Exercice 9. Segmenter une chaîne de caractères en une suite de mots. Lorsque l'on fait une recherche sur le web avec un *moteur de recherche* tel que celui de `Google` ou de `Qwant`, il arrive fréquemment que l'on oublie un ou des blancs entre les mots. Exemple : on pourrait entrer "lacigale" au lieu de "la cigale" ou, en forçant le trait, on pourrait entrer "lacigaleayantchantétoutl'été".

Vous constatez que le moteur de recherche rectifie de lui-même. Google nous dit *"Résultats pour la cigale ayant chanté tout l'été"*. Quant à Qwant, il ne vous dit rien du tout et parcourt le web à la recherche de "la cigale ayant chanté tout l'été"[1].

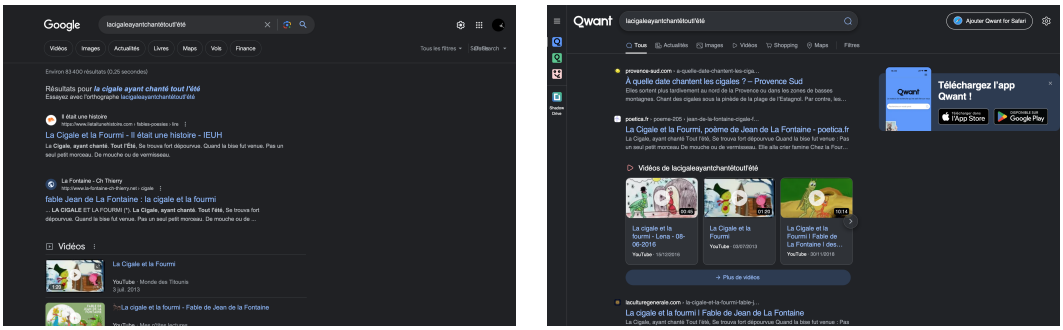

FIGURE 1 – Recherche de "lacigaleayantchantétoutl'été" dans les moteurs `Google` et `Qwant`.

Nous nous intéressons à la décomposition d'une chaîne de caractères en une suite de mots qui composent cette chaîne de caractères.

Exemple : "lacigaleayantchantétoutl'été" décomposée en "la cigale ayant chanté tout l été".

Une chaîne C de n caractères est donnée ainsi qu'un dictionnaire de mots. Nous voulons répondre à la question : " la chaîne C peut-elle être segmentée (décomposée) en mots du dictionnaire ? " Puis, si la réponse est positive, nous voulons afficher une telle segmentation.

Exemple : avec le dictionnaire `"alla" "ayant" "bise" "c" "chanté" "chez" "cigale" "crier" "défaut" "dépourvue" "elle" "est" "famine" "fort" "fourmi" "fut" "l" "la" "là" "moindre" "n" "pas" "prêteuse" "quand" "sa" "se" "son" "tout" "trouva" "venue" "voisine" "été"`
la chaîne "lacigaleayantchantétoutlété" est segmentable mais
la chaîne "lacigaleayantdansétoutlété" ne l'est pas car le mot "dansé" n'est pas dans le dictionnaire.

La première idée qui vient est d'analyser la chaîne dans le sens de la lecture en isolant un mot en début de chaîne et en progressant de mot en mot. Cette approche *gloutonne* ne permet pas de répondre à la question posée. Sur l'exemple ci-dessus, le fait d'isoler le mot " l " en début de la chaîne conduit à poser la question d'une segmentation de la chaîne "acigaleayantchantétoutlété". Cette chaîne ne peut être segmentée car aucun mot du dictionnaire ne peut être isolé au début de cette chaîne.

Solution par programmation dynamique. Supposons le problème résolu... Si la chaîne est *segmentable* elle se termine par un mot du dictionnaire. Qu'en est-il du préfixe, c'est-à-dire de la chaîne qui précède ce dernier mot ? Il est, lui-même, segmentable. Et qu'en est-il si la chaîne n'est pas segmentable ? Ce second cas est la situation où il n'existe aucun mot finissant la chaîne tel que le préfixe est segmentable.

