





## 1. Exercice 1



## Enoncé

## Solution 1

## Solution 2 : avec la méthode `items`

Sur le réseau social TipTop, on s'intéresse au nombre de « like » des abonnés. Les données sont stockées dans des dictionnaires où les clés sont les pseudos et les valeurs correspondantes sont les nombres de « like » comme ci-dessous :

### Script Python

```
{'Bob': 102, 'Ada':  
201, 'Alice': 103,  
'Tim': 50}
```

Écrire une fonction

`top_like` qui :

- Prend en paramètre un dictionnaire likes non vide dont les clés sont des chaînes de caractères et les valeurs associées sont des entiers ;
- Renvoie un tuple dont :
  - La première valeur est la clé du dictionnaire associée à la valeur maximale ; en cas d'égalité sur plusieurs

clés, on  
choisira la plus  
petite suivant  
l'ordre  
alphabétique

- La seconde  
valeur est la  
valeur  
maximale  
présente dans  
le dictionnaire.

Exemples :

#### Script Python

```
>>> top_like({'Bob':  
102, 'Ada': 201,  
'Alice': 103, 'Tim':  
50})  
(('Ada', 201)  
>>> top_like({'Alan':  
222, 'Ada': 201, 'Eve':  
222, 'Tim': 50})  
(('Alan', 222)
```

#### Script Python

```
def top_like(likes):  
    top_pseudo = None  
    top_nb_likes = 0  
    for pseudo in likes:  
        nb_likes =  
likes[pseudo]  
        if nb_likes >  
top_nb_likes or  
nb_likes ==  
top_nb_likes and  
pseudo < top_pseudo:  
            top_pseudo =  
pseudo  
            top_nb_likes =  
nb_likes  
    return top_pseudo,  
top_nb_likes
```

#### Script Python

```
def top_like(likes):
    top_pseudo =
None
    top_nb_likes = 0
    for pseudo,
nb_likes in
likes.items():
        if nb_likes >
top_nb_likes or
nb_likes ==
top_nb_likes and
pseudo < top_pseudo:
            top_pseudo
= pseudo
            top_nb_likes
= nb_likes
    return
top_pseudo,
top_nb_likes
```

## 2. Exercice 2



Enoncé	Solution
--------	----------

Recopier et compléter sous Python la fonction suivante en respectant la spécification. On ne recopiera pas les commentaires.

```
1  def
2  dichotomie(
3  x):
4      """
5      tab :
6      tableau
7      d'entiers
8      trié dans
9      l'ordre
10     croissant
11     x :
12     nombre
13     entier
14     La
15     fonction
16     renvoie
17     True si
    tab
    contient
    x et False
    sinon
    """
    debut
    = ...
    fin
    = ...
    while
    debut <=
    fin:
        milieu
        = ...
        if x
        ==
        tab[milieu]:
```

```
return ...
    if x > ...:
        debut
= ...
    else:
        fin = ...
return ...
```

Exemples :

### Script Python

```
>>>
dichotomie([15,
16, 18, 19, 23,
24, 28, 29, 31,
33],28)
True
>>>
dichotomie([15,
16, 18, 19, 23,
24, 28, 29, 31,
33],27)
False
```

```
1 def
2 dichotomie(
3 x):
4     """
5     tab :
6     tableau
7     d'entiers
8     trié dans
9     l'ordre
10    croissant
11    x :
12    nombre
13    entier
14    La
15    fonction
16    renvoie
17    True si
    tab
    contient
    x et False
    sinon
    """
```

```
    debut
    = 0
```

```
    fin =
    len(tab) -
    1
```

```
    while
    debut <=
    fin:
```

```
        milieu =
        (debut +
        fin) // 2
```

```
        if x
        ==
        tab[milieu]:
```

```
            return
            True
```

```
            if x
            >
            tab[milieu]:
```

```
                debut =
                milieu +
                1
```

```
            else:
```

```
                fin =
                milieu - 1
```

```
                return
                False
```