#include<stdio.h>

#include<stdlib.h>

typedef struct link {

int coeff;

int pow;

struct link \* next;

} my\_poly;

void my\_create\_poly(my\_poly \*\*);

void my\_show\_poly(my\_poly \*);

void my\_add\_poly(my\_poly \*\*, my\_poly \*, my\_poly \*);

int main(void)

{

int ch;

do

{

my\_poly \* poly1, \* poly2, \* poly3;

printf("\nCreation of first expression:\n");

my\_create\_poly(&poly1);

printf("\nFirst expression:\n");

my\_show\_poly(poly1);

printf("\nCreation of second expression:\n");

my\_create\_poly(&poly2);

printf("\nSecond Expression:\n");

my\_show\_poly(poly2);

my\_add\_poly(&poly3, poly1, poly2);

my\_show\_poly(poly3);

printf("\nDo you want to continue the expression (Y = 1/N = 0): ");

scanf("%d", &ch);

} while (ch);

return 0;

}

void my\_create\_poly(my\_poly \*\* node)

{

int flag;

int coeff, pow;

my\_poly \* tmp\_node;

tmp\_node = (my\_poly \*) malloc(sizeof(my\_poly));

\*node = tmp\_node;

do

{

printf("\nEnter Coeff:");

scanf("%d", &coeff);

tmp\_node->coeff = coeff;

printf("\nEnter Pow:");

scanf("%d", &pow);

tmp\_node->pow = pow;

tmp\_node->next = NULL;

printf("\nContinue adding more terms to the polynomial list?(Y = 1/N = 0): ");

scanf("%d", &flag);

if(flag)

{

tmp\_node->next = (my\_poly \*) malloc(sizeof(my\_poly));

tmp\_node = tmp\_node->next;

tmp\_node->next = NULL;

}

} while (flag);

}

void my\_show\_poly(my\_poly \* node)

{

printf("\nThe polynomial expression is:\n");

while(node != NULL)

{

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

node = node->next;

if(node != NULL)

printf(" + ");

}

}

void my\_add\_poly(my\_poly \*\* result, my\_poly \* poly1, my\_poly \* poly2)

{

my\_poly \* tmp\_node;

tmp\_node = (my\_poly \*) malloc(sizeof(my\_poly));

tmp\_node->next = NULL;

\*result = tmp\_node;

while(poly1 && poly2)

{

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

{

tmp\_node->pow = poly1->pow;

tmp\_node->coeff = poly1->coeff;

poly1 = poly1->next;

}

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

{

tmp\_node->pow = poly2->pow;

tmp\_node->coeff = poly2->coeff;

poly2 = poly2->next;

}

else

{

tmp\_node->pow = poly1->pow;

tmp\_node->coeff = poly1->coeff + poly2->coeff;

poly1 = poly1->next;

poly2 = poly2->next;

}

if(poly1 && poly2)

{

tmp\_node->next = (my\_poly \*) malloc(sizeof(my\_poly));

tmp\_node = tmp\_node->next;

tmp\_node->next = NULL;

}

}

while(poly1 || poly2)

{

tmp\_node->next = (my\_poly \*) malloc(sizeof(my\_poly));

tmp\_node = tmp\_node->next;

tmp\_node->next = NULL;

if(poly1)

{

tmp\_node->pow = poly1->pow;

tmp\_node->coeff = poly1->coeff;

poly1 = poly1->next;

}

if(poly2)

{

tmp\_node->pow = poly2->pow;

tmp\_node->coeff = poly2->coeff;

poly2 = poly2->next;

}

}

printf("\nAddition Process is:");

}