

Concours blanc 1 : Python

Nom et Prénom :

Classe :

On rappelle que les fonctions `ln` et `| |` font partie de la bibliothèque `numpy` et s'écrivent `\np.log()` et `np.abs()` :

Exercice 1 :

On reprend l'énoncé de l'exercice 1 partie 2 du CB de ce jour.

Soit f la fonction définie sur $[0; +\infty[$ par :

$$f(x) = \begin{cases} \frac{\ln(1+2x)}{x} - 1 & \text{si } x \in]0; +\infty[\\ 1 & \text{si } x = 0 \end{cases}$$

On a montré que f est continue sur R^+ et strictement décroissante sur R^+ . Elle réalise donc une bijection de R^+ sur $f(R^+)$.

Comme de plus $0 \in f(R^+)$, alors f s'annule en un unique point : on appelle ce point α . Enfin il a été démontré que $1 < \alpha < 2$.

1. On cherche à appliquer maintenant la méthode de la dichotomie pour trouver une valeur approchée de α à 10^{-5} près. On remarque que $f(1)$ et $f(2)$ sont de signes opposés.

En utilisant la fonction f à définir ci-dessous, compléter le 2ème programme qui permet de déterminer une valeur approchée de α à 10^{-5} près .

```
import numpy as np

def f(x):
    return ...

a=1
b=2
while ..... 10**(-5) :
    m=...
    if .....<0 :
        b=.....
    else:
        .....
print("Une valeur approchée de alpha à 10^{-5} pres est : ",a)
```

2. On reprend la partie 3 de l'exercice 1 :

On considère la suite $(u_n)_{n \in \mathbb{N}}$ définie par la donnée de son premier terme $u_0 = 1$ et par :

$$\forall n \in \mathbb{N}, \quad u_{n+1} = \ln(1 + 2u_n)$$

Il a alors été montré que :

$$\forall n \in \mathbb{N}, \quad |u_{n+1} - \alpha| \leq \frac{2}{3} |u_n - \alpha|$$

On souhaite utiliser cette nouvelle inégalité pour calculer une valeur approchée de α à 10^{-5} près.

Compléter ci-dessous le programme qui permet d'effectuer le calcul :

```
import numpy as np

u=1
c=1
while      c ..... 10**(-5) :
    u=.....
    .....

print("Une valeur approchée de alpha à 10^{-5} pres est : ",.....)
```

Exercice 2 :

Soit $(u_n)_{n \in \mathbb{N}}$ définie par $\forall n \in \mathbb{N}, u_n = (3n + 1)!$

1. Ecrire une fonction `suite_u()` , d'argument un entier $n \geq 1$, qui renvoie la valeur du terme u_n , sans utiliser la fonction factoriel de la bibliothèque `math`.

```
def suite_u(n):
    u=.....
    for ..... :
        .....
    return ..... .....
```

2. On définit la suite $(v_n)_{n \in \mathbb{N}}$ par : $\forall n \in \mathbb{N}^*, \quad v_n = \frac{1}{2n(n+1)} \sum_{k=1}^n (1 + ku_k)$

Ecrire une fonction `suite_v()` , d'argument un entier $n \geq 1$, qui renvoie la valeur du terme v_n .

```
def suite_v(n):

    .....

    return
```