**SSN COLLEGE OF ENGINEERING**

**DEPARTMENT OF COMPUTER SCIENCE**

**ASSIGNMENT 3**

**POLYNOMIAL MANIPULATION USING LINKED LIST**

**NAME: S.MADHUMITHA**

**ROLLNO:185001086**

**CSE -B**

**PROGRAM:**

**FILE 1:VARIABLE**

#ifndef \_variable\_h variable.h

#define variable.h

struct poly{

int c;

int e;

struct poly\* next;

};

typedef struct poly node;

#endif

**FILE 2:FUNCTIONS**

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

#include”variable.h”

#ifndef \_functions\_h functions.h

#define \_functions\_h

node\* create(){

node\* newn;

newn=(node\*)malloc(sizeof(struct poly));

newn->next=NULL;

return newn;

}

void insertlast(node\* hd,int cf,int ex){

node\* t,\*p;

p=(node\*)malloc(sizeof(struct poly));

t=hd->next;

p->c=cf;p->e=ex;

while(t->next!=NULL){

t=t->next;

}

p->next=t->next;

t->next=p;

}

void insertpos(node\* hd,int c1,int e1){

node\*t,\*p,\*prev;int flag=0;

prev=hd;

p=(node\*)malloc(sizeof(struct poly));

p->c=c1;p->e=e1;

t=hd->next;

if(t==NULL){insertfirst(hd,c1,e1);flag=1;}

while(t!=NULL){

if((t->e)>e1){

t=t->next;prev=prev->next;

continue;

}

else if(t->e==e1){t->c=t->c+c1;flag=1;

break;

}

else if(t->e<e1){

prev->next=p;

p->next=t;flag=1;break;

}break;

}

if(flag==0)insertlast(hd,c1,e1);

return;

}

void insertfirst(node\* hd,int cf,int ex){

node\* p;p=(node\*)malloc(sizeof(struct poly));

p->c=cf;p->e=ex;

hd->next=p;

p->next=NULL;

}

void mult(node\* a,node\* b,node \*r){

int cf=0,ex=0;

node \*temp;

a=a->next;

b=b->next;

while(a!=NULL){

temp=b;

while(temp!=NULL){

cf=a->c\*temp->c;

ex=a->e+temp->e;

insertpos(r,cf,ex);

temp=temp->next;}

a=a->next;}

return;

}

void add(node\* p, node\* q,node \*r){

p=p->next;q=q->next;

printf("adding...");

while(p!=NULL && q!=NULL){

if(p->e==q->e){

if(p->c+q->c!=0){

r->e=p->e;

r->c=p->c+q->c;

p=p->next;

q=q->next;

}

else{p=p->next;

q=q->next;

continue;

}

}

else if(p->e > q->e)

{

r->e=p->e;

r->c=p->c;

p=p->next;

}

else if(q->e >p->e){

r->e=q->e;

r->c=q->c;

q=q->next;

}

r->next=malloc(sizeof(struct poly));

r=r->next;

r->next=NULL;

}

if(p!=NULL){

while(p!=NULL){

r->c=p->c;

r->e=p->e;

r->next=malloc(sizeof(struct poly));

r=r->next;

r->next=NULL;

p=p->next;

}

}

if(q!=NULL){

while(q!=NULL){

r->c=q->c;

r->e=q->e;

r->next=malloc(sizeof(struct poly));

r=r->next;

r->next=NULL;

q=q->next;

}

}

r->next=NULL;

return;

}

void print(node\* temp){

temp=temp->next;

while(temp!=NULL){

printf(" coefficient:%d ", temp->c);

printf(" exponent:%d ", temp->e);

printf("\n");

temp=temp->next;

}

}

void printadd(node\* a){

node \*tmp;

tmp=a;

while(tmp->next!=NULL){

printf("coef: %d ", tmp->c);

printf("exp: %d ", tmp->e);

printf("\n");

tmp=tmp->next;

}

}

#endif

**FILE 3:MAIN FUNCTION**

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

int main(){

int i=0,c,e;

int choice;

char ch='s';

node \*p1,\*p2,\*p3,\*p4;

p1=create();

p2=create();

p3=create();

p4=create();

printf("polynomial1\n");

while(ch=='s'||ch=='S'){

printf("coeff");scanf("%d",&c);

printf("exponent");scanf("%d",&e);

if(i==0)insertfirst(p1,c,e);

else

insertlast(p1,c,e);

printf("choice");

scanf(" %c",&ch);i=i+1;

}

printf("polynomial2\n");

ch='s';

i=0;

while(ch=='s'||ch=='S'){

printf("coeff");scanf("%d",&c);

printf("exponent");scanf("%d",&e);

if(i==0)insertfirst(p2,c,e);

else

insertlast(p2,c,e);

printf("choice");

scanf(" %c",&ch);i=i+1;

}ch='s';

while(ch=='s' ||ch=='S')

{printf("MENU\n1.POLYNOMIAL ADDITION\n2.POLYNOMIAL MULTIPLICATION\n");

scanf("%d",&choice);

switch(choice){

case 1:add(p1,p2,p3);

printadd(p3);

break;

case 2:mult(p1,p2,p4);

print(p4);}printf("continue or exit");

scanf(" %c",&ch);}

printf("thank you");

return 0;

}

**OUTPUT:**

polynomial1

coeff3

exponent12

choices

coeff8

exponent8

choices

coeff-22

exponent4

choices

coeff3

exponent1

choices

coeff-7

exponent0

choicea

polynomial2

coeff7

exponent14

choices

coeff-10

exponent9

choices

coeff-8

exponent8

choices

coeff6

exponent5

choices

coeff-9

exponent1

choice0

MENU

1.POLYNOMIAL ADDITION

2.POLYNOMIAL MULTIPLICATION

2

coefficient:21 exponent:26 **(POLYNOMIAL MULTIPLICATION)**

coefficient:56 exponent:22

coefficient:-30 exponent:21

coefficient:-24 exponent:20

coefficient:-154 exponent:18

coefficient:-62 exponent:17

coefficient:-64 exponent:16

coefficient:21 exponent:15

coefficient:-49 exponent:14

coefficient:241 exponent:13

coefficient:176 exponent:12

coefficient:-30 exponent:10

coefficient:-158 exponent:9

coefficient:56 exponent:8

coefficient:18 exponent:6

coefficient:156 exponent:5

coefficient:-27 exponent:2

coefficient:63 exponent:1

continue or exit s

MENU

1.POLYNOMIAL ADDITION

2.POLYNOMIAL MULTIPLICATION

1

adding...coef: 7 exp: 14 **(POLYNOMIAL ADDITION)**

coef: 3 exp: 12

coef: -10 exp: 9

coef: 6 exp: 5

coef: -22 exp: 4

coef: -6 exp: 1

coef: -7 exp: 0

continue or exit n

thank you

Process returned 0 (0x0) execution time : 76.022 s

Press any key to continue.