TP 02 – Exercices supplémentaires

Exercice 1 **[Suite définie par récurrence]** Soit $(u_n)_{n\in\mathbb{N}}$ la suite définie par

$$u_0 = 2 \qquad \text{et} \qquad \forall n \in \mathbb{N},\ \ u_{n+1} = 3u_n^2 - 1$$

Écrire une fonction prenant en argument d'entrée un entier naturel n et renvoyant u_n en sortie. *On vérifiera que la valuation de la fonction en* `3` *donne* `362`.

Entrée [1]:
```
def suite1(n):
    u = 2
    for k in range(1, n+1):
        u = 3*u**2-1
    return u
```

Entrée [2]:
```
suite1(3)
```

Out [2]:
```
362
```

Exercice 2 **[Suite récurrente linéaire d'ordre 2]** Soit $(u_n)_{n\in\mathbb{N}}$ la suite définie par

$$u_0 = 1,\ u_1 = 1 \qquad \text{et} \qquad \forall n \in \mathbb{N},\ \ u_{n+2} = u_{n+1} + u_n$$

Écrire une fonction prenant en argument d'entrée un entier naturel n et renvoyant u_n en sortie. *On vérifiera que la valuation de la fonction en* `0` *et en* `1` *renvoie bien* `1` *et que la valuation de la fonction en* `4` *donne* `5`.

Entrée [3]:
```
def suite2(n):
    if n == 0:
        return 1
    elif n == 1 :
        return 1
    else :
        u,v = 1,1
        for k in range(2, n+1):
            u,v=v,u+v
        return v
```

Entrée [4]:
```
suite2(0)
suite2(1)
suite2(4)
```

Out [4]:
```
1
1
5
```

Exercice 3 **[Suite définie par une somme]** Soit $(u_n)_{n\in\mathbb{N}}$ la suite définie par

$$\forall n \in \mathbb{N}, \quad u_n = \sum_{k=0}^{n} \frac{k}{2^k}$$

Écrire une fonction prenant en argument d'entrée un entier naturel n et renvoyant u_n en sortie. *On vérifiera que la valuation de la fonction en* `1` *renvoie* `0.5` *et que la valuation de la fonction en* `4` *donne* `1.625`.

Entrée [5]:
```
def suite3(n):
    S = 0
    for k in range(0, n+1):
        S = S + k/2**k
    return S
```

Entrée [6]:
```
suite3(1)
suite3(4)
```

Out [6]:
```
0.5
1.625
```

Exercice 4 **[Suite définie par un produit]** Soit $(u_n)_{n\geqslant 2}$ la suite définie par

$$\forall n \in\geqslant 2, \quad u_n = \prod_{k=2}^{n} \left(1 - \frac{1}{k^3}\right)$$

Écrire une fonction prenant en argument d'entrée un entier naturel n et renvoyant u_n en sortie. *On vérifiera que la valuation de la fonction en* `2` *renvoie* `0.875` *et que la valuation de la fonction en* `4` *donne environ* `0.83`.

Entrée [7]:
```
def suite4(n):
    P = 1
    for k in range(2, n+1):
        P = P * (1-1/k**3)
    return P
```

Entrée [8]:
```
suite4(2)
suite4(4)
```

Out [8]:
```
0.875
0.8294270833333334
```

Exercice 5 **[Suites imbriquées]** Soient $(u_n)_{n\in\mathbb{N}}$ et $(u_n)_{n\in\mathbb{N}}$ les suites définies par

$$u_0 = 1,\ v_0 = 0 \qquad \forall n \in \mathbb{N},\quad u_{n+1} = u_n + v_n \text{ et } v_{n+1} = u_n + 2v_n.$$

Écrire une fonction prenant en argument d'entrée un entier naturel n et renvoyant u_n et v_n en sortie. *On vérifiera que la valuation de la fonction en* `0` *renvoie* `(1,0)` *et que la valuation de la fonction en* `4` *donne* `(13,21)`.

