

Informatique TP 10 section 3

Gatt Guillaume

#Question 11

```
def somme(P,Q) :  
    if len(P)>len(Q) :  
        k=len(P)-len(Q)  
        L=[0 for i in range(k)]  
        Q= Q +L[ : ]  
    elif len(P)<len(Q) :  
        k=len(Q)-len(P)  
        L=[0 for i in range(k)]  
        P= P +L[ : ]  
    S=[]  
    for i in range(len(P)) :  
        S.append(P[i] +Q[i])  
    return S
```

#Question 12

```
def mult_scal(P,a) :  
    L=[]  
    for i in range(len(P)) :  
        L.append(a*P[i])  
    return L
```

#Question 13

```
def prod(P,Q) :  
    L=[0 for i in range(len(P)+len(Q)-1)]  
    for i in range(len(P)) :  
        for j in range(len(Q)) :  
            L[i+j]+=P[i]*Q[j]  
    return L
```

#Question 14

```
def Lagrange(A,B) :  
    n=len(A)  
    L=[0]  
    for i in range(n):  
        P=[B[i]]  
        for j in range(n):  
            if j!= i :  
                P=prod(P,[-A[j]/(A[i]-A[j]),1/(A[i]-A[j])])  
        L=somme(L,P)  
    return L
```

#Question 15

```
def matrice_poly(A):  
    M=[]  
    for i in range(len(A)) :  
        L=[]  
        for j in range(len(A)):  
            L.append(A[i]**j)  
        M.append(L)  
    return M
```

```
def Lagrange2(A,B) :
    M=matrice_poly(A)
    Y=[[B[i]] for i in range(len(A))]
    return gauss(M,Y)
```

Question 13 :

La fonction **prod(P,Q)** est de complexité avec n la taille de P et m la taille de Q $O(n \times m)$

Question 14 :

La fonction **Lagrange(A,B)** a une complexité $O(n^2)$ avec n la taille de A

Question 15 :

La matrice M associé est :

$$M = \begin{pmatrix} 1 & a_1 & a_1^2 & \dots & a_1^{n-1} \\ 1 & a_2 & a_2^2 & \dots & a_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_{n-1} & (a_{n-1})^2 & \dots & (a_{n-1})^{n-1} \\ 1 & a_n & a_n^2 & \dots & a_n^{n-1} \end{pmatrix}$$