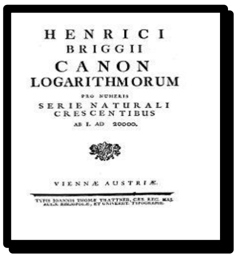
Lundi 21 Mars 2022, TA **Algorithme de Briggs : sert à calculer des valeurs approchées de logarithmes de nombres entiers.**	# Algorithmes de Briggs	**Chapitres : fonction ln, suites et calculs limites** **Programmation Python** **Compétences : Calculer, raisonner, représenter, modéliser**



Partie 1 : Histoire des Maths, qui était Henry Briggs ?

Mathématicien anglais. On sait peu de choses sur sa vie .

Il est né dans le Yorkshire en Angleterre en 1561 et mort à Oxford, au Nord Ouest de Londres en 1630.

Son nom est attaché à la découverte des logarithmes décimaux (appelés aussi logarithmes vulgaires ou briggsiens).

Le caractère instrumental de ce nouvel outil mathématique lui valut une large et rapide diffusion auprès des utilisateurs confrontés à des calculs longs et compliqués.

À partir de 1596, Briggs enseigna la géométrie à Londres. Il s'intéresse à l'astronomie qui nécessite de lourds calculs.

Il est le premier à déceler la puissance des logarithmes tout juste inventés par Neper pour simplifier les calculs complexes.

A l'été 1615, il entreprend le voyage à Edimbourg pour rencontrer Neper.

Neper, ce célèbre théologien, physicien, astronome et mathématicien écossais.

La notation *Log* comme abréviation de logarithme apparaît en 1616 dans une traduction anglaise de l'œuvre de Neper.

De leurs discussions, il ressortira l'adoption d'un logarithme en base 10 et tel que log(1)=0.

En effet, un des objectifs de Briggs était obtenir le logarithme de 10 comme une puissance de 10. Après avoir examiné diverses options, il a décidé que le log 10 = 1. Ceci ne signifiait pas seulement la naissance de **logarithme décimal**, mais aussi le concept de *base du logarithme*.
Briggs a amélioré le travail de Napier et publiera finalement de nombreuses tables de logarithmes à base 10.

C'est Briggs qui se chargera de la construction de tables du logarithme de plus en plus précises : 14 décimales pour tous les nombres compris en 1 et 20000, puis 15 décimales pour toutes les fonctions trigonométriques, pour chaque centième de degré.

Leurs travaux sont résumés dans l'ouvrage, *Arithmetica logarithmica* (1624).

Ces tables ont été rééditées à Londres sous leur forme définitive en 1663 et cet ouvrage est resté d'usage jusqu'au début du 19e siècle.

À partir de 1620, Briggs est professeur à Oxford. Il laissera peu de travaux portant sa propre marque, mais son activité pour populariser le **logarithme** mérite un grand respect.

Partie 2 : Etude d'une suite, programmation Python

Objectif : L'algorithme de Briggs permet d'obtenir des valeurs approchées de logarithmes népérien de nombres entiers à l'aide de fonctions élémentaires comme la racine carrée.

Cette activité propose de découvrir et d'implémenter l'algorithme de Briggs avec Python pour calculer des valeurs de logarithmes.

Principe de l'algorithme de Briggs

On considère un réel $a > 0$, et on souhaite évaluer $ln(a)$.

1) On pose $u_0 = a$ et pour tout $n \in N$, $u_{n+1} = \sqrt{u_n}$.

a) Démontrer par récurrence que pour tout $n \in \mathbb{N}$, $u_n = e^{\frac{ln(a)}{2^n}}$.

b) En déduire que (u_n) converge et que $\lim\limits_{n \to +\infty} u_n = 1$.

2) a) En calculant de deux façons le nombre dérivé de la fonction ln en 1, démontrer que $\lim\limits_{x \to 1} \frac{ln(x)}{x-1} = 1$.

b) En déduire que $\lim\limits_{n \to +\infty} \frac{\ln(u_n)}{u_n - 1} = 1$ et donc que $ln(u_n) \approx u_n - 1$.

c) Justifier que, pour tout $n \in \mathbb{N}$, $2^n ln(u_n) = ln(a)$, et en déduire que pour n suffisamment grand, on a $ln(a) \approx 2^n(u_n - 1)$.

