

Nano dico :

Consigne :

Trier des mots

Exercice 1 :

Décrire l'algorithme qui permettra de trier une liste de mots simple, sans doublons.

L'algorithme sera exécuté en 2 grandes étapes, la première va trier notre liste de mot à l'inverse de l'ordre alphabétique, ensuite la suppression des doublons comme vu en cours stocker dans un nouveau tableau qui va inverser l'ordre des mots et qui seront donc dans l'ordre alphabétique.

Exercice 2 :

Ecrire le pseudo code correspondant à cet algorithme.

Variables :

TABLEAU Words

TABLEAU newList

DEBUT

Inverser l'ordre du tableau Words

Pour i <- 0 à Words[taille]

ENTIER indexAtWord = -1

ENTIER word <- words[i]

Pour j <- 0 à newList[Taille]

SI newList[j] égal à null ou newList[j] égal à word

IndexAtWord <- j

FIN SI

FIN POUR

SI indexAtWord >= 0

newList[indexAtWord] = word

FIN SI

FIN POUR

Afficher (newList)

FIN

Exercice 3 :
Implémenter en Java.

```
import java.util.Arrays;
import java.util.Collections;

class dico {
    public static void main(final String[] args) {

        // Déclaration
        String[] words = {"Basketball", "Football", "Rugby", "Tennis", "Football"}
;
        String[] newList = new String[words.length];

        // Inverser l'ordre du tableau
        Arrays.sort(words, Collections.reverseOrder());

        // Main code
        for (int i = 0; i < words.length; i++) {
            int indexAtWord = -1;
            String word = words[i];

            for (int j = 0; j < newList.length; j++) {
                if (newList[j] == null || newList[j].equals(word)) {
                    indexAtWord = j;
                }
            }

            if (indexAtWord >= 0) {
                newList[indexAtWord] = word;
            }
        }

        // Final Code
        System.out.println(Arrays.toString(newList));
    }
}
```