Polynômes : P, Q

$P+Q$

$P-Q$

Multiplier par un nombre : $k \cdot P$

```
def produit_par_nombre(P, k):
    ⋮
    return Q
```

Exemple : $P = 2 - 3X + 4X^2 - X^5$ → [2, -3, 4, 0, 0, -1]

$k = 1{,}5$

$k \cdot P = 1{,}5(2 - 3X + 4X^2 - X^5)$

[1,5·2, 1,5·(-3), 1,5·4, 1,5·0, 1,5·0, 1,5·(-1)]

Multiplier deux polynômes : $p \cdot q$

$$p = 1 + 2x - 3x^2 + 4x^3 - 2x^4$$

$$q = 2 - x - 3x^2 + 2x^3 - 4x^5 + x^6$$

$p = [1, 2, -3, 4, -2]$ $n = 5$

$q = [2, -1, -3, 2, 0, -4, 1]$ $m = 7$

		0	1	2	3	4	5	6
		2	-1	-3	2	0	-4	1
0	1	2	-1	-3	2	0	-4	1
1	2	4	-2	-6	4	0	-8	2
2	-3	-6	3	9	-6	0	12	-3
3	4	8	-4	-12	8	0	-16	4
	-2	-4	2	6	-4	0	8	-2

q : 0 1 2 3 4 5 6 — $j = 5$

$i = 1$

$c = p[i] * q[j]$

$= 2 * (-4)$

$= -8$

```
n = len(p)
m = len(q)
i = 0
while i < n:
    j = 0
    while j < m:
        c = p[i]*q[j]
        j += 1
    i += 1
```

$i + j$

résultat : $c_0 + c_1 x + \cdots + c_9 x^9 + c_{10} x^{10}$

$r = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$ ← $\text{len}(r) = 11$

$= m + n - 1$

```
n = len(p)
m = len(q)
r = [0 for k in range(m+n-1)]
i = 0
j = 0
while i < n:
    while j < m:
        r[i+j] += p[i] * q[j]
        j += 1
    j = 0
    i += 1
```

Horner $p = [1, -1, 2, -3, 4, -2]$ ($a_0, a_1, a_2, a_3, a_4, a_5$)

$p = a_0 + a_1 x + \dots + a_5 x^5$

$a = -3$

$-2x^5 + 4x^4 - 3x^3 + 2x^2 - x + 1$ | $x+3$

	-2	4	-3	2	-1	1
3		-6	-6	-27	-75	-228
	-2	-2	-9	-25	-76	-227
		q				r, le reste

```
def horner(p, a):
    n = len(p)
    i = n-1
    q = []
    while i >= 0:
        ....
        i = i-1
```