// C++ program for addition of two polynomials

// using Linked Lists

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

//using namespace std;

// Node structure containing power and coefficient of variable

struct Node

{

int coeff;

int pow;

struct Node \*next;

};

// Function to create new node

void create\_node(int x, int y, struct Node \*\*temp)

{

struct Node \*r, \*z;

z = \*temp;

if(z == NULL)

{

r =(struct Node\*)malloc(sizeof(struct Node));

r->coeff = x;

r->pow = y;

\*temp = r;

r->next = (struct Node\*)malloc(sizeof(struct Node));

r = r->next;

r->next = NULL;

}

else

{

r->coeff = x;

r->pow = y;

r->next = (struct Node\*)malloc(sizeof(struct Node));

r = r->next;

r->next = NULL;

}

}

// Function Adding two polynomial numbers

void polyadd(struct Node \*poly1, struct Node \*poly2, struct Node \*poly)

{

while(poly1->next && poly2->next)

{

// If power of 1st polynomial is greater than 2nd, then store 1st as it is

// and move its pointer

if(poly1->pow > poly2->pow)

{

poly->pow = poly1->pow;

poly->coeff = poly1->coeff;

poly1 = poly1->next;

}

// If power of 2nd polynomial is greater than 1st, then store 2nd as it is

// and move its pointer

else if(poly1->pow < poly2->pow)

{

poly->pow = poly2->pow;

poly->coeff = poly2->coeff;

poly2 = poly2->next;

}

// If power of both polynomial numbers is same then add their coefficients

else

{

poly->pow = poly1->pow;

poly->coeff = poly1->coeff+poly2->coeff;

poly1 = poly1->next;

poly2 = poly2->next;

}

// Dynamically create new node

poly->next = (struct Node \*)malloc(sizeof(struct Node));

poly = poly->next;

poly->next = NULL;

}

while(poly1->next || poly2->next)

{

if(poly1->next)

{

poly->pow = poly1->pow;

poly->coeff = poly1->coeff;

poly1 = poly1->next;

}

if(poly2->next)

{

poly->pow = poly2->pow;

poly->coeff = poly2->coeff;

poly2 = poly2->next;

}

poly->next = (struct Node \*)malloc(sizeof(struct Node));

poly = poly->next;

poly->next = NULL;

}

}

// Display Linked list

void show(struct Node \*node)

{

while(node->next != NULL)

{

printf("%dx^%d", node->coeff, node->pow);

node = node->next;

if(node->next != NULL)

printf(" + ");

}

}

// Driver program

int main()

{

struct Node \*poly1 = NULL, \*poly2 = NULL, \*poly = NULL;

// Create first list of 5x^2 + 4x^1 + 2x^0

FILE \*fptr;

fptr = fopen("a.txt","r");

if(fptr == NULL)

{

printf("Whoops!\nFile does not exist!\n");

exit(1);

}

int count = 0;

int i;

while(!feof(fptr))

{

fscanf(fptr,"%d",&i);

count++;

//printf("%d\n",i );

}

fseek(fptr, 0, SEEK\_SET);

int \*a = (int \*)malloc(count \* sizeof(int));

if(a == NULL)

{

printf("Error! memory not allocated.");

exit(0);

}

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

{

fscanf(fptr,"%d",a+i);

}

for(i=0;i<count;i=i+2)

{

create\_node(a[i],a[i+1],&poly1);

}

// Create second list of 5x^1 + 5x^0

FILE \*f1ptr;

f1ptr = fopen("b.txt","r");

if(f1ptr == NULL)

{

printf("Whoops!\nFile does not exist!\n");

exit(1);

}

int count1 = 0;

//int i;

while(!feof(f1ptr))

{

fscanf(f1ptr,"%d",&i);

count1++;

//printf("%d\n",i );

}

fseek(f1ptr, 0, SEEK\_SET);

int \*b = (int \*)malloc(count1 \* sizeof(int));

if(b == NULL)

{

printf("Error! memory not allocated.");

exit(0);

}

for (int i = 0; i < count1; i++)

{

fscanf(f1ptr,"%d",b+i);

}

for(i=0;i<count1;i=i+2)

{

create\_node(b[i],b[i+1],&poly2);

}

printf("1st Number: ");

show(poly1);

printf("\n2nd Number: ");

show(poly2);

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

// Function add two polynomial numbers

polyadd(poly1, poly2, poly);

// Display resultant List

printf("\nAdded polynomial: ");

show(poly);

return 0;

}