Nous avons la propriété : la chaîne est segmentable si et seulement si elle se termine par un mot du dictionnaire tel que le préfixe de la chaîne est lui-même segmentable.

Soit n la longueur de la chaîne. On rappelle que la chaîne totale est le n-préfixe de la chaîne.
La propriété devient : le n-préfixe de la chaîne est segmentable si et seulement si il se termine par un mot du dictionnaire tel que le préfixe de la chaîne est lui-même segmentable.

Notons $s(n)$ la vérité de la proposition "le n-préfixe de la chaîne C est segmentable". Alors :

$$s(n) \quad = \quad \exists i,\ 0 \leq i < n \text{ tel que } C[i:n] \in D \text{ et } s(i) = \text{ vrai}$$

1. Sans aucun rapport avec notre problème : remarquons que les résultats retournés par les deux moteurs de recherche sont très différents.

Nous généralisons en notant $s(j)$ la vérité de la proposition " le j-préfixe de la chaîne C est segmentable. » Avec cette notation, la chaîne C est segmentable si et seulement si $s(n) =$ vrai.
L'expression récurrente de la propriété $s(j)$ est

$$s(j) \quad = \quad \exists i, 0 \leq i < j \text{ tel que } C[i:j] \in D \text{ et } s(j) = \text{ vrai}$$

La base de la récurrence est $s(0) =$ vrai car la chaîne vide se décompose en 0 mot du dictionnaire.
L'équation de récurrence est donc

Base $j = 0 : s(0) =$ vrai
Hérédité $1 \leq j < n+1 : s(j) \quad = \quad \exists i, 0 \leq i < j$ tel que $C[i:j] \in D$ et $s(i) =$ vrai

Ce problème est très proche d'un exercice déjà traité en travaux dirigés. Lequel ?

Tous les problèmes de tailles i présents en partie droite de l'équation de récurrence sont de taille strictement inférieure à la taille j du problème à résoudre. Ainsi, connaissant la valeur du cas de base $j = 0$, nous pouvons calculer toutes les valeurs $s(j)$ par valeurs j croissantes. Nous atteindrons la valeur $j = n$ et saurons si la chaîne est segmentable. Si cette chaîne est segmentable, nous pourrons construire la chaîne segmentée (à condition d'avoir mémorisé les valeurs $\arg s(j)$).

Représentation du dictionnaire : le dictionnaire sera représenté par une table de hachage – `HashMap`. Voir par exemple : `https://docs.oracle.com/javase/8/docs/api/java/util/Hashtable.html`.

Questions.

1. Écrire une fonction `calculerA(String C, HashMap<String,Boolean> D)` qui calcule les deux tableaux $S[0:n]$ et $A[0:n]$ de terme général $S[j] = s(j)$ et $A[j] = a(j) = \arg s(j)$. Cette fonction ne retourne que le tableau A.

   Nous posons $\arg s(j) = -1$ si et seulement si $s(j) =$ faux et nous posons $\arg s(0) = 0$.
2. Écrire une fonction `String segmentation(String C, int[] A, int j)` qui retourne une segmentation de la chaîne C.

Exemples d'exécution du programme `Segmentation` dont l'ébauche est donnée page suivante et dans le dossier partagé de l'unité :

```
% java Segmentation lacigaleayantchantétoutlété
Segmentation de lacigaleayantchantétoutlété :  la cigale ayant chanté tout l été

% java Segmentation lafourmiayantchantétoutlété
Segmentation de lafourmiayantchantétoutlété :  la fourmi ayant chanté tout l été

% java Segmentation lacigalelafourmiayantchantétoutlété
Segmentation de lacigalelafourmiayantchantétoutlété :  la cigale la fourmi ayant chanté tout l été

% java Segmentation létéayantcigaletoutlafourmi
Segmentation de létéayantcigaletoutlafourmi :  l été ayant cigale tout la fourmi

% java Segmentation lacigaleayantdansé
lacigaleayantdansé n'est pas segmentable (car le mot "dansé" n'est pas dans le dictionnaire)
```

