A. BENADDI
Prépa 2ème Année ECG

Feuille1 Rappel

1. Exercices (l'indispensable)

1) Créer une liste L en python qui contient les nombres k^3, pour $k \in [\![0;n]\!]$, après avoir demandé à l'utilisateur de saisir un valeur pour l'entier naturel n. Puis compléter le programme en python, pour qu'il calcule $\sum_{k=0}^{n} k^3$, avec n une valeur donnée dans $\mathbb{N}$.
2) Écrire une fonction en python qui prend en argument d'entrée une liste de réels et renvoie la liste obtenue en calculant, si possible, l'inverse de chaque réel de la liste.
3) Écrire un programme en python qui demande une liste de coefficients ainsi qu'une liste de notes et qui renvoie la moyenne pondérée des notes selon les coefficients.
4) Écrire une fonction qui prend en arguments d'entrée une liste ainsi que deux indices i et j, puis qui renvoie la nouvelle liste obtenue après échange des éléments d'indices i et j, si les indices sont dans l'intervalle $[\![0,n]\!]$, avec n la longueur de la liste.
5) Écrire une fonction qui prend en arguments d'entrée une liste et un élément x, puis qui renvoie la liste obtenue après suppression de toutes les occurrences de $x$dans cette liste.
6) Écrire une fonction qui prend en arguments d'entrées une liste, un indice i et un élément x, puis qui renvoie la liste obtenue après insertion de l'élément x à l'indice i, en décalant le reste vers la droite.
7) Ecrire une fonction qui étant une liste renvoie le maximum et son nombre d'occurrences.
8) Ecrire une fonction qui étant une liste renvoie une liste qui ne contient aucun doublon.
9) Ecrire une fonction qui étant donnée une liste, d'au moins deux éléments, renvoie les deux valeurs les plus proches de la liste.
10) Écrire un programme qui demande la valeur d'un entier naturel n à l'utilisateur et affiche ensuite la valeur de $\sum_{0\leqslant i<j\leqslant n}(i+j)$

2. Quelques extraits de concours

Exercice 1

Représenter les premiers termes de la suite $(u_n)_n$ définie par $u_n = \dfrac{n!}{n^n}$.

Exercice 2

Pour $n \in \mathbb{N}^*$, on considère la fonction $f_n : x \longmapsto x^n + x - 1$. On peut montrer, que pour tout $n \in \mathbb{N}^*$, l'équation $f_n(x) = 0$ possède une unique solution dans $[0;1]$, que l'on notera α_n.

a. Écrire un programme Python qui demande une valeur de $n \in \mathbb{N}^*$ à l'utilisateur et affiche un graphique représentant la courbe de f_n, en précisant avec une croix rouge le point de coordonnées $(\alpha_n, 0)$.

b. Écrire un programme qui demande une valeur de $n \in \mathbb{N}^*$ à l'utilisateur et affiche un graphique représentant toutes les courbes des fonctions f_k , en précisant avec une croix rouge le point de coordonnées $(\alpha_k, 0)$, pour $k \in [\![1;n]\!]$.

Exercice 3 (Ecricome2018)

On considère la suite $u_n)_{n\geq 0}$ définie par :

$$u_0 = 0, \quad u_1 = 1 \quad \text{et} \quad \forall n \in \mathbb{N}, \quad u_{n+2} = u_{n+1} + u_n,$$

a. Ecrire une fonction python afin qu'elle, prenne en argument un entier $n \geq 2$, elle calcule et renvoie la valeur du terme u_n de la suite $(u_n)_{n\geq 0}$.

b. Ecrire une fonction python qui renvoie le premier rang p à partir duquel $\dfrac{u_{n+1}}{u_n}$ est une valeur approchée du nombre d'or à 10^{-10} près.

Rappel nombre d'or $\rho = \dfrac{1+\sqrt{5}}{2}$.

Exercice 4 (Edhec 2021)

On se propose d'étudier la suite $(u_n)_{n\in\mathbb{N}}$, définie par la donnée de $u_0 = 1$ et par la relation, valable pour tout entier naturel n : $u_{n+1} = \sqrt{u_n^2 + u_n}$

Pour tout entier naturel n non nul, on pose $S_n = \sum_{k=0}^{n-1} u_k$. On montrer que pour tout entier naturel n, u_n est bien défini et supérieur ou égal à 1 et diverge.

a) Ecrire un script *python* afin qu'il permette de déterminer et d'afficher la plus petite valeur de n pour laquelle on a $S_n > 1000$.

b) On montre que $S_n = u_n^2 - 1$ et qu'un équivalent de S_n pour n au voisinage de $+\infty$ est $S_n \underset{n\to+\infty}{\sim} \dfrac{n^2}{4}$.

Ecrire un script *python* afin qu'il fasse le même travail que celui de la question **a)** sans calculer S_n .

Exercice 5 (Ecricome 2023)

1. Montrer que, pour tout entier naturel n, l'intégrale $\int_0^{+\infty} e^{-t} sin^n(t)dt$ converge. On note alors, pour tout $n \in \mathbb{N}$,

$$I_n = \int_0^{+\infty} e^{-t} sin^n(t)dt$$

2. (a) Calculer I_0 .
   (b) Soit n un entier naturel supérieur ou égal à 2
   i. A l'aide d'intégrations par parties, montrer que pour tout réel A strictement positif :

$$\int_0^A e^{-t} sin^n(t)dt = -e^{-A} sin^{n-1}(A)\,(sin(A) + ncos(A)) - n \int_0^A e^{-t} sin^n(t)dt + n(n-1) \int_0^A e^{-t} cos^2(t) sin^{n-2}(t)dt.$$

   ii. En déduire que $I_n = \dfrac{n(n-1)}{n^2+1} I_{n-2}$.

   (a) Compléter la fonction Python suivante pour que, prenant en argument un entier naturel n, elle calcule et renvoie la valeur de I_{2n}.

```
def calcul(n):
    I=.........
    for k in range(1,n+1):
        I = I * ...........
    return ..........
```

(b) On a représenté ci-dessous la suite $(\sqrt{n} I_{2n})_{n\geqslant 1}$ à l'aide du programme Python précédent.
Que peut-on conjecturer pour un équivalent de I_{2n} lorsque n tend vers $+\infty$?