**AIM:** To write a c program for Polynomial Addition, Polynomial Subtraction and Polynomial Differentiation

ADDITION AND SUBTRACTION:

**ALGORITHM:**

**Step 1:** Start.

**Step 2:** Create a structure to represent a term in a polynomial, containing coefficients and exponents.

**Step 3:** Define a linked list node to hold the polynomial terms.

**Step 4:** Implement functions to create and manipulate linked lists representing polynomials.

**Step 5:** Traverse both polynomial linked lists simultaneously.

**Step 6:**  Adding or Subtracting corresponding terms and storing the result in a new linked list.

**Step 7:**  Stop.

DIFFERENTIATION:

**ALGORITHM:**

**Step 1:** Start.

**Step 2:** Traverse the linked list representing the polynomial.

**Step 3:** For each term, differentiate the term using the power rule (multiply the coefficient by the exponent and decrease the exponent by 1).

**Step 4:** If the exponent becomes 0 after differentiation, remove the term from the linked list.

**Step 5:** If the resulting coefficient is zero, remove the term from the linked list.

**Step 6:**  Continue until all terms have been differentiated.

**PROGRAM:**

#include <stdio.h>

#include <stdlib.h>

struct poly

{

int coeff;

int pow;

struct poly \*Next;

};

typedef struct poly Poly;

void Create(Poly \*List);

void Display(Poly \*List);

void Addition(Poly \*Poly1, Poly \*Poly2, Poly \*Result);

void Subtraction(Poly \*Poly1, Poly \*Poly2, Poly \*Result);

int main()

{

Poly \*Poly1 = malloc(sizeof(Poly));

Poly \*Poly2 = malloc(sizeof(Poly));

Poly \*Result = malloc(sizeof(Poly));

Poly1->Next = NULL;

Poly2->Next = NULL;

printf("Enter the values for first polynomial :\n");

Create(Poly1);

printf("The polynomial equation is : ");

Display(Poly1);

printf("\nEnter the values for second polynomial :\n");

Create(Poly2);

printf("The polynomial equation is : ");

Display(Poly2);

Printf(“Enter 1 for addition of two polynomials \n Enter 2 for polynomial subtraction”);

int choice;

do {

printf(“Enter your choice”);

scanf(“%d”,&choice);

switch (choice)

{

case1: Addition(Poly1, Poly2, Result);

printf("\nThe polynomial equation addition result is : ");

Display(Result);

case2: Subtraction(Poly1, Poly2, Result);

printf("\nThe polynomial equation subtraction result is : ");

Display(Result);

}

}while(choice<=2 & choice>=1);

return 0;

}

void Create(Poly \*List)

{

int choice;

Poly \*Position, \*NewNode;

Position = List;

do

{

NewNode = malloc(sizeof(Poly));

printf("Enter the coefficient : ");

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

printf("Enter the power : ");

scanf("%d", &NewNode->pow);

NewNode->Next = NULL;

Position->Next = NewNode;

Position = NewNode;

printf("Enter 1 to continue : ");

scanf("%d", &choice);

} while(choice == 1);

}

void Display(Poly \*List)

{

Poly \*Position;

Position = List->Next;

while(Position != NULL)

{

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

Position = Position->Next;

if(Position != NULL && Position->coeff > 0)

{

printf("+");

}

}

}

void Addition(Poly \*Poly1, Poly \*Poly2, Poly \*Result)

{

Poly \*Position;

Poly \*NewNode;

Poly1 = Poly1->Next;

Poly2 = Poly2->Next;

Result->Next = NULL;

Position = Result;

while(Poly1 != NULL && Poly2 != NULL)

{

NewNode = malloc(sizeof(Poly));

if(Poly1->pow == Poly2->pow)

{

NewNode->coeff = Poly1->coeff + Poly2->coeff;

NewNode->pow = Poly1->pow;

Poly1 = Poly1->Next;

Poly2 = Poly2->Next;

}

else if(Poly1->pow > Poly2->pow)

{

NewNode->coeff = Poly1->coeff;

NewNode->pow = Poly1->pow;

Poly1 = Poly1->Next;

}

else if(Poly1->pow < Poly2->pow)

{

NewNode->coeff = Poly2->coeff;

NewNode->pow = Poly2->pow;

Poly2 = Poly2->Next;

}

NewNode->Next = NULL;

Position->Next = NewNode;

Position = NewNode;

}

while(Poly1 != NULL || Poly2 != NULL)

{

NewNode = malloc(sizeof(Poly));

if(Poly1 != NULL)

{

NewNode->coeff = Poly1->coeff;

NewNode->pow = Poly1->pow;

Poly1 = Poly1->Next;

}

if(Poly2 != NULL)

{

NewNode->coeff = Poly2->coeff;

NewNode->pow = Poly2->pow;

Poly2 = Poly2->Next;

}

NewNode->Next = NULL;

Position->Next = NewNode;

Position = NewNode;

}

}

void Subtraction(Poly \*Poly1, Poly \*Poly2, Poly \*Result)

{

Poly \*Position;

Poly \*NewNode;

Poly1 = Poly1->Next;

Poly2 = Poly2->Next;

Result->Next = NULL;

Position = Result;

while(Poly1 != NULL && Poly2 != NULL)

{

NewNode = malloc(sizeof(Poly));

if(Poly1->pow == Poly2->pow)

{

NewNode->coeff = Poly1->coeff - Poly2->coeff;