La Cigale et la Fourmi – Jean de la Fontaine (1621 - 1695)

La Cigale, ayant chanté
tout l'été,
Se trouva fort dépourvue
Quand la bise fut venue.
Pas un seul petit morceau
De mouche ou de vermisseau.
Elle alla crier famine
Chez la Fourmi sa voisine,
La priant de lui prêter
Quelque grain pour subsister
Jusqu'à la saison nouvelle.
Je vous paierai, lui dit-elle,
Avant l'août, foi d'animal,
Intérêt et principal.
La Fourmi n'est pas prêteuse ;
C'est là son moindre défaut.
Que faisiez-vous au temps chaud ?
Dit-elle à cette emprunteuse.
Nuit et jour à tout venant
Je chantais, ne vous déplaise.
Vous chantiez ? J'en suis fort aise :
Et bien ! Dansez maintenant.

```java
import java.util.Arrays;
import java.util.HashMap;
class Segmentation{
    static int[] calculerA(String C, HashMap<String,Boolean> D){ int n = C.length();
    /* calcule S[0:n+1] de terme général S[j] = s(j) et
    A[0:n+1] de terme général A[j] = arg s(j), et retourne A. */
        boolean[] S = new boolean[n+1];
        int[] A = new int[n+1];
        // base de la récurrence j = 0
        [...]
        // cas général : 1 ≤ j < n+1
        // s(j) = "il existe i, 0 ≤ i < j tel que S[i:j] est un mot du dictionnaire D et
        // S[0:i] est segmentable."
        [...]
        return A;
    }

    static String segmentation(String C, int[] A, int j){
    /* retourne une segmentation de C.substring(0,j) (le j-préfixe de la chaîne C) */
        [...]
    }
    static HashMap<String,Boolean> dictionnaire(){
        HashMap<String, Boolean> D = new HashMap<String, Boolean>();
        D.put("la",true); D.put("ci",true); D.put("cigale",true); D.put("et",true);
        D.put("fourmi",true); D.put("le",true); D.put("ayant",true);
        D.put("chant",true); D.put("chanté",true); D.put("tout",true);
        D.put("l",true); D.put("été",true);
        return D;
    }
    public static void main(String[] Args){
        if (Args.length != 1){
            System.out.println("Usage : java Segmentation chaîne");
            return;
        }

        HashMap<String,Boolean> D = dictionnaire();

        String C = Args[0];
        int n = C.length();
        int[] A = calculerA(C,D);
        if (A[n] == -1)
            System.out.println(C + " n'est pas segmentable");
        else
            System.out.println("\nSegmentation de " + C + " : " + segmentation(C,A,n));

// Les deux lignes ci-dessous affichent le tableau A et le dictionnaire D.
// Les mettre en en commentaire si l'on ne veut pas les voir.
        System.out.println("\nA = " + Arrays.toString(A));
        System.out.println("D = " + D.toString());
    }
}
/*
% javac Segmentation.java
% java Segmentation lacigaleayantchanté

Segmentation de lacigaleayantchanté :  la cigale ayant chanté

A = [0, 0, 0, -1, 2, -1, -1, -1, 2, -1, -1, -1, -1, 8, -1, -1, -1, -1, 13, 13]
D = {ayant=true, chanté=true, chant=true, fourmi=true, la=true, ci=true, le=true, tout=true,
l=true, été=true, cigale=true, et=true}
% java Segmentation lacigaleayantdansé
lacigaleayantdansé n'est pas segmentable

A = [0, 0, 0, -1, 2, -1, -1, -1, 2, -1, -1, -1, -1, 8, -1, -1, -1, -1, -1]
D = {ayant=true, chanté=true, chant=true, fourmi=true, la=true, ci=true, le=true, tout=true,
l=true, été=true, cigale=true, et=true}
%
*/

```