

Informatique TB

Corrigé

2023

Préliminaires

1.

```
def ber(p):  
    if random()<=p:  
        return 1  
    else:  
        return 0
```

Comme la valeur de `True` est 1 et celle de `False` est 0, on peut écrire le code

```
def ber(p):  
    return (random()<=p)
```

I Trafic routier dans un tunnel

2.

`len(T)` renvoie 8.

3. :

`T[1]` renvoie 1.

4. :

`[0, 0, 1, 0, 1, 1, 0, 1]`

5. `def tunnel (T):`

```
L=len(T)*[0]  
for i in range (0, len(T) -1):  
    if T[i]==1 and T[i +1]==0:  
        L[i+1]=1  
        L[i]=0  
    if T[i]==1 and T[i +1]==1:  
        L[i]=1  
return (L)
```

6. `def tunnel2(T,p):`

```
L=len(T)*[0]  
for i in range (0, len(T) -1):  
    if T[i]==1 and T[i +1]==0:  
        L[i+1]=1  
        L[i]=0  
    if T[i]==1 and T[i +1]==1:
```

```

        L[i]=1
    if T[0]==0:
        L[0]=ber(p)
    return (L)

```

```

7. def evol2(T,n,p):
    L=T
    print (L)
    for i in range (n):
        L=tunnel2(L,p)
    print (L)

```

II Motif sur un coquillage

8.

0	0	1	0	1	0	0	1	1	1	0
0	1	1	0	1	1	1	0	1	0	0

9. On constate que la case i vaut 1 ssi le nombre de voisins est impair.

```

def coq(T):
    L=len(T)*[0]
    for i in range (1, len(T) -1):
        if (T[i-1]+T[i]+T[i+1])%2==1:
            L[i]=1
    return (L)

```

```

10. def evol4 (T,n):
    A=[T]
    L=T
    for i in range (n):
        L=coq(L)
        A.append(L)
    return (A)

```

Pour obtenir l'image du coquillage

```

import numpy as np
from PIL import Image

T=401*[0]
T[200]=1
A=np.array(evol4(T,401),dtype=np.uint8)
A=A*255
# Convert `a` to a PIL Image
img = Image.fromarray(A)
img.show()

```

11. La proposition 3 est la seule qui donne le résultat attendu.

```

12. def voisins(M,i,j):
    compteur=0
    n=len(M) #la matrice est supposée carrée
    for k in [-1,0,1]:
        for h in [-1,0,1]:
            if 0<=i+k<=n-1 and 0<=j+h<=n-1 and (h,k)!=(0,0):
                compteur+=M[i+k][j+h]
    return compteur

```

```
def epi(T,p):
    n=len(T) # la matrice est supposée carrée
    M=[[0]*n for i in range(n)]
    for i in range(n):
        for j in range(n):
            if T[i][j]==1:
                M[i][j]=1
            else:
                M[i][j]=ber(p*voisins(T,i,j))
    return M
```