

Exercice 2

L'objectif de cet exercice est la manipulation des polynômes creux à une seule variable. Un polynôme creux est un polynôme dont certains coefficients sont nuls.

Un polynôme est construit à partir de monômes.

Un monôme est une expression de la forme ax^n ou a ($a \neq 0$) est le coefficient du monôme et $n(n \geq 0)$ son degré.

Un monôme est représenté par un dictionnaire à un élément dont la clé est le degré n et la valeur est le coefficient a .

Exemple : Le monôme $8x^2$ est représenté par le dictionnaire $\{2 : 8\}$.

Un polynôme creux est alors défini comme une association de monômes de degrés différents.

Exemple : Le polynôme $-x^4 + 8x^2 - 5x$ est représenté par le dictionnaire $\{2 : 8, 1 : -5, 4 : -1\}$.

Le dictionnaire $\{0 : 1, 5 : 1, 8 : 1\}$ représente le polynôme $x^8 + x^5 + 1$.

1. Ecrire une fonction, nommée **degree**, qui retourne le degré d'un polynôme.
2. Ecrire une fonction, nommée **evaluer**, qui retourne la valeur du polynôme pour un réel x_0 donné.
3. Ecrire une fonction, nommée **coeff**, qui retourne le coefficient du monôme de degré i .
4. Ecrire une fonction, nommée **addition**, qui retourne le polynôme somme de deux polynômes.
5. Ecrire une fonction, nommée **produit**, qui retourne le polynôme produit de deux polynômes.
6. Ecrire une fonction, nommée **affiche**, qui retourne la chaîne représentant l'expression du polynôme ordonné par ordre décroissant.
Pour le polynôme représenté par $\{4 : 4, 0 : 4, 12 : 6, 9 : 1, 7 : -1\}$, la chaîne retournée est :
' $6x^{**12} + x^{**9} - x^{**7} + 4x^{**4} + 4$ '
7. Ecrire une fonction, nommée **primitive**, qui retourne le polynôme représentant la primitive. On suppose que la constante d'intégration est nulle.



Filière : 1 ^{ère} LFI	Matière : Programmation Python		Enseignant : Dr. A. BEN HMIDA SAKLY
Date : 08 / 06 / 2024	Nbr de Crédits : 3	Coefficient : 1.5	Documents autorisés : Non
Durée de l'examen : 1h30	Régime d'évaluation : MR		Nombre de pages : 08
	EX (50%) + DS (20%) + OR (5%) + TP (25%)		
Nom & Prénom :			Matricule :
Signature :	Code confidentiel :		Classe : N° Place :

NOTE : Répondre directement sur les feuilles de l'examen /

Note

/ 20

Exercice 1

- 1.1. Écrire une fonction **paritysort** qui prend en argument une liste d'entiers **lis** et qui renvoie une liste contenant les valeurs de **lis** et tel que tous les nombres pairs se trouvent à gauche des nombres impairs.

Exemple:

`lis=[1,2,3,4,5,6,8,10]`

`print(paritysort(lis))` affiche (par exemple) `[2,4,6,8,10,1,3,5]`.

- 1.2. Modifier la fonction pour que les valeurs ayant la même parité soient triées par ordre croissant.

Exemple:

`lis=[5,8,4,2,3,1,10,9,7]`

`print(paritysort(lis))` affiche `[2,4,8,10,1,3,5,7,9]`

Ne rien écrire ici

- 1.3. Écrire une fonction **shift** qui prend en argument une liste d'entiers **lis** et qui renvoie une copie de cette liste où toutes les valeurs sont décalées d'une case vers la droite (et la dernière valeur se retrouve en position 0).

Exemple:

```
lis=[1000,1,2,3]
print(shift(lis)) affiche [3,1000,1,2].
```

- 1.4. Écrire une fonction **circulaire** qui prend en argument deux listes d'entiers **lis1** et **lis2** et qui renvoie True si **lis2** est une permutation circulaire de **lis1**, et False sinon (ou si les listes n'ont pas la même longueur).

Exemple:

```
lis1=[1,2,3,4,5] lis2=[3,4,5,1,2] lis3=[3,5,4,1,2]
circulaire(lis1,lis2) = True
circulaire(lis1, lis3) = False
```