

1 Combien de fois la fonction **print** est appelée dans les scripts suivants ?

1.

```
1 for i in range(10):
2     for j in range(3):
3         print(i ** 2 + j)
```

2.

```
1 for i in range(3):
2     for j in range(4):
3         for k in range(5):
4             print(i + j + k)
```

3.

```
1 for i in range(7):
2     j = 0
3     while j < i:
4         print(i + j)
5         j = j + 1
```

4. On donnera le résultat en fonction de n .

```
1 for i in range(n):
2     for j in range(1, n):
3         print(j)
```

5. On donnera le résultat en fonction de n .

```
1 for i in range(n):
2     for j in range(i):
3         print(j)
```

6. On donnera le résultat en fonction de n .

```
1 for i in range(n):
2     for j in range(i, n):
3         print(j)
```

2 Après le code Python qui suit, quelles sont les valeurs finales de x et y ?

```
1 x = 4
2 while x > 0:
3     y = 0
4     while y < x:
5         y = y + 1
6         x = x - 1
```

3 Écrire en compréhension la liste des produits $a * b$ pour a et b entre 1 et 100 avec a pair et b impair.

4 À quoi est égale la liste suivante ?

```
1 >>>L = [i * j for i in range(10) for j in range(i) if j > i]
```

5 ♡ Étude et amélioration du tri bulle.

1. On suppose que la liste L que l'on doit trier à l'aide du tri bulle est déjà ordonnée dans l'ordre croissant. Combien de parcours fait-on ? Combien de comparaisons seront effectuées au total ? Quelle est la complexité dans ce cas favorable ?
2. On suppose que la liste L est déjà triée dans l'ordre décroissant. Combien de parcours fait-on ? Combien de comparaisons seront effectuées au total ? Quelle est la complexité dans ce cas défavorable ?
3. À la fin du premier parcours de la liste, montrer que le plus grand élément est forcément placé à la fin de la liste. De même, à la fin du parcours numéro j , les j derniers éléments de la liste sont bien placés, il est alors inutile de les comparer à nouveau. Modifier l'algorithme présenté dans le cours pour tenir compte de cela.
4. Que faut-il modifier dans l'algorithme du tri bulle initial pour trier une liste dans décroissant ?

6 ♡ Recherche des deux points les plus proches. On se donne une liste de nombres L et l'on souhaite chercher les deux éléments de la liste, x et y , les plus proches, c'est-à-dire tels que $|x - y|$ soit minimale. En python, la valeur absolue s'obtient avec la commande `abs`. Écrire une fonction qui prend en paramètre une liste et renvoie les deux indices correspondants aux éléments qui réalisent cette distance minimale. On utilisera deux boucles `for` imbriquées. Quelle est la complexité de l'algorithme sachant que l'opération comptée est la soustraction ?

7 On considère les deux fonctions suivantes :

```
1 def mystere(n):
2     s = 0
3     for k in range(1, n + 1):
4         s = s + factorielle(k)
5     return(s)
6
7 def factorielle(n):
8     p = 1
9     for k in range(2, n + 1):
10         p = p * k
11     return(p)
```

1. Quelle valeur renvoie `mystere(4)` ?
2. Déterminer le nombre de multiplications qu'effectue `mystere(n)`.
3. Proposer une amélioration de la fonction `mystere` afin d'obtenir une complexité linéaire.