/\*

GOPIKRISHNA V

S3 CSE A

52

Polynomial Addition of One Variable

\*/

#include<stdio.h>

#include<stdlib.h>

struct node

{

int exp;

int coef;

struct node \*link;

}\*phead,\*qhead,\*rhead=NULL,\*p,\*q,\*r;

void display(struct node \*\*head,int poly)

{

struct node \*ptr;

printf("Polynomial %d Expression = ",poly);

ptr = \*head;

if (\*head == NULL)

{

printf("Empty List ");

}

else

{

while(ptr != NULL)

{

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

ptr = ptr -> link;

}

}

printf("\b \n");

}

void insert(struct node \*\*head, int coef, int exp)

{

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

temp -> exp = exp;

temp -> coef = coef;

temp -> link = NULL;

ptr = \*head;

if (\*head == NULL)

{

\*head = temp;

}

else

{

while( ptr -> link != NULL)

{

ptr = ptr -> link;

}

ptr -> link = temp;

}

}

void polyadd()

{

p=phead;

q=qhead;

rhead=NULL;

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

{

if(p->exp==q->exp)

{

insert(&rhead,p->coef+q->coef,p->exp);

p=p->link;

q=q->link;

}

else

{

if(p->exp > q->exp)

{

insert(&rhead,p->coef,p->exp);

p=p->link;

}

else

{

insert(&rhead,q->coef,q->exp);

q=q->link;

}

}

}

while(p!=NULL)

{

insert(&rhead,p->coef,p->exp);

p=p->link;

}

while(q!=NULL)

{

insert(&rhead,q->coef,q->exp);

q=q->link;

}

display(&rhead,0);

}

void main()

{

int i,len,coef,exp;

printf("## ONE VARIABLE POLYNOMIAL ADDITION ##\n");

printf("FIRST POLYNOMIAL\n");

printf("Length of Linked List = ");

scanf("%d",&len);

printf("\n");

for (i = 1; i <= len; i++)

{

printf("Element %d --> Coefficient >> ",i);

scanf("%d",&coef);

printf("Element %d --> Exponent >> ",i);

scanf("%d",&exp);

printf("\n");

insert(&phead,coef,exp);

}

display(&phead,1);

printf("\nSECOND POLYNOMIAL\n");

printf("Length of Linked List = ");

scanf("%d",&len);

printf("\n");

for (i = 1; i <= len; i++)

{

printf("Element %d --> Coefficient >> ",i);

scanf("%d",&coef);

printf("Element %d --> Exponent >> ",i);

scanf("%d",&exp);

printf("\n");

insert(&qhead,coef,exp);

}

display(&qhead,2);

polyadd();

}

**OUTPUT**

