POLYNOMIAL ADDITION USING

LINKED LIST

By

VAISHAK S(R3-258)

**ALGORITHM**

**Input:** 2 polynomials ‘first’ and ‘second’

**Output:** Resultant polynomial ‘result’

Create a structure Node with variables 'coeff' and 'pow' of type int and a pointer 'next' of type struct Node.

**readPolynimial(struct Node\* poly )**

1.Create a node 'temp' of type Node

2.\*poly=temp

3.do

{

4. Read the coefficient and exponent from

the user to coeff and pow respectively

5.temp->coeff =coeff

6.temp->pow = exp

7.temp-> next = NULL

8.if(cont)

{

create new node

temp = temp->next;

temp->next = NULL;

}while(cont)

**displayPolynomial(struct Node\*poly)**

1.if(poly!=NULL)

2.then print poly and power in the correct format

3.poly=poly->next

4.if(poly!=NULL)

print "+"

5.Go to step 1

**addPolynomials( )**

1.Create a node temp

2.\*result=temp

3.while(first && second)

{

if(first->pow > second->pow)

{

temp->coeff = first->coeff;

temp->pow = first->pow;

first = first->next;

}

else if(first->pow < second->pow)

{

temp->coeff = second->coeff;

temp->pow = second->pow;

second = second->next;

}

else

{

temp->coeff = first->coeff + second->coeff;

temp->pow = first->pow;

first = first->next;

second = second->next;

}

if(first && second)

{

temp->next = (struct Node)malloc(sizeof(struct Node));

temp = temp->next;

temp->next = NULL;

}

}

while(first || second)

{

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

temp = temp->next;

temp->next = NULL;

if(first)

{

temp->coeff = first->coeff;

temp->pow = first->pow;

first = first->next;

}

else if(second)

{

temp->coeff = second->coeff;

temp->pow = second->pow;

second = second->next;

}

}

}

6.Stop

**main( )**

1.Start

2.Create 3 nodes first,second and third and set them as NULL

3.Call function readPolynomial( first and second from user )

4.Call function displayPolynomial(first and the second polynomial )

5.Call function addPolynomial( add the first and the second to get the resultant polynomial )

6.Call function readPolynomial( resultant polynomial)

7.Stop

**PROGRAM**

#include<stdio.h>

#include<stdlib.h>

struct Node

{

int coeff;

int pow;

struct Node\* next;

};

void readPolynomial(struct Node\*\* poly)

{

int coeff, exp, cont;

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

\*poly = temp;

do

{

printf("\n Coeffecient: ");

scanf("%d", &coeff);

printf("\n Exponent: ");

scanf("%d", &exp);

temp->coeff = coeff;

temp->pow = exp;

temp-> next = NULL;

printf("\nHave more terms? 1 for y and 0 for no: ");

scanf("%d", &cont);

if(cont)

{

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

temp = temp->next;

temp->next = NULL;

}

}while(cont);

}

void displayPolynomial(struct Node\* poly)

{

printf("\nPolynomial expression is: ");

while(poly != NULL)

{

printf("%dX^%d", poly->coeff, poly->pow);

poly = poly->next;

if(poly != NULL)

printf("+");

}

}

void addPolynomials(struct Node\*\* result, struct Node\* first, struct Node\* second)

{

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

temp->next = NULL;

\*result = temp;

while(first && second)

{

if(first->pow > second->pow)

{

temp->coeff = first->coeff;

temp->pow = first->pow;

first = first->next;

}

else if(first->pow < second->pow)

{

temp->coeff = second->coeff;

temp->pow = second->pow;

second = second->next;

}

else

{

temp->coeff = first->coeff + second->coeff;

temp->pow = first->pow;

first = first->next;

second = second->next;

}

if(first && second)

{

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

temp = temp->next;

temp->next = NULL;

}

}

while(first || second)

{

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

temp = temp->next;

temp->next = NULL;

if(first)

{

temp->coeff = first->coeff;

temp->pow = first->pow;

first = first->next;

}

else if(second)

{

temp->coeff = second->coeff;

temp->pow = second->pow;

second = second->next;

}

}

}

void main()

{

struct Node\* first = NULL;

struct Node\* second = NULL;

struct Node\* result = NULL;

printf("\nEnter the corresponding data:-\n");

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

readPolynomial(&first);

displayPolynomial(first);

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

readPolynomial(&second);

displayPolynomial(second);

addPolynomials(&result, first, second);

displayPolynomial(result);

}