

```

'''
Série 1 : rappel
Goupe : SP3/BG
'''
# Exercice 2
def saisie_deg():
    while 1:
        try:
            n=int(input("n="))
            if n>0:
                return n
        except ValueError: # python exception tree
            print("Erreur : n invalid")

def saisie_poly(n):
    p = []
    for i in range(n+1):
        while 1:
            try:
                c = float(input("p[{}]".format(i)))
                if (i<n)or(i==n and c!=0): # != not equal
                    p.append(c)
                    break #stop while
            except :
                continue

    return p

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

def opp_poly(p): return [-c for c in p]

def add_poly(p1,p2):
    p1m, p2m = list(p1), list(p2)
    n1 , n2 = len(p1),len(p2)
    if (n2>n1):
        p1m += [0] *(n2-n1)
    else:
        p2m += [0]*(n1-n2)
    return [p1m[i] + p2m[i] for i in range(len(p1m))]

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

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

    return p

def suite_poly(p,k):
    assert type(k)==int and k>=0, "k invalid"
    assert type(p)==list, "p invalid"

    if k == 0: spk = p
    elif k==1 : spk = opp_poly(derive(p))
    else:
        sp_k = list(p)
        sp_k1 = opp_poly(derive(sp_k))
        for i in range(2,k+1):
            q = mul_poly(sp_k1 , sp_k)
            sp_k2 = add_poly(mul_poly(q, sp_k1) , sp_k)
            sp_k = list(sp_k1)
            sp_k1 = list(sp_k2)
        spk = sp_k2

    return spk

```