Entrée [9]:
```
def suite5(n):
    u,v = 1,0
    for k in range(1,n+1):
        u,v = u+v, u+2*v
    return(u,v)
```

Entrée [10]:
```
suite5(0)
suite5(4)
```

Out [10]:
```
(1,0)
(13,21)
```

Exercice 6 **[Suite récurrente linéaire d'ordre supérieur]** Soit $(u_n)_{n\in\mathbb{N}}$ la suite définie par

$$u_0 = 1, \qquad \text{et} \qquad \forall n \in \mathbb{N},\quad u_{n+1} = \sum_{k=0}^{n} \frac{u_k}{k^2+1}$$

Écrire une fonction prenant en argument d'entrée un entier naturel n et renvoyant u_n en sortie. *On vérifiera que la valuation de la fonction en* `0` *et en* `1` *renvoie bien* `1` *et que la valuation de la fonction en* `4` *donne* `1.98`.

Pour simplifier l'écriture de ce programme, on peut remarquer que,

$$\forall n \in \mathbb{N}^*, u_{n+1} = \sum_{k=0}^{n-1} \frac{u_k}{k^2+1} + \frac{u_n}{n^2+1} = u_n + \frac{u_n}{n^2+1} = \frac{u_n \times (n^2+2)}{n^2+1}$$

Entrée [11]:
```
def suite6(n):
    if n == 0 :
        return(1)
    else :
        u = 1 #ici u représente u1
        for k in range(2, n+1):
            u = (u*((k-1)**2+2))/((k-1)**2+1)
        return u
```

Entrée [12]:
```
suite6(0)
suite6(1)
suite6(4)
```

Out [12]:
```
1
1
1.98
```

Exercice 7 **[Autour des coefficients binomiaux]** Pour tout $n \in \mathbb{N}^*$ et pour tout $k \in \{1, \dots, n\}$, on définit

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

L'objectif de cet exercice est d'étudier plusieurs techniques de calculs de coefficients binomiaux.

1. Première méthode : à partir de la définition.

   (a) Écrire une fonction, appelée `factorielle`, qui prend en argument un entier `n` et qui renvoie la valeur de `n!`. *On vérifiera que* `factorielle(5)` *renvoie* `120`.

Entrée [13]:
```
def factorielle(n):
    P=1
    for k in range(1,n+1):
        P=P*k
    return(P)
```

Entrée [14]:
```
factorielle(5)
```

Out [14]:
```
120
```

   (b) En déduire une fonction, appelée `binom1`, qui prend en argument deux entiers `n` et `k`, et qui renvoie la valeur du coefficient binomial de $\binom{n}{k}$. *On vérifiera que* `binom1(2,5)` *renvoie* `10.0`.

Entrée [15]:
```
def binom1(k,n):
    c = factorielle(n)/(factorielle(k)*factorielle(n-k))
    return c
```

Entrée [16]:
```
binom1(2,5)
```

Out [16]:
```
10.0
```

2. Deuxième façon.

   (a) Montrer que,

$$\forall n \in \mathbb{N}^*, \forall k \in \{1, \dots, n\}, \qquad \binom{n}{k} = \frac{n(n-1) \times \cdots \times (n-k+1)}{k!}$$

   À partir de l'égalité précédente, on en déduit que

$$\forall n \in \mathbb{N}^*, \forall k \in \{1, \dots, n\}, \qquad \binom{n}{k} = \prod_{i=1}^{k} \frac{n-i+1}{i}$$

   (b) En déduire une nouvelle fonction, appelée `binom2`, qui prend en argument deux entiers `n` et `k`, et qui renvoie la valeur du coefficient binomial de $\binom{n}{k}$. *On vérifiera que* `binom1(2,5)` *renvoie* `10.0`.

Entrée [17]:
```
def binom2(k,n):
    P = 1
    for i in range(1, k+1):
        P = P*((n-i+1)/i)
    return P
```

Entrée [18]:
```
binom2(2,5)
```

Out [18]:
```
10.0
```