Data Structures Odyssey: Exploring the Foundations of Computing

Ex. No.:03	Polynomial Manipulation	Date:21/03/2024

Write a C program to implement the following operations on Singly Linked List.

(i) Polynomial Addition

(ii) Polynomial Subtraction

(iii) Polynomial Multiplication

Algorithm:

1) Start
2) Define a structure for a polynomial term with coefficients and exponents.
3) Create functions for the following operations: a. Polynomial addition:
- Initialize a result polynomial.
- Traverse both input polynomials simultaneously.
- Add coefficients of terms with the same exponent and append to the result.
- Append any remaining terms from both polynomials to the result.
4) Polynomial subtraction:
- Initialize a result polynomial.
- Traverse both input polynomials simultaneously.
- Subtract coefficients of terms with the same exponent and append to the result.
- Append any remaining terms from both polynomials to the result.
5) Polynomial multiplication:
- Initialize a result polynomial.
- Multiply each term of the first polynomial with each term of the second polynomial.
- Add the coefficients of terms with the same exponent and append to the result. 6) Test the operations by performing addition, subtraction, and multiplication on sample polynomials.
7) Stop

PROGRAM:

```
#include <stdio.h>

#include <stdlib.h> struct

node

{ int coeff; int

pow; struct node

*Next;

};

 struct node *Poly1,*Poly2,*Result; void Create(struct node *List); void

Display(struct node *List); void Addition(struct node *Poly1,struct node

*Poly2,struct node *Result); int main()

{

Poly1=(struct node*)malloc(sizeof(struct node));

Poly2=(struct node*)malloc(sizeof(struct node));

Result=(struct node*)malloc(sizeof(struct node));


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);

Addition(Poly1, Poly2, Result); printf("\nThe

polynomial equation addition result is : ");

Display(Result); return 0;

}
```

```
void Create(struct node *List)
{
int choice; struct node *Position,
*NewNode; Position = List; do
{
NewNode = malloc(sizeof(struct node));
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(struct node *List)
{
struct node *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(struct node *Poly1, struct node *Poly2, struct node *Result)
{
struct node *Position; struct
node *NewNode; Poly1 =
Poly1->Next;
Poly2 = Poly2->Next;
Result->Next = NULL; Position =
Result; while(Poly1 != NULL && Poly2
!= NULL)
{
NewNode = malloc(sizeof(struct node)); 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(struct node)); 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;

}

}
```

Program 2:

```
#include<stdio.h> #include<stdlib.h>
```

```
struct node
{    int
coeff;    int
expo;
   struct node *next;
};

struct node* insert(struct node *head,int co,int exp)
{
   struct node *temp;
   struct node *newnode=malloc(sizeof(struct node));
newnode->coeff=co;    newnode->expo=exp;
   newnode->next=NULL;



if(head==NULL || exp>head->expo)
   {
      newnode->next=head;
head=newnode;
   }
else
   {
      temp=head;
      while(temp->next!=NULL &&temp->next->expo>=exp)
         temp=temp->next;        newnode->next=temp-
>next;
      temp->next=newnode;
   }
   return head;
}
struct node* create(struct node *head)
{    int n,i;
int coeff;
int expo;
   printf("Enter the no of terms:");
scanf("%d",&n);
   for(i=0;i<n;i++)
   {
      printf("Enter the coeefficient for term %d:",i+1);
scanf("%d",&coeff);

      printf("Enter the exponent for term %d:",i+1);
scanf("%d",&expo);

      head=insert(head,coeff,expo);
   }
   return head;
}
   void print(struct node* head)
   {
```

```
        if(head==NULL)
printf("No Polynomial");
        else
        {
           struct node *temp=head;
           while(temp!=NULL)
           {
              printf("%dx^%d",temp->coeff,temp->expo);
              temp=temp->next;
if(temp!=NULL)
printf("+");


            else
                  printf("\n");
           }
        }
    }

    void polyAdd(struct node *head1, struct node *head2)
    {
        struct node *ptr1=head1;
struct node *ptr2=head2;         struct
node *head3=NULL;
        while(ptr1!=NULL && ptr2!=NULL)
        {
           if(ptr1->expo == ptr2->expo)
           {
           head3=insert(head3,ptr1->coeff+ptr2->coeff,ptr1->expo);
ptr1=ptr1->next;                ptr2=ptr2->next;
           }
           else if(ptr1->expo > ptr2->expo)
           {
              head3=insert(head3,ptr1->coeff,ptr1->expo);
              ptr1=ptr1->next;
           }
           else if(ptr1->expo < ptr2->expo)
           {
              head3=insert(head3,ptr2->coeff,ptr2->expo);
              ptr2=ptr2->next;
           }
        }
        while(ptr1!=NULL)
        {
           head3=insert(head3,ptr1->coeff,ptr1->expo);
           ptr1=ptr1->next;
        }
        while(ptr2!=NULL)
        {
           head3=insert(head3,ptr2->coeff,ptr2->expo);
           ptr2=ptr2->next;
        }
```

```
      printf("Added Polynomial is: ") ;
print(head3);
   }
   int main()
   {
      struct node *head1=NULL;
struct node *head2=NULL;
      printf("Enter first polynomial\n");

      head1=create(head1);
      printf("Enter second polynomial\n");
head2=create(head2);
polyAdd(head1,head2);        return 0;

   }
```

OUTPUT:

```
aiml231501167@cselab:~$ gcc program3.c
aiml231501167@cselab:~$ ./a.out
Enter the values for first polynomial :
Enter the coefficient : 2
Enter the power : 2
Enter 1 to continue : 1
Enter the coefficient : 3
Enter the power : 3
Enter 1 to continue : 0
The polynomial equation is : 2x^2+3x^3
Enter the values for second polynomial :
Enter the coefficient : 2
Enter the power : 2
Enter 1 to continue : 1
Enter the coefficient : 5
Enter the power : 3
Enter 1 to continue : 0
The polynomial equation is : 2x^2+5x^3
The polynomial equation addition result is : 4x^2+8x^3aiml231501167@cselab:~$
```

RESULT: Thus the program was successfully executed.