

INFO-H-100 – Informatique – Prof. Th. Massart
1^{ère} année du grade de Bachelier en Sciences de l'Ingénieur
Interrogation de Juin

Remarques préliminaires

- On vous demande de répondre à **chaque question sur une feuille séparée**.
- N'oubliez pas d'inscrire votre nom, prénom et numéro de matricule sur chaque feuille.
- Vous disposez de 2 heures 45 minutes et vous ne pouvez pas utiliser de notes.
- La réponse à la question doit comprendre, si approprié, le code *Python* structuré et conforme aux règles de bonne pratique et conventions ainsi que des commentaires pertinents.
- Vous pouvez ajouter des fonctions si cela vous semble nécessaire.
- Sauf mention contraire, vous ne pouvez utiliser aucune fonction de librairies (pas d'import).

Solution de la question 4

```
a = [[3,4,[9,1,3],4,5,[3,[1,2],4,[2,1]]],7,8,[9,1,5,[5,2,3],7]]
print(a)

if(type(a) is list):
    print('ok')

print(type(a))

def profondeur(liste):
    prof = 1 #profondeur par défaut d'une liste vaut 1
    for elem in liste:
        if(type(elem) is list):
            #si un des éléments de la liste est une liste on appelle récursivement
            #notre fonction profondeur pour calculer la profondeur de celui-ci
            p = profondeur(elem) + 1
            if p > prof:
                #on vérifie que la profondeur est plus grande que la profondeur
                #actuelle. En effet, si notre liste contient plusieurs listes nous calculons
                #différentes profondeurs, il faut conserver la valeur max
                prof = p
    return prof

print(profondeur(a))
```

Solution de la question 5

```
def get_longest_sentence(filename):
    string = ""
    length = 0
    f = open(filename,"r+")
    for line in f.readlines():
        words = line.split(" ")
        if length < len(words):
            length = len(words)
            string = line
    f.close()
    return (string,length)

print(get_longest_sentence("text.txt"))
```

Solution de la question 6

```
import datetime

def annee(personne):
    return personne["Birthdate"][2]

def mois(personne):
    return personne["Birthdate"][1]
```

```

def jour(personne):
    return personne["Birthdate"][0]

def date_de_naissance(personne):
    return datetime.date(annee(personne),mois(personne),jour(personne))

def relation_ordre(personne1,personne2):
    return date_de_naissance(personne1) > date_de_naissance(personne2)

def tri_selection(ls):
    """ Procedure du tri par selection """
    for i in range(len(ls) - 1):
        pos = pos_min(ls, i)
        echange(ls, i, pos)

def echange(ls, i1, i2):
    """ Echange dans la liste 'ls', les valeurs des indices i1 et i2"""
    ls[i1], ls[i2] = ls[i2], ls[i1]

def pos_min(ls,i):
    mini = i
    for j in range(i,len(ls)):
        if relation_ordre(ls[mini],ls[j]):
            mini = j
    return mini

#####

def tri_insertion(ls):
    for i in range(1, len(ls)):
        j = pos_insert(ls[i], ls, i)
        deplacer(ls, i, j)

def deplacer(ls,i,j):
    """ Deplace dans la liste 'ls' la valeur en position i vers la position j"""
    val = ls[i]
    del ls[i]
    ls.insert(j,val)

def pos_insert(val, ls, n):
    """ Renvoi la position d'insertion de la valeur 'val' parmi les n 1er elements de la liste"""
    j = 0
    while j < n and relation_ordre(val,ls[j]):
        j += 1
    return j

#####

if __name__ == "__main__":

    liste = [{"Lastname" : "Dupont", "Firstname" : "Pierre", "Birthdate" : (20,12,1980)},
              {"Lastname" : "Dyrshvak", "Firstname" : "Stevie", "Birthdate" : (18,8,1990)},
              {"Lastname" : "Ninak", "Firstname" : "Heiashi", "Birthdate" : (3,1,1984)},
              {"Lastname" : "Leloup", "Firstname" : "Marie", "Birthdate" : (16,3,1989)}]

    print("Avant :",liste)
    tri_selection(liste)
    print("Apres :",liste)

    liste = [{"Lastname" : "Dupont", "Firstname" : "Pierre", "Birthdate" : (20,12,1980)},
              {"Lastname" : "Dyrshvak", "Firstname" : "Stevie", "Birthdate" : (18,8,1990)},
              {"Lastname" : "Ninak", "Firstname" : "Heiashi", "Birthdate" : (3,1,1984)},
              {"Lastname" : "Leloup", "Firstname" : "Marie", "Birthdate" : (16,3,1989)}]

    print("Avant :",liste)
    tri_insertion(liste)
    print("Apres :",liste)

```