

Partiel blanc - Ingénierie Informatique

Exercice 1 : Trouver l'inconnue (*cours*)

Soit le code Python suivant :

```
x = 69

def magic(x):
    x = x + 42

magic(x)
print("x vaut", x)
```

Quel sera le message affiché sur la console ?

On supposera par la suite que le module numpy est importé par :
import numpy as np

Exercice 2 : Méthode de Newton (*facile*)

Vous êtes-vous déjà demandé comment votre calculatrice arrivait à trouver les racines d'une fonction ? Plusieurs méthodes existent, une en particulier est la méthode de Newton.

Soit f la fonction dont on veut calculer une racine et x_0 une valeur de départ quelconque, on applique la suite récurrente suivante :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

On applique cette suite jusqu'à obtenir une image suffisamment proche de 0, ici on prendra $|f(x_n)| < 10^{-5}$.

Écrivez la fonction Python *newton(x)* qui permet de calculer les racines de la fonction $x \rightarrow -\cosh(x) + x^2$.

*Indication : numpy dispose de toutes les fonctions mathématiques, on pourra utiliser *np.cosh()*, *np.sinh()* et *np.abs()*.*

Exercice 3 : Tasser un tableau 1D (*moyen*)

Dans certaines applications, il peut être nécessaire de tasser un tableau, c'est-à-dire de déplacer tous les zéros à la fin du tableau et de conserver l'ordre des valeurs.

Exemple :

$$[0, 1, 2, 0, 3] \Rightarrow [1, 2, 3, 0, 0]$$

Écrivez la fonction Python `tasser(tab, n)` qui tasse un tableau numpy `tab` de taille `n` et renvoie le nombre de zéros contenus dans le tableau.

Exercice 4 : Petit circuit imprimé (*difficile*)

On souhaite pouvoir simuler le comportement d'un circuit imprimé. Pour cela, on considère le circuit comme un tableau numpy à deux dimensions.

Pour simplifier les choses, on suppose que notre circuit ne dispose que de 3 types de composants :

- Le composant **NOP** : il ne change pas la direction du courant et n'affiche rien. On le code comme un **0** dans le tableau.
- Les composants **DIR** : ils permettent de changer la direction du courant dans le circuit. On les code comme des **chiffres allant de 1 à 4** permettant d'aller respectivement vers le **haut**, la **droite**, le **bas** et la **gauche**.
- Les composants **TXT** : ils permettent d'afficher du texte dans la console et ne changent pas la direction du courant. On les code comme **une lettre** dans le circuit.

On considère deux conditions essentielles :

- Le courant commence en haut à gauche vers la droite.
- On considère que le circuit a terminé son exécution quand le courant sort du tableau.

Exemple :

$$\begin{bmatrix} 0, & 'M', & 'a', & 'i', & 3], \\ ['t', & 'n', & 'e', & 'x', & 4] \end{bmatrix} \Rightarrow \text{Affichera "Maixent".}$$

Écrivez la fonction `executer(circuit, l, h)` qui permet d'exécuter n'importe quel circuit de largeur `l` et hauteur `h`.

Indication : on pourra utiliser le kwarg `end` de `print` pour ne pas sauter de ligne, c'est-à-dire, `print(..., end="")`.

Bonus : Concevez un circuit qui affiche "Banane" avec 4 composants TXT.