Implémentation en langage Python

3) a) Dans cette question, on pose $a = 2$. On veut déterminer la plus petite valeur de n telle que $|u_n - 1| < 10^{-2}$, c'est à dire u_n "proche" de 1 à 10^{-2} près.

Etape 1 : Compléter l'algorithme pour que l'affichage ressemble au tableau de droite.

```
N ← 0
U ← 2
Tant que |u_n − 1| ≥ _ _ _ _
    N ← _ _ _ _
    U ← _ _ _ _
    Si |u_n − 1| ≥ 10^-2
        Res = « faux »
    Sinon
        Res = « _ _ _ _ »

Afficher ("n : ", n , "u : ", u, res)
```

Etape 2 : Compléter, écrire et exécuter le programme en langage Python.

```
from math import sqrt
n=0
u=2
while _ _ _ _ _ _ _ _ _ _ _ _:
    n = _ _ _ _ _
    u = _ _ _ _ _
    if abs(u-1) > =0.01 :
        res = "faux"
    else :
        res = _ _ _ _ _

print ("n : ", n , "u : ", u, res)
```

Etape 3 : Compléter le tableau

n	u_n	$\lvert u_n - 1 \rvert < 10^{-2}$
0	2	Faux
1	1,414213562	Faux
.....	....	....
.....	....	....
.....	....	....
.....	....	....
.....	....	....
.....	....	....
.....	....	....

En Python :

_ La syntaxe **from math import sqrt** permet d'utiliser la fonction **sqrt** qui permet de calculer une racine carrée

_ On peut utiliser la fonction **abs** pour le calcul d'une valeur absolue.

b) Quelle valeur approchée de $ln(2)$ peut-on en déduire ?

4) On considère un réel $a > 0$ et un entier $m \in \mathbb{N}$.

On souhaite obtenir une valeur approchée de $ln(a)$ pour le rang n tel que $|u_n - 1| < 10^{-m}$.

Compléter et écrire une fonction Python Briggs qui reçoit en argument la valeur a et l'entier m, et qui renvoie la valeur approchée de $ln(a)$ obtenue avec l'algorithme de Briggs.

```
from math import sqrt
def Briggs(a,m):
    n = 0
    u = a
    while abs(u-1)_ _ _ _ _ _ _ _
        n=_ _ _ _ _
        u=_ _ _ _ _ _
    return _ _ _ _ _ _
```

5) Utiliser la fonction Briggs avec $m = 9$ pour obtenir une valeur approchée de $\ln(2)$.

6) La syntaxe $from\ math\ import\ log$ permet d'utiliser la fonction Python log qui correspond à la fonction mathématique ln.

Effectuer une saisie pour obtenir la valeur de $ln(2)$. Vérifier la cohérence avec le résultat de la question précédente.

Partie 3 : Correction

Principe de l'algorithme de Briggs

1) On considère un réel $a > 0$, et on souhaite évaluer $ln(a)$.

On pose $u_0 = a$ et pour tout $n \in N, u_{n+1} = \sqrt{u_n}$.

a) Démontrer par récurrence que pour tout $n \in \mathbb{N}$, $\boldsymbol{u_n = e^{\frac{ln(a)}{2^n}}}$.

Notons $P(n)$ la propriété " $u_n = e^{\,ln(a)/2^n}$ ".

- Initialisation :
  $e^{\,ln(a)/2^0} = e^{\,ln(a)} = a = u_0$ donc $P(0)$ est vraie.
- Hérédité :
  Supposons que $P(n)$ soit vraie pour un $n \geq 0$ fixé.
  Alors $u_{n+1} = \sqrt{u_n} = \sqrt{e^{\,ln(a)/2^n}} = e^{\,\frac{1}{2}ln(a)/2^n} = e^{\,ln(a)/2^{n+1}}$
  $P(n+1)$ est donc vraie.
- Conclusion :
  La propriété $P(n)$ étant initialisée pour $n = 0$ et héréditaire pour $n \geq 0$, elle est vraie pour tout $n \geq 0$.

b) En déduire que $(u_n)n \geq 0$ converge et que $\lim\limits_{n \to +\infty} u_n = 1$.

