

Corrections questions Pierre pompidor (recherches, Bash, Python)

1 point : Recherche à partir du dossier d'accueil les fichiers dont les noms finissent par ".py" :

```
find ~ -name "*.py"
```

D'autres solutions plus compliquées avec un ls récursif et un grep sont possibles...

1 point : Expression régulière qui vérifie qu'une chaîne de caractères commence et se termine par un chiffre :

```
"^[0-9].*[0-9]$" ou "^[0-9].+[0-9]$" ou "^\\d.*\\d$" ou "^\\d.+\\d$"
```

0.5 point : En bash, quels sont les deux mots de début et de fin d'un bloc d'instructions d'une structure itérative :

```
do et done
```

1 point : Ecrire les deux instructions Bash qui créent une variable nommée *i* initialisée avec 1, puis affiche la valeur de cette variable :

```
i=1
echo $i
```

2 points : Script Bash qui sans utiliser la commande *ls* affiche tous les fichiers du répertoire courant dont les noms commencent par le premier paramètre donné au script et se terminent par son second paramètre :

```
for fichier in $1*$2
do
    echo $fichier
done
```

il est possible que des étudiants se lancent dans

```
for fichier in *
do
    echo $fichier | grep "$1.*$2"
done
```

1 point pour la boucle sur le contenu du dossier et 1 point pour l'utilisation des paramètres

1,5 points : Code Python qui affiche tous les noms des fichiers (répertoires ou fichiers réguliers) du répertoire dans lequel s'exécute le script.

```
import os
for fichier in os.listdir() :
    print(fichier)
```

ou

```
import os
for fichier in os.popen("ls") :
    print(fichier)
```

0,5 point pour la boucle for ... in ... (ou une autre boucle valide)

0,5 point pour l'utilisation de listdir() ou de popen("ls")

0,5 point pour l'importation du module os

0,5 point : Même chose en numérotant à partir de 1 ces fichiers.

```
import os
num = 1
for fichier in os.listdir() :
    print(num, fichier)
    num += 1
```

2.5 points : Fonction *intersection()* qui reçoit en paramètre deux listes et qui renvoie une liste contenant une seule fois les éléments en commun :

```
def intersection(liste1, liste2) :
    encommun = []
    for v in liste1 :
        if v in liste2 and v not in encommun :
            encommun.append(v)
    return encommun
```

0.5 point pour la fonction et son retour

0.5 point pour la création de la liste vide

0.5 point pour le (les ?) parcours de liste

0.5 point pour le test d'appartenance

0.5 point pour le not in