NewNode->pow = Poly1->pow;

Poly1 = Poly1->Next;

Poly2 = Poly2->Next;

}

else if(Poly1->pow > Poly2->pow)

{

NewNode->coeff = Poly1->coeff;

NewNode->pow = Poly1->pow;

Poly1 = Poly1->Next;

}

else if(Poly1->pow < Poly2->pow)

{

NewNode->coeff = -(Poly2->coeff);

NewNode->pow = Poly2->pow;

Poly2 = Poly2->Next;

}

NewNode->Next = NULL;

Position->Next = NewNode;

Position = NewNode;

}

while(Poly1 != NULL || Poly2 != NULL)

{

NewNode = malloc(sizeof(Poly));

if(Poly1 != NULL)

{

NewNode->coeff = Poly1->coeff;

NewNode->pow = Poly1->pow;

Poly1 = Poly1->Next;

}

if(Poly2 != NULL)

{

NewNode->coeff = -(Poly2->coeff);

NewNode->pow = Poly2->pow;

Poly2 = Poly2->Next;

}

NewNode->Next = NULL;

Position->Next = NewNode;

Position = NewNode;

}

}

**OUTPUT:**

ADDITION-

Enter the values for first polynomial :

Enter the coefficient : 2

Enter the power : 2

Enter 1 to continue : 1

Enter the coefficient : 6

Enter the power : 1

Enter 1 to continue : 1

Enter the coefficient : 5

Enter the power : 0

Enter 1 to continue : 0

The polynomial equation is : 2x^2+6x^1+5x^0

Enter the values for second polynomial :

Enter the coefficient : 3

Enter the power : 2

Enter 1 to continue : 1

Enter the coefficient : -2

Enter the power : 1

Enter 1 to continue : 1

Enter the coefficient : -1

Enter the power : 0

Enter 1 to continue : 0

The polynomial equation is : 3x^2-2x^1-1x^0

The polynomial equation addition result is : 5x^2+4x^1+4x^0

SUBTRACTION-

Enter the values for first polynomial :

Enter the coefficient : 3

Enter the power : 2

Enter 1 to continue : 1

Enter the coefficient : 4

Enter the power : 1

Enter 1 to continue : 1

Enter the coefficient : -2

Enter the power : 0

Enter 1 to continue : 0

The polynomial equation is : 3x^2+4x^1-2x^0

Enter the values for second polynomial :

Enter the coefficient : -7

Enter the power : 2

Enter 1 to continue : 1

Enter the coefficient : -10

Enter the power : 1

Enter 1 to continue : 1

Enter the coefficient : 17

Enter the power : 0

Enter 1 to continue : 0

The polynomial equation is : -7x^2-10x^1+17x^0

The polynomial equation subtraction result is : 10x^2+14x^1-19x^0

DIFFERENTIATION:

#include <stdio.h>

#include <stdlib.h>

struct poly

{

int coeff;

int pow;

struct poly \*Next;

};

typedef struct poly Poly;

void Create(Poly \*List);

void Display(Poly \*List);

void Differentiation(Poly \*Poly1, Poly \*Result);

int main()

{

Poly \*Poly1 = malloc(sizeof(Poly));

Poly \*Result = malloc(sizeof(Poly));

Poly1->Next = NULL;

printf("Enter the values for polynomial :\n");

Create(Poly1);

printf("The polynomial equation is : ");

Display(Poly1);

Differentiation(Poly1, Result);

printf("\nThe polynomial differentiation equation is : ");

Display(Result);

return 0;

}

void Create(Poly \*List)

{

int choice;

Poly \*Position, \*NewNode;

Position = List;

do

{

NewNode = malloc(sizeof(Poly));

printf("Enter the coefficient : ");

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

printf("Enter the power : ");

scanf("%d", &NewNode->pow);

NewNode->Next = NULL;

Position->Next = NewNode;

Position = NewNode;

printf("Enter 1 to continue : ");

scanf("%d", &choice);

} while(choice == 1);

}

void Display(Poly \*List)

{

Poly \*Position;

Position = List->Next;

while(Position != NULL && Position->pow >= 0)

{

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

Position = Position->Next;

if(Position != NULL && Position->coeff > 0)

{

printf("+");

}

}

}

void Differentiation(Poly \*Poly1, Poly \*Result)

{

Poly \*Position;

Poly \*NewNode;

Poly1 = Poly1->Next;

Result->Next = NULL;

Position = Result;

while(Poly1 != NULL)

{

NewNode = malloc(sizeof(Poly));

NewNode->coeff = Poly1->coeff \* Poly1->pow;

NewNode->pow = Poly1->pow - 1;

Poly1 = Poly1->Next;

NewNode->Next = NULL;

Position->Next = NewNode;

Position = NewNode;

}

}

**OUTPUT**

Enter the values for polynomial :

Enter the coefficient : 3

Enter the power : 5

Enter 1 to continue : 1

Enter the coefficient : -2

Enter the power : 3

Enter 1 to continue : 1

Enter the coefficient : 1

Enter the power : 1

Enter 1 to continue : 1

Enter the coefficient : 5

Enter the power : 0

Enter 1 to continue : 0

The polynomial equation is : 3x^5-2x^3+1x^1+5x^0

The polynomial differentiation equation is : 15x^4-6x^2+1x^0

**RESULT:**

Hence the program polynomial addition, polynomial subtraction and polynomial differentiation is implemented.