

DS LAB-APPLICATIONS OF LINKED LIST-2

1BM19CS079

LIKITHA B

Addition of Two Polynomials-

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#include<math.h>
```

```
struct node{
```

```
float cf;
```

```
float px;
```

```
float py;
```

```
int flag;
```

```
struct node *link;
```

```
};
```

```
typedef struct node *NODE;
```

```
NODE getnode()
```

```
{
```

```
    NODE x;
```

```
    x=(NODE)malloc(sizeof(struct node));
```

```
    if(x==NULL)
```

```
    {
```

```
printf("out of memory");  
exit(0);  
}  
return x;  
}
```

```
NODE insert_rear(float cf,float x,float y,NODE head)  
{  
    NODE temp,cur;  
    int flag;  
    temp=getnode();  
    temp->cf=cf;  
    temp->px=x;  
    temp->py=y;  
    temp->flag=0;  
    cur=head->link;  
    while(cur->link!=head)  
        cur=cur->link;  
    cur->link=temp;  
    temp->link=head;  
    return head;  
}
```

```
NODE read_poly(NODE head){  
  
    int i;  
  
    float cf,px,py;  
  
    printf("Enter the coefficient as -999 to end the polynomial\n");  
  
    for(i=1;;i++){  
  
        printf("enter the %d term\n",i);  
  
        printf("coeff:\n");  
  
        scanf("%f",&cf);  
  
        if(cf== -999)  
  
            break;  
  
        printf("pow x:\n");  
  
        scanf("%f",&px);  
  
        printf("pow y:\n");  
  
        scanf("%f",&py);  
  
        head=insert_rear(cf,px,py,head);  
  
    }  
  
    return head;  
  
}  
  
void display(NODE head){  
  
    NODE temp;  
  
    if(head->link==head)
```

```

{
printf("polynomial doesn not exist\n");
return;
}
temp=head->link;
while(temp!=head)
{
printf("%5.2fx^%3.1fy^%3.1f\t",temp->cf,temp->px,temp->py);
temp=temp->link;
}
printf("\n");
}

```

```

NODE add_poly(NODE h1,NODE h2, NODE h3){
NODE p1,p2;
int x1,x2,y1,y2,cf1,cf2,cf;
p1=h1->link;
while(p1!=h1){
x1=p1->px;
y1=p1->py;
cf1=p1->cf;
p2=h2->link;

```

```
while(p2!=h2)
{
x2=p2->px;
y2=p2->py;
cf2=p2->cf;
if(x1==x2 && y1==y2)
break;
p2=p2->link;

}
if(p2!=h2){
cf=cf1+cf2;
p2->flag=1;
if(cf!=0)
h3=insert_rear(cf,x1,y1,h3);
}
else
h3=insert_rear(cf1,x1,y1,h3);
p1=p1->link;
}
p2=h2->link;
while(p2!=h2)
```

```
{  
if(p2->flag==0)  
{  
h3=insert_rear(p2->cf,p2->px,p2->py,h3);  
}  
p2=p2->link;  
}  
return h3;  
}  
  
int main()  
{  
NODE h1,h2,h3;  
h1=getnode();  
h2=getnode();  
h3=getnode();  
h1->link=h1;  
h2->link=h2;  
h3->link=h3;  
  
printf("Enter the first polynomial\n");  
h1=read_poly(h1);  
  
printf("Enter the second polynomial\n");  
h2=read_poly(h2);
```

```
h3=add_poly(h1,h2,h3);

printf("the first polynomial\n");

display(h1);

printf("the second polynomial\n");

display(h2);

printf("the sum of the polynomials\n");

display(h3);

return 0;

}
```

```
Enter the first polynomial
Enter the coefficient as -999 to end the polynomial
enter the 1 term
coeff:
5
pow x:
2
pow y:
1
enter the 2 term
coeff:
4
pow x:
2
pow y:
1
enter the 3 term
coeff:
8
pow x:
5
pow y:
9
enter the 4 term
coeff:
6
pow x:
7
pow y:
4
enter the 5 term
coeff:
```

```
pow y:
4
enter the 5 term
coeff:
1
pow x:
8
pow y:
9
enter the 6 term
coeff:
-999
Enter the second polynomial
Enter the coefficient as -999 to end the polynomial
enter the 1 term
coeff:
5
pow x:

8
pow y:
9
enter the 2 term
coeff:
3
pow x:
9
pow y:
3
enter the 3 term
coeff:
```



```

pow y:
3
enter the 3 term
coeff:
7
pow x:
4
pow y:
5
enter the 4 term
coeff:
8
pow x:
9
pow y:
3
enter the 5 term
coeff:
7
pow x:
4
pow y:
7
enter the 6 term
coeff:
5
pow x:
1
pow y:
0
enter the 7 term
coeff:

```

