DATE:6.01.15

**ASSIGNMENT NO.18**

->PROBLEM STATEMENT:

Write a program in C to add two polynomial equations using linked list and display the result after addition.

**->ALGORITHM:**

**Data Structure Definition**

[ The data structure Node represents a single node in the linked list]

*Structure Node*

*( coef* that hold the coefficient of the polynomial.

*expo* that keeps the exponent of the corresponding coefficient

*next* that holds the address of the next Node

*)*

1. **Algorithm for function *createpoly()***

Purpose : To create a polynomial equation.

Step 1:- [ Initialize the ‘head’ pointer]

head:= NULL.

Step 2:- [ Take input from user to insert into linked list ]

1. Input coefficient.

Step 3:- Do

1. Input exponent
2. Allocate memory for a new Node and Assign its pointer as new\_node
3. new\_node → coef := coef
4. new\_node →expo := expo
5. new\_node → next := NULL
6. If ( head = NULL) Then
7. head := new\_node
8. Else
9. ptr → next := new\_node

[ end of If…..then…..else……structure]

1. ptr := new\_node
2. Input coefficient [such that num ≠ -999]
3. While (coef ≠ -999)

[ end of while….. loop]

Step 4:- Return head

**(2) Algorithm for function *display\_poly()*.**

Purpose:- To display the polynomial equation.

Step 1:- [ Check for existence of polynomial equation]

If (head = NULL) then

1. Display “No polynomial equation found.”
2. Return.

[ end of If…..then…….structure]

Step 2:- ptr := head.

Step 3:- While ( ptr→next ≠ NULL)

1. If (ptr→expo = 0)
2. Display “ptr→coef”.
3. Else
4. Display “ptr→coef ,ptr→expo”.

[End of If……then…..Else…….structure]

1. ptr := ptr→next.

[ end of while… loop]

Step 4:- Display “ptr→coef ,ptr→expo”.

Step 5:- Return head.

**(3) Algorithm for function *sumpoly().***

Purpose:- Toadd two polynomial eq;uations and display their sum.

Step 1:- ptr1 := head1

Step 2:- ptr2 := head2

Step 3:- sumpoly := (node\*)malloc(sizeof(node))

Step 4:- sumpoly\_head := sumpoly

Step 5:- While (ptr1 ≠ NULL AND ptr2 ≠ NULL)

1. If ((ptr1→expo) = (ptr2→expo)) then
2. sumpoly→coef := ptr1→coef + ptr2→coef
3. sumpoly→expo := ptr1→expo
4. ptr1 := ptr1→next
5. ptr2 := ptr2→next
6. Else If((ptr1→expo) > (ptr2→expo)) then
7. sumpoly→coef : = ptr1→coef
8. sumpoly→expo := ptr1→expo
9. ptr1 := ptr1→next
10. Else
11. sumpoly→coef := ptr2→coef
12. sumpoly→expo := ptr2→expo
13. ptr2 := ptr2→next

[End of If….then….Else…..If…..structure]

1. sumpoly→next := (node \*)malloc(sizeof(node))
2. sumpoly := sumpoly→next

[End of while….loop]

Step 6:- While (ptr1 ≠ NULL)

1. sumpoly→coef := ptr1→coef
2. sumpoly→expo := ptr1→expo
3. ptr1 := ptr1→next
4. sumpoly→next := (node\*)malloc(sizeof(node))
5. sumpoly := sumpoly→next

[End of While…..loop]

Step 7:- While (ptr2 ≠ NULL)

1. sumpoly→coef := ptr2→coef
2. sumpoly→expo := ptr2→expo
3. ptr2 := ptr2→next
4. sumpoly→next := (node\*)malloc(sizeof(node))
5. sumpoly := sumpoly→next

[End of While…..loop]

Step 8:- sumpoly→next := NULL

Step 9:- Return head.

->CODE:

/\* ADDITION TWO POLYNOMIALS \*/

# include<stdio.h>

# include<conio.h>

/\* Defination of straucture \*/

typedef struct node

{

int coef;

int expo;

struct node \*next;

}node;

struct node \*head1,\*head2; // Declared head pointres as GLOBAL

/\* Function of getnode() \*/

struct node \*getnode(int x, int y)

{

node \*p;

p = (node\*)malloc(sizeof(node));

p->coef = x;

p->expo = y;

p->next = NULL;

return p;

}

/\* Function for create a polynomaial \*/

struct node \*createpoly(node \*head)

