

Devoir surveillé n°3 – correction

**EXERCICE 1 :** (3pt) Entourer la bonne réponse à chaque question.

- 1) Si Décalage s'écrit "44 c3 a9 63 61 6c 61 67 65" en UTF-8 en hexadécimal, quelle peut être l'écriture de Glacé?  
a) 67 6c 61 63 a9 c3  
b) 47 6c 61 63 c3 a9  
c) 67 6c 61 63 c3 a9  
d) 47 6c 61 63 a9 c3
- 2) Lequel de ces nombres peut être représenté de façon exacte en virgule flottante :  
a) 17,1  
b) 8,55  
c) 31,75  
d) 64,95
- 3) Quelle commande permet de créer un dossier sous Linux?  
a) rmdir  
b) mkdir  
c) ls  
d) cat
- 4) On considère les instructions suivantes :

```
>>> score['Alice'] = 4  
>>> score['Bob'] = 7  
>>> score['Alice'] = 15
```

Quelle est la valeur de score après ces instructions?

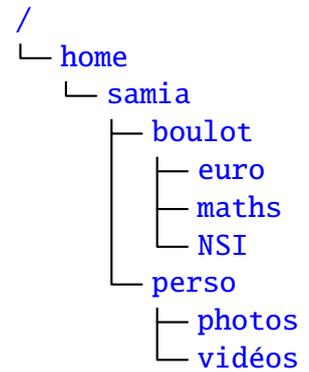
- a) {'Alice': 19, 'Bob': 7}
  - b) {'Alice': 15, 'Bob': 7}
  - c) {'Alice': 4, 'Bob': 7, 'Alice': 15}
  - d) [4, 7, 15]
- 5) On considère le dictionnaire dico = {'Alice': 5, 'Bob': 3, 'Charlie': 6}  
Quelle est la valeur de [k for k in dico if dico[k] > 5]?
- a) 'Charlie'
  - b) [6]
  - c) ['Alice', 'Charlie']
  - d) ['Charlie']
- 6) On considère le code suivant :

```
def venomous(animaux):  
    reponse = dict()  
    for animal in animaux:  
        if animal in reponse:  
            reponse[animal] = reponse[animal] + 1  
        else:  
            reponse[animal] = 1  
    return reponse  
  
enclos = ['poule', 'poule', 'coq', 'poussin', 'poule', 'poussin']
```

Quelle est la valeur de venomous(enclos)?

- a) {'poule': 3, 'coq': 1, 'poussin': 2}
- b) {'poule': 1, 'coq': 1, 'poussin': 1}
- c) {'poule': 1, 'poule': 1, 'coq': 1, 'poussin': 1, 'poule': 1, 'poussin': 1}
- d) {3: 'poule', 1: 'coq', 2: 'poussin'}

**EXERCICE 2 :** (3pt) On donne ci-contre l'architecture des dossiers de Samia sur l'ordinateur de la maison. Les fichiers ne sont pas affichés.



- 1) a) Donnez le chemin absolu du dossier NSI.  
`/home/samia/boulot/NSI`
  - b) Le dossier courant dans le terminal est perso. Donnez le chemin relatif permettant d'aller dans NSI.  
`../boulot/NSI`
  - c) Le dossier courant est maintenant NSI. Dans quel dossier va se trouver le fichier `memo.txt` après la commande :  
`touch ../../memo.txt` **dans le dossier samia**
- 2) Après vérification, Samia découvre que les droits sont `rw-rw-rw-` pour tous ses dossiers et fichiers dans son dossier personnel (`samia`).
    - a) Son frère qui a son propre compte et son propre groupe. Pourquoi est-ce qu'il peut aller voir et modifier tous les documents de Samia? **Les droits des autres utilisateurs sont `rw-`. Cela veut dire que tout le monde a le droit de lecture et d'écriture.**
    - b) Elle veut être la seule à pouvoir lire et écrire dans `perso`. Elle veut aussi avoir les droits d'exécution.  
 Quelle code octal doit elle utiliser dans cette commande? `chmod 700 perso`
    - c) Samia crée un dossier `partage` dans son dossier personnel. Voici une partie du contenu de son dossier personnel :

