Les polynômes – Partie 1

```python
def degre(p):
    return len(p) - 1

def pretty(polynome):
    deg = degre(polynome)
    while polynome[deg] == 0 and deg > 0:
        polynome.pop()
        deg = deg - 1
    return polynome

def somme(p, q):
    deg = max(degre(p), degre(q))
    s = [0] * (deg + 1)
    k = 0
    while k < degre(p) + 1:
        s[k] = p[k]
        k = k + 1
    j = 0
    while j < degre(q) + 1:
        s[j] = s[j] + q[j]
        j = j + 1
    return pretty(s)
```

Exercice 1

Les questions suivantes concernent la fonction `somme`.

a) Que représentent les paramètres `p` et `q` passés à la fonction `somme` ?
`p` et `q` sont des listes de nombres représentant des polynômes.
L'élément d'indice i est le coefficient du terme en x^i.

b) À quoi correspond la variable `deg` ? Pourquoi utilise-t-on la fonction `max` ?
`deg` est le degré maximal entre les deux polynômes. On utilise `max` afin que la liste `somme` soit suffisamment longue pour contenir tous les coefficients.

c) Que représente la liste `s` créée dans la fonction `somme` ?
`s` est la liste qui contiendra les coefficients du polynôme `somme` de `p` et `q`.

d) Quel est le rôle de la première boucle `while` ?
La première boucle copie les coefficients du polynôme `p` dans la liste `s`.

e) Quel est le rôle de la seconde boucle `while` ?
La seconde boucle ajoute à `s` les coefficients du polynôme `q`.

Exercice 2

On considère les polynômes suivants :

```python
p = [2, 1, 3]
q = [5, -1]
```

a) Quel est le degré du polynôme `p` ? du polynôme `q` ?
Degré de `p` : 2 (polynôme $2 + x + 3\,x^2$)
Degré de `q` : 1 (polynôme $5 - x$)

b) Dans la fonction `somme`, quelle est la valeur de `deg` ?
`deg` = 2

c) Compléter les deux lignes suivantes en indiquant le contenu de `s` après chaque boucle :
Après la première boucle : `s = [2, 1, 3]`.
Après la seconde boucle : `s = [7, 0, 3]`.

d) Quel polynôme (expression mathématique) correspond à la liste retournée ?
Le polynôme correspondant est : $7 + 0\,x + 3\,x^2$, soit $3\,x^2 + 7$.

Exercice 3

a) Le programme fonctionne-t-il si `p` et `q` n'ont pas le même degré ? Justifier.
Oui, le programme fonctionne car la liste `s` est créée avec la taille du plus grand degré, et seules les valeurs existantes sont copiées ou additionnées.

b) Comment écrire le polynôme nul ? Que se passe-t-il si l'un des polynômes est nul ?
Le polynôme nul : `[0]`.
Si un polynôme est nul, la somme renvoyée est l'autre polynôme, ce qui est cohérent mathématiquement.

Exercice 4

a) Réécrire la fonction `somme` en utilisant des boucles `for` à la place des boucles `while`.

```
def somme(p, q):
        deg = max(degre(p), degre(q))
        s = [0] * (deg + 1)
        for i in range(len(p)):
                s[i] = p[i]
        for i in range(len(q)):
                s[i] = s[i] + q[i]
        return s
```

b) Proposer une version de la fonction qui ne fait appel qu'à une seule boucle `for`.

```
def somme(p, q):
        deg = max(degre(p), degre(q))
        s = [0] * (deg + 1)
        for i in range(deg):
                if i < len(p):
                        s[i] += p[i]
                if i < len(q):
                        s[i] += q[i]
        return s
```

c) Ecrire une fonction qui calcule la différence de deux polynômes (`p - q`).
Pour calculer la différence `p - q`, il suffit de remplacer l'addition de `q[i]` par une soustraction.

Exercice 5

a) Quel est le rôle de la fonction `pretty` ?
La fonction `pretty` sert à nettoyer la représentation d'un polynôme stocké sous forme de liste, en supprimant les coefficients nuls inutiles en fin de liste.
Autrement dit, elle enlève les zéros correspondant à des termes de plus haut degré inexistants.

b) Expliquer la signification du `and` à la ligne 3 de la fonction `pretty`.
L'expression complète est vraie uniquement si les deux conditions sont vraies en même temps :
1. `polynome[deg] == 0`, le coefficient de plus haut degré est nul ;
2. `deg > 0`, on n'est pas au degré 0 (polynôme constant).
La boucle s'exécute donc tant que le coefficient de plus haut degré est nul et que le polynôme n'est pas réduit à un seul terme.

c) On a `p = [0, 0, 2, 1, 0, 0, 3, 0, 0, 0]`. Que retourne `pretty(p)` ?
`p = [0, 0, 2, 1, 0, 0, 3]`

```
1 def produit(p, q):
2     deg = degre(p) + degre(q)
3     mult = [0] * (deg + 1)
4     for i in range(degre(p)+1):
5         for j in range(degre(q)+1):
6             mult[i+j] = mult[i+j] + p[i]*q[j]
7     return mult
```