{

node \*ptr1,\*ptr2;

int coef,expo;

printf("\nEnter the particulars of the polynomial:-");

printf("\n\tEnter coefficient : ");

scanf("%d",&coef);

do

{

printf("\tEnter the exponent : ");

scanf("%d",&expo);

ptr1 = getnode(coef,expo);

if (head == NULL)

head = ptr1;

else

ptr2->next = ptr1;

ptr2 = ptr1;

printf("\n\tEnter the coefficient (-999 to STOP) : ");

scanf("%d",&coef);

}while(coef != -999);

return head;

}

/\* Function for display a polynomial \*/

void display\_poly(node \*head)

{

node \*ptr;

ptr = head;

if (head == NULL)

{

printf("\nNo polynomial found. Create a polynomial first.");

getch();

return;

}

printf("\n\tYour polynomial is :- ");

while (ptr->next != NULL)

{

if(ptr->expo == 0)

printf(" %d +",ptr->coef);

else

printf(" %dx^%d + ",ptr->coef,ptr->expo);

ptr = ptr->next;

}

printf(" %dx^%d ",ptr->coef,ptr->expo);

}

/\* Function for addition of two polynomial equation \*/

void sum\_poly(node \*head1,node \*head2)

{

node \*ptr1,\*ptr2,\*sumpoly,\*sumpoly\_head;

ptr1 = head1;

ptr2 = head2;

sumpoly = (node\*)malloc(sizeof(node));

sumpoly\_head = sumpoly;

while((ptr1 != NULL) && (ptr2 != NULL))

{

if ((ptr1->expo) == (ptr2->expo))

{

sumpoly->coef = ptr1->coef + ptr2->coef;

sumpoly->expo = ptr1->expo;

ptr1 = ptr1->next;

ptr2 = ptr2->next;

}

else if((ptr1->expo) > (ptr2->expo))

{

sumpoly->coef = ptr1->coef;

sumpoly->expo = ptr1->expo;

ptr1 = ptr1->next;

}

else

{

sumpoly->coef = ptr2->coef;

sumpoly->expo = ptr2->expo;

ptr2 = ptr2->next;

}

sumpoly->next = (node \*)malloc(sizeof(node));

sumpoly = sumpoly->next;

}

while (ptr1 != NULL)

{

sumpoly->coef = ptr1->coef;

sumpoly->expo = ptr1->expo;

ptr1 = ptr1->next;

sumpoly->next = (node\*)malloc(sizeof(node));

sumpoly = sumpoly->next;

}

while (ptr2 != NULL)

{

sumpoly->coef = ptr2->coef;

sumpoly->expo = ptr2->expo;

ptr2 = ptr2->next;

sumpoly->next = (node\*)malloc(sizeof(node));

sumpoly = sumpoly->next;

}

sumpoly->next = NULL;

printf("\nAfter addition polynomial is :-");

sumpoly = sumpoly\_head;

while(sumpoly->next->next !=NULL)

{

if (sumpoly->expo == 0)

printf(" %d + ",sumpoly->coef);

else

printf(" %d x^%d + ",sumpoly->coef,sumpoly->expo);

sumpoly = sumpoly->next;

}

if(sumpoly->expo == 0)

printf(" %d ",sumpoly->coef);

else

printf(" %d x^%d",sumpoly->coef,sumpoly->expo);

}

/\* Main function \*/

void main()

{

clrscr();

printf("\n\t\tPROGRAM FOR POLYNOMIAL ADDITION\n");

printf("\nEnter your 1st polynomial :-");

head1 = createpoly(head1);

display\_poly(head1);

printf("\nEnter your 2nd polynomial :-");

head2 = createpoly(head2);

display\_poly(head2);

sum\_poly(head1,head2);

getch();

}

**->OUTPUT :**

/\* ================== OUTPUT =============================>

PROGRAM FOR POLYNOMIAL ADDITION

Enter your 1st polynomial :-

Enter the particulars of the polynomial:-

Enter coefficient : 12

Enter the exponent : 2

Enter the coefficient (-999 to STOP) : 12

Enter the exponent : 3

Enter the coefficient (-999 to STOP) : 12

Enter the exponent : 4

Enter the coefficient (-999 to STOP) : -999

Your polynomial is :- 12x^2 + 12x^3 + 12x^4

Enter your 2nd polynomial :-

Enter the particulars of the polynomial:-

Enter coefficient : 13

Enter the exponent : 2

Enter the coefficient (-999 to STOP) : 13

Enter the exponent : 3

Enter the coefficient (-999 to STOP) : -999

Your polynomial is :- 13x^2 + 13x^3

After addition polynomial is :- 25 x^2 + 25 x^3 + 12 x^4

\*/