

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

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 d'une heure et vous ne pouvez pas utiliser de notes.
- Si du code vous est demandé,
 - la réponse à la question doit comprendre le code *Python* structuré et conforme aux règles de bonne pratique et conventions,
 - sauf mention contraire, vous ne pouvez utiliser aucune fonction de bibliothèques (pas d'`import`)
 - veillez à découper votre réponse en fonctions de manière pertinente
 - veillez à utiliser des structures de données appropriées.

Question 1 - Théorie (7 points)

Expliquez succinctement ce que fait et ce qu'imprime le code suivant en justifiant vos réponses.

a)

```
print [a for a in range(10) if a%2==0]
```

Proposition de solution : Imprime la liste constituée, formée par les nombres pairs de l'intervalle $[0..10[$, c'est-à-dire $[0, 2, 4, 6, 8]$

b) Sachant que `plist` contient les coefficients d'un polynôme (l'indice correspondant au degré), à quelle opération correspond la fonction suivante ?

```
def fun1(plist):  
    res=[]  
    for i in range(1,len(plist)):  
        res+=[i*plist[i]]  
    return res
```

Proposition de solution : renvoie la liste donnant les coefficients du polynôme de la dérivé première du polynôme donné en paramètre.

c) Sachant que `plist` contient les coefficients d'un polynôme, à quelle opération correspond la fonction suivante ?

```
def fun2(plist,c,i):  
    res= [0]*i  
    for i in plist:  
        res += [i * c]  
    return res
```

Proposition de solution : correspond à la multiplication du polynôme par le monôme $c.x^i$

Remarque : Le code donné à l'examen était

```
def fun2(plist,c,i):  
    res= [0]*i  
    for i in plist:  
        res += i * c  
    return res
```

Sans la correction à la ligne 4, il fallait alors répondre que le code était erroné puisque `res` référence une liste et ensuite on additionne cette liste à un entier `i*c`, ce qui n'a pas de sens en python.

- d) Sachant que `liste` est une liste d'éléments contenant `[2, 5, 7, 9, 7, 6, 4, 2, 2, 1, 0, 0]`, que fait le code suivant ?

```
dic = {}
for item in liste:
    junk=dic.setdefault(item, 0)
liste = dic.keys()
print liste
```

Proposition de solution : imprime `[2, 5, 7, 9, , 6, 4, 1, 0]` ou une liste contenant les mêmes valeurs mais dans un autre ordre. En effet, d'après la définition de `setdefault`, `junk` reçoit la valeur de l'élément de clé `item` de la liste ou 0 si cet élément est absent de la liste **ET** dans ce dernier cas, rajoute une nouvelle entrée dans le dictionnaire `dic` avec la clé `item` et la valeur 0. On constate donc que `junk` recevra toujours des valeurs 0 (mais n'est pas utilisé plus loin) et que `dic` va avoir des entrées dont les clés correspondent aux valeurs de la liste avec des valeurs 0 associées. A la fin, seules les clés nous intéressent. Donc à la fin la `liste` imprimée contient les valeurs initialement dans la liste mais en ayant supprimé les valeurs multiples. En gros, cela revient à prendre un multienemble (plusieurs fois a même valeur possible) et à en faire un ensemble (une fois chaque valeur possible).

Le but de la dernière sous question, est de vérifier votre compréhension de ce que fait un code, ayant à votre disposition l'aide mémoire qui vous explique le fonctionnement de la ou des fonctions ou méthodes utilisées (ex : `setdefault`).

Question 2 - Pratique (13 points)

On vous demande de trier les notes des étudiants pour un cours de façon décroissante.

- La note d'un étudiant est représentée par un tuple contenant son nom complet et une note sur 20. Par exemple ('César Jules', 19).
- Les notes d'un cours sont représentées par une liste dont le premier élément est le nom du cours et les éléments suivants sont les notes individuelles des étudiants inscrits à ce cours. Par exemple ['Algèbre', ('Doe John', 12), ('César Jules', 19), ('Jobs Steve', 17)]
- Les notes d'une année sont représentées par une liste de notes de cours. Cette liste est triée selon le nom du cours par ordre alphabétique.

On vous demande d'écrire une fonction `tri_notes` qui reçoit le nom d'un cours et les notes d'une année, et qui trie, en place, les notes du cours données de façon décroissante. Pour cela, vous devez écrire un tri vu au cours et vous ne pouvez pas utiliser la fonction `sort` de Python.

Voici un exemple complet d'utilisation de votre fonction :

```
>>> notes = [['Algèbre', ('Doe John',12), ('Cesar Jules',19), ('Jobs Steve',17)],
             ['Informatique', ('Jobs Steve',11), ('Doe John',8), ('Cesar Jules',15)],
             ['Physique', ('Jobs Steve',11), ('Doe John',16)]]
>>> tri_notes('Informatique',notes)
>>> print notes
[['Algèbre', ('Doe John',12), ('Cesar Jules',19), ('Jobs Steve',17)],
 ['Informatique', ('Cesar Jules',15), ('Jobs Steve',11), ('Doe John',8)],
 ['Physique', ('Jobs Steve',11), ('Doe John',16)]]
```

Proposition de solution :

```
def plus_grande_note(note1,note2):
    if note1[1] > note2[1]:
        return note1
    else:
        return note2

def pos_max(ls, i):
    """La position du max de ls en commençant en i.
    Pre: i dans ls
    """
    res = i
    while i < len(ls):
        if plus_grande_note(ls[i],ls[res]):
            res = i
        i += 1
    return res

def echange(ls, i1, i2):
    ls[i1], ls[i2] = ls[i2], ls[i1]

def tri_selection(ls):
    for i in range(1, len(ls) - 1): #car il ne faut pas trier le premier element
        pos = pos_max(ls, i)
        echange(ls, i, pos)

def tri_notes(cours, notes):
    for liste in notes:
        if liste[0] == cours:
            tri_selection(liste)
```

BON TRAVAIL !