Méthodes numériques : polynômes

Exercice 1 (0 points)

On donne le code ci-dessous :

```
p=[1,2,3]
q=p
p[1]=-2
print(q)
print(p)
```

a) Quelles sont les listes affichées à la console après exécution de ce code ?

b) Donner une explication.

a) [1,-2,3] pour print(q)
[1,-2,3] pour print(p)

Il n'y a qu'une seule liste en mémoire !

b) p → [1,2,3]
q ↗

La variable p contient une référence à la liste stockée en mémoire.
La variable q se réfère au même objet après l'affectation q = p.

Exercice 2 (0 points)

On donne le code ci-dessous :

```
p=[1,2,3]
q=p[:]
p[1]=-2
print(q)
print(p)
```

a) Quelles sont les listes affichées à la console après exécution de ce code ?

b) Donner une explication.

a) [1,2,3] pour print(q) [1,-2,3] pour print(p)

b) L'expression p[:] crée une copie (shallow / peu profonde) de la liste p. Cela crée un nouvel objet en mémoire.
L'affectation q = p[:]

Exercice 3 (0 points)

On donne l'extrait de code ci-dessous :

```
def horner(p, a):
    q = p[:]
    i = (len(p) - 1) - 1
    while i >= 0:
        q[i] += q[i + 1]*a
        i = i - 1
    return q[1:], q[0]
p = [-1,2,-3]
a = 2
print(horner(p, a))
```

$-3x^2+2x-1 \mid x-2$

	−3	2	−1
2		−6	−8
	−3	−4	−9

a) Quel est l'affichage à la console après exécution de cet extrait de code ?

b) Quelle est la valeur référencée par `q` à la fin de l'exécution du corps de la boucle, au moment où `i` vaut 0 ?

a) $([-4, -3], -9)$

b) $q = [-1, 2, -3]$

$i = 1$

$q = [-1, 2+(-3)\cdot 2, -3] = [-1, -4, -3]$

$i = 0$

$q = [-1+2(-4), -4, -3] = [-9, -4, -3]$

$i = -1$

Au moment où $i = 0$ à la fin d'une exécution du corps de la boucle, q est une référence à la liste $[-1, -4, -3]$ stockée en mémoire.

Exercice 4 (0 points)

Compléter le code ci-dessous

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




    return r
```

```
    while i < n:
        j = 0
        while j < m:
            r[i+j] += p[i] * q[j]
            j = j+1
        i = i+1
```

Exercice 5 (0 points)

Écrire une fonction somme qui additionne les coefficients des polynômes dans l'ordre décroissant des degrés. Cela signifie en particulier que l'instruction d'incrémentation de l'indice de la boucle qui se trouve dans la fonction est la suivante : `i = i - 1`

```
def somme(p, q):
    if len(p) < len(q):
        p, q = q, p
    s = p[:]
    i = len(q) - 1
    while i >= 0:
        s[i] += q[i]
        i = i - 1
    return s
```