<code>drw-rw-rw-</code>	<code>2</code>	<code>samia</code>	<code>samia</code>	<code>4096</code>	<code>3</code>	<code>janv.</code>	<code>08:47</code>	<code>boulot</code>
<code>drw-rw-rw-</code>	<code>2</code>	<code>samia</code>	<code>samia</code>	<code>4096</code>	<code>8</code>	<code>févr.</code>	<code>21:32</code>	<code>partage</code>
<code>drw-----</code>	<code>2</code>	<code>samia</code>	<code>samia</code>	<code>4096</code>	<code>7</code>	<code>janv.</code>	<code>16:01</code>	<code>perso</code>

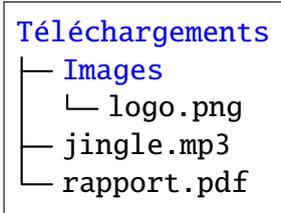
On rappelle que le `d` en début de ligne signifie qu'on voit un dossier et n'a pas de lien direct avec les droits.

Samia rajoute sa sœur au groupe `samia` et utilise la commande `chmod 640 partage`. Que pourra et ne pourra pas faire sa sœur dans ce dossier? Justifier.  
**Les droits du groupe `samia` seront `r--`. Elle pourra uniquement lire dans ce dossier.**
  - 3) a) Samia n'arrive pas à exécuter le script `setup.sh`. Après des recherches sur Internet, on lui a dit qu'elle devait se rajouter les droits d'exécution.  
 Quelle commande doit-elle taper pour se rajouter, à elle seule, les droits d'exécutions, sans changer les autres droits au fichier?  
`chmod u+x setup.sh`
  - b) Elle décide de s'entraîner sur les droits des fichiers. Après plusieurs essais, elle arrive à donner les droits `r-x-w-rw-` au fichier `essai.txt`. Quels seront les droits après l'exécution de la commande `chmod u+w-x,g+r,o-w essai.txt`?  
`rw-rw-r--`

**EXERCICE 3 :** (7pt) Cet exercice porte sur les structures de données (dictionnaires)

Afin d'organiser les dossiers et les fichiers sur un disque dur, une structure arborescente est utilisée. Les fichiers sont dans des dossiers qui sont eux-mêmes dans d'autres dossiers, etc. Dans une arborescence, chaque dossier peut contenir des fichiers et des dossiers, qui sont identifiés par leur nom. Le contenu d'un dossier est modélisé par la structure de données **dictionnaire**. Les clés de ce dictionnaire sont des chaînes de caractères donnant le nom des fichiers et des dossiers contenus.

Le dossier appelé Téléchargements contient deux fichiers rapport.pdf et jingle.mp3, et un dossier Images contenant simplement le fichier logo.png. Il est représenté ci-contre. Ce dossier Téléchargements est modélisé en Python par le dictionnaire suivant :



```
{"Images": {"logo.png": 36}, "rapport.pdf": 450, "jingle.mp3": 4800}
```

Les valeurs numériques sont exprimées en ko (kilo-octets). "logo.png": 36 signifie que le fichier logo.png occupe un espace mémoire de 36 ko sur le disque dur.

On rappelle, ci-dessous, quelques commandes sur l'utilisation d'un dictionnaire :

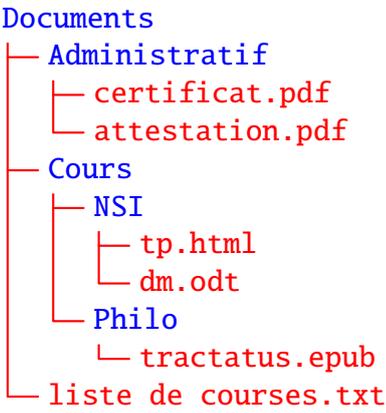
- dico = {} ou dico = dict() crée un dictionnaire vide appelé dico,
- dico[cle] = contenu met la valeur contenu pour la clé cle dans le dictionnaire dico,
- dico[cle] renvoie la valeur associée à la clé cle dans le dictionnaire dico,
- cle in dico renvoie un booléen indiquant si la clé cle est présente dans le dictionnaire dico,
- for cle in dico: parcourt l'ensemble des clés du dictionnaire.

