

## Devoir d'informatique

### Corrigé

#### Exercice 1

```
n = int(input('Saisir un entier n:'))
if n % 2 == 0 :
    p = (n+4)/(2*n)
else :
    p = 5*n - 1
print(p)
```

#### Exercice 2

1. 

```
L = range(2,101,2)
# ou
L = [2*k for k in range(1,51)]
# ou
L = []
for k in range(1,51):
    L.append(2*k)
# ou
L = []
for k in range(1,101):
    if k%2 == 0 : # si k est pair
        L.append(k)
```
2. 

```
L[2] = 17
```
3. 

```
L[-1] = 4
#ou
L[49] = 4 # car la liste a 50 éléments
```
4. 

```
len(L)
```
5. 

```
a in L # renvoie True si a est dans L et False sinon.
```

#### Exercice 3

1.

| Étape          | Calcul à faire        |
|----------------|-----------------------|
| Initialisation | $u_0 = 2$             |
| $k = 0$        | $u_1 = \ln(3u_0 + 0)$ |
| $k = 1$        | $u_2 = \ln(3u_1 + 1)$ |
| $k = 2$        | $u_3 = \ln(3u_2 + 2)$ |

2. 

```
import numpy as np
```

```
def SuiteU(n):
    u = 2
    for k in range(0,n) :
        u = np.log(3*u+k)
    return u
```

3. # Solution 1

```
u = 2
n = 0
while u < 1000 :
    u = np.log(3*u+n)
    n = n+1
print(n)
```

# Solution 2

```
u = 2
while SuiteU(n) < 1000 :
    n = n+1
print(n)
```

4. # Solution 1

```
S = 0
u = 2
for k in range(0,101) :
    S = S + u
    u = np.log(3*u+k)
print(S)
```

# Solution 2

```
u = 2
S = 2
for k in range(0,100) :
    u = np.log(3*u+k)
    S = S + u
print(S)
```

```
# Solution 3
S = 0
for k in range(0,101) :
    S = S + SuiteU(k)
print(S)
```

### Exercise 4

```
import numpy.random as rd
n = 1
A = rd.randint(0,n+1) # Abscisse à l'instant n = 1
while A != 0 : # ou while A > 0 :
    n = n+1
    A = rd.randint(0,n+1)
print(n)
```

### Exercise 5

```
1. # Solution 1
def maximum(L):
    m = L[0]
    for k in range(len(L)) :
        if L[k] > m :
            m = L[k]
    return m

# Solution 2
def maximum(L):
    m = L[0]
    for x in L :
        if x > m :
            m = x
    return m

2. def hist(L,n):
    H = []
    for k in range(n+1):
        Nk = 0
        for i in range(len(L)):
            if L[i] == k :
                Nk = Nk + 1
        H.append(Nk)
    return H

# ou
def hist(L,n):
    H = []
    for k in range(n+1):
        Nk = L.count(k)
        H.append(Nk)
    return H

3. def croissant(L) :
    H = hist(L, maximum(L))
    C = []
    for k in range(len(H)) :
        Nk = H[k]
        C = C + Nk * [k] # on ajoute Nk fois le nombre k
    return C
```