

Addition of polynomials.

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

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

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

```
struct node
```

```
{
```

```
    float cf; float px, py;
```

```
    int flag;
```

```
    struct node *link;
```

```
};
```

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

```
NODE getnode()
```

```
{
```

```
    NODE x;
```

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

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

```
    {
```

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

```
        exit (0);
```

```
    }
```

```
    return x;
```

```
}
```

```
NODE insert-rear ( float cf, float px, float y, NODE first)
```

```
{ NODE temp, cur;
```

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

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

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

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

```
    temp->flag = 0;
```

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

```

    } return temp; }
    cur = first;
    while (cur->link != NULL)
    {
        cur = cur->link;
    }
    cur->link = temp;
    return first;
}

NODE read read-poly(NODE first)
{
    int i;
    float cf, px, py;
    pf("Enter -999 to end the polynomial:\n");
    for (i = 1; i++)
    {
        pf("Enter o/p d term: ", i);
        pf(" Co-efficient ");
        scanf("%f", &cf);
        (cf != -999)
        {
            break;
        }
        pf("Power of x:");
        pf("%d", &px);
        pf("Power of y:");
        pf("%d", &py);
        first = insert-xas(cf, px, py, first);
    }
    return first;
}

```

```
void display(NODE first)
{
```

```
    NODE temp;
```

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

```
    {
        printf("Polynomial does not exist");
    }
```

```
    else
```

```
    { temp = first;
```

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

```
    {
```

```
        printf("%d.%d.%d/x^%d.%d.%d/y^%d.%d.%d)lt + " temp->cf,
               temp->px, temp->py);
```

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

```
    }
```

```
        printf("%d.%d.%d/x^%d.%d.%d/y^%d.%d.%d)\n", temp->cf, temp->
               px, temp->py);
```

```
    }
```

```
}
```

```
NODE add_poly(NODE f1, NODE f2, NODE f3)
```

```
{
    NODE p1, p2;
```

```
    int x1, x2, y1, y2, cf1, cf2, cof;
```

```
    p1 = f1;
```

```
    while (p1 != NULL)
```

```
    {
        x1 = p1->px;
```

```
        y1 = p1->py;
```

```
        cf1 = p1->cf;
```

```
        p2 = f2;
```

```
        while (p2 != NULL)
```

```
        {
```


$x2 = p2 \rightarrow px;$

$y2 = p2 \rightarrow py;$

$f2 = p2 \rightarrow f;$

if $(x1 == x2 \ \&\& \ y1 == y2)$

{

break;

$p2 = p2 \rightarrow \text{link};$

}

if $(p2 != \text{NULL})$

{

$cof = cf1 + cf2;$

$p2 \rightarrow \text{flag} = 1$

if $(cof != 0)$

{

$f3 = \text{insert_rear}(cof, x1, y1, f3);$

}

}

else.

{

$f3 = \text{insert_rear}(cf1, x1, y1, f3);$

}

$p1 = p1 \rightarrow \text{link};$

}

$p2 = f2;$

while $(p2 != \text{NULL})$

{

if $(p2 \rightarrow \text{flag} == 0)$

{

$f3 = \text{insert_rear}(p2 \rightarrow cf, p2 \rightarrow px, p2 \rightarrow py, f3);$

}

$p2 = p2 \rightarrow \text{link};$

}

return $f3;$

}

```
int main()
```

```
{
```

```
    NODE f1, f2, f3;
```

```
    f1 = NULL;
```

```
    f2 = NULL;
```

```
    f3 = NULL;
```

```
    printf("Enter first poly: \n");
```

```
    f1 = read_poly(f1);
```

```
    printf("Enter second poly \n");
```

```
    f2 = read_poly(f2);
```

```
    f3 = add_poly(f1, f2, f3);
```

```
    printf("The first poly is:");
```

```
    display(f1);
```

```
    printf("The second poly is:");
```

```
    display(f2);
```

```
    printf("The sum of 2 polynomials is:");
```

```
    display(f3);
```

```
    return 0;
```

```
}
```