

1. Expliquez le fonctionnement de cette commande (1 point)

```
last > ~/connexions
```

2. Que fait ce script ? (1,5 points)

```
for i in $*
do
    for j in *
    do
        if [[ $j == *$i* ]]
        then
            echo $j
        fi
    done
done
```

3. Quel est le résultat et le fonctionnement de cette application d'expression régulière ? (1.5 points)

```
chaine = "2 3 5 7 et aussi 11 13 17 19"
resultat = re.search(".*(\d+)", chaine)
print(resultat.group(0))
print(resultat.group(1))
```

4. Script système Python à écrire (6 points)

Vous administrez un serveur web de ventes en ligne et celui-ci produit des statistiques mensuelles sur les ventes que vous réalisez.

Vous disposez d'un fichier nommé *promotions.log* qui liste (en fin de chaque mois) pour chaque produit mis en promotion (prix baissé) l'augmentation de ses ventes. Chaque ligne du fichier est composé du nom du produit et de la liste des promotions qui l'ont concerné (chaque promotion est suivie du pourcentage d'augmentation de ses ventes).

En voici un exemple d'extrait :

```
produit1:10=15,20=26
produit2:20=32,30=41
produit3:20=14,25=22,40=51
...
```

Par exemple, le prix du *produit1* a été abaissé de 10% et ses ventes ont alors augmenté de 15%, et en promotion de 20%, ses ventes ont alors augmenté de 26%.

Ecrivez un script Python *promotions.py* qui affiche pour la promotion passée en paramètre, la liste des augmentations des ventes. **Ce script doit mettre en œuvre un dictionnaire dont les clefs seront les promotions.** Par exemple, si le script est appelé ainsi :

```
python3 promotions.py 20
```

le script affiche :

```
Voici les augmentations de ventes des produits dont les prix ont été abaissés de 20% :
26% (produit1)
32% (produit2)
14% (produit3)
moyenne : 24%
```