

## TP Algorithmique programmation

### TP 11 : Révisions

**Exercice 1.** Écrire un programme **Conjugaison** prenant en paramètre un verbe du premier groupe à l'infinitif et affichant la conjugaison de ce verbe au présent de l'indicatif. Exemples d'utilisation du programme :

```
$ java Conjugaison chanter
je chante
tu chantes
il chante
nous chantons
vous chantez
elles chantent
```

```
$ java Conjugaison
Usage : Conjugaison <verbe en er>
```

```
$ java Conjugaison finir
Erreur : je ne sais pas conjuguer finir
```

On demande d'utiliser un tableau contenant les pronoms ("je", "tu", ..., "elles") et un tableau contenant les suffixes ("e", "es", ..., "ent").

**Exercice 2.** On se propose de réaliser en Java un correcteur orthographique utilisant un dictionnaire de mots contenu dans un fichier texte (un mot par ligne). Le programme sera constitué d'une seule classe **Correcteur** comprenant une variable dictionnaire. Pour charger le dictionnaire en mémoire, on utilisera la classe suivante :

```
import java.io.*;

public class LineReader {
    public static LineReader in = new LineReader();
    private BufferedReader buffer;

    public LineReader(String file) {
        try {

            buffer = new BufferedReader(
```

```

        new InputStreamReader(new FileInputStream(file)));
    } catch (FileNotFoundException e) {
        buffer = null;
    }
}
private LineReader() {
    buffer = new BufferedReader(new InputStreamReader(System.in));
}
public String readLine() {
    try {
        return buffer.readLine();
    } catch (IOException e) {
        return null;
    }
}
}
}

```

Usage : `LineReader.in.readLine()` retourne une ligne lue sur l'entrée standard (null si l'on entre par Ctrl-D le caractère de fin de texte); pour lire dans un fichier toto, on crée d'abord un objet de la classe `LineReader` :

```

LineReader lr = new LineReader("toto");
String lignelue = lr.readLine();

```

On suppose le nombre de mots du dictionnaire borné par un entier `TAILLEMAX`, de façon à ce qu'on puisse définir la variable dictionnaire comme un tableau de taille fixe. Donner une version de la classe `Correcteur` comportant :

- un constructeur `Correcteur()` créant le tableau dictionnaire;
- un constructeur `Correcteur(String nomFichierDictionnaire)` chargeant le dictionnaire en mémoire;
- une méthode `int position(String m)` retournant l'index du mot `m` dans le tableau dictionnaire (-1 s'il ne s'y trouve pas); suivant que le tableau dictionnaire est ou non trié, la méthode utilisera une recherche dichotomique ou séquentielle.

Pour tester l'implémentation du dictionnaire, on pourra utiliser la méthode suivante :

```

public static void main(String[] args) {
    Correcteur c = new Correcteur(args[0]);
    String mot;
    while ((mot = LineReader.in.readLine()) != null) {
        boolean motTrouve = c.position(mot) >= 0;
        System.out.println( "Mot " + mot
            + (motTrouve?" trouvé":" non trouvé"));
    }
}

```

Tester la classe Correcteur pour différentes tailles de dictionnaires, triés ou non ; pour pouvoir comparer les temps de chargement et de recherche, on fera afficher le temps écoulé (utiliser `System.currentTimeMillis`).