L'adresse d'un fichier ou d'un dossier correspond au nom de tous les dossiers à parcourir depuis la racine afin d'accéder au fichier ou au dossier. Cette adresse est modélisée en Python par la liste des noms de dossier à parcourir pour y accéder.

Exemple: L'adresse du dossier: /home/pierre/Documents/ est modélisée par la liste ["home", "pierre", "Documents"].

1) Dessiner l'arbre donné par le dictionnaire suivant, qui correspond au dossier Documents donné ci-dessous.

```
Documents = {
  "Administratif":{
    "certificat.pdf ": 1500,
    "attestation.pdf ": 850
  },
  "Cours": {
    "NSI": {
      "tp.html": 60,
      "dm.odt": 345
    },
    "Philo": {"tractatus.epub": 2600}
  },
  "liste de courses.txt ": 24
}
```



2) a) On reprend la variable Documents de la question précédente. On considère les instructions suivantes :

```
>>> dossier = Documents
>>> dossier = dossier["Cours"]
>>> dossier = dossier["NSI"]
```

Compléter le résultat de l'évaluation de la variable dossier après l'exécution des instructions ci-dessus :

```
>>> dossier
{'tp.html': 60, 'dm.odt': 345}
```

- b) On donne la fonction `parcourir` suivante qui prend en paramètres un dossier racine et une liste représentant une adresse, et qui renvoie le contenu du dossier cible correspondant à l'adresse.

Exemple : Si la variable `Documents` contient le dictionnaire de l'exemple de la question 1 alors `parcourir(Documents, ["Cours", "Philo"])` renvoie le dictionnaire `{"tractatus.epub": 2600}`.

Compléter le code de la fonction :

```
def parcourir(racine, adr):
    dossier = racine
    for nom_dossier in adr:
        dossier = dossier[nom_dossier]
    return dossier
```

- c) Soit la fonction suivante :

```
def afficher(racine, adr, nom_fichier):
    dossier = parcourir(racine, adr)
    print(dossier[nom_fichier])
```

Qu'affiche l'instruction `afficher(Documents, ["Cours", "NSI"], "tp.html")` sachant que la variable `Documents` contient le dictionnaire de la question 1 ?

60

- 3) a) La fonction `ajouter(racine, adr, nom_fichier, taille)` suivante ajoute au dictionnaire `racine`, à l'adresse `adr`, la clé `nom_fichier` associée à la valeur `taille`. Une ligne de la fonction donnée ci-dessous contient une erreur. Laquelle? Proposer une correction.

```
def ajouter_fichier(racine, adr, nom_fichier, taille):
    dossier = parcourir(racine, adr)
    taille = dossier[nom_fichier]
```

Il faut changer la dernière ligne : `dossier[nom_fichier] = taille`

- b) Écrire une fonction `ajouter_dossier(racine, adr, nom_dossier)` pour créer un dictionnaire représentant un dossier vide appelé `nom_dossier` dans le dictionnaire `racine` à l'adresse `adr`. On pourra utiliser `parcourir`.

```
def ajouter_dossier(racine, adr, nom_dossier):
    dossier = parcourir(racine, adr)
    dossier[nom_dossier] = dict()
```

- 4) Écrire une fonction `calcule_taille` qui prend en paramètre un dictionnaire `dossier` modélisant le contenu du répertoire `dossier` et qui renvoie le total de l'espace mémoire occupé par les fichiers contenus dans le dossier. On considère que le répertoire `dossier` ne contient que des fichiers et **aucun sous-dossier**.

```
def calcule_taille(dossier):
    taille = 0
    for fichier in dossier:
        taille = taille + dossier[fichier]
    return taille
```