

INFO-H-100 – Informatique – Prof. Th. Massart
1^{ère} année du grade de bachelier en sciences de l'ingénieur
Examen de septembre

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).

Question 1 - Python (1 points)

Python est un langage interprété avec typage dynamique fort. Expliquez.

Question 2 - Python (1 points)

Que fait le code suivant ? Expliquez.

```
d = {1: {1: 36, 4: 44, 6: 61}, 3: {2: 44, 4: -9, 7: 0}, 7: {1: 8, 7: 21}}
new= []
for i in range(max(d.keys())+1):
    if i in d:
        new.append(d[i].get(i,0))
    else:
        new.append(0)
print(new)
```

Question 3 - Diagrammes d'état (1 points)

Soit le code suivant :

```
known = {0 : 0, 1 : 1}

def fib_dict(n):
    if n in known:
        res = known[n]
    else:
        res = fib_dict(n-1) + fib_dict(n-2)
        known[n] = res
    return res

print(fib_dict(2))
```

Donnez les diagrammes d'état des variables et objets au moment de l'exécution de chaque `return` lors de l'exécution de la fonction.

Question 4 - Runtime et complexité (2 points)

La fonction traduction prend en paramètre deux dictionnaires `lang1` et `lang2` dont les clés et valeurs sont des chaînes de caractères (`string`). Elle renvoie un dictionnaire `res` dont les clés et valeurs sont

également des string tel que si x est la clé liée à la valeur y dans `lang1` et que y est la clé liée à la valeur z dans `lang2`, le dictionnaire résultant contiendra une clé x dont la valeur est z .

La fonction renvoie également deux listes de tuples `L1` et `L2`. La première contenant les couples (clé, valeurs) appartenant au premier dictionnaire (`lang1`) et auxquels il n'y a aucune correspondance dans le deuxième dictionnaire (`lang2`) et la deuxième contenant les tuples qui appartiennent seulement au deuxième dictionnaire.

```
def traduction(lang1, lang2):
    res={}
    l1=[]
    l2=[]
    for key in lang1:
        if lang1[key] in lang2:
            res[key] = lang2[lang1[key]]
        else:
            l1.append((key, lang1[key]))
    for key in lang2:
        if key not in lang1.values():
            l2.append((key, lang2[key]))
    return res, l1, l2
```

Donnez la complexité moyenne **et** maximale de cette fonction (en temps d'exécution) en justifiant bien votre réponse.

Question 5 - Récursivité (5 points)

Nous vous demandons d'écrire une fonction **réursive** calculant la profondeur d'une liste passée en paramètre. La liste pourra contenir des nombres et/ou des listes imbriquées, chacune de ces sous-listes pourra également contenir des nombres et/ou des listes imbriquées.

Par profondeur nous entendons le niveau de profondeur de la liste la plus imbriquée. La profondeur d'une liste ne contenant que de simples éléments est de 1.

Voici quelques exemples :

- `[1,2,3]` - profondeur 1 ;
- `[1,2,[4,6]]` - profondeur 2, il y a une liste imbriquée dans la liste ;
- `[[1,2,[5,[2,9],7],4,[2,9]],5,3,2,[2,0]]` - profondeur 4, profondeur de la liste la plus imbriquée.

Indice :

```
if (type(variable) is list):
    print("'variable' est une liste.")
```

Question 6 - Fichiers (4 points)

Écrire une fonction qui reçoit un nom de fichier et renvoie un tuple formé d'un string et d'un entier, respectivement la ligne contenant le plus grand nombre de mots dans le fichier et le nombre de mots correspondant. Un mot est considéré comme étant tout ensemble de caractères séparés des autres par des espaces.

Exemple :

*The doctor was twisting something at the back of Ender's head.
Suddenly a pain stabbed through him like a needle from his neck to his groin.
Ender felt his back spasm, and his body arched violently backward ; hi head struck the bed.
He could feel his legs thrashing, and his hands were clenching each other, wringing each other tightly.
"Deedee !" shouted the doctor.*

```
>>> print(get_longest_sentence("text.txt"))
('He could feel his legs thrashing, and his hands were clenching each other,
wringing each other tightly.', 17)
```

Question 7 - Tri (6 points)

On représente une personne par un dictionnaire dont les clefs sont "Lastname", "Firstname", et "Birthdate". Les trois clefs sont des chaînes de caractères. Les valeurs associées à "Lastname" et "Firstname" sont des chaînes de caractères. La valeur associée à "Birthdate" est un tuple de trois nombres qui correspondent respectivement au jour, au mois et à l'année de la naissance.

On vous demande d'écrire une fonction qui trie, de façon *croissante*, une liste de personnes en fonction de leur date de naissance. Pour cela, vous pouvez utiliser soit le tri par insertion, soit le tri par sélection. **Veillez à préciser sur votre copie le tri utilisé.**

Par exemple :

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