Exercice 6

a) Quel est le rôle de la liste `mult` ?
`mult` contient les coefficients du polynôme produit, initialisés à zéro.

b) Pourquoi utilise-t-on deux boucles `for` imbriquées ?
Les deux boucles permettent de multiplier chaque coefficient de `p` par chaque coefficient de `q`.

c) À quoi correspond l'indice `i + j` dans la liste `mult` ?
L'indice `i + j` correspond au degré du terme obtenu en multipliant x^i par x^j.

d) À la ligne 6, pourquoi n'écrit-on pas simplement `mult[i+j] = p[i]*q[j]` ?
Cette instruction ajoute le produit `p[i] ]· q[j]` au coefficient déjà calculé, cela évite d'écraser une valeur précédemment obtenue.

Exercice 7

Les questions suivantes concernent la fonction `produit`.
On considère les polynômes suivants :

```
p = [1, 2]
q = [3, 4]
```

a) Écrire les expressions mathématiques correspondant à `p` et `q`.
$p(x) = 1 + 2\,x$; $q(x) = 3 + 4\,x$

b) Quelle est la valeur de `deg` ?
`deg` $= 1 + 1 = 2$

c) Compléter le tableau des calculs effectués par le programme.
Calculs :

```
i = 0, j = 0 -> mult[0] = mult[0] + 1 x 3 = 3
i = 0, j = 1 -> mult[1] = mult[1] + 1 x 4 = 4
i = 1, j = 0 -> mult[1] = mult[1] + 2 x 3 = 10
i = 1, j = 1 -> mult[2] = mult[2] + 2 x 4 = 8
```

d) Donner la valeur finale de la liste `mult`.
`mult = [3, 10, 8]`

e) Vérifier que le résultat correspond bien au produit mathématique des deux polynômes.
$(1 + 2\,x)(3 + 4\,x) = 3 + 4\,x + 6\,x + 8\,x^2 = 3 + 10\,x + 8\,x^2 =$`[3, 10, 8]`

```
def horner(Dividende, diviseur):
    Dividende = Dividende[::-1]
    quotient = [0] * degre(Dividende)
    quotient[0] = Dividende[0]
    reste = 0
    for k in range(1, degre(Dividende)):
        quotient[k] = Dividende[k] - diviseur[0]*quotient[k-1]
    reste = Dividende[degre(Dividende)] - diviseur[0]*quotient[
       degre(Dividende)-1]
    quotient = quotient[::-1]
    return quotient, reste
```

Exercice 8

Les questions suivantes concernent la fonction `horner`.

a) Que représente la liste `Dividende` ?
`Dividende` est une liste représentant les coefficients d'un polynôme.

b) Quel type de polynôme représente `diviseur` ?
`diviseur` représente un polynôme de degré 1, de la forme $x - a$.

c) Pourquoi inverse-t-on la liste `Dividende` au début de la fonction ?
L'inversion permet de travailler à partir du coefficient dominant, comme dans la méthode de Horner.

d) Que représente la liste `quotient` ?
`quotient` est la liste des coefficients du polynôme quotient.

e) Pourquoi la taille de la liste quotient est-elle `degre(Dividende)` ?
Le degré du quotient est égal au degré du dividende moins 1.
`degre(quotient) = degre(Dividende) - 1`.

f) Expliquer le rôle de l'instruction : `quotient[0] = Dividende[0]`.
Le premier coefficient du quotient est égal au coefficient dominant du dividende.

g) À quoi correspond la boucle `for` ?
La boucle permet de calculer successivement les coefficients du quotient en utilisant les résultats précédents.

h) Quel est le rôle de la dernière ligne calculant `reste` ?
Le reste est le dernier calcul de la méthode de Horner.

Exercice 9

Les questions suivantes concernent la fonction `horner`. On considère :

```
1 Dividende = [-5, 1, -3, 2]
```

On divise par le polynôme $x - 2$.

a) Quelle est la valeur passée dans `diviseur` ?
`diviseur = [-2, 1]`

b) Donner la liste `Dividende` après l'instruction d'inversion.
Dividende inversé : `[2, -3, 1, -5]`

c) Compléter les valeurs successives du tableau `quotient`.
Calcul du quotient :
`quotient[0]` $= 2$
`quotient[1]` $= -3 - (-2) \cdot 2 = -3 + 2 \cdot 2 = -3 + 4 = 1$
`quotient[2]` $= 1 - (-2) \cdot 1 = 1 + 2 \cdot 1 = 1 + 2 = 3$

d) Donner la valeur finale du `quotient` et du `reste`.
Quotient (avant inversion) : `[2, 1, 3]`
Quotient final : `[3, 1, 2]`
Reste : $-5 - (-2) \cdot 3 = -5 + 2 \cdot 3 = -5 + 6 = 1$

e) Vérifier le résultat par un calcul mathématique.

	2	-3	1	-5
2		4	2	6
	2	1	3	1

$2x^3 - 3x^2 + x - 5 = (x - 2)(2x^2 + x + 3) + 1$