T.D. n°8 Listes et graphes

Exercice 1

1) Compléter la fonction suivante :

```
def appartient(x,L):
    y=…
    for i in range(len(L)):
        if …
            y=…
    return y
```

afin qu'elle retourne True si x appartient à la liste L, et False sinon. (Sans utiliser ni x in L, ni L.count)

2) Tester cette fonction avec L = [2,5,1,4,7] et x = 4, puis x = 3.

Exercice 2

1) Compléter le programme suivant, afin qu'il déterminer et affiche :

_ le maximum des valeurs prises par L

_ la place de ce maximum (ou la première des places s'il est présent plusieurs fois)

```
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 …
        max=…
        place=…
print('maximum : ',max)
print('place : ',place)
```

2) Dans cette question, on supposera que la liste n'est pas constante.

On ajoute les lignes suivantes au programme précédent :

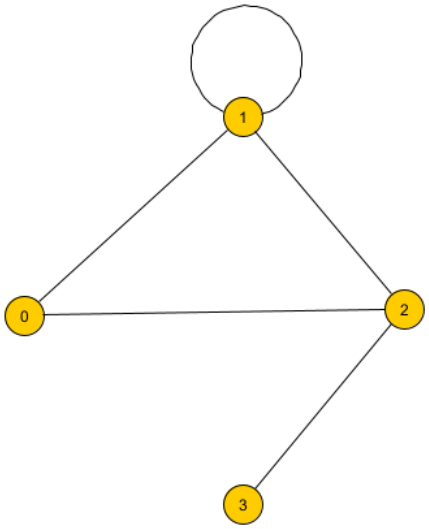
```
n=0
while L[n]==max:
    n=n+1
place2=n;max2=L[n]
for i in … :
    if … and …:
        max2=…
        place2=…
print('maximum2 : ',max2)
print('place2 : ',place2)
```

a) Sans utiliser Spyder, donner les valeurs de place2 et max2 après la quatrième ligne pour la liste L=[10,10,3,5,7,1,10,2].

b) Compléter ces lignes de programme afin de déterminer la deuxième valeur la plus grande prise par L (différente de la valeur du maximum), et la première place où cette valeur est prise.

Exercice 3

On considère le graphe suivant :



Ecrire en Python sa liste d'adjacence, ainsi que sa matrice d'adjacence.

Exercice 4

1) Compléter la fonction suivante pour qu'elle prenne en entrée la liste d'adjacence d'un graphe et qu'elle renvoie la matrice d'adjacence. Vérifier avec le graphe précédent.

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

2) Compléter la fonction suivante pour qu'elle prenne en entrée la matrice d'adjacence d'un graphe, et qu'elle renvoie la liste d'adjacence. Vérifier avec le graphe précédent.

```
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]…:
                …
    return(L)
```

Exercice 5

Ecrire une fonction degré(M,s) qui prend en entrée la matrice d'adjacence d'un graphe non orienté et le numéro d'un sommet s et renvoie le degré du sommet s.
(On rappelle que, si M est une matrice, M[i] désigne la i-ème ligne de M).
Vérifier avec le graphe précédent.

Exercice 6

Ecrire une fonction est_connexe(M) qui prend en entrée la matrice d'adjacence d'un graphe non orienté et renvoie True ou False selon qu'il est connexe ou non.
Vérifier avec le graphe précédent.