```

4
pow y:
7
enter the 6 term
coeff:
5
pow x:
1
pow y:
0
enter the 7 term
coeff:
-9999
pow x:
7
pow y:
4
enter the 8 term
coeff:
-999
the first polynomial
5.00x^2.0y^1.0 4.00x^2.0y^1.0 8.00x^5.0y^9.0 6.00x^7.0y^4.0 1.00x^8.0y^9.0
the second polynomial
5.00x^8.0y^9.0 3.00x^9.0y^3.0 7.00x^4.0y^5.0 8.00x^9.0y^3.0 7.00x^4.0y^7.0
5.00x^1.0y^0.0 -9999.00x^7.0y^4.0
the sum of the polynomials
5.00x^2.0y^1.0 4.00x^2.0y^1.0 8.00x^5.0y^9.0 -9993.00x^7.0y^4.0 6.00x^8.0y^9.0
3.00x^9.0y^3.0 7.00x^4.0y^5.0 8.00x^9.0y^3.0 7.00x^4.0y^7.0 5.00x^1.0y^0.0

...Program finished with exit code 0
Press ENTER to exit console.

```

PROGRAM-2

Evaluation Of polynomial-

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <math.h>
```

```
struct node
```

```
{
```

```
    float cf;
```

```
    float px;
```

```
    float py;
```

```
    struct node *link;
```

```
};
```

```
typedef struct node *NODE;
```

```
NODE getnode()
```

```
{
```

```
    NODE x;
```

```
    x=(NODE)malloc(sizeof(struct node));
```

```
    if (x==NULL)
```

```
    {
```

```
        printf("Memory full\n");
```

```
        exit(0);
```

```
    }
```

```
    return x;
```

```
}
```

```
NODE insert_rear(float cf,float x,float y,NODE first)
```

```
{
```

```
    NODE temp,cur;
```

```
    temp=getnode();
```

```
    temp->cf=cf;
```

```
    temp->px=x;
```

```
    temp->py=y;
```

```
    temp->link=NULL;
```

```
    if (first==NULL)
```

```
    {
```

```
        return temp;
```

```
    }
```

```
    cur=first;
```

```
    while (cur->link!=NULL)
```

```
    {
```

```
        cur=cur->link;
```

```
    }
```

```
    cur->link=temp;
```

```
    return first;
```

```
}
```

```
NODE read_poly(NODE first)
```

```

{
    int i;

    float cf,px,py;

    printf("Enter -999 to end the polynomial:\n");
    for (i=1;;i++)
    {
        printf("Enter %d term:\n",i);
        printf("Coefficient:\n");
        scanf("%f",&cf);
        if (cf==-999)
        {
            break;
        }
        printf("Power of x:\n");
        scanf("%f",&px);
        printf("Power of y:\n");
        scanf("%f",&py);
        first=insert_rear(cf,px,py,first);
    }
    return first;
}

float evaluate_polynomial(NODE first)

```

```

{
    float x,y,sum=0;

    NODE polynomial;

    printf("Enter the values of x and y:\n");

    scanf("%f%f",&x,&y);

    polynomial=first;

    while (polynomial!=NULL)
    {
        sum=sum+polynomial->cf*pow(x,polynomial->px)*pow(y,polynomial->py);

        polynomial=polynomial->link;
    }

    return sum;
}

void display(NODE first)
{
    NODE temp;

    if (first==NULL)
    {
        printf("Polynomial does not exist\n");
    }

    else
    {

```

```

temp=first;
while (temp->link!=NULL)
{
printf("(%5.2fx^%3.2fy^%3.2f)\t+",temp->cf,temp->px,temp->py);
temp=temp->link;
}

printf("(%5.2fx^%3.2fy^%3.2f)\n",temp->cf,temp->px,temp->py);
}
}

int main(){
    NODE first;

    float res;

    first=NULL;

    printf("Enter the polynomial: \n");

    first = read_poly(first);

    res = evaluate_polynomial(first);

    printf("Polynomial is: \n");

    display(first);

    printf("Result is %f\n",res);

    return 0;
}

```

```

Enter the polynomial:
Enter -999 to end the polynomial:
Enter 1 term:
Coefficient:
7
Power of x:
5
Power of y:
6
Enter 2 term:
Coefficient:
6
Power of x:
2
Power of y:
4
Enter 3 term:
Coefficient:
3
Power of x:
7
Power of y:
9
Enter 4 term:
Coefficient:
5
Power of x:
4
Power of y:
3
Enter 5 term:

```

```

7
Power of y:
9
Enter 4 term:
Coefficient:
5
Power of x:
4
Power of y:
3
Enter 5 term:
Coefficient:
2
Power of x:
6
Power of y:
1
Enter 6 term:
Coefficient:
-999
Enter the values of x and y:
5
6
Polynomial is:
( 7.00x^5.00y^6.00)      +( 6.00x^2.00y^4.00)      +( 3.00x^7.00y^9.00)      +( 5.00x^4.00y^3.00)
( 2.00x^6.00y^1.00)
Result is 2362981744640.000000

...Program finished with exit code 0
Press ENTER to exit console.

```