

## QCM 1

### Question 1

On définit un dictionnaire : `d = { 'couleur': 'vert', 'taille': 42, 'marque': 'le coq sportif' }`  
Quelle est la valeur de l'expression `d.keys()` ?

#### Réponses

- A `['couleur', 'taille', 'marque']`
- B `[('couleur', 'vert'), ('taille', 42), ('marque', 'le coq sportif')]`
- C `['vert', 42, 'le coq sportif']`
- D `['couleur': 'vert', 'taille': 42, 'marque': 'le coq sportif']`

### Question 2

Quelle est la complexité du tri par sélection ?

#### Réponses

- A inconnue
- B linéaire
- C quadratique
- D exponentielle

### Question 3

On exécute le script suivant :

```
inventaire = {'pommes': 430, 'bananes': 312,
             'oranges' : 274, 'poires' : 137}
```

```
stock = 0
for fruit in inventaire.keys():
    if fruit != 'bananes':
        stock = stock + inventaire[fruit]
```

Que contient la variable `stock` à la fin de cette exécution ?

#### Réponses

- A `{430, 274, 137}`
- B `312`
- C `841`
- D `{ 'pommes', 'oranges', 'poires' }`

### Question 4

On a défini

```
repertoire = [ {'nom': 'Francette', 'poste': 412},
               {'nom': 'Jeanne', 'poste': 222},
               {'nom': 'Éric', 'poste': 231} ]
```

Quelle expression permet d'accéder au poste d'Éric ?

#### Réponses

- A `repertoire[2]['poste']`
- B `repertoire['poste'][2]`
- C `repertoire['Éric']['poste']`
- D `repertoire['Éric']`

### Question 5

Quel est l'ordre de grandeur du coût du tri par insertion (dans le pire des cas) ?

#### Réponses

- A l'ordre de grandeur du coût dépend de l'ordinateur utilisé
- B linéaire en la taille du tableau à trier
- C quadratique en la taille du tableau à trier
- D indépendant de la taille du tableau à trier

**Question 6**

La variable `sequence` contient une liste de lettres, éventuellement répétées, choisies parmi 'A', 'B', 'C', 'D'. On veut créer un dictionnaire `effectifs` associant à chaque lettre le nombre de fois qu'elle apparaît dans la liste `sequence`.

Par exemple si `sequence` contient ['A', 'B', 'B', 'D', 'B', 'A'],

`effectifs` doit contenir {'A':2, 'B':3, 'C':0, 'D':1}.

Parmi les scripts suivants, lequel réalise cet objectif ?

**Réponses****A**

```
effectifs = {'A':0, 'B':0, 'C':0, 'D':0}
for lettre in sequence:
    effectifs[lettre] = effectifs[lettre] + 1
```

**B**

```
effectifs = {}
for lettre in sequence:
    effectifs[lettre] = effectifs[lettre] + 1
```

**C**

```
effectifs = {'A':0, 'B':0, 'C':0, 'D':0}
for lettre in effectifs.keys():
    effectifs[lettre] = len([lettre in effectifs])
```

**D**

```
effectifs = {}
for lettre in effectifs.keys():
    effectifs[lettre] = len([lettre in effectifs])
```