

Sts-Ciel 2 Co-anim - Suites numériques

Se loguer avec l'identifiant : **exam02.eleve** Mot de passe :

Ce DS est composé de 3 exercices indépendants. Les 3 fichiers .py constitués sont à déposer en fin d'épreuves dans le répertoire : **Examens(Z :)/exam02/copies**

Pour lancer PYZO, aller dans le répertoire **Examens(Z :)/exam02/sujets** et double-cliquer sur le raccourci :

Pyzo_win10_STI.cmd

Vous trouvez également dans ce répertoire le texte de ce sujet, en version pdf.

EXERCICE 1. :

On considère la suite (u_n) définie par son premier terme $u_0 = 2$ et par la relation $u_{n+1} = 2u_n - 1$

⇒ Ecrire dans un fichier nommé *suite_votreNom.py*, le code python d'une fonction nommée *suite()*. Elle prend en argument un entier n et retourne la valeur du nombre $S_{100} = \sum_{i=0}^{100} u_n$. On donne ci-contre une exécution en exemple :

```
>>> suite(1)
5
```

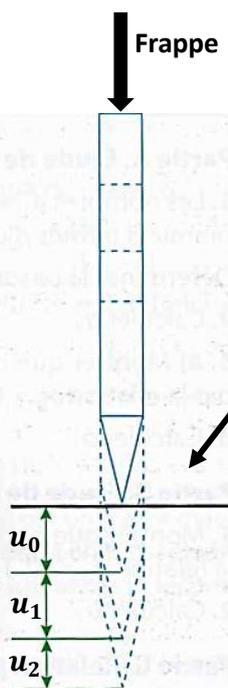
Le fichier sera à rendre dans le dossier
Examens(Z :)/exam02/copies

```
def suite(n) :
    s = 2
    u = 2
    for i in range(1,n+1) :
        u = 2*u-1
        s = s + u
    return s
```

CORRIGE

EXERCICE 2. : UTILISATION DES SUITES POUR PREVOIR L'ENFONCEMENT D'UN PIEU DE VIGNE

Un vigneron décide d'installer ses piquets à l'aide d'un enfonceur pendulaire à mat éleveur branché sur la prise de force du tracteur.



Un pieu est introduit initialement dans un trou de 10 cm de profondeur. A chaque frappe, il s'enfonce un peu plus.

Soit la suite (u_n) définie par :

$$u_0 = 10,$$

$u_1 =$ **déplacement** vertical du pieu après la 1^{ère} frappe,

$u_2 =$ **déplacement** vertical du pieu après la 2^{nde} frappe (voir figure ci-contre),

.....

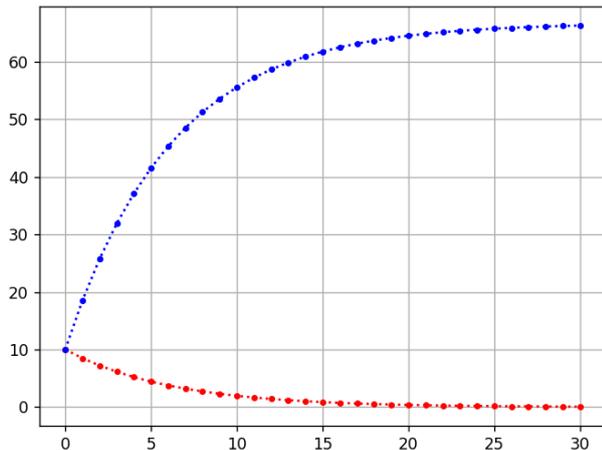
$u_n =$ déplacement vertical du pieu après la n^{ième} frappe.

Le sol étant de plus en plus dense, lors de l'enfoncement, entre 2 frappes successives, le déplacement vertical diminue à chaque fois de 15 %.

Ainsi u_1 est 15 % inférieur à u_0 , u_2 est 15 % inférieur à u_1 , etc u_{n+1} est 15 % inférieur à u_n

La relation de récurrence exprimant u_{n+1} en fonction de u_n est donc $u_{n+1} = 0,85 u_n$ avec $u_0 = 10$
 L'enfoncement total du pieu après n coup est égal à : $E_n = u_0 + u_1 + u_2 + \dots + u_n$

Le code python donné ci-contre est incomplet. En l'exécutant, on trace la courbe donnée ci-dessous, qui donne l'évolution des termes u_n de la suite (u_n) (déplacement à chaque frappe) et des termes E_n qui donnent l'enfoncement total du pieu.



⇒ Ecrire le code python complet dans un fichier nommé *vigne_votreNom.py*. Ce fichier sera à rendre dans le dossier **Examens(Z :)/exam02/copies**

```
from matplotlib.pyplot import plot,grid,show

def frappe(n) :
    X = [0]
    U = [10]
    E = [10]
    u = 10
    e = 10
    for i in range(1,n+1) :
        u = u * 0.85
        e = e + u
        X.append(i)
        U.append(u)
        E.append(e)
    return X,U,E

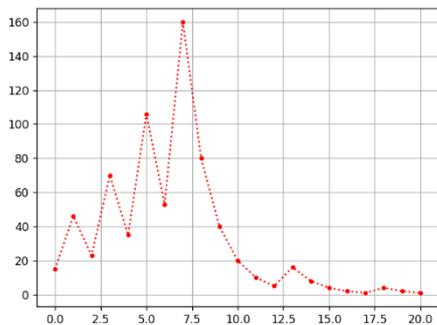
# Programme principal
X,U,E = frappe(30)
plot(X,U,"r:.")
plot(X,E,"b:.")
grid(True)
show()
```

EXERCICE 3. : CONJECTURE DE SYRACUSE

On considère la suite (u_n) définie par son premier terme $u_0 = N$ et par les relations :

$$u_{n+1} = \frac{u_n}{2} \quad \text{si } u_n \text{ est pair}$$

$$u_{n+1} = 3u_n + 1 \quad \text{si } u_n \text{ est impair}$$



Le code python donné ci-contre est incomplet. En l'exécutant, on trace la courbe donnée ci-dessous, qui donne l'évolution des termes u_n de la suite (u_n) de u_0 à u_{20} pour un 1^{er} terme $u_0 = 15$

⇒ Ecrire le code python complet dans un fichier nommé *syracuse_votreNom.py*. Ce fichier sera à rendre dans le dossier **Examens(Z :)/exam02/copies**

Aide : Pour savoir si un nombre i est pair ou impair, on réalise l'opération $i\%2$. Si ce résultat est égal à 0, i est un nombre pair, s'il est égal à 1, i est impair. Par exemple :

```
if i%2 == 0 :
    print("je suis un nombre paire")

if i%2 == 1 :
    print("je suis un nombre impaire")
```

```
def syracuse(n,N) :
    X = [0]
    Y = [N]
    u = N
    i = 0
    while i !=n :
        i = i + 1
        if u%2 == 0 : u = u/2
        else : u = 3*u+1
        X.append(i)
        Y.append(u)
    return X,Y

# Programme principal
X,Y = syracuse(n=20,N=15)
plot(X,Y,"r:.")
grid(True)
show()
```