Pratham Prakash Gudadhe

BT22CSE054

Assignment 1

Assignment 1

#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->coe==*coe*){

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

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

*struct* node \*head3=NULL;

    t1=*head1*;

    t2=*head2*;

    while(t1!=NULL && t2!=NULL){

        if(t1->exp > t2->exp){

            head3=insert(head3,t1->coe,t1->exp);

            t1=t1->link;

        }

        else if(t2->exp > t1->exp){

            head3=insert(head3,t2->coe,t2->exp);

            t2=t2->link;

        }

        else if(t1->exp==t2->exp){

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

            t1=t1->link;

            t2=t2->link;

        }

    }

       while (t1 != NULL) {

        head3 = insert(head3, t1->coe, t1->exp);

        t1 = t1->link;

        }

        // Add the remaining terms from t2, if any

        while (t2 != NULL) {

        head3 = insert(head3, t2->coe, t2->exp);

        t2 = t2->link;

        }

    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=addition(head1,head2);

    printf("Addition = ");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;

}

