

1. Question système (1,5 points)

Le plus simple est de créer un répertoire ayant tous les droits pour les "autres" dont vous ne communiquerez le nom qu'à vos collègues. Vous enlèverez le droit de lecture aux "autres" au répertoire père de celui-ci (pour que le nom de ce répertoire ne puisse être connu), mais pas le droit d'accès :

```
mkdir ~/RepertoireMystere
chmod 777 ~/RepertoireMystere
chmod 701 ~
```

Vous ne pouvez pas créer de groupes car vous n'êtes pas administrateur, et la création de clefs de connexions, de dépôt GIT..., est d'une part une procédure plus lourde, et d'autre part va conférer des droits bien trop importants.

2. Expliquez le fonctionnement de cette commande (1,5 points)

```
ls -a ~ | grep "^\."
```

La première commande liste **aussi** les noms des fichiers/dossiers cachés du **répertoire d'accueil** que la seconde commande ne garde que s'ils commencent par un point (a priori des noms de fichiers/dossiers "système").

3. Quel est le cas de figure où le script bash suivant va provoquer une erreur ? (1 point)

Dans le cas où le nom du fichier/dossier testé contient une espace.

Par exemple si un dossier s'appelle "mon dossier", la ligne `if [-d $i]` va devenir `if [-d mon dossier]` et l'opérateur `-d` ne va fonctionner que sur "mon". La variable `$i` doit donc être protégée : `if [-d "$i"]`

4. Script système Python à écrire (6 points)

```
#!/usr/bin/env python3
import sys, os, re
versions = {}
if len(sys.argv) > 2 :
    liste = os.popen("ls -l "+sys.argv[1]) # Autre possibilité : liste = os.listdir(sys.argv[1])
    for executable in liste :
        resultat = re.search(".* ([^\d+])([.\d+]*)$", executable)
        # si utilisation de os.listdir la regexp est plus simple : "([^\d+])([.\d+]*)$"
        # voire même : (.+?)([.\d+]*)$
        # Il y a d'autres possibilités notamment pour la partie qui extrait la version : (\d+\.?.*\d*)
        # et il reste un problème pour les noms de fichiers/dossiers qui se terminent par .
        if resultat :
            nomExe = resultat.group(1)
            numVersion = resultat.group(2)
            if nomExe in versions :
                if numVersion not in versions[nomExe] :
                    versions[nomExe].append(numVersion)
            else :
                versions[nomExe] = [numVersion]
    if sys.argv[2] in versions :
        print(versions[sys.argv[2]])
else :
    print("Usage : ./versions.py repertoire nomDeLExecutable")
```

Et bien sûr si vous ne deviez pas construire de dictionnaire, cela serait encore plus simple :

```
versions = [] # on a juste besoin d'une liste
for executable in liste :
    resultat = re.search(sys.argv[2]+"([.\d+]*)$", executable)
    if resultat :
        if resultat.group(1) not in versions :
            versions.append(resultat.group(1))
print(versions)
```