**Program 7:** **Design, Develop and Implement Program in C for the following operations on singly linked List**

**a. Represent and Evaluate a Polynomial P(x) = 6x5 - 4x3 +3x2 +2x -2 = 0**

**b. Find the sum of two polynomials POLY1(x) and POLY2(x) and store the result in POLYSUM(x).**

**Support the program with appropriate functions for each of the above operations**

#include<stdio.h>

#include<malloc.h>

struct node

{

int degree;

int coeff;

struct node \*link;

}; typedef struct node \* NODE;

**//\*\*\*\*\*\*\*\*\*\*\* Function to Insert at begining \*\*\*\*\*\*\*\*\***

NODE insfront(NODE first)

{

NODE newnode;

newnode = (NODE)malloc(sizeof(struct node));

printf("\nEnter the Degree and coeffiecient \n");

scanf("%d%d", &newnode->degree, &newnode->coeff);

newnode->link = first;

first = newnode;

return first;

}

**//\*\*\*\*\*\*\*\*\*\*\* Function to Insert at begining \*\*\*\*\*\*\*\*\***

NODE insans(NODE reslt,int deg, int sum)

{

NODE newnode;

newnode = (NODE)malloc(sizeof(struct node));

newnode->degree = deg;

newnode->coeff= sum;

newnode->link = reslt;

reslt = newnode;

return reslt;

}

**//------------------------------------------------------**

void display(NODE result)

{

printf("The sum of two polynomials are \n");

while(result != NULL)

{

printf("X^%d %d +",result->degree, result->coeff);

result = result->link;

}

}

**//-----------------------------------------------------------------------------------------------------------------**

NODE add(NODE first, NODE second, NODE result)

{

NODE temp1 = first, temp2 = second;

while( temp1 != NULL && temp2 != NULL)

{

if(temp1->degree == temp2->degree )

{

result = insans(result, temp1->degree, temp1->coeff + temp2->coeff);

temp1 = temp1->link;

temp2 = temp2->link;

}

else if(temp1->degree < temp2->degree )

{

result = insans(result, temp1->degree, temp1->coeff);

temp1 = temp1->link;

}

else

{

result = insans(result, temp2->degree, temp2->coeff);

temp2 = temp2->link;

}

}

while( temp1 != NULL)

{

result = insans(result, temp1->degree, temp1->coeff);

temp1 = temp1->link;

}

while( temp2 != NULL)

{

result = insans(result, temp2->degree, temp2->coeff);

temp2 = temp2->link;

}

}

//---------------------------------------------------------------------------------------------------------

void main()

{

NODE first = NULL,NODE second,result = NULL;

int ch;

clrscr();

for(;;)

{

printf(" Enter 1: First Poly 2: Second Ploy 3:Addition 3:Display\n");

scanf("%d",&ch);

switch(ch)

{

case 1: first = insfront(first);

break;

case 2: second = insfront(second);

break;

case 3: result = add(first,second,result);

break;

case 4: display(result);

break;

default: exit(0);

}

}

}