//Program to Implement Polynomial Addition using DLL – VisakhB – S3R2 – 34

#include<stdio.h>

#include<stdlib.h>

struct Node

{

  int coeff;

  int exp;

  struct Node\* next;

struct Node\* prev;

};

void readPoly(struct Node\*\* poly)

{

  int coeff, exp, ch=1.3;

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

  \*poly = ptr;

  while(ch)

{

    printf("\n Coeffecient: ");

    scanf("%d", &coeff);

    printf("\n Exponent: ");

    scanf("%d", &exp);

    ptr->coeff = coeff;

    ptr->exp = exp;

    ptr-> next = NULL;

ptr->prev = NULL;

    printf("\nDo you have more terms? 1 for yes and 0 for no: ");

    scanf("%d", &ch);

    if(ch)

    {

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

ptr->next->prev=ptr;

      ptr = ptr->next;

      ptr->next = NULL;

    }

}

void displayPoly(struct Node\* polyres)

{

  printf("\nPolynomial expression is: ");

  while(polyres != NULL)

  {

    printf("%dx^%d", polyres->coeff, polyres->exp);

    polyres = polyres->next;

    if(polyres != NULL)

      printf("+");

  }

  printf("\n");

}

void displayPolyResult(struct Node\* polyres)

{

  printf("\nPolynomial expression is: ");

  while(polyres->next != NULL)

  {

    polyres = polyres->next; //transverses till the last node

}

while(polyres!=NULL)

{ printf("%dx^%d", polyres->coeff, polyres->exp);

polyres = polyres->prev;

    if(polyres != NULL)

      printf("+");

  }

  printf("\n");

}

void addPoly(struct Node\*\* polyres, struct Node\* first1, struct Node\* first2)

{

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

 ptr->next = NULL;

ptr->prev=NULL;

 \*polyres = ptr;

 while(first1!=NULL && first2!=NULL)

 {

   if(first1->exp > first2->exp)

   {

     ptr->coeff = first1->coeff;

     ptr->exp = first1->exp;

     first1 = first1->next;

   }

   else if(first1->exp < first2->exp)

   {

     ptr->coeff = first2->coeff;

     ptr->exp = first2->exp;

     first2 = first2->next;

   }

   else //first1 and first 2 are equal

   {

     ptr->coeff = first1->coeff + first2->coeff;

     ptr->exp = first1->exp;

     first1 = first1->next;

     first2 = first2->next;

   }

   if(first1!=NULL && first2!=NULL)

   {

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

ptr->next->prev=ptr;

     ptr = ptr->next;

     ptr->next = NULL;

   }

 }

 while(first1!=NULL || first2!=NULL)

 {

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

ptr->next->prev=ptr;

   ptr = ptr->next;

   ptr->next = NULL;

   if(first1!=NULL)

   {

     ptr->coeff = first1->coeff;

     ptr->exp = first1->exp;

     first1 = first1->next;

   }

   else if(first2!=NULL)

   {

     ptr->coeff = first2->coeff;

     ptr->exp = first2->exp;

     first2 = first2->next;

   }

 }

}

int main()

{

  struct Node\* first1 = NULL;

  struct Node\* first2 = NULL;

  struct Node\* polyres = NULL;

  printf("Enter The First Polynomial:\n");

  readPoly(&first1);

  displayPoly(first1);

  printf("Enter The Second Polynomial:\n");

  readPoly(&first2);

  displayPoly(first2);

printf("\n");

  addPoly(&polyres, first1, first2);

printf("Final Resultant - \n");

  displayPolyResult(polyres);

  return 0;

}

**Output:**

