

# Travail dirigé 01

**Cours:** Prog. Web Serveur 2 (420-109-GG)

**Sujets:** Organisation de données en Java

---

1. Écrire en langage Java un programme permettant de trouver le nombre des pics dans un tableau de valeurs entières.

*Exemple :*

```
numbers = [1, 2, 3, 2, 1, 3, 5, 2, 3, 1]
numPeaks = 3
```

2. Soit un tableau des chaînes de caractères. Créer un nouveau tableau contenant des valeurs de type (String, Int) permettant de calculer le nombre d'occurrences de chaque chaîne dans le premier tableau.

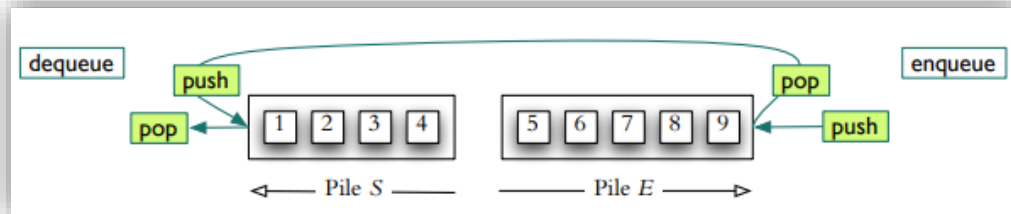
*Exemple :*

```
strings = ["tuples", "are", "awesome", "tuples", "are", "cool",
           "tuples", "tuples", "tuples", "shades"]
countedStrings = [["tuples", 5], ["are", 2], ["awesome", 1], ["cool", 1],
                  ["shades", 1]]
```

3. Écrire en langage Java une fonction prenant en entrée une liste et deux indices i et j pour inverser l'ordre de tous les éléments entre ces deux indices dans cette liste.
4. Écrire en langage Java un programme permettant de placer un nouvel élément inséré au fond d'une pile.
5. En informatique, une file (en anglais Queue), est une structure de données basée sur le principe du premier entré, premier sorti.
  1. En utilisant une liste chaînée LinkedList, créer une classe Queue d'objets de type Person (firstName, lastName, score).
  2. Implémenter les trois méthodes put, get et isEmpty
  3. Trouver la personne avec le meilleur score et afficher son nom complet.

6. Même questions que le numéro 5 en utilisant deux piles Stack au lieu d'une liste LinkedList pour créer la file d'objets de type Person.

Indice : implanter une queue (file FIFO) en utilisant deux piles E et S («entrée» et «sortie»).



7. Créer un classe Java Employee (champs : fullName, age, hireDate)
- Déclarer une liste d'employés (List<Employee>)
  - Définir quelques enregistrements dans cette liste
  - Écrire une fonction permettant de retourner les employés dont l'âge est supérieur à une valeur donnée
  - Trier les employés par âge (du plus jeune au plus vieux).
8. **(Optionnel)** On souhaite implémenter un annuaire électronique en Java, qui donne pour chaque nom de contact, ses contacts (phoneNumber et address).
- Définir la classe Contact, composée des 2 attributs phoneNumber et address
  - Implémentant un constructeur par défaut ainsi que des accesseurs pour ses attributs.
  - Définir la classe Directory qui gère l'annuaire électronique à l'aide d'une collection de type HashMap. Cette collection associe à chaque nom, un objet de type Contact. La classe Annuaire implémente les méthodes suivantes :
- add(String N, Contact C) : Ajout d'un nouveau contact

- `displayContacts(String N)`: Affichage des coordonnées d'un contact donné.
  - `updateContacts(String N)`: Modification des coordonnées d'un contact donné.
  - `delete(String N)`: Suppression d'un contact donné.
  - `getNames()`: Affichage de la liste des contacts (tous les noms).
  - `getPhones()`: Affichage de la liste des numéros de téléphone disponibles dans l'annuaire.
  - `getAddresses()`: Affichage de la liste des adresses disponibles dans l'annuaire.
  - `getDirectory()`: récupérer la totalité de l'annuaire à partir d'un fichier.
  - `save()` : sauvegarder l'annuaire dans un fichier
- d. Écrire un programme principal, présentant un menu pour la gestion d'un annuaire électronique.