DS 7 – Maths & Info

Partie Informatique – 1 heure

N'hésitez pas à justifier et commenter vos fonctions. Le soin sera pris en compte.

Dans ce devoir, on modélise un polynôme $\sum_{k=0}^{n} a_k X^k$ par la liste $[a_0, a_1, \dots, a_n]$ (dans cet ordre !)

Ainsi le polynôme $1 + 3X + X^4$ est modélisé par la liste $L = [1, 3, 0, 0, 4]$ et inversement la liste $L = [0, 2, 1, 3]$ correspond au polynôme $2X + X^2 + 3X^3$.

1. Quelle est la longueur (minimale) d'une liste représentant un polynôme de degré n ?
2. Donner la liste correspondant au polynôme $P = X^2 - 1$.
   Donner les polynômes correspondant aux listes $[1, 3, 0, 5]$ et $[2, 1, -1, 0, 0]$.

Partie I – Racines

3. Compléter la fonction Python suivante qui prend en argument une **liste** correspondant à un polynôme P et un flottant a, et qui renvoie la valeur de $P(a)$.

```python
def eval(P, a):
    s = 0
    for k in range(......):
        s = s + ................
    return s
```

4. Écrire une fonction Python `dérive(L)` qui prend en argument une liste correspondant à un polynôme P et qui renvoie la liste correspondant au polynôme P' dérivé de P.

5. En déduire une fonction Python `multiplicité(P, a)` qui prend en argument une liste représentant un polynôme P non nul et un flottant a, et qui renvoie l'ordre de multiplicité de a en tant que racine de P. La fonction renverra simplement 0 si a n'est pas racine de P.

6. On suppose que les hypothèses du théorème des valeurs intermédiaires sont vérifiées.
   Écrire une fonction `cherche_racine(P, a, b, eps)` qui donne une valeur approchée à `eps` près d'une racine de P sur l'intervalle $[a, b]$. Comment s'appelle le procédé utilisé ?
   *Attention, P est modélisé par une **liste**, il faudra utiliser la fonction `eval` pour obtenir une image comme $P(a)$.*

Partie II – Opérations sur les polynômes

7. (a) À quelle opération sur les listes correspond la multiplication par X ? On s'appuiera sur quelques exemples à partir de polynômes simples.
   (b) En déduire une fonction `multiplication_par_X(P)` qui prend en argument une liste représentant un polynôme P, et qui renvoie la liste correspondant au polynôme $X \times P$. *Deux lignes suffisent pour cette fonction !*

8. En déduire une fonction Python `mult_monome(P, k)` qui prend en argument une liste correspondant à un polynôme P et un entier k, et qui renvoie la liste correspondant à $X^k P$.

9. Que fait la fonction suivante, en terme d'opérations sur les polynômes correspondants ? (Ici P est une liste modélisant un polynôme et c est un flottant.)

```python
def mystere(P, c):
    R = []
    for e in P:
        R.append(e * c)
    return R
```

   On pourra illustrer sa réponse par un exemple.

10. Il n'est pas facile de modéliser informatiquement l'addition de deux polynômes car les listes correspondantes ne sont pas nécessairement de même taille.
    Une astuce consiste à ajouter des 0 à la fin des listes, ce qui ne change pas la valeur du polynôme.
    (a) Écrire une fonction Python `astuce(P1, P2)` qui prend deux listes $P1, P2$, pas nécessairement de la même longueur et qui renvoie de listes de même longueur, dont les coefficients sont les mêmes que ceux de $L1, L2$ au début, puis des 0 à la fin.
    Par exemple : `astuce([1,2],[1,4,0,2])` peut renvoyer [1,2,0,0], [1,4,0,2]
    (b) En déduire une fonction Python `somme(P1, P2)` qui prend en argument deux listes représentant deux polynômes P_1, P_2 et renvoie une liste qui correspond au polynôme $P_1 + P_2$.

11. **Bonus – si le temps le permet :** À l'aide des fonctions écrites précédemment, écrire une fonction `produit(P, Q)` qui réalise le produit de deux polynômes P et Q représentés par des listes.