Table des matières


```python
import numpy as np
N = 10 # Constante. On travaille dans IR_N[X]
```

Dans tout le texte, on fixe un entier $N > 0$ et on travaille dans l'ensemble :

$$\mathbf{R}_N[X] = \{P \in \mathbf{R}[X] \quad \deg(P) \leq N\}.$$

I Choix de la structure de données

1. Si $P = a_N X^N + \cdots + a_1 X + a_0$ est un polynôme à coefficients réels de degré au plus N, on décide de le représenter en python par la liste `L` de longueur `N+1` de ses coefficients rangés par *degré croissant* : `L = [a0,...,aN]`.

2. Par exemple, dans $\mathbf{R}_2[X]$, :
- si $P = X^2 - 3X + 2$, sa représentation sous python est `[2,-3,1]`.
- si $P = X - 3$, sa représentation sous python est `[-3,1,0]`.

[Q1.] Écrire une fonction `RN(L,d=N)` (la variable `d` prend ainsi par défaut la valeur `N` déclarée au début du script) prenant en entrée une liste de flottants `L` et qui renvoie en sortie la liste `L` à laquelle on concatène éventuellement à droite une liste de `0` de sorte que la liste de sortie soit de longueur exactement `d+1`. En particulier, si la longueur de `L` excède `N+1`, la fonction renvoie la liste `L`. Par exemple, `RN([1,2],3)` renvoie `[1,2,0,0]`, mais `RN([1,2])` renvoie `[1,2,0]` si on avait fixé `N=2`. Enfin `RN([1,2,2,2,2],3)` renvoie `[1,2,2,2,2]`.

[Q2.] Écrire une fonction `deg(P)` qui prend en entrée une liste de longueur `N+1` représentant un polynôme $P \in \mathbf{R}_N[X]$ et qui :
- renvoie en sortie le degré de P si $P \neq 0$.
- renvoie `-1` si P est nul (on convient que le polynôme nul est de degré -1 sous python).

[Q3.] Écrire une fonction `eval(P,a)` prenant en entrée une liste de flottants représentant un polynôme P, un flottant `a`, et renvoyant en sortie la valeur du réel $P(a)$.

II Opérations sur les polynômes

II 1) Multiplication par un scalaire

[Q4.] Écrire une fonction `dot(alpha,P)` qui prend en entrée un flottant `alpha`, une liste représentant polynôme P et renvoyant en sortie la liste représentant le polynôme αP.

II 2) Somme

[Q5.] Écrire une fonction `somme1(P,Q)` qui prend en entrée deux listes représentant des polynômes P et Q de $\mathbf{R}_N[X]$, et renvoyant en sortie la liste représentant $P+Q$.

[Q6.] Écrire une fonction `somme2(P,Q)` qui prend en entrée deux listes représentant des polynômes P et Q de degré quelconque, et renvoyant en sortie la liste représentant $P+Q$ vue comme un élément d'un espace $\mathbf{R}_d[X]$ adéquat.

II 3) Produit

[Q7.] Écrire une fonction `produit(P,Q)` qui prend en entrée deux listes représentant des polynômes P et Q, et renvoyant en sortie la liste représentant PQ à l'aide de la formule donnant le produit de deux polynômes (dans un espace $\mathbf{R}_d[X]$ adéquat).

[Q8.] (*Autre manière de programmer le produit*).

1. Soit $k \in \mathbf{N}$. Quel est l'effet de la multiplication par le monôme X^k sur [la liste des coefficients d'] un polynôme P ?
2. Programmer une fonction `shift(k,L)` qui prend en entrée un entier k, une liste de flottants L et renvoyant en sortie la liste obtenue par concaténation des listes $\underbrace{[0,\dots,0]}_{k \text{ items}}$ et L.
3. Utiliser les fonctions `dot`, `shift` et `somme1` ou `somme2` pour programmer le produit de deux polynômes de $\mathbf{R}_N[X]$.

[Q9.]