Comme $\lim\limits_{n\to+\infty} \frac{ln(a)}{2^n} = 0$ et $\lim\limits_{X\to 0} e^X = 1$, on en déduit (par composition des limites) que $\lim\limits_{n\to+\infty} u_n = 1$.

2) a) En calculant de deux façons le nombre dérivé de la fonction ln en 1, démontrer que $\lim\limits_{x\to 1} \frac{ln(x)}{x-1} = 1$.

La fonction dérivée de ln est la fonction inverse, donc $ln'(1) = \frac{1}{1} = 1$

D'autre part, en écrivant ce nombre dérivé comme limite d'un taux d'accroissement, on a:

$$\lim_{x\to 1} \frac{ln(x) - ln(1)}{x-1} = 1$$

$$\lim_{x\to 1} \frac{ln(x)}{x-1} = 1$$

b) En déduire que $\lim\limits_{n\to+\infty} \frac{ln(u_n)}{u_n - 1} = 1$.

Comme $\lim\limits_{n\to+\infty} u_n = 1$ et $\lim\limits_{x\to 1} \frac{ln(x)-ln(1)}{x-1} = 1$, on en déduit par composition de limites que $\lim\limits_{n\to+\infty} \frac{ln(u_n)}{u_n-1} = 1$.

c) Justifier que $2^n ln(u_n) = ln(a)$, et en déduire que pour n suffisamment grand, on a $ln(a) \approx 2^n(u_n - 1)$.

$$2^n ln(u_n) = 2^n ln(e^{\,ln(a)/2^n}) = 2^n \times \frac{ln(a)}{2^n} = ln(a)$$

On a donc $\frac{ln(u_n)}{u_n-1} = \frac{ln(a)}{2^n(u_n-1)}$.

Comme $\lim\limits_{n\to+\infty} \frac{ln(u_n)}{u_n-1} = 1$, on en déduit que $\lim\limits_{n\to+\infty} 2^n(u_n - 1) = ln(a)$

Finalement, pour n suffisamment grand, on a $ln(a) \approx 2^n(u_n - 1)$.

Implémentation en langage Python

3) Dans cette question, on pose $a = 2$.
Compléter le tableau suivant jusqu'à obtenir une valeur u_n telle que :

$$|u_n - 1| < 10^{-2}$$

Quelle valeur approchée de $ln(2)$ peut-on en déduire ?

```python
from math import sqrt

n=0
u=2

while abs(u-1)>=0.01:
    n=n+1
    u=sqrt(u)
    if abs(u-1)>=0.01 :
        res = "faux"
    else :
        res = "vrai"

    print ("n : ", n , "u : ", u, res)
```

n	u_n	$\|u_n - 1\| < 10^{-2}$
0	2	Faux
1	1,414213562	Faux
2	1,189207115	Faux
3	1,090507733	Faux
4	1,044273782	Faux
5	1,021897149	Faux
6	1,010889286	Faux
7	1,005429901	Vrai

On en déduit que $ln(2) \approx 2^7(u_7 - 1) \approx 0{,}6950273424$

4) On considère un réel $a > 0$ et un entier $m \in \mathbb{N}$.

On souhaite obtenir une valeur approchée de $ln(a)$ pour le rang n tel que $|u_n - 1| < 10^{-m}$.

Compléter et écrire une fonction Python Briggs qui reçoit en argument la valeur a et l'entier m, et qui renvoie la valeur approchée de $ln(a)$ obtenue avec l'algorithme de Briggs.

```python
from math import sqrt

def Briggs(a,m):
    n=0
    u=a
    while abs(u-1)>=10**-m:
        n=n+1
        u=sqrt(u)
    return 2**n*(u-1)
```

5) Utiliser la fonction Briggs avec $m = 9$ pour obtenir une valeur approchée de $\ln(2)$.

```python
#Effectuer ici l'appel à la fonction Python Briggs
Briggs(2,9)
```

```
0.6931471824645996
```

6) La syntaxe $from\ math\ import\ log$ permet d'utiliser la fonction Python log qui correspond à la fonction mathématique ln.
Effectuer une saisie pour obtenir la valeur de $ln(2)$. Vérifier la cohérence avec le résultat de la question précédente.

```python
#Effectuer ici la saisie pour calculer ln(2)
from math import log
log(2)
```

```
0.6931471805599453
```