlib.h>

#include <math.h>

struct polynomial

{

    int coff;

    int expo;

    struct polynomial\* next;

};

typedef struct polynomial poly;

void 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->expo == 0 && head->coff == 0 && head->next == NULL) {

            head->expo = e;

            head->coff = c;

        } 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) {

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

                    temp->expo = head->expo;

                    temp->coff = head->coff;

                    temp->next = head->next;

                    head->expo = e;

                    head->coff = c;

                    head->next = temp;

                    free(newnode);

                } else {

                    prev->next = newnode;

                    newnode->next = curr;

                }

            }

        }

    }

}

void creation(poly\*head){

    int a=1;

    while(a==1){

    int e, c;

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

        scanf("%d %d", &c, &e);

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

        newnode->expo = e;

        newnode->coff = c;

        newnode->next = NULL;

        if (head->expo == 0 && head->coff == 0 && head->next == NULL) {

            head->expo = e;

            head->coff = c;

        } 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) {

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

                    temp->expo = head->expo;

                    temp->coff = head->coff;

                    temp->next = head->next;

                    head->expo = e;

                    head->coff = c;

                    head->next = temp;

                    free(newnode);

                } else {

                    prev->next = newnode;

                    newnode->next = curr;

                }

            }

        }

        printf("Want you to continue: 1/0\n");

        scanf("%d",&a);

        if(a==0){

            break;

        }

    }

    }

void show(poly\* head) {

    while (head != NULL) {

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

        if (head->next != NULL) {

            printf("+");

        }

        head = head->next;

    }

}

int value(poly\*head,int x){

int sum=0;

while(head!=NULL){

    sum+=head->coff\*(pow(x,head->expo));

    head=head->next;

}

return sum;

}

void revshow(poly\*head,int n){

char arr[n];

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

{

    arr[i]="head->coffx^head->expo";

}

for (int i = n; i > 0; i--)

{

printf("%c+",arr[i]);

}

 }

int main() {

    int n;

    // printf("enter the number of terms in polynomial ");

    // scanf("%d", &n);

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

    head->coff = 0;

    head->expo = 0;

    head->next = NULL;

    // create(head, n);

    creation(head);

    show(head);

    int x;

    printf("\nEnter the value of x: \n");

    scanf("%d",&x);

    printf("The value of polynomial is %d\n",value(head,x));

}