1. Soit $N \in \mathbf{N}$. Quels sont les coefficients du polynôme $(1+X)^N$?
2. Utiliser la fonction `produit` pour écrire une fonction `pascal(N)` donnant la liste des lignes 0 à $N-1$ du triangle de Pascal. Par exemple :

```
console
In [1]: pascal(5)
Out[1]: [[1, 0, 0, 0, 0, 0],
         [1, 1, 0, 0, 0, 0],
         [1, 2, 1, 0, 0, 0],
         [1, 3, 3, 1, 0, 0],
         [1, 4, 6, 4, 1, 0]]
```

III Autour de la méthode de Hörner

III 1) Présentation et programmation

Soit $P = a_nX^n + \cdots + a_1X + a_0$ et $\alpha \in \mathbf{R}$

La méthode de Hörner est une méthode permettant d'économiser le nombre d'opérations dans le calcul de $P(\alpha)$ en remarquant que si α^k est calculé, il peut-être réinvesti dans le calcul des $a_\ell\alpha^\ell$ pour $\ell \geq k$.

Plus explicitement, on pose le calcule de $P(\alpha)$ en écrivant :

$$P(\alpha) = \Big(\big((a_n\alpha + a_{n-1})\alpha + a_{n-2}\big)\ \ldots\Big)\alpha + a_0.$$

Précisément, l'expression ci-dessus se résume formellement de la manière suivante : si on définit pour tout entier k tel que $0 \leq k \leq n$ les fonctions affines f_k par :

$$x \mapsto f_k(x) = \alpha x + a_{n-k},$$

il se trouve que l'on a :

$$P(\alpha) = u_n$$

où les nombres $u_0, \ldots, u_n$ sont définis par récurrence par :

$$(H) \quad \begin{cases} u_0 = f_0(0) \\ \forall k \in \{0, \ldots, n-1\} \quad u_{k+1} = f_{k+1}(u_k) = \alpha u_k + a_{n-k-1} \end{cases}$$

■ **Remarque 1.** On a une suite récurrente. Néanmoins, à chaque étape, le calcul d'un terme se fait en fonction du précédent à travers une fonction f_k dépendant du rang du terme calculé (alors que classiquement, on applique la même fonction f à tout rang).

[Q10.] Exemple : $n = 3$, et $P = a_3X^3 + a_2X^2 + a_1X + a_0$. Compléter le tableau suivant :

$f_0(x) =$		$u_0 =$	
$f_1(x) =$		$u_1 =$	
$f_2(x) =$		$u_2 =$	
$f_3(x) =$		$u_3 =$	

et vérifier que l'on a bien $u_3 = P(\alpha)$.

[Q11.] Programmer une fonction `horner(P,a)` qui prend en entrée un polynôme et qui renvoie en sortie la valeur de $P(a)$ calculée par la méthode de Hörner.

Pour comparer les temps de calcul entre `eval(P,a)` et `horner(P,a)`, on pourra choisir les polynôme et flottant P, a de son choix et taper dans la console :

```
console
In [2]: %timeit eval(P,a)
In [3]: %timeit horner(P,a)
```

III 2) Factorisation d'un polynôme par $X - \alpha$

Soit $n \in \mathbf{N}^\star$ donné et $P = a_n X^n + \cdots + a_1 X + a_0$ ($a_n \neq 0$) et mettons que α soit racine de P, si bien que pour des coefficients $b_{n-1}, \ldots, b_0$, on a : $a_n X^n + \cdots + a_1 X + a_0 = (X - a)(b_{n-1} X^{n-1} + \cdots + b_1 X + b_0)$

[Q12.]

1. En procédant par identification, trouver une relation entre les nombres b_j et les nombres u_k definis par (H).
2. En déduire une fonction `factorise(P,a)` qui prend en entrée un polynôme P, un flottant a racine de P, et qui renvoie en sortie la liste des coefficients du polynôme Q tel que $P = (X - a)Q$.

[Q13.] Donner la factorisation par $X - 1$ du polynôme représenté par :

```
[-69, 6, 7, 9, 1, 6, 2, 2, 7, 5, 6, 2, 1, 2, 4, 9].
```