Assignment 2

Pratham Prakash Gudadhe

BT22CSE054

CSE SEC A

POLYNOMIAL MULTIPLICATION

#include<stdio.h>

#include<stdlib.h>

*struct* node{

*int* coe;

*int* exp;

*struct* node \*link;

};

*struct* node \*insert(*struct* node \**head*,*int* *coe*,*int* *exp*){

*struct* node \*temp;

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

    new->coe=*coe*;

    new->exp=*exp*;

    new->link=NULL;

    if(*head*==NULL || *exp* > *head*->exp){

        new->link=*head*;

*head*=new;

    }

    else if(*exp* == *head*->exp){

*head*->coe+=*coe*;

        free(new);

    }

    else{

        temp=*head*;

        while(temp->link!=NULL && temp->link->exp >= *exp*){

            temp=temp->link;

        }

        if(temp->exp==*exp*){

            temp->coe+=*coe*;

            free(new);

        }

        else{

        new->link=temp->link;

        temp->link=new;

        }

    }

    return *head*;

}

*struct* node\* createlist(*struct* node \**head*){

*int* n,exp,coe;

    printf("Enter Number of terms = ");

    scanf("%d",&n);

    for(*int* i=0;i<n;i++){

        printf("Enter coefficient = ");

        scanf("%d",&coe);

        printf("Enter Exponent = ");

        scanf("%d",&exp);

*head*=insert(*head*,coe,exp);

    }

    return *head*;

}

*void* printdata(*struct* node \**head*){

    if(*head*==NULL){

        printf("Linked List is empty");

    }

*struct* node \*ptr=NULL;

    ptr=*head*;

    while(ptr!=NULL){

        printf("%dx^%d",ptr->coe,ptr->exp);

        if(ptr->link!=NULL){

            printf(" + ");

        }

        ptr=ptr->link;

    }

}

*struct* node \*multiply(*struct* node \**head1*,*struct* node \**head2*){

*struct* node \*t1=NULL,\*t2=NULL;

    t1=*head1*;

    t2=*head2*;

*struct* node \*head3=NULL;

    while(t2!=NULL){

        while(t1!=NULL){

          head3=insert(head3,(t1->coe)\*(t2->coe),(t1->exp)+(t2->exp));

          t1=t1->link;

        }

        t2=t2->link;

        t1=*head1*;

    }

    return head3;

}

*void* nodedetail(*struct* node \**head*){

*struct* node \*ptr=*head*;

    while(ptr!=NULL){

        if(ptr->link==NULL){

            printf("%d,%d,NULL\n",ptr->coe,ptr->exp);

        }

        else{

            printf("%d,%d,%p\n",ptr->coe,ptr->exp,ptr->link);

        }

        ptr=ptr->link;

    }

}

*int* main(){

*struct* node \*head1=NULL;

*struct* node \*head2=NULL;

*struct* node \*head3=NULL;

    printf("Enter for Polynomial 1 :-\n\n");

    head1=createlist(head1);

    printf("Enter for Polynomial 2 :-\n\n");

    head2=createlist(head2);

    printf("Polynomial 1 = ");printdata(head1);

    printf("\n");

    printf("Polynomial 2 = ");printdata(head2);

    printf("\n");

    head3=multiply(head1,head2);

    printf("Multiplication = ");printdata(head3);

    printf("\n\n");

    printf("Polynomial A :-\n");nodedetail(head1);printf("\n");

