

Exercice 1

1.

(a)

i.

```
def somme(n):  
    s = 0  
    for k in range(1, n+1):  
        s = s + 2*k*k/k**2  
    return s
```

ii.

```
def somme_nieux(n):  
    s = 0 ; p = 1  
    for k in range(1, n+1):  
        p = 2*k  
        s = s + p/k*k  
    return s
```

(5)

i.

```

def seuil (n):
    n=1
    while Somme(n) < M:
        n=n+1
    return n

```

ii.

$$\forall n \in \mathbb{N}^*, S_{n+1} = \sum_{k=1}^{n+1} \frac{2^k}{k^2} = \sum_{k=1}^n \frac{2^k}{k^2} + \frac{2^{n+1}}{(n+1)^2} = S_n + \frac{2^{n+1}}{(n+1)^2} \quad \checkmark$$

$$S_1 = \frac{2^1}{1^2} \text{ donc } \underline{S_1 = 2}$$

iii.

```

def seuil_mieux (n):
    n=1; S=2
    while S < n:
        n=n+1
        S = S + 2^n/n^2
    return n

```

(2)

(a) on peut prendre rd.randrange(2) ou rd.randint(0,1)

de la bibliothèque random (import random as rd)

→ ils renvoient des ns aléatoires 0, ou 1 avec équiprobabilité

(b)

```

import random as rd
def jeu()
    n=1
    while rd.randrange(2) != 1:
        n=n+1
    return n

```