# DM 03 – Corrigé Informatique

**Exercice 1** On considère la suite $(u_k)_{k\in\mathbb{N}}$ définie par

$$\forall k \in \mathbb{N}, \qquad u_k = \frac{1}{k^2+1}$$

Écrire une fonction, appelée `exo1`, qui prend en argument un entier `n` et qui renvoie la liste de tous les termes de la suite (depuis $u_0$) jusqu'au `n`-ième. *On vérifiera que l'évaluation de* `exo1` *en* `3` *donne une liste finissant par* `0.1`.

Entrée [1]:
```
def exo1(n):
    L=[]
    for k in range(0, n+1):
        u = 1/(k**2+1)
        L.append(u)
    return(L)
```

Entrée [2]:
```
exo1(3)
```

Out [2]:
```
[1.0, 0.5, 0.2, 0.1]
```

**Exercice 2** On considère la suite $(u_k)_{k\in\mathbb{N}}$ définie par

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

1. Écrire une fonction, appelée `exo2`, qui prend en argument un entier `n` et qui renvoie le terme $u_n$ de la suite. *On vérifiera que l'évaluation de* `exo2` *en* `4` *donne environ* `1.99`.

Entrée [3]:
```
import numpy as np

def exo2(n):
    u = 0
    for k in range(1, n+1):
        u = np.sqrt(u+2)
    return(u)
```

Entrée [4]:
```
exo2(4)
```

Out [4]:
```
1.9903694533443939
```

2. Écrire une fonction, appelée `listeexo2`, qui prend en argument un entier `n` et qui renvoie la liste de tous les termes de la suite (depuis $u_0$) jusqu'au `n`-ième. *On vérifiera que l'évaluation de* `listeexo2` *en* `3` *donne une liste finissant par environ* `1.96`.

Entrée [5]:
```
import numpy as np

def listeexo2(n):
    u=0
    L=[0]
    for k in range(1, n+1):
        u = np.sqrt(u+2)
        L.append(u)
    return(L)
```

Entrée [6]:
```
listeexo(3)
```

Out [6]:
```
[0, 1.4142135623730951, 1.8477590650225735, 1.9615705608064609]
```

**Exercice 3** On considère la suite $(S_n)_{n\in\mathbb{N}}$ définie par

$$\forall n \in \mathbb{N}, \qquad S_n = \sum_{k=0}^{n} k$$

1. Donner, pour tout $k \in \mathbb{N}$, une relation entre $S_{k+1}$ et $S_k$.

   On a,
$$\forall k \in \mathbb{N}, \qquad S_{k+1} = S_k + (k+1)$$

2. Écrire une fonction, appelée `sommeentiers`, qui prend en argument un entier `n` et qui renvoie la valeur de la somme $S_n$. *On vérifiera que l'évaluation de* `sommeentiers` *en* `6` *donne* `21`.

Entrée [7]:
```
def sommeentiers(n):
    S=0
    for k in range(0,n):
        S = S + k+1
    return(S)
```

Entrée [8]:
```
sommeentiers(6)
```

Out [8]:
```
21
```

**Exercice 4** Écrire un programme calculant $S = \sum_{k=1}^{1000} \frac{1}{k^2}$. Vérifier que $S \approx 1.64$.

Entrée [9]:
```
S=0
for k in range(0,1000):
    S = S + 1/(k+1)**2
print(S)
```

Out [9]:
```
1.6439345666815615
```

**Exercice 5** Trouver, pour tout $k \in \mathbb{N}$, une relation entre $(k+1)!$ et $k!$. En déduire le script d'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 [10]:
```
def factorielle(n):
    P=1
    for k in range(1,n+1):
        P=P*k
    return(P)
```

Entrée [11]:
```
factorielle(5)
```

Out [11]:
```
120
```