

```

'''
Série 1 : Rappel
Date : 11-09-2023
Groupe : ipein/sml/GB
'''

# Exercice 2
## Q1
def saisie_deg():
    while 1:
        try:
            n=int(input("donner un entier positif: "))
        except :
            continue
        if n>0:
            break
    return n

## Q2
def saisie_poly(n):
    p = [None] * (n+1)
    # boucle n+1 fois
    for i in range(n+1):
        # step 1: saisie coef de deg i
        while 1:
            try:
                coef = float(input("P[{}]=".format(i)))
            except :# si exception
                #repete la saisie
                continue
            else:# si pas d'exceptions
                if i<n or (i == n and coef != 0):
                    p[i] = coef
                    break # arrêter while

    return p

## Q3
def derive(p):
    return [i * p[i] for i in range(1,len(p))]

## Q4
def opp_poly(p):
    return [-coef for coef in p]

## Q5
def add_poly(p1,p2):
    n1,n2 = len(p1),len(p2)
    p3,p4 = list(p1),list(p2)
    # opérateur ternaire
    if (n2 > n1):
        p3 = p1 +[0]*(n2-n1)
    else :
        p4 = p2 + [0]*(n1-n2)
    return [p3[i]+p4[i] for i in range(len(p4))]

## Q6
def mul_poly(p1, p2):
    n1,n2 = len(p1),len(p2)
    p = [0] * (n1+n2-1)

    for i in range(n1):
        for j in range(n2):
            p[i+j] += p1[i] * p2[j]

    return p

def poly(p,k):
    assert type(p)==list and len(p)>1, "poly invalid"
    assert k>=0 , "k incorrect"
    if k==0:
        return p
    elif k==1:
        return opp_poly(derive(p))
    else:#k>1
        for i in range(2,k+1):
            sp0 = p
            sp1 = opp_poly(derive(p))
            q = mul_poly(sp1,sp0)
            sp2 = add_poly(mul_poly(q , sp1) , sp0)
        return sp2

## prog principal
'''

```

```
n = saisie_deg()
print("n=",n)
p = saisie_poly(n)
print(p)
'''
n=6
p =[1,1,-5,1,-2,0,6]
d = derive(p)
print(add_poly(p,p))
```