PCSI 3
2020/2021

Lycée Sainte-Geneviève
Samedi 11 janvier 2020

DS 1 d'informatique

2 heures

Lisez attentivement les consignes ci-dessous

- *Les documents, calculatrices et autres appareils électroniques sont interdits.*
- *La qualité de la rédaction sera prise en compte dans l'évaluation.*
- *La présentation de la copie sera prise en compte dans l'évaluation :*
  - ▷ *Encadrez les résultats principaux.*
  - ▷ *Soignez votre écriture.*
  - ▷ *Maintenez une marge dans vos copies et aérez vos copies.*
  - ▷ *Numérotez vos copies.*
- ***Vos programmes doivent être clairs****.*
  - ▷ *Choisissez judicieusement les noms de vos variables ainsi que les noms de vos fonctions auxiliaires.*
  - ▷ *Aérez vos programmes.*
  - ▷ *Faites des indentations suffisamment grandes.*
  - ▷ ***Indiquez les indentations par des traits verticaux****.*
  - ▷ *Si nécessaire, commentez vos programmes en utilisant une couleur secondaire.*

Polynômes

Le but de ce problème est de créer un ensemble de fonctions permettant de manipuler les polynômes en `Python`*.*

- *Un polynôme sera codé sous la forme d'une liste, plus précisément de la liste de ses coefficients.*
- *Par exemple, le polynôme* $5 - 4X + 3X^2$ *sera représenté par la liste* `[5, -4, 3]`*.*
- *La représentation d'un polynôme n'est pas unique, puisque la liste* `[5, -4, 3, 0, 0]` *(par exemple) représente le même polynôme.*
- *Le polynôme nul pourra être représenté par la liste vide.*
- *L'unique liste sans zéros à sa fin représentant un polynôme* P *sera appelée* représentation normale *de* P*.*

Dans la suite, on écrira `polynôme` *pour désigner une liste dont les éléments sont des nombres.*

On suppose que l'environnement dans lequel sont écrites les fonctions possède deux variables `PLUS_INF` *et* `MOINS_INF` *auxquelles sont affectés respectivement les objets* `float("infinity")` *et* `-float("infinity")` *représentant* $+\infty$ *et* $-\infty$*.*

On rappelle que la commande

```
[0]*N
```

crée une liste composée de `N` *éléments tous égaux à* `0`*.*

Partie I – Exemples

1. (a) À quel polynôme correspond la liste `[1]` ?

(b) À quel polynôme correspond la liste `[0, 1]` ?

(c) À quel polynôme correspond la liste `[0, 0, 0, 1, 0, 0]` ?

2. (a) Donner le `polynôme` qui est la représentation normale de $X^3 + 2X^2 + 3X + 4$.

(b) Donner la représentation normale du polynôme nul.

Partie II – Premières fonctions

3. Écrire une fonction `maxCoeffs(P)` qui prend en argument un `polynôme` `P` et qui renvoie le maximum des valeurs absolues des coefficients de P.

4. Écrire une fonction `estNul(P)` qui prend en argument un `polynôme` `P` et qui renvoie `True` si le polynôme est nul et `False` sinon.

5. Écrire une fonction `degre(P)` qui renvoie le degré d'un `polynôme` `P`.

6. Écrire une fonction `formeNormale(P)` qui prend en argument un `polynôme` `P` et qui renvoie la représentation normale de `P`.

Partie III – Premières opérations

7. (a) Écrire une fonction `evaluation(P, alpha)` qui prend en argument un `polynôme` `P` et un nombre `alpha` et qui renvoie l'évaluation $P(\alpha)$ de P en α.

(b) Quelle est la complexité de votre fonction ?

8. Écrire une fonction `scalairisation(mu, P)` qui prend en argument un nombre `mu` et un `polynôme` `P` et qui renvoie un `polynôme` représentant la scalairisation μP.

9. Écrire une fonction `somme(P, Q)` qui prend en argument deux `polynômes` `P` et `Q` et qui renvoie un `polynôme` représentant leur somme $P + Q$.

Partie IV – Dérivation

10. Écrire une fonction `derive(P, k)` qui prend en argument un `polynôme` `P` et un entier naturel `k` et qui renvoie la dérivée k-ième $P^{(k)}$ de P.

Partie V – Produit

11. (a) Écrire une fonction `produitParX(P)` qui prend en argument un `polynôme` `P` et qui renvoie un `polynôme` représentant le produit XP.

(b) Écrire une fonction `produit(P, Q)` qui prend en argument deux `polynômes` `P` et `Q` et qui renvoie un `polynôme` représentant leur produit PQ.

Partie VI – Arithmétique polynomiale

Définition

Soient P et Q deux polynômes.

On dit que P divise Q quand il existe un polynôme R tel que $Q = PR$.

12. Écrire une fonction `divisionEuclidienne(A, B)` qui prend en argument deux `polynômes` `A` et `B` et qui renvoie

- si $B \neq 0$: un `tuple (Q, R)` de `polynômes` où `Q` est le quotient dans la division euclidienne de A par B et où `R` est le reste.
- si $B = 0$: `None`

13. Écrire une fonction `valuation(P, alpha)` qui prend en argument un `polynôme` `P` et un nombre `alpha` et qui renvoie le plus grand entier k tel que $(X - \alpha)^k$ divise P.

On conviendra que si $P = 0$ alors la fonction renvoie $+\infty$.

Partie VII – Recherche de racines

14. Écrire une fonction `racine(P, a, b, eps)` qui prend en argument un polynôme `P`, trois nombres flottants `a`, `b` et `eps` tels que

- `a` $<$ `b`,
- `P(a)` et `P(b)` sont de signes stricts contraires

et qui renvoie un nombre flottant `x` tel que

- `a` $\leqslant$ `x` $\leqslant$ `b`,
- `x` est `eps`-proche d'une racine de P, *ie* il existe une racine α de P vérifiant $|\mathtt{x} - \alpha| \leqslant \mathtt{eps}$.

$\boxed{\textit{On procèdera par dichotomie.}}$

L'existence d'une racine de P dans $[a, b]$ est assurée par le théorème des valeurs intermédiaires et par le fait que $P(a)$ et $P(b)$ sont de signes contraires.

15. (a) Soit $P \in \mathbb{R}[X]$ un polynôme unitaire de degré impair.

(i) Déterminer un réel $b > 0$, dépendant des coefficients de P, tel que $P(b) > 0$.

(ii) Déterminer un réel $a < 0$, dépendant des coefficients de P, tel que $P(a) < 0$.

(b) Écrire une fonction `racineImpair(P, eps)` qui prend en argument un polynôme `P` de degré impair et un nombre flottant `eps`, et qui renvoie un nombre flottant `x` qui est `eps`-proche d'une racine de P.

FIN DU SUJET.