Algorithme: lire\_une\_phrase\_terminant\_par\_point

Var:

Cc, Cp : carboniser // Cc = caractereCourant, Cp = caracterePrecedent

l\_ph, nbr\_m, nbr\_v: entier

// l\_ph = longueur de la phrase, nbr\_m = nombre de mots, nbr\_v = nombre de voyelle,

Debut

// Initialisation des compteurs

l\_ph := 0

nbr\_m := 0

nbr\_v := 0

// Lecture de la phrase caractère par caractère

TANT\_QUE (Cc n'est pas un point) FAIRE

// Lire le caractère suivant

Cc := LireCaractere()

// Incrémenter le compteur de longueur de phrase

l\_ph := l\_ph + 1

// Vérifier si le caractère courant est une lettre

SI (Cc est une lettre) ALORS

// Incrémenter le compteur de voyelles si c'est une voyelle

SI (Cc est une voyelle) ALORS

nbr\_v := nbr\_v + 1

FIN\_SI

// Vérifier si le caractère précédent était un espace (nouveau mot)

SI (Cp est un espace) ALORS

nbr\_m := nbr\_m+ 1

FIN\_SI

FIN\_SI

// Stocker le caractère courant comme caractère précédent pour la prochaine itération

Cp := Cc

FIN\_TANT\_QUE

// Incrémenter le nombre de mots pour le dernier mot après le point

nbr\_m := nbr\_m + 1

// Afficher les résultats

Afficher ("Longueur de la phrase :", l\_ph)

Afficher( "Nombre de mots :", nbr\_m)

Afficher ("Nombre de voyelles :", nbr\_v)