T.D. n°8 : Correction

Exercice 1

```
def appartient(x,L):
    y=False
    for i in range(len(L)):
        if L[i]==x:
            y=True
    return y
```

Exercice 2

```
import numpy.random as rd
L=rd.randint(1,100,20)
print('L =',L)

max=L[0]
place=0
for i in range(1,len(L)):
    if L[i]>max:
        max=L[i]
        place=i
print('maximum : ',max)
print('place : ',place)

n=0
while L[n]==max:
    n=n+1

place2=n;max2=L[n]
for i in range(n+1,len(L)):
    if L[i]>max2 and L[i]<max:
        max2=L[i]
        place2=i
print('maximum2 : ',max2)
print('place2 : ',place2)
```

Exercice 3

```
L=[[1,2],[0,1,2],[0,1,3],[2]]
M=np.array([[0,1,1,0],[1,1,1,0],[1,1,0,1],[0,0,1,0]])
```

Exercice 4

```
def mat_adj(L):
    n=len(L)
    M=np.zeros((n,n))
    for i in range(n):
        for j in L[i]:
            M[i,j]=1
    return M

def lst_adj(M):
    n=len(M)
    L=[[] for i in range(n)]
    for i in range(n):
        for j in range(n):
            if M[i,j]==1:
                L[i].append(j)
    return(L)
```

Exercice 5

```
def degré(M,s):
    return np.sum(M[s])
```

Exercice 6

```
def Est_connexe(M):
    n=len(M)
    Id=np.eye(n,n)
    S=Id
    for k in range(1,n):
        S=S+al.matrix_power(M,k)
    if np.min(S)> 0:
        return(True)
    else:
        return(False)
```