Les fonctions et récursivité

Série 3

Exercice 1

On se propose de calculer une valeur approchée de la constante K de Catalan en utilisant la formule suivante :

$$K=1-\frac{1}{3^2}+\frac{1}{5^2}-\frac{1}{7^2}+\dots$$

*Ecrire une fonction **val_app(epsilon)** qui permet de retourner une valeur approchée de la constante K en utilisant la formule ci-dessus et en s'arrêtant dès que la valeur absolue de la différence entre deux somme successives devienne inférieure ou égale à une erreur epsilon donnée en paramètre.*

⇒ *Corrigé*

```
def val_app(epsilon) :
        elm1=1
        elm2=elm1-1/(3**2)
        signe=1
        val=5
        while abs(elm2-elm1)>epsilon :
                elm1=elm2
                elm2=elm1+signe*(1/(val**2))
                val+=2
                signe*=(-1)
        return elm2
```

Exercice 2

Soit la formule suivante qui permet de déterminer une valeur approchée de Cos(x) :

$$Cos(x)=1-\frac{x^2}{2!}+\frac{x^4}{4!}-\frac{x^6}{6!}+\cdots$$

*Ecrire une fonction **Calcul_Cos(x)** qui permet de :*

- *Saisir un réel x appartenant à l'intervalle [-1, 1],*
- *Calculer et afficher une valeur approchée de Cos(x) en utilisant la formule donnée ci-dessus. Le calcul s'arrête lorsque la différence entre deux termes consécutifs devient inférieure à 10^{-4}.*

⇒ *Corrigé*

```
def Calcul_Cos(x) :
    elm1=1
    elm2=elm1-(x**2/2)
    signe=1
    puis=2
    f=2
    while abs(elm2-elm1)>0.0001:
        f=f*(puis+1)*(puis+2)
        puis+=2
        elm1=elm2
        elm2=elm1+signe*(x**puis/f)
        signe*=(-1)
    return elm2
```

Exercice 3

Soit la suite U définie par :

-U_0 est un entier positif pris au hasard (avec $3<U_0<40$

-$Un=U_{n-1}/2$ si U_{n-1} est pair, sinon $Un=3*U_{n-1}+1$ (n>0)

cette suite aboutit au cycle redondant formé par les trois termes 4,2,1 à partir d'un certain rang.

- ***Exemple :***

Pour $U_0=3$

*$U_1=10$ $U_2=5$ $U_3=16$ $U_4=8$ **$U_5=4$ $U_6=2$ $U_7=1$** $U_8=4$ $U_9=2$ $U_{10}=1$,*

Donc la suite U entre dans le cycle redondant 4,2,1 à partir du $6^{ème}$ terme(rang=6)

*Ecrire une fonction Python permettant de déterminer le rang à partir duquel la suite U aboutit au cycle redondant **4, 2** et **1***

⇒ *Corrigé*

```
import random
def rang() :
    u0=random.randrange(3,40)
    pos=1
    while u0!=4:
        if u0%2==0:
            Un=u0//2
        else:
            Un=3*u0+1
        u0=Un
        pos+=1
    return pos
```

Exercice 4

Ecrivez un programme Python permettant de calculer la limite à ε près de la suite définie par la relation de récurrence :

$U_0 = 2$ et $U_{n+1} = U_n + 2/U_n$, $n \geq 0$.

On arrête d'itérer quand l'intervalle entre 2 termes consécutifs devient strictement inférieur à ε.

⇒ *Corrigé*

```
def limite(epsilon) :
    U0=2
    Un=U0+2/U0
    while abs(Un-U0)>epsilon :
        U0=Un
        Un=U0+2/U0
    return Un
```

Exercice 5

Ecrivez un programme Python permettant de calculer la nième terme de la suite définie par :

- $F_0=1$, $F_1=2$
- $F_n= 4F_{n-1} + 3F_{n-2}$ $(n \geq 2)$

⇒ **Corrigé**

```
def suite(n) :
        F0,F1=1,2
        for i in range(2,n)
                Fn=4*F1+3*F0
                F0=F1
                F1=Fn
        return Fn
```

Exercice 6

La suite de Fibonacci est définie comme suit :

$$Un\begin{cases} 1 \ \ si\, n < 2 \\ U_{n-1} + U_{n-2} \ sinon \end{cases}$$

Ecrire une fonction récursive calculant Fib(n)

⇒ **Corrigé**

```
def Fib(n) :
        if n<2 :
                return 1
        else :
                return  Fib(n-1)+ Fib(n-2)
```

Exercice 7

Soit la suite définie par :

$$Un\begin{cases} 1 \ \ si\, n < 2 \\ 3U_{n-1} + U_{n-2} \ sinon \end{cases}$$

Ecrire une fonction récursive permettant de calculer le nième terme de la suite.

$\Rightarrow$ *Corrigé*

```
def Suite(n) :
        if n<2 :
                return 1
        else :
                return  3*Suite (n-1)+ Suite (n-2)
```

Exercice 8

Soient u et v les deux suites définies par :

$$\begin{cases} u_0 = a \\ v_0 = b \end{cases} \quad et \quad \forall n > 0 \begin{cases} u_n = u_{n-1}^2 + 2v_{n-1} \\ v_n = v_{n-1} + 3u_{n-1} \end{cases}$$

Ecrire deux fonctions CalculerU (a,b,n) et CalculerV(a,b,n) pour calculer respectivement les deux termes Un et Vn des deux suites.

$\Rightarrow$ *Corrigé*

```
def CalculerU(a,b,n) :
        if n==0 :
                return a
        else :
                return  (CalculerU(a,b,n-1))**2+ 2*CalculerV(a,b,n-1)
def CalculerV(a,b,n) :
        if n==0 :
                return b
        else :
                return  CalculerV(a,b,n-1)+ 3*CalculerU(a,b,n-1)
```

Exercice 9

Considérons la méthode suivante pour calculer X^n :

$$X^n = \begin{cases} X^{n/2} * X^{n/2} & si\_n\_est\_pair \\ X^{n/2} * X^{n/2} * X & si\_n\_est\_impair \end{cases}$$

n/2 représente la division entière de n par 2.

Ecrire une fonction récursive pour calculer X^n

⇒ Corrigé

```
def puissance(x,n) :
        if n==1:
                return x
        if n%2==0 :
                return puissance(x,n//2)* puissance(x,n//2)
        else :
                return  puissance(x,n//2)* puissance(x,n//2)*puissance(x,1)
```