# 1 Listes et range

Quelles valeurs contiennent les listes suivantes ?

```
[1, 1+1, 4-1]
len(['a', 'b'])
[1>0, True, 'a' in 'abc']
[]
```

# 2 Ajout dans une séquence

On souhaite ajouter à la fin d'un itérable (de type chaîne ou liste d'entiers) 2 fois un même élément (un caractère ou un entier). Par exemple :

```
$$$ ajout_doublon('coucou', '!')
'coucou!!'
```

Commenter la correction des fonctions suivantes :

```
def ajout_doublon1(s:str, x:str) -> str:
    return s + 2*x
```

```
def ajout_doublon2(l:list[int], x:int):
   res = []
   for elt in l:
       res.append(elt)
   res.append([x] * 2)
   return res
```

```
def ajout_doublon3(l:list[int], x:int):
    return l + 2 * [x]
```

```
def ajout_doublon4(s:str, x:str):
    s.append(2 * x)
```

```
def ajout_doublon5(l:list[int], x:int):
  res = []
  for elt in l:
      res.append(elt)
  res.append(x)
  return res.append(x)
```

# 3 Code mystère

On donne la fonction suivante :

```
def mystere(l:list[int]) -> int:
    """
    precondition: non vide, contient des elts positifs ou nuls
    """
    res = 0
    for elt in l:
        if elt > res:
            res = elt
    return res
```

1. Dérouler avec un tableau de la mémoire l'exécution de `mystere([4, 3, 4, 6])`.
2. Compléter la documentation de cette fonction.

# 4 Écriture de tests et de code

Pour chacune des fonctions décrites ci-dessous, écrire la documentation complète, en particulier les tests, ainsi que le code.

3. Une fonction `nombre_occurrences` qui prend en paramètre un caractère et une chaîne `chaine` et qui renvoie le nombre d'occurrences de `caracteres` dans `chaine` :

   ```
   $$$ nombre_occurrences('a', 'abracadabra')
   5
   ```

4. Une fonction `nombre_occurrences2` qui prend en paramètre une chaine `caracteres` non vide et une chaîne `chaine` et qui renvoie le nombre d'occurrences cumulé des caractères de `caracteres` dans `chaine` :

   ```
   $$$ nombre_occurrences2('afbcg', 'abracadabra')
   8
   ```

5. Une fonction `produit` qui prend en paramètre une liste d'entiers et qui renvoie le produit des éléments qu'elle contient.

   ```
   $$$ produit([2, 3, -5])
   -30
   ```

6. Une fonction `pairs_et_positifs` qui prend en paramètre une liste d'entiers `l` et renvoie une nouvelle liste contenant uniquement les entiers strictement positifs et pairs de `l`, dans le même ordre. Que faudrait-il changer pour obtenir à la fois les entiers pairs et les entiers strictement positifs ?

   ```
   $$$ pairs_positifs([2, 3, -6, 4, 5, 12])
   [2, 4, 12]
   ```

7. Écrire une fonction `incremente_chiffres` qui prend en paramètre une chaîne de caractères `chaine` et renvoie une chaîne contenant dans le même ordre les caractères de `chaine`, mais tels que les chiffres ont été remplacés par leur valeur incrémentée. Par exemple :

   ```
   $$$ incremente_chiffres('3%$6')
   '4%$7'
   ```

   On utilisera `str.isdigit()`.

   ```
   >>> 'a'.isdigit()
   False
   >>> '42'.isdigit()
   True
   ```