一元多项式加减法

加法：算法描述：减法与加法是类似的，只要将a + b 转变为a + （-b）即可，所以我们这边只以加法为例。

一元多项式的格式是固定的，不妨将链表的头节点定为一元多项式最高幂次项的系数，将节点的值定为幂次，将幂次为0的项定为头节点，向右延伸。在到达最高幂次项之前，所有的幂次位都要设为节点，系数可以用0表示。通过函数实现节点的相加。

代码：

class poly:

\_\_a = [0]\*20 #存放第一个输入的多项式和运算结果

\_\_b = [0]\*20#存放输入的多项式

\_\_result = [0]\*20#结果

def \_\_Input(self,f):

n = input('依序输入二项式的系数和指数（指数小于10）：').split()

for i in range(int(len(n)/2)):

f[ int(n[2\*i+1])] = int(n[2\*i])

self.\_\_output(f)

def \_\_add(self,a,b): #加法函数

return [a[i]+b[i] for i in range(20)]

def \_\_minus(self,a,b): #减法函数

return [a[i]-b[i] for i in range(20)]

def \_\_mul(self,a,b):

self.\_\_result = [0]\*20

for i in range(10):#第一个循环：b分别于a[0]到a[9]相乘

for j in range(10): #第二个循环：b[j]\*a[i]

self.\_\_result[i+j] = int(self.\_\_result[i+j]) + int(a[i]\*b[j])

return self.\_\_result

def \_\_output(self,a):#输出多项式

b = ''

for i in range(20):

if a[i]> 0:

b = b+'+'+str(a[i])+'X^'+str(i)

if a[i]<0:

b = b+"-"+str(-a[i])+'X^'+str(i)

print(b[1::])

def control(self):

print ("二项式运算：\n")

self.\_\_Input(self.\_\_a)

while True:

operator = input('请输入运算符（结束运算请输入‘#’）')#self.Input(self.a)

if operator =='#':

return 0

else:

self.\_\_b = [0]\*20

self.\_\_Input(self.\_\_b)

self.\_\_a = {'+':self.\_\_add(self.\_\_a,self.\_\_b),'-':self.\_\_minus(self.\_\_a,self.\_\_b),'\*':self.\_\_mul(self.\_\_a,self.\_\_b)}.get(operator)

print ('计算结果：',end='')

self.\_\_output(self.\_\_a)

POLY = poly() #初始化类

测试：