ECT1

TP 12 : les vecteurs

Exercice 1 : Révision

Soit f définie par $f(x)=\frac{x-1}{x-3}$

1- Déterminer Df

2- Compléter la fonction Python qui renvoie l'image d'un nombre x et "impossible" si x n'est pas dans l'ensemble de définition de f.

```
def f(x):
    if x==3 :
        return ……
    else :
        return ……
```

Exercice 2 : Un nouveau type array

On désire construire un tableau d'une ligne et de plusieurs colonnes appelé vecteur

Pour cela il faut d'abord importer la bibliothèque numpy dans tous les exercices de ce TP

1- On veut créer le tableau suivant :

3	5	12

Pour cela aller à la console et taper `V=np.array([3,5,12]);V`. Évaluer le résultat

Remarque : on aurait pu écrire aussi `V=np.array((3,5,12))`

2- Évaluer à la console V[0], V[1], V[2]

Remarque pour extraire un élément les [ ] sont obligatoires

Comme pour les listes, les numéros des colonnes débute à 0 ainsi V[0] affiche le 1er terme .

3- A vous maintenant , créer le tableau suivant :

1	2	3	4	5	6

Selon la taille du tableau cela peut devenir fastidieux . Heureusement qu'il existe des moyens plus rapide.

4- Évaluer à la console `Z=np.zeros(7);Z`

5- Évaluer à la console `U=np.ones(7);U`

6- Évaluer à la console `U[2]=5;U`

7- Évaluer à la console `U[1:3]`

8- Évaluer séparément à la console `np.sum(U);np.prod(U); np.min(U); np.max(U)`

Les vecteurs

nom = `np.array([`u_0`,`u_1`,..,` u_n`])` crée le vecteur *nom*=(u_0 , u_1 , …, u_n)

`nom[0]` affiche le 1er élément, `nom[1]` affiche le 2ème élément…
`nom[i:j]` extrait le vecteur (u_i , u_{i+1} , …, u_{j-1})
`nom[i]=nombre` modifie le terme d'indice i
`np.sum(`nom`)` renvoie la somme des éléments de nom
`np.prod(`nom`)` renvoie le produit des éléments de nom
`np.min(`nom`)` renvoie le plus petit éléments de nom
`np.max(`nom`)` renvoie le plus grand éléments de nom
`np.zeros(n)` crée un vecteur à n composantes toutes égales à 0
`np.ones(n)` crée un vecteur à n composantes toutes égales à 1

<u>Exercice 3</u> : Applications aux suites

1- Soit (u_n) une suite définie par $u_0=1$ et $\forall\, n \in \mathbb{N},\ u_{n+1}=e^{-u_n}$

Écrire un programme Python qui demande *n* à l'utilisateur et qui affiche un vecteur contenant tous les termes de la suite jusqu'au terme de rang n

2- Soit (u_n) une suite définie par $u_1=2$ et $\forall\, n \in \mathbb{N}^*,\ u_{n+1}=3u_n-1$

Écrire un programme Python qui demande *n* à l'utilisateur et qui affiche un vecteur contenant tous les termes de la suite jusqu'au terme de rang n

3- Soit (u_n) une suite définie par $u_0=1$, $u_1=1$ et $\forall\, n \in \mathbb{N}$, $u_{n+2}=u_{n+1}+u_n$

Écrire un programme Python qui demande *n* à l'utilisateur et qui affiche un vecteur contenant tous les termes de la suite jusqu'au terme de rang n

Programme type pour afficher tous les terme d'une suite récurrente dans un vecteur

```
import numpy as np # toujours importer numpy
n=int(input('n='))
u=np.zeros(nombre de case) # nombre de case dépend du rang du 1er terme
u[0],u[1]=                # valeurs du ou des premiers termes
for k in range(debut n°case à calculer , fin n°case +1)  :
      u[k]=    #utiliser le brouillon et k désigne le numéro de la case

print(u) #sans indentation
```

Les numéros des cases débutent à 0

Exercice 4 : Applications aux sommes

Soit (u_n) une suite définie par $u_1=2$ et $\forall\, n \in \mathbb{N}^*,\ u_{n+1}=\frac{1}{2}{u_n}^2-1$

Écrire un programme Python qui demande n à l'utilisateur et qui affiche $S=\sum_{k=1}^{n} u_k$ en utilisant les vecteurs.