**DAY 7 – POLYNOMIAL**

9. Write a menu driven C program. to representation polynomials using

linked list and perform (i) polynomial addition and (ii) polynomial

multiplication.

**PROGRAM**

#include<stdio.h>

#include<stdlib.h>

struct node

{

    int coef;

    int expo;

    struct node \*link;

};

struct node \*create(struct node \*);

struct node \*insert\_s(struct node \*,int,int);

struct node \*insert(struct node \*,int,int);

void display(struct node \**ptr*);

void poly\_add(struct node \*,struct node \*);

void poly\_mult(struct node \*,struct node \*);

void main( )

{

    struct node \*start1=NULL,\*start2=NULL;

    printf("\*\*\*\*\* Enter polynomial 1 \*\*\*\*\*\n");

    start1=create(start1);

    printf("\*\*\*\*\* Enter polynomial 2 \*\*\*\*\*\n");

    start2=create(start2);

    printf("Polynomial 1 is : ");

    display(start1);

    printf("Polynomial 2 is : ");

    display(start2);

    poly\_add(start1, start2);

    poly\_mult(start1, start2);

}

*//structure - create*

struct node \*create(struct node \**start*)

{

    int i,n,ex;

    int co;

    printf("Enter the number of terms : ");

    scanf("%d",&n);

    for(i=1;i<=n;i++)

    {

        printf("term %d Coeficient: ",i);

        scanf("%d",&co);

        printf("term %d Expontnt: ",i);

        scanf("%d",&ex);

*start*=insert\_s(*start*,co,ex);

    }

    return *start*;

}

*//structure - insert*

struct node \*insert\_s(struct node \**start*,int *co*,int *ex*)

{

    struct node \*ptr,\*tmp;

    tmp=(struct node \*)malloc(sizeof(struct node));

    tmp->coef=*co*;

    tmp->expo=*ex*;

*//list empty or exp greater than first one*

    if(*start*==NULL || *ex* > *start*->expo)

    {

        tmp->link=*start*;

*start*=tmp;

    }

    else

    {

        ptr=*start*;

        while(ptr->link!=NULL && ptr->link->expo >= *ex*)

        ptr=ptr->link;

        tmp->link=ptr->link;

        ptr->link=tmp;

    }

    return *start*;

}

*//structure - insert*

struct node \*insert(struct node \**start*,int *co*,int *ex*)

{

    struct node \*ptr,\*tmp;

    tmp=(struct node \*)malloc(sizeof(struct node));

    tmp->coef=*co*;

    tmp->expo=*ex*;

*//If list is empty*

    if(*start*==NULL)

    {

        tmp->link=*start*;

*start*=tmp;

    }

    else */\*Insert at the end of the list\*/*

    {

        ptr=*start*;

        while(ptr->link!=NULL)

        ptr=ptr->link;

        tmp->link=ptr->link;

        ptr->link=tmp;

    }

    return *start*;

}

*//display*

void display(struct node \**ptr*)

{

    if(*ptr*==NULL)

    {

        printf("Zero polynomial\n");

        return;

    }

    while(*ptr*!=NULL)

    {

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

*ptr*=*ptr*->link;

        if(*ptr*!=NULL)

        printf(" + ");

        else

        printf("\n");

    }

}

void poly\_add(struct node \**p1*,struct node \**p2*)

{

    struct node \*start3;

    start3=NULL;

    while(*p1*!=NULL && *p2*!=NULL)

    {

    if(*p1*->expo > *p2*->expo)

    {

    start3=insert(start3,*p1*->coef,*p1*->expo);

*p1*=*p1*->link;

    }

    else if(*p2*->expo > *p1*->expo)

    {

    start3=insert(start3,*p2*->coef,*p2*->expo);

*p2*=*p2*->link;

    }

    else if(*p1*->expo==*p2*->expo)

    {

    start3=insert(start3,*p1*->coef+*p2*->coef,*p1*->expo);

*p1*=*p1*->link;

*p2*=*p2*->link;

    }

    }

*/\*if poly2 has finished and elements left in poly1\*/*

    while(*p1*!=NULL)

    {

    start3=insert(start3,*p1*->coef,*p1*->expo);

*p1*=*p1*->link;

    }

*/\*if poly1 has finished and elements left in poly2\*/*

    while(*p2*!=NULL)

    {

    start3=insert(start3,*p2*->coef,*p2*->expo);

*p2*=*p2*->link;

    }

    printf("Added polynomial is : ");

    display(start3);

}

void poly\_mult(struct node \**p1*, struct node \**p2*)

{

 struct node \*start3;

 struct node \*p2\_beg = *p2*;

 start3=NULL;

 if(*p1*==NULL || *p2*==NULL)

 {

 printf("Multiplied polynomial is zero polynomial\n");

 return;

 }

 while(*p1*!=NULL)

 {

*p2*=p2\_beg;

 while(*p2*!=NULL)

 {

 start3=insert\_s(start3,*p1*->coef\**p2*->coef,*p1*->expo+*p2*->expo);

*p2*=*p2*->link;

 }

*p1*=*p1*->link;

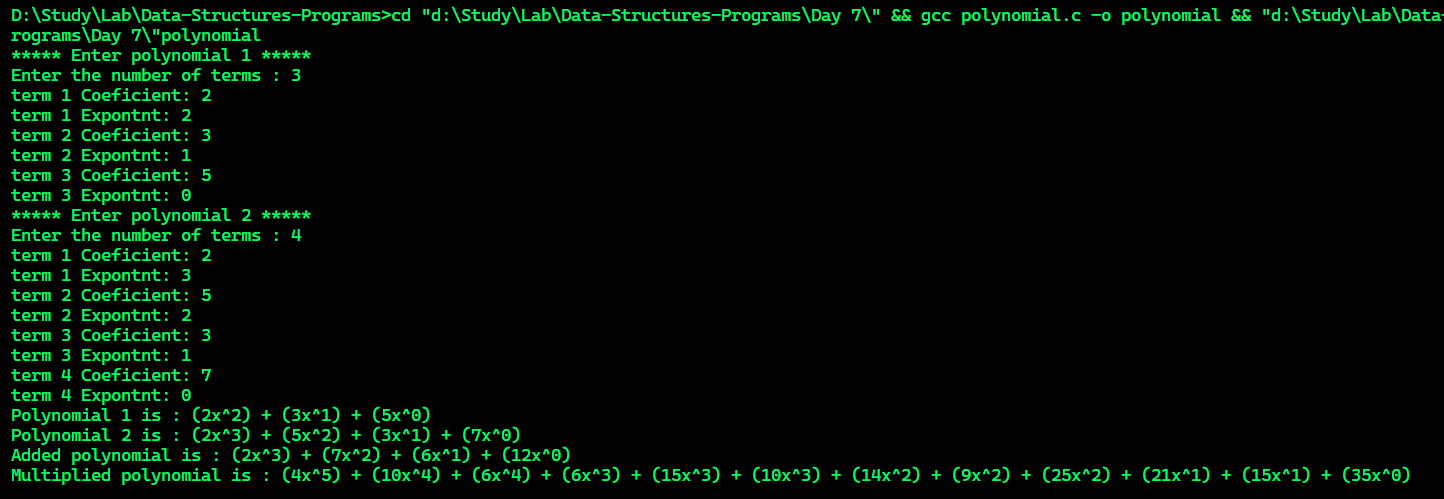
 }

 printf("Multiplied polynomial is : ");

 display(start3);

}

**OUTPUT**

****