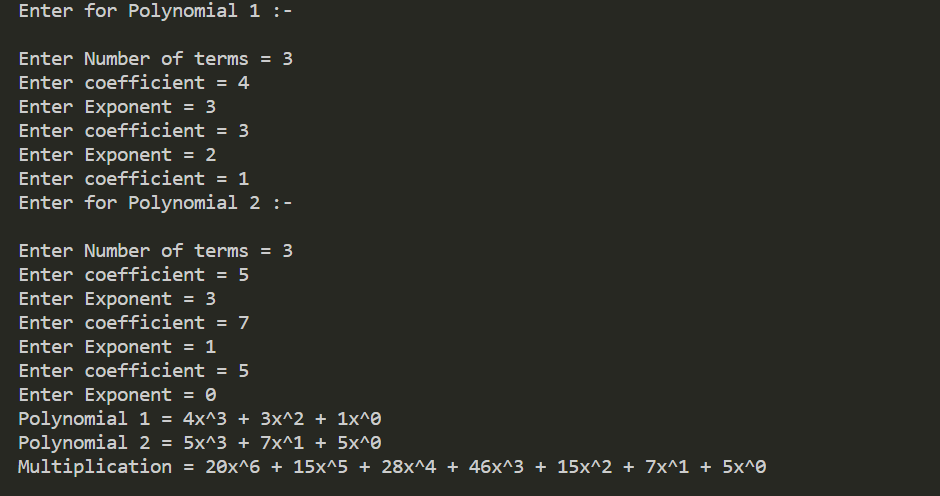
    printf("Polynomial B :-\n");nodedetail(head2);printf("\n");

    printf("Resultant Linked List :-\n");nodedetail(head3);

    return 0;

}

OUTPUTS :-



LONG INTEGER ADDITION

#include<stdio.h>

#include<stdlib.h>

*struct* node{

*int* data;

*struct* node \*link;

};

*struct* node \*insert(*struct* node \**head*,*int* *data*){

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

    temp->data=*data*;

    temp->link=NULL;

    if(*head*==NULL){

*head*=temp;

    }

    else{

*struct* node \*ptr;

        ptr=*head*;

        while(ptr->link!=NULL){

            ptr=ptr->link;

        }

        ptr->link=temp;

    }

    return *head*;

}

*struct* node \*createlist(*int* *data*){

*struct* node \*head=NULL;

    while(*data*>0){

        head=insert(head,*data*%10);

*data*/=10;

    }

    return head;

}

*struct* node \*add(*struct* node \**head1*,*struct* node \**head2*){

*struct* node \*t1,\*t2;

*struct* node \*head3=NULL;

    t1=*head1*;

    t2=*head2*;

*int* carry=0;

    while(t1!=NULL && t2!=NULL){

        if(t1->data+t2->data+carry>9){

            head3=insert(head3,(t1->data+t2->data+carry)%10);

            carry=1;

        }

        else{

            head3=insert(head3,t1->data+t2->data+carry);

            carry=0;

        }

        t1=t1->link;

        t2=t2->link;

    }

    while(t2!=NULL){

            if(t2->data+carry > 9){

                head3=insert(head3,(t2->data+carry)%10);

                carry=1;

            }

            else{

                head3=insert(head3,t2->data+carry);

            }

            t2=t2->link;

    }

    while(t1!=NULL){

            if(t1->data+carry > 9){

                head3=insert(head3,(t1->data+carry)%10);

                carry=1;

            }

            else{

                head3=insert(head3,t1->data+carry);

            }

            t1=t1->link;

    }

    return head3;

}

*void* printlist(*struct* node \**head*){

*struct* node \*temp=*head*;

    while(temp!=NULL){

        printf("%d",temp->data);

        temp=temp->link;

    }

}

*struct* node \*reverse(*struct* node \**head*) {

*struct* node \*current = *head*;

*struct* node \*next, \*prev = NULL;

    while (current != NULL) {

        next = current->link;

        current->link = prev;

        prev = current;

        current = next;

    }

*head* = prev;

    return *head*;

}

*int* main(){

*int* x,y;

*struct* node \*head1=malloc(sizeof(*struct* node));

*struct* node \*head2=malloc(sizeof(*struct* node));

*struct* node \*head3=malloc(sizeof(*struct* node));

    printf("Enter first number = ");

    scanf("%d",&x);

    printf("Enter second number = ");

    scanf("%d",&y);

    head1=createlist(x);

    head2=createlist(y);

    head3=add(head1,head2);

    printf("\nResult = ");

    head3=reverse(head3);

    printlist(head3);

    return 0;

}

